

V.S.B. ENGINEERING COLLEGE, KARUR
Academic Year: 2018-2019 (Odd Semester) Department
of Computer Science and Engineering Question Bank
(2013 Regulation)

S.No.	Name of the Subject / Lab	Semester	Page No.
1	Cryptography and Network Security	VII	2
2	Graph Theory and Applications	VII	24
3	Grid and Cloud Computing	VII	48
4	Resource Management Techniques	VII	102
5	Service Oriented Architecture	VII	137
6	Information Retrieval	VII	187

CS6701 - Cryptography and Network Security

UNIT – I PART-A

1. Define security attack, security mechanism and security services.

Security attack: any action that compromises the security of information owned by an organization. Security mechanism: a mechanism that is designed to detect, prevent or recover from a security attack. Security services: a service that enhances the security of the data processing systems and the information transfers of an organization.

2. Mention the different types of security services.

- Authentication
- Confidentiality
- Data integrity
- Non repudiation
- Access control
- Availability

3. Define passive attack and active attack.

Passive attacks are in the nature of eavesdropping, or monitoring of transmissions. The types of passive attack are Release of message content Traffic analysis Active attacks involve some modification of data stream or creation of a false stream. The types of active attack are Masquerade Replay Modification Denial of service

4. Define cryptanalysis and cryptology.

Cryptanalysis: techniques used for deciphering or decrypting a message without the knowledge of the enciphering or encrypting details is said to be Cryptanalysis.

Cryptology: the study of cryptography and cryptanalysis together is called cryptology

5. What is Brute force attack?

Trying out all the possible keys on a piece of cipher text until an intelligible translation to plain text is obtained

6. Mention the various types of cryptanalytic attack.

- Known plaintext
- Cipher text only
- Chosen plaintext
- Chosen Cipher text

7. Find $11^7 \text{ mod } 13$.

$$11^2 = 121 \text{ mod } 13 = 4$$

$$11^4 = 4 * 4 = 16 \text{ mod } 13 = 3$$

$$11^7 = 11 * 3 * 4 = 132 \text{ mod } 13 = 2$$

Answer is: $11^7 \text{ mod } 13 = 2$

8. Define the two basic building blocks of encryption techniques. Substitution technique – it is one in which the letters of the plaintext are replaced by other letters or by numbers or symbols.

Transposition technique – it is one which performs some sort of permutation on the plaintext letters.

9. Differentiate Passive Attack from Active Attack.

Passive Attack:

- Doesn't modify the content of the message.
- Traffic Analysis.
- Release of message contents.

Active Attack:

- Modification of data (or) producing false stream.
- Replay attack.
- Denial of service.

10. Difference between Cryptography and Steganography.

Cryptography:

Cryptography is a method of storing and transmitting data in a particular form so that only those for whom it is intended can read and process it. Cryptography is closely related to the disciplines of cryptology and cryptanalysis. Cryptography includes techniques such as microdots, merging words with images, and other ways to hide information in storage or transit.

Steganography:

It is the hiding of a secret message within an ordinary message and the extraction of it at its destination. Steganography takes cryptography a step farther by hiding an encrypted message so that no one suspects it exists. Ideally, anyone scanning your data will fail to know it contains encrypted data.

11. Define the following terms:

- Plaintext:** the original message to be transmitted.
- Cipher text:** the coded (encrypted) message or the scrambled message.
- Encryption / Enciphering :** process of converting plain text to cipher text.
- Decryption/ Deciphering:** process of converting cipher text to plain text.

12. What are the two basic functions used in encryption algorithm?

All the encryption algorithms are based on two general principles:

Substitution: In which each element in the plaintext (bit, letter, group of bits or letters) is mapped into another element.

Transposition: In which elements in the plaintext are rearranged.

The fundamental requirement is that no information be lost (that is that all operations are reversible). Most systems, referred to as product systems, involve multiple stages of substitutions and transpositions

13. List out the problems of one time pad.

- There is a practical problem of making large quantities of random keys.
- Key distribution and protection is also major problem with one time pad

14. Mention few mono-alphabetic and poly-alphabetic ciphers.

- Mono-alphabetic ciphers:** - playfair cipher, hill cipher, Caesar cipher
- Poly-alphabetic ciphers:** - vigenere cipher, one time pad cipher

15. Derive the Euler's Totient Function.

The Euler's totient function or ϕ () function is a very important number theoretic function having a deep relationship to prime numbers and the so-called order of integers. The totient $\phi(n)$ of a positive integer n greater than 1 is defined to be the number of positive integers less than n that are co-prime to n . $\phi(1)$ is defined to be 1.

16. Define the following term:

- Group**
- Ring**

A group is defined as: a set of elements, together with an operation performed on pairs of these elements such that:

- The operation, when given two elements of the set as arguments, always returns an element of the set as its result. It is thus fully defined and closed over the set.
- One element of the set is an identity element. Thus, if we call our operation op , there is some element of the set e such that for any other element of the set x , $e op x = x op e = x$.

- Every element of the set has an inverse element. If we take any element of the set p , there is another element q such that $p \text{ op } q = q \text{ op } p = e$.
- The operation is associative. For any three elements of the set, $(a \text{ op } b) \text{ op } c$ always equals $a \text{ op } (b \text{ op } c)$.

Rings:

A ring is a set of elements with two operations, one of which is like addition, the other of which is like multiplication, which we will call add and mul. It has the following properties:

- The elements of the ring, together with the addition operation, form a group.
- Addition is commutative. That is, for any two elements of the set p and q , $p \text{ add } q = q \text{ add } p$. (The word Abelian is also used for "commutative", in honor of the mathematician Niels Henrik Abel.)
- The multiplication operation is associative.
- Multiplication distributes over addition: that is, for any three elements of the group a , b , and c , $a \text{ mul } (b \text{ add } c) = (a \text{ mul } b) \text{ add } (a \text{ mul } c)$.

17. What is steganography? Mention few techniques in it.

Steganography is a technique for hiding the original message. Some of the related techniques are

- Character marking
- Invisible ink
- Pin punctures
- Typewriter correction ribbon

18. Convert the given text “Anna University” into cipher text using Rail-fence.

Plaintext : anna university

Rail-fence :

A	N	U	I	E	S	T
N	A	N	V	R	I	Y

Ciphertext : ANUIESTNANVRIY

19. What are the aspects required for Network security model?

Using this model requires us to:

- Design a suitable algorithm for the security transformation
- Generate the secret information (keys) used by the algorithm
- Develop methods to distribute and share the secret information
- Specify a protocol enabling the principals to use the transformation and secret information for a security service

20. Write short notes on Euclidean Algorithm

- An efficient way to find the GCD(a,b)
- Uses theorem that:

$$\text{GCD}(a,b) = \text{GCD}(b, a \text{ mod } b)$$

- Euclidean Algorithm to compute GCD(a,b) is:

EUCLID(a,b)

1. $A = a; B = b$
2. if $B = 0$ return $A = \text{gcd}(a, b)$
3. $R = A \text{ mod } B$
4. $A = B$
5. $B = R$
6. goto 2

21. What is Polynomial Division?

- Can write any polynomial in the form:

$$f(x) = q(x)g(x) + r(x)$$

can interpret $r(x)$ as being a remainder

$$r(x) = f(x) \bmod g(x)$$

- If have no remainder say $g(x)$ divides $f(x)$
- If $g(x)$ has no divisors other than itself & 1 say it is irreducible (or prime) polynomial
- Arithmetic modulo an irreducible polynomial forms a field

22. Give short notes on Chinese Remainder Theorem.

Suppose n_1, n_2, \dots, n_k are positive integers which are pairwise coprime. Then, for any given set of integers a_1, a_2, \dots, a_k , there exists an integer x solving the system of simultaneous congruences.

$$x \equiv a_1 \pmod{n_1}$$

$$x \equiv a_2 \pmod{n_2}$$

⋮

$$x \equiv a_k \pmod{n_k}$$

23. Draw Additive Modulo of 5 and Multiplicative modulo of 7.

+	0	1	2	3	4	5	6	7
0	0	1	2	3	4	5	6	7
1	1	2	3	4	5	6	7	0
2	2	3	4	5	6	7	0	1
3	3	4	5	6	7	0	1	2
4	4	5	6	7	0	1	2	3
5	5	6	7	0	1	2	3	4
6	6	7	0	1	2	3	4	5
7	7	0	1	2	3	4	5	6

×	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6
2	0	2	4	6	1	3	5
3	0	3	6	2	5	1	4
4	0	4	1	5	2	6	3
5	0	5	3	1	6	4	2
6	0	6	5	4	3	2	1

24. What is Galois Field?

- GF(p) is the set of integers $\{0, 1, \dots, p-1\}$ with arithmetic operations modulo prime p
- These form a finite field
 - since have multiplicative inverses
 - Hence arithmetic is “well-behaved” and can do addition, subtraction, multiplication, and division without leaving the field GF(p)

PART - B

1. **Illustrate and discuss about the various substitution techniques with example.**
2. **Describe in detail about OSI Security Architecture.**

3. $A \equiv 11 \pmod{37}$ $A \equiv 42 \pmod{49}$
Find the value 'A' using Chinese Remainder Theorem. Suppose add the value 678 with 'A' value, what we do by method of CRT.
4. Derive Fermat's and Euler's Theorem with suitable example.
5. Encrypt the message "PAY" using Hill Cipher with the following matrix and show the decryption to get the original plain text.

$$K = \begin{bmatrix} 17 & 17 & 5 \\ 21 & 18 & 21 \\ 2 & 2 & 19 \end{bmatrix}$$

6. Describe briefly on Network Security model with neat diagram.
7. Explain in detail on Symmetric Cipher model with suitable example.

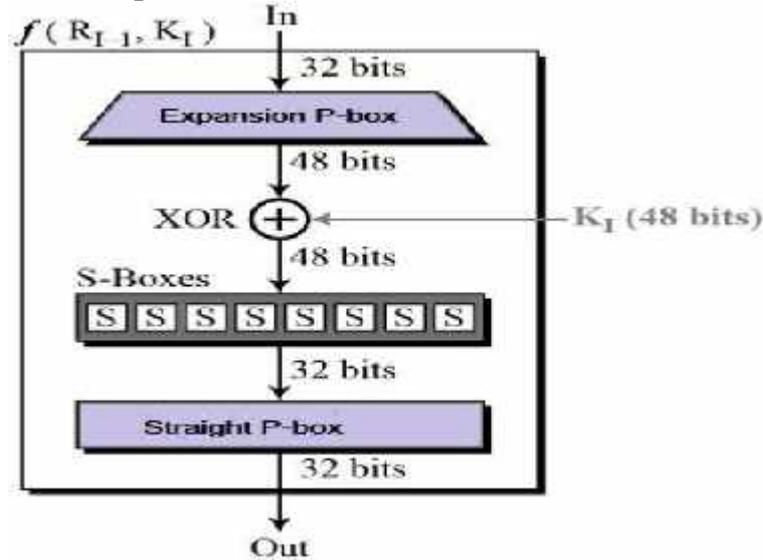
UNIT – II

PART-A

1. **Define stream cipher and block cipher.**
A stream cipher is one that encrypts a digital data stream one bit or one byte at a time.
A block cipher is one in which a block of plaintext is treated as a whole and used to produce a cipher text block of equal block.
2. **Define symmetric key cryptography and public key cryptography.**
In symmetric key cryptography, only one key is used for encryption and In public key cryptography, two keys (public key and private key) are used. When one key is used for encryption, then the other must be used for decryption. The public key is known to all the participants but the private key is kept secret by the owner.
3. **Define Euler's totient function (used in RSA algorithm).**
It is the number of positive integers that are less than 'n' and relatively prime to 'n'. Where n is the product of two prime numbers (p & q). It is represented as $F(n)$ and it is expressed as $F(n) = F(pq) = (p-1)(q-1)$.
4. **Why do we need Diffie Hellman algorithm?**
It is used for exchanging the secret keys between the sender and the receiver. It allows two users to exchange a key securely.
5. **What are the attacks that can be performed in the networks?**
 - Disclosure
 - Traffic analysis
 - Masquerade
 - Content modification
 - Sequence modification
 - Timing modification
 - Source repudiation
 - Destination repudiation
6. **Define DES. What are the advantages and disadvantages?**
The data encryption standard (DES) is a common standard for data encryption and a form of secret key cryptography (SKC), which uses only one key for encryption and decryption. Public key cryptography (PKC) uses two keys, i.e., one for encryption and one for decryption. DES is an implementation of a Feistel Cipher. It uses 16 round Feistel structure. The block size is 64-bit. Though, key

length is 64-bit, DES has an effective key length of 56 bits, since 8 of the 64 bits of the key are not used by the encryption algorithm (function as check bits only).

7. What are the techniques involved for each round in DES?



Expansion/permutation: $E()$, XOR, Substitution/choice: S-box(), Permutation: $P()$

8. Define Block cipher.

A block cipher is a method of encrypting text (to produce ciphertext) in which a cryptographic key and algorithm are applied to a block of data (for example, 64 contiguous bits) at once as a group rather than to one bit at a time. The main alternative method, used much less frequently, is called the stream cipher.

9. Differentiate cryptanalysis attack and Brute force attack

cryptanalysis attack	Brute force attack.
<ol style="list-style-type: none"> 1. Require $2^{55.1}$ operations 2. If DES had 15 rounds or few, attack requires less effort 	<ol style="list-style-type: none"> 1. Require 2^{55} operations 2. If DES had 15 rounds or few, attack requires more effort

10. List the types of Block cipher modes of operations

- Electronic Codebook (ECB)
- Cipher Block Chaining (CBC)
- Propagating Cipher Block Chaining (PCBC)
- Cipher Feedback (CFB)
- Output Feedback (OFB)
- Counter (CTR)

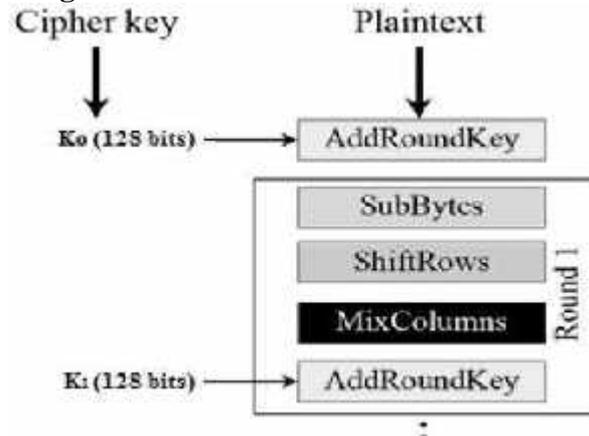
11. Define AES.

AES is an iterative rather than Feistel cipher. It is based on ‘substitution–permutation network’. It comprises of a series of linked operations, some of which involve replacing inputs by specific outputs (substitutions) and others involve shuffling bits around (permutations).

Interestingly, AES performs all its computations on bytes rather than bits. Hence, AES treats the 128 bits of a plaintext block as 16 bytes. These 16 bytes are arranged in four columns and four rows for processing as a matrix –

Unlike DES, the number of rounds in AES is variable and depends on the length of the key. AES uses 10 rounds for 128-bit keys, 12 rounds for 192-bit keys and 14 rounds for 256-bit keys. Each of these rounds uses a different 128-bit round key, which is calculated from the original AES key.

12. Draw the block diagram for AES



13. What are the transformation functions for each round in AES?

- Substitution bytes
- Shift rows
- Mix columns
- Add round key

14. What are the approaches to attack the RSA algorithm?

Brute attack

- Mathematical attack
- Timing attack
- Hardware fault based attack
- Chosen cipher text attack

15. How to manage the keys?

Encryption is an effective way to secure data, but the encryption keys used must be carefully managed to ensure data remains protected and accessible when needed.

- Symmetric key distribution using Symmetric encryption
- Symmetric key distribution using Asymmetric encryption

16. Define Elliptic curve cryptography

Elliptic curve cryptography (ECC) is an approach to public-key cryptography based on the algebraic structure of elliptic curves over finite fields. ECC requires smaller keys compared to non-ECC cryptography (based on plain Galois fields) to provide equivalent security.

17. List four general characteristics of schema for the distribution of the public key.

- Public announcement
- Publicly available directory
- Public-key authority
- Public-key certificates

18. Where is the miller-rabin algorithm is used?

Miller–Rabin primality test. The Miller–Rabin primality test or Rabin–Miller primality test is a primality test: an algorithm which determines whether a given number is prime, similar to the Fermat primality test and the Solovay–Strassen primality test.

$$X^2 \equiv y^2 \pmod{n}$$

19. Perform encryption and decryption using the RSA algorithm.

1. $p = 3; q = 11, e = 7; M = 5$
2. $p = 5; q = 11, e = 3; M = 9$
3. $p = 7; q = 11, e = 17; M = 8$
4. $p = 11; q = 13, e = 11; M = 7$
5. $p = 17; q = 31, e = 7; M = 2$

Hint: Decryption is not as hard as you think; use some finesse.

1. $n = p \times q = 3 \times 11 = 33$
 $j(n) = (p-1) \times (q-1) = 2 \times 10 = 20$
 $\gcd(j(n), e) = \gcd(20, 7) = 1$
 $\therefore d \equiv e^{-1} \pmod{j(n)}$
 $d \times e \pmod{j(n)} = 1$
 $7d \pmod{20} = 1$
 $\therefore d = 3$

So: Public Key $pu = \{e, n\} = \{7, 33\}$
Private Key $pr = \{d, n\} = \{3, 33\}$

Encryption:

$$C = M^e \pmod{n} = 5^7 \pmod{33} = 14$$

Decryption: $M = C^d \pmod{n} = 14^3 \pmod{33} = 5$

20. What is the effect on the cipher text of a error in block P1 of plain text?. Assume CBC mode. What is the error at the reciever?

If a bit of a plain text block P1 is in error the entire cipher text block will be effected and will be erroneous. (Though, the encryption algorithm is correctly encrypting what is given to it.) All subsequent cipher blocks will also be effected each cipher text block is fed to next stage and XOR with next plain text block. However, at the receiver, only the block P1 of plain text recovered reproduces the same bit error. All the subsequent plain text blocks are reproduced correctly.

21. Define Public key certificate

In cryptography, a public key certificate (also known as a digital certificate or identity certificate) is an electronic document used to prove ownership of a public key.

22. Define RC5 algorithm

The RC5 encryption algorithm is a fast, symmetric block cipher suitable for hardware or software implementations. A novel feature of RC5 is the heavy use of data-dependent rotations. RC5 has a variable-length secret key, providing flexibility in its security level.

23. What are the uses of Public Key Cryptography

- Encryption/ Decryption
- Digital Signature
- Key exchange

24. Define RSA Cryptosystem-(three scholars Ron Rivest, Adi Shamir, and Len Adleman)

Two aspects of the RSA cryptosystem, firstly generation of key pair and secondly encryption-decryption algorithms. RSA does not directly operate on strings of bits as in case of symmetric key encryption.

Generate the RSA modulus (n)

- Select two large primes, p and q.
- Calculate $n=p*q$. For strong unbreakable encryption, let n be a large number, typically a minimum of 512 bits.

Find Derived Number (e)

- Number e must be greater than 1 and less than $(p - 1)(q - 1)$.
- There must be no common factor for e and $(p - 1)(q - 1)$ except for 1. In other words two numbers e and $(p - 1)(q - 1)$ are coprime.

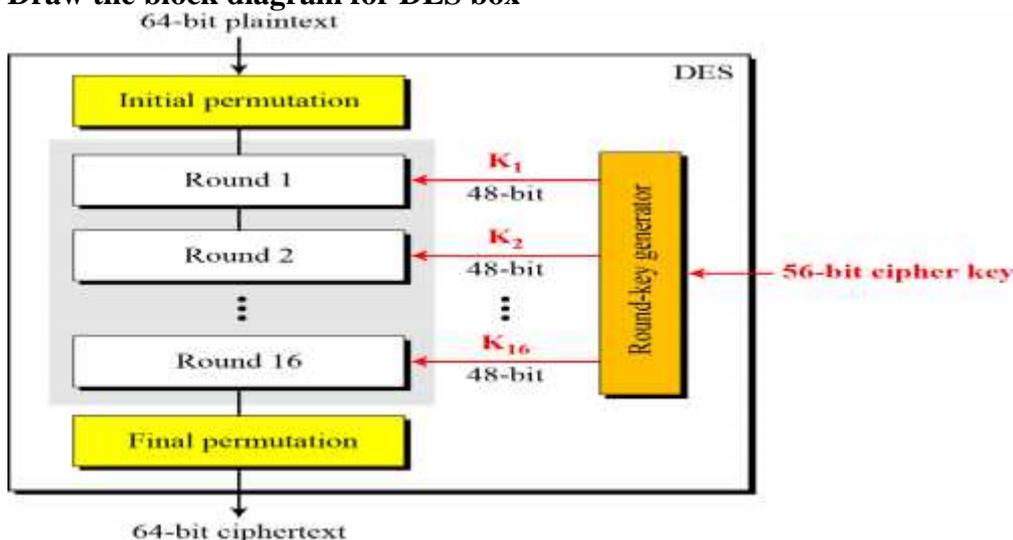
Form the public key

- The pair of numbers (n, e) form the RSA public key and is made public.
- Interestingly, though n is part of the public key, difficulty in factorizing a large prime number ensures that attacker cannot find in finite time the two primes (p & q) used to obtain n. This is strength of RSA.

Generate the private key

- Private Key d is calculated from p, q, and e. For given n and e, there is unique number d.
- Number d is the inverse of e modulo $(p - 1)(q - 1)$. This means that d is the number less than $(p - 1)(q - 1)$ such that when multiplied by e, it is equal to 1 modulo $(p - 1)(q - 1)$.

25. Draw the block diagram for DES box



PART - B

- 1. Identify the possible threats for RSA algorithm and list their counter measures**
- 2. Perform decryption and encryption using RSA algorithm with $p=3$, $q=11$, $e=7$ and $N=5$.**
- 3. Draw the general structure of DES and explain the encryption decryption process.**
- 4. Explain the generation sub key and S Box from the given 32-bit key by Blowfish.**
- 5. In AES, how the encryption key is expanded to produce keys for the 10 rounds**
- 6. Explain AES algorithm with block diagram**
- 7. Describe about RC4 and RC5 algorithm.**
- 8. Mention the strengths and weakness of DES algorithm.**

UNIT – III
PART - A

PART - B

1. Define hash Function and Explain its properties in cryptography.

1. **What is message authentication?**

It is a procedure that verifies whether the received message comes from assigned source has not been altered. It uses message authentication codes, hash algorithms to authenticate the message

2. **Define the classes of message authentication function.**

Message encryption: The entire cipher text would be used for authentication.

Message Authentication Code: It is a function of message and secret key produce a fixed length value.

Hash function: Some function that map a message of any length to fixed length which serves as authentication.

3. **What are the requirements for message authentication?**

The requirements for message authentication are

Disclosure: Release of message contents to any person or process not processing the appropriate cryptographic key

Traffic Analysis: Discovery of the pattern of traffic between parties. In a connection oriented application, the frequency and duration of connections could be determined. In either a connection oriented or connectionless environment, the number and length of messages between parties could be determined.

Masquerade: Insertion of messages into the network from a fraudulent source. This includes the creation of messages by an opponent that are purported to come from an authorized entity. Also included are fraudulent acknowledgements of message receipt or no receipt by someone other than the message recipient.

Content modification: Changes to the contents of a message, including insertion, deletion, transposition, and modification.

Sequence modification: Any modification to a sequence of messages between parties, including insertion, deletion, and modification.

Timing modification: Delay or replay of messages. In a connection oriented application, an entire session or sequence of messages could be a replay of some previous valid session, or individual messages in the sequence could be delayed or replayed. In connectionless application, an individual message could be delayed or replayed.

Source repudiation: Denial of transmission of message by source.

Destination repudiation: Denial of receipt of message by destination.

4. **What you meant by hash function?**

Hash function accept a variable size message M as input and produces a fixed size hash code $H(M)$ called as message digest as output. It is the variation on the message authentication code.

5. **Differentiate MAC and Hash function?**

MAC: In Message Authentication Code, the secret key shared by sender and receiver. The MAC is appended to the message at the source at a time which the message is assumed or known to be correct.

Hash Function: The hash value is appended to the message at the source at time when the message is assumed or known to be correct. The hash function itself not considered to be

secret.

6. Write Any three hash algorithm.

- MD5 (Message Digest version 5) algorithm.
- SHA_1 (Secure Hash Algorithm).
- RIPEMD_160 algorithm.

7. What are the requirements of the hash function?

H can be applied to a block of data of any size. H produces a fixed length output. H(x) is relatively easy to compute for any given x, making both hardware and software implementations practical

$$h = H(M)$$

M = Variable length Message

H(M) = Fixed length hash value

8. What you meant by MAC?

MAC is Message Authentication Code. It is a function of message and secret key which produce a fixed length value called as MAC.

$$\text{MAC} = \text{Ck}(M)$$

Where,

M = variable length message

K = secret key shared by sender and receiver.

CK (M) = fixed length authenticator.

9. Differentiate internal and external error control.

Internal error control:

In internal error control, an error detecting code also known as frame check sequence or checksum.

External error control:

In external error control, error detecting codes are appended after encryption.

10. What is the meet in the middle attack?

This is the cryptanalytic attack that attempts to find the value in each of the range and domain of the composition of two functions such that the forward mapping of one through the first function is the same as the inverse image of the other through the second function-quite literally meeting in the middle of the composed function.

11. What is the role of compression function in hash function?

The hash algorithm involves repeated use of a compression function f, that takes two inputs and produce a n-bit output. At the start of hashing the chaining variable has an initial value that is specified as part of the algorithm. The final value of the chaining variable is the hash value usually $b > n$; hence the term compression.

12. Distinguish between direct and arbitrated digital signature?

Direct digital signature Arbitrated Digital Signature The direct digital signature involves only the communicating parties.

- The arbiter plays a sensitive and crucial role in this digital signature.
- This may be formed by encrypting the entire message with the sender's private key.
- Every signed message from a sender x to a receiver y goes first to an arbiter A, who subjects the message and its signature to a number of tests to check its origin and content.

13. What are the properties a digital signature should have?

- It must verify the author and the data and time of signature.
- It must authenticate the contents at the time of signature.
- It must be verifiable by third parties to resolve disputes.

•

14. What requirements should a digital signature scheme should satisfy?

- The signature must be bit pattern that depends on the message being signed.
- The signature must use some information unique to the sender, to prevent both forgery and denial.
- It must be relatively easy to produce the digital signature.
- It must be relatively easy to recognize and verify the digital signature. It must be computationally infeasible to forge a digital
- Signature, either by constructing a new message for an existing digital signature or by constructing a fraudulent digital signature for a given message.

It must be practical to retain a copy of the digital signature in storage.

15. What types of attacks are addressed by message authentication?

Content modification: Changes to the contents of the message

Sequence Modification: Any modification to a sequence of messages between parties including insertion, deletion and recording

Timing Modification: Delay or replay of message

16. What is the difference between a message authentication code and a one-way hash function?

The difference between a MAC and a one way hash function is that unlike a MAC a hash code does not use a key but is a function only of the input message.

17. Is it necessary to recover the secret key in order to attack a MAC algorithm?

A number of keys will produce the correct MAC and the opponent has no way of knowing which the correct key is. On an average $2^{(n-k)}$ keys produce a match. Therefore attacks do not require the discovery of the key.

18. What is the function of a compression function in a hash function?

The hash function involves repeated use of a compression function. The motivation is that if the compression function is collision resistant, then the hash function is also collision resistant function. So a secure hash function can be produced.

19. What are the two types of certificates?

Two types of certificates are,

- i. Forward Certificate
- ii. Reverse Certificate

20. What is public key certificate?

The public-key authority could be somewhat of a bottleneck in the system, for a user must appeal to the authority for a public key for every other user that it wishes to contact. As before, the directory of names and public keys maintained by the authority is vulnerable to tampering.

21. What are the requirements for the use of a public key certificate scheme?

- Any participant can read a certificate to determine the name and public key of the certificate's owner.
- Any participant can verify that the certificate originated from the certificate authority and is not counterfeit.
- Only the certificate authority can create and update certificates.
- Any participant can verify the currency of the certificate.

22. What is the use of digital signature and it is two approaches?

Data appended to, or a data unit that allows a recipient if the data unit to prove the source and integrity of the data unit and protect against forger.

- RSA Approach
- DSS Approach

23. What is birthday attack?

A birthday attack is a name used to refer to class of brute force attacks. It gets its name from

the surprising result that the probability that two or more people in a group of 23 share the same birthday is greater than 0.5. such a result is called a birthday paradox.

24. What do you mean by one way property in hash function?

For any given value h , it is computationally infeasible to find x such that $H(x) = h$.

25. What is digital signature?

Digital signature is an authentication mechanism that enables the creator of a message to attach a code that acts as a signature.

26. What is one way property?

A function that maps an arbitrary length message to a fixed length message digest is a one-way property hash function if it is a one-way function.

27. Write any two differences between MD5 and secure hash algorithm.

MD5	SHA
Pad message so its length is $448 \bmod 512$	Pad message so its length is a multiple of 512 bits
Initialize the 4 word buffer (A, B, C, D)	Initialize the 5 bit buffer (A, B, C, D, E)
Process the message in 16 word chunks using 3 rounds of 16 bit operations each on chunk and buffer	Process the message in 16 word chunks using 4 rounds of 20 bit operations

28. What are the performance difference between MD5, SHA-512, and RIPEMD-16?

- MD5 produces a 128 bit hash value. SHA-512 produces 160 bit hash values
- Brute force attack is harder
- Not vulnerable to known attacks
- Slower than MD5
- All designed as simple and compact
- SHA-1 optimized for big-endian CPUs vs RIPEMD-160 & MD5 optimized for little-endian CPUs

2. Explain Digital signature with ElGamal public key cryptosystems.

3. Explain Secure hash algorithm in detail.

4. Explain the process of deriving eighty 64 bit words from the 1024-bits for processing of a single block and also discuss single round function in SHA-512 algorithm. Show the values of W_{16} , W_{17} , W_{18} and W_{19} .

5. Alice chooses $Q=101$ and $P=7879$. Assume $(q,p,\text{and } y)$: Alice public key, Alicxe selects $h=3$ and calculates g . Alice choose $x=75$ as the private key and calculates y . Now Alice can send a message to bob. Assume that $H(M) = 22$ and alic chooses secret no $K=50$. Verify the signature.

6. Explain in detail about MD5 algorithm with necessary diagrams.

UNIT – IV

PART-A

1. What is an intruder?

Accessing a network unauthorized is called intrusion.

2. What is intrusion deduction system?

An Intrusion Deduction System (IDS) is a system for deduction unauthorized access to the system

3. What are audit reports? Give its two forms?

□ Audit report is a fundamental tool for intrusion deducting. Two forms of audit are:

1. Native audit records

2. Detective specific audit records.

4. Define malicious program.

A program that is intentionally included or inserted in a system for harmful purpose is malicious program.

5. What is a virus?

A virus is a piece of program code that can infect other programs by modifying them.

6. What is a worm?

A worm is a program designed to copy itself and send copies from a computer to other computer across the network.

7. Enlist four types of viruses?

1. Parasitic virus.
2. Memory resident virus.
3. Boot sector virus.
4. Stealth virus.

8. What is Trojan horse?

A Trojan horse is a computer program that appears to be useful but that actually does damage.

9. What is a logic bomb?

A logic bomb is a software embedded in some legitimate programs and is set to explode under certain conditions..

10. What are the steps in virus removal process?

- a) Detection of virus
- b) Identification of virus
- c) Removal of traces of virus.

11. What is generic decryption technology?

A generic decryption technology can detect most complex polymorphic viruses with fast scanning speed.

12. What is Denial Of Service?

A denial of service is an attempt to prevent a genuine user of service from using it.

13. What are the design goals of firewalls?

1. All the traffic must pass through it.
2. Only authorized traffic is allowed to pass.
3. Firewall itself is immune to penetration.

14. Enlist commonly used firewalls from threats of security

- a) Packet filtering router
- b) Application level gateway
- c) Circuit level gateway

15. Who is masquerader and who is clandestine user?

1. **Masquerader:** An unauthorized user who penetrates a system access control and exploit an user.

2. **Clandestine user:** A user who seizes supervisory control of system to suppress audit collection.

16. What is mentioned by a trusted system?

A trusted system is a computer and operating system that can be verified to implement a given security policy. Typically, the focus of a trusted system is data access control. A policy is implemented that dictates what objects may be accessed.

17. Define honey pots.

A honey pot is a trap set to detect, deflect or in some manner counteract attempts at unauthorized use of information systems.

18. List out the types of viruses.

Types of viruses are;

- a) Parasitic virus
- b) Memory-resident virus
- c) Boot sector virus
- d) Stealth virus
- e) Polymorphic virus

19. What are the major issues derived by porras about the design of a distributed intrusion deduction system?

Porras points out following major issues :

- a) System may need to deal with different audit record formats.
- b) One or more nodes in the network will serve as collision and analysis points for the data from the systems on the networks.
- c) Either centralized or decentralized architecture can be used.

20. What are three main components involved in the distributed intrusion detection system ?

Components :

- a) **Host agent module:** An audit collection module operating as a background process on a monitored system.
- b) **Lan monitor agent module:** Same as host agent except that it analysis LAN traffic and reports.
- c) **Central manager module:** Receives reports from LAN monitor and hos agents and processes and correlates these reports to deduction intrusion.

21. Define intruder. Name three different classes of intruders.

An intruder is a person who attempts to gain unauthorized access to a system, to damage that system, or to disturb data on that system. Classes of intruders: masquerader, misfeasor, clandestine user.

22. What do you mean by Trojan horses?

Trojan horse is a computer program that appears to be useful but that actually does damage.

23. What is honey pot?

A System placed there just so it will be attacked, so attackers waste time, and so their attacks can be analyzed.

24. Write down the role of security standards.

Standard allows products from multiple vendors to communicate, giving the purchaser more flexibility in equipment selection in use.

25. Write down the system security standards?

Security standards development and publication are done by Internet architecture board, Internet engineering task force and Internet engineering steering group.

26. Define: Intrusion.

Intrusion is an illegal act of entering, seizing or taking possession of another's property.

27 Give few examples of worms.

Example of worm is the Morris worm and My doom.

28. Mention the two levels of hackers.

Two level's of hackers are criminal hackers , disgruntled employees, ideological hackers and underemployed adult hackers etc.

29. What are the effects of malicious software? Write an two?

The effects of malicious viruses on a computer system include occupation of disk space. Tancle 2 virus will change icons on a computer screen.

30. What are Zombies?

Zombies are computer connected to internet that has been compromised by a hacker, computer virus or Trojan horse and can be used to perform malicious tasks under remote direction.

31. Difference between spyware and virus

Sno	Delete	Truncate
1.	Spyware is specific unwanted software that collects user information without appropriate notice and consent.	A virus is a specific software that can be distributed (spread) from computer to computer usually by e-mail.
2.	Spyware tries to stick to the computer.	Virus spreads throughout the system i.e. one computer to other.

32. What problem was Kerberos designed to address?

The problem that Kerberos addressed is this : assume an open distributed environment in which users at workstations wish to access services on servers distributed throughout the network. We would like for servers to be able to restrict access to authorized users and to be able to authenticate requests for service In this environment a workstation cannot be trusted to identify its users correctly to Network services.

33. List four requirements defined for Kerberos.

The four requirements define for Kerberos are: **Secure, Reliable, Transparent and Scalable.**

34. What entities constitute a full-service Kerberos environment?

A full service environment consists of a Kerberos server, a number of clients and a number of application servers.

35. List out the requirements of Kerberos.

Kerberos requirements are secure, reliable, transparent ans scalable.

36. What are the principle differences between Kerberos version 4 and version 5?

- 1)Kerberos V.4 requires DES and V.5 allows many encryption techniques
- 2)V.4 requires use of IP and V.5 allows other network protocols.
- 3)Version 5 has a longer ticket life time
- 4)Version 5 allows tickets to be renewed
- 5)Version 5 can accept any symmetric-key algorithm
- 6)Version 5 uses a different protocol for describing data types
- 7)Version 5 has more overhead than Version 4

37. Define :malicious software

Malicious software is any software that gives partial to full control of your computer to do whatever the malware creator wants. Malware can be a virus, worm , trojan, adware, spyware, root kit , etc.

38. Differentiate macro virus and boot virus.

A macro virus is platform independent virtually all of the macro viruses in fact MS word documents. Macro viruses take advantages of a feature found in word and other office applications such as Microsoft excel , namely the macro. Boot sector virus infects a master boot record or boot record and spreads when a system is booted from the disk containing the virus.

39. When are the certificates revoked in X.509?

The certificate should be revoked before expiry because of following reasons:

- 1) User's private key is compromised
- 2) User is not certified by CA
- 3) CA's certificate is compromised.

40. What are the advantages of intrusion detection system over firewall?

- a) Monitoring and analysis for user and system activity
- b) Auditing of system configurations and vulnerabilities
- c) Assessing the integrity of critical system and data files
- d) Recognition of activity patterns reflecting known attacks
- e) Statistical analysis for abnormal activity patterns
- f) Operating system audit trail management, with recognition of user activity reflecting policy violations.

41. List down the difference between viruses and worms.

Virus: A computer virus is a program that is loaded on your computer without your knowledge and runs without your permission. A virus is designed to reproduce itself through legitimate processes in computer programs and operating systems; therefore, a virus requires a host in order to replicate. Viruses are often capable of mutating or changing while they are replicating themselves.

Worm: A worm is a small piece of software that uses security holes within networks to replicate itself. The worm scans the network for another computer that has a specific security hole. It copies itself to the new machines exploiting the security hole, and then starts replicating from that system as well. Once infected, The worm may send itself to everyone in your address book.

PART - B

1. Write short note on : Firewalls
2. Write short note on : Viruses
3. Elaborately explain Kerberos authentication mechanism with suitable diagrams
4. Explain statistical anomaly detection and rule based intrusion detection
5. Describe any two advanced anti-virus techniques in detail.
6. Discuss the architecture of distributed intrusion detection system with the necessary diagrams. Illustrate the three common types of firewalls with diagrams
7. Explain Kerberos Version 4 in detail
8. Explain the characteristics and types of firewall.
9. Write brief notes on the following:
 - i) classification of viruses
 - ii) Worm counter measures.
10. Explain the SET operations
11. Explain the intrusion detection techniques
12. Explain the trusted systems

UNIT – V
PART-A

1. Why does PGP generate a signature before applying compression?

The signature is generated before compression due to 2 reasons :

1.It is preferable to sign an uncompressed message so that one can store only the uncompressed message together with the signature for future verification.

2.Even if one were willing to generate dynamically a recompressed message for verification, PGP's compression algorithm presents a difficulty.

2. Why is Radix 64 conversion useful for email generation?

The Radix 64 conversion is performed before the segmentation of the messages take place. The use of radix 64 is that it converts the input stream to 33%. The radix 64 converts the input stream to a radix 64 format.

3. What is MIME?

Multipurpose Internet Mail Extension (MIME) is an extension to the RFC 822 framework that is intended to address some of the problems and limitations of these uses of SMTP.

4. What is S/MIME ?

Secure/Multipurpose Internet Mail Extension is a security enhancement to the MIME Internet e-mail format standard , based on technology from RSA Data Security. It is ability to sign and / or encrypt messages.

5. What Services are provided by IPSec?

Services provided by IPSec.

- a) Access control
- b) Connectionless integrity
- c) Data origin authentication
- d) Rejection of replayed packets.

6. What is reply attack?

A reply attack is one which an attacker obtains a copy of an authentication packet and the later transmits it to the intended destination.

7. What is the difference between Transport mode and Tunnel mode?

The main differences between Transport mode and Tunnel mode are:

Transport mode	Tunnel mode
It provides protection for upper layer protocols.	It provides protection to the entire IP packet.
Used for end-to-end communication between two host.	It is used when one or both ends of an SA is a security gateway , such as firewall or router that implement IPSec.
AH: Authentication IP payload and selected portions of IP header and IPv6 extension header .	Authentication entire inner IP packet Plus selected portions of outer IP header and outer IPv6 extension headers

8. What is the difference between an SSL connection and an SSL session?

A connection is a transport that provides a suitable type of service. For SSL, such connections are peer-to-peer relationships. The connections are transient. An SSL session is an association between a client and a server. Sessions are created by the Handshake Protocol. Session define a set of cryptographic security parameters, which can be shared among multiple connections.

9. Why does ESP include a padding field?
 Padding field is added to the ESP to provide partial traffic flow confidentiality by concealing the actual length of the payload.
10. **Why is the segmentation and reassembly function in PGP needed?**
 E-mail facilities often are restricted to a maximum message length. To accommodate this restriction, PGP automatically subdivides a message that is too large into segments that are small enough to send via e-mail. The segmentation is done after all of the other processing, including the radix-64 conversion. Thus, the session key component and signature component appear only once, at the beginning of the first segment.
11. **How does PGP use the concept of trust?**
 PGP provide a convenient means of using trust, associating trust with public keys, and exploiting trust information. Each entry in the public-key ring is a public key certificate. Associated with each such entry is a key legitimacy field that indicates the extent to which PGP will trust that this is a valid public key for this user; the higher the level of trust , the stronger is the binding of this user ID to this key.
12. What are the applications involved in IP Security?
 Application of IP security
 a)Provide secure communication across private and public LAN.
 b)Secure remote access over the Internet.
 c)Secure communication to other organization.
13. **Mention four SSL protocols**
 Four SSL protocols are
 1.Handshaking protocol: Establish communication variables
 2.Change cipher spec protocol: Alert to a change in communication variables
 3.Alert protocol: Messages important to SSL connections
 4.Application encryption protocol: Encrypt /decrypt application data
14. What is MIME and S/MIME?
 Multipurpose Internet Mail Extensions(MIME) is a supplementary protocol that allows non-ASCII data to be sent through email. Secure/Multipurpose Internet Mail Extension extends the protocols of MIME by adding digital signatures and encryption to them. S/MIME is not restricted to mail; it can be used with any transport mechanism that transports MIME data, such as HTTP.
15. **Define TLS?**
 Transport Layer Security(TLS) is a protocol that encrypts and delivers mail securely.TLS encryption requires the use of a digital certificate, which contains identity information about the certificate owner as well as a public key, used for encrypting communications.
16. **What do you mean by S/MIME?**
 S/MIME is a security enhancement to the MIME Internet e-mail format standard, based on technology from RSA Data Security. S/MIME provides the cryptographic security services for electronic messaging applications: authentication, message integrity, non-repudiation of origin, privacy and data security.
17. **What are the different types of MIME?**
 MIME types are:
 Text, Multipart, Message type, Image type, Video type, Audio type and Application type.
18. **What protocols comprise SSL?**
 SSL Record Protocol, SSL Handshake Protocol, SSL Change Cipher Spec etc.
19. List out the services provided by PGP.
 Services provided by PGP are digital signature , message encryption, compression, e-mail compatibility and segmentation.

20. Expand and define SPL.

The Security Parameter Index (SPI) is an identification tag added to the header while using IPsec for tunneling the IP traffic. This tag helps the Kernel discern Between two traffic streams where different encryption rules and algorithms may be in use.

21. Define :SET.

Secure electric transaction(SET) is an open encryption and security specification designed to protect credit card transactions on Internet.

22. What do you mean by PGP ?

PGP stands for Pretty Good Privacy. It was developed originally by Phil Zimmerman. PGP is open-source. Although PGP can be used for protecting data in long-term storage, it is used primarily for email security.

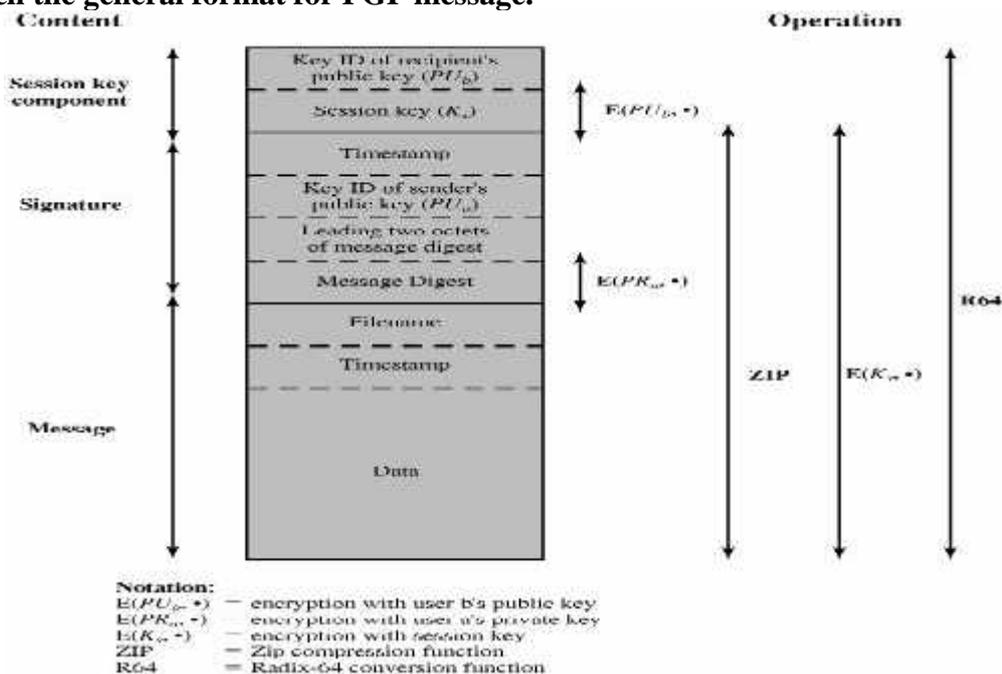
23. What is optimally asymmetric encryption padding?

Optimally Asymmetric Encryption Padding(OAEP): OAEP is main standard padding for RSA public key encryption : a way to format that message before encryption in order to reach a higher security level.

24. What are the protocols used to provide IP Security ?

Authentication header (AH) protocol and Encapsulating Security Payload(ESP) used to provide IP security.

25. Sketch the general format for PGP message.



26. What is tunnel mode in IP security?

Tunnel mode provides protection to entire IP packets. Security fields are added IP packets and entire packet is new IP packet with a new IP header. Entire new IP packet travels through a tunnel from one point to other over IP network. No routers over the network are able to detect inner IP header. Since original packet is encapsulated by new larger packet having different source and destination address

PART - B

1. Write short note: Web security

2. Write short note: SSL.

3. **Explain about the PKI.**
4. **Explain pretty good privacy in detail.**
5. **For what purpose Zimmerman developed PGP? Brief the various services provided by PGP. Discuss the threats faced by an e-mail and explain its security requirements to provide a secure e-mail service.**
6. **Explain Secure Socket Layer (SSL) in detail.**
7. **Draw and explain the IP security architecture**

CS6702 GRAPH THEORY AND APPLICATIONS 2 MARKS QUESTIONS AND ANSWERS

UNIT I INTRODUCTION

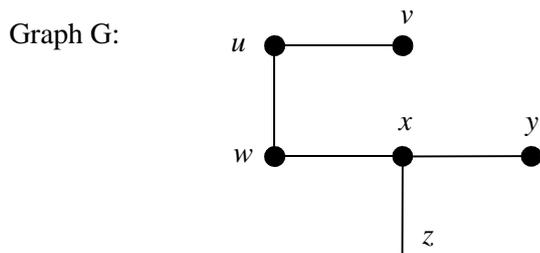
1. Define Graph.

A graph $G = (V, E)$ consists of a set of objects $V = \{v_1, v_2, v_3, \dots\}$ called **vertices** (also called **points** or **nodes**) and other set $E = \{e_1, e_2, e_3, \dots\}$ whose elements are called **edges** (also called **lines** or **arcs**).

The set $V(G)$ is called the **vertex set** of G and $E(G)$ is the **edge set** of G .

For example :

A graph G is defined by the sets $V(G) = \{u, v, w, x, y, z\}$ and $E(G) = \{uv, uw, wx, xy, xz\}$.

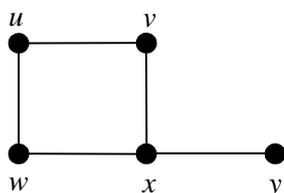


A graph with p -vertices and q -edges is called a **(p, q) graph**. The $(1, 0)$ graph is called **trivial graph**.

2. Define Simple graph.

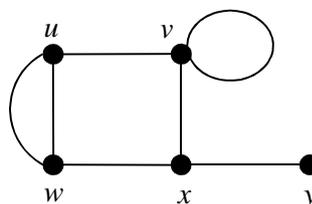
- An edge having the same vertex as its end vertices is called a self-loop.
- More than one edge associated a given pair of vertices called parallel edges.
- A graph that has neither self-loops nor parallel edges is called simple graph.

Graph G:



Simple Graph

Graph H:



Pseudo Graph

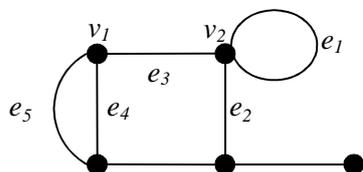
3. Write few problems solved by the applications of graph theory.

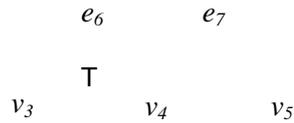
- Konigsberg bridge problem
- Utilities problem
- Electrical network problems
- Seating problems

4. Define incidence, adjacent and degree.

When a vertex v_i is an end vertex of some edge e_j , v_i and e_j are said to be *incident* with each other. Two non parallel edges are said to be *adjacent* if they are incident on a common vertex. The number of edges incident on a vertex v_i , with self-loops counted twice, is called the *degree* (also called valency), $d(v_i)$, of the vertex v_i . A graph in which all vertices are of equal degree is called *regular graph*.

Graph G:

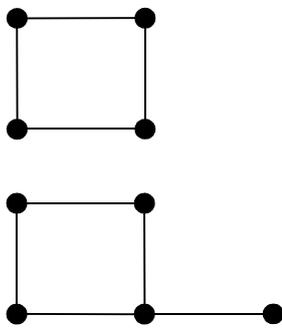




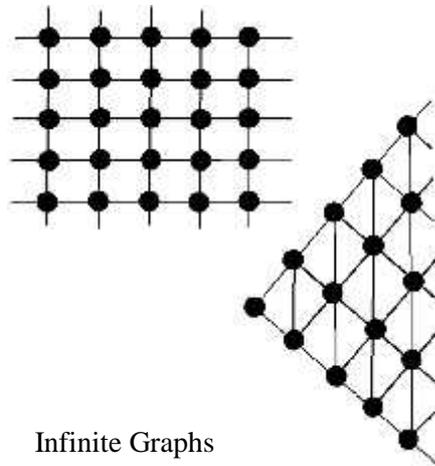
The edges e_2, e_6 and e_7 are incident with vertex v_4 .
 The edges e_2 and e_7 are adjacent.
 The edges e_2 and e_4 are not adjacent.
 The vertices v_4 and v_5 are adjacent.
 The vertices v_1 and v_5 are not adjacent.
 $d(v_1) = d(v_3) = d(v_4) = 3. d(v_2) = 4. d(v_5) = 1.$

5. What are finite and infinite graphs?

A graph with a finite number of vertices as well as a finite number of edges is called a *finite* graph; otherwise, it is an *infinite* graph.



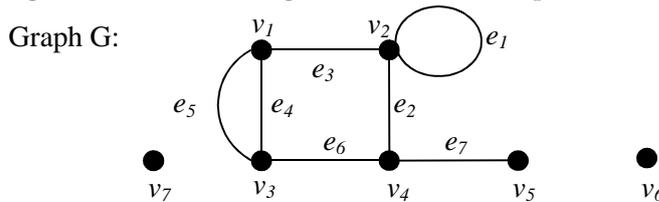
Finite Graphs



Infinite Graphs

6. Define Isolated and pendent vertex.

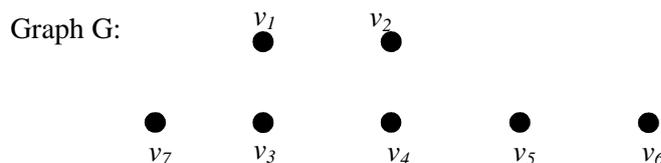
A vertex having no incident edge is called an *isolated vertex*. In other words, isolated vertices are vertices with zero degree. A vertex of degree one is called a *pendant vertex* or an *end vertex*.



The vertices v_6 and v_7 are *isolated vertices*.
 The vertex v_5 is a *pendant vertex*.

7. Define null graph.

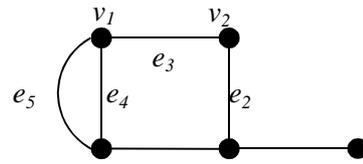
In a graph $G=(V, E)$, If E is empty (Graph without any edges) Then G is called a *null graph*.

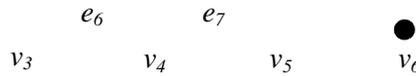


8. Define Multigraph

In a multigraph, no loops are allowed but more than one edge can join two vertices, these edges are called **multiple edges** or parallel edges and a graph is called **multigraph**.

Graph G:





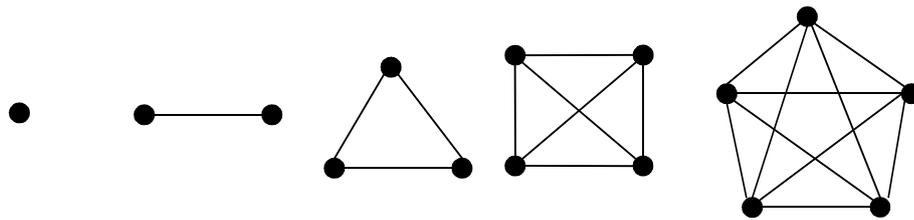
The edges e_5 and e_4 are **multiple** (parallel) edges.

9. Define complete graph

A simple graph G is said to be **complete** if every vertex in G is connected with every other vertex. *i.e.*, if G contains exactly one edge between each pair of distinct vertices.

A complete graph is usually denoted by K_n . It should be noted that K_n has exactly $n(n-1)/2$ edges.

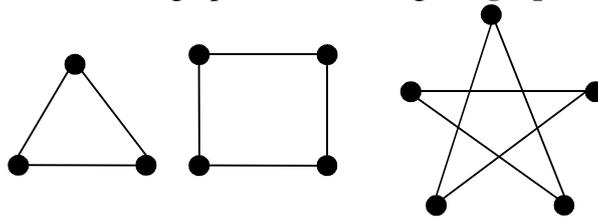
The complete graphs K_n for $n = 1, 2, 3, 4, 5$ are shown in the following figure.



10. Define Regular graph

A graph in which all vertices are of **equal degree**, is called a **regular graph**.

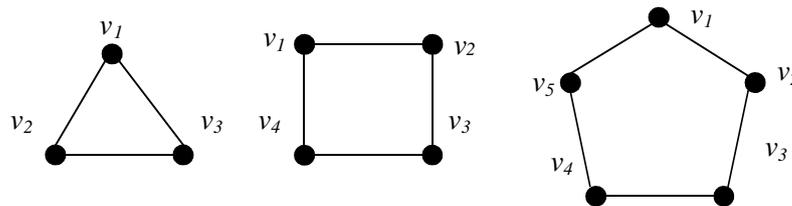
If the degree of each vertex is r , then the graph is called a regular graph of degree r .



11. Define Cycles

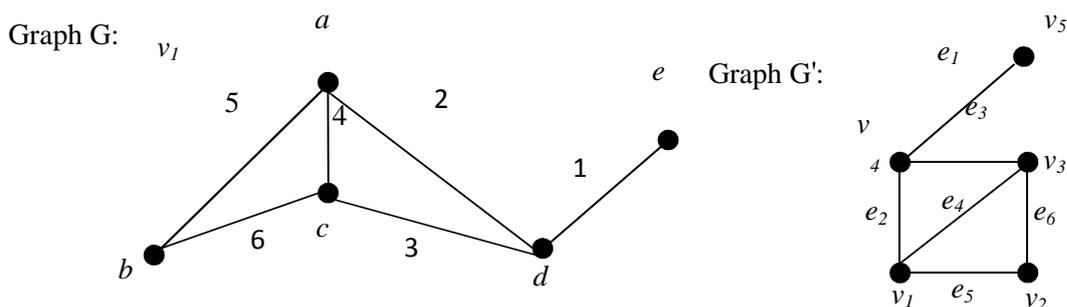
The cycle C_n , $n \geq 3$, consists of n vertices v_1, v_2, \dots, v_n and edges $\{v_1, v_2\}, \{v_2, v_3\}, \dots, \{v_{n-1}, v_n\}$, and $\{v_n, v_1\}$.

The cycles c_3, c_4 and c_5 are shown in the following figures



12. Define Isomorphism.

Two graphs G and G' are said to be **isomorphic** to each other if there is a one-to-one correspondence between their vertices and between their edges such that the incidence relationship is preserved.



Correspondence of vertices

- $f(a) = v_1$
- $f(b) = v_2$
- $f(c) = v_3$
- $f(d) = v_4$
- $f(e) = v_5$

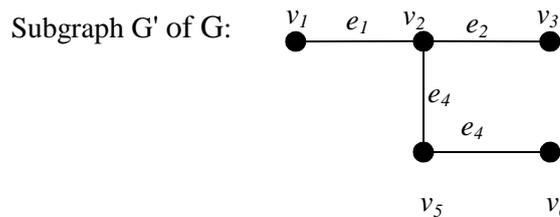
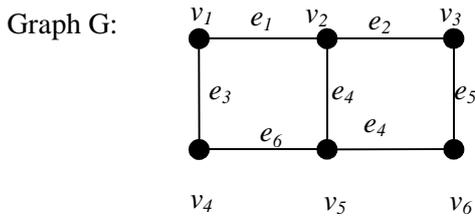
Correspondence of edges

- $f(1) = e_1$
- $f(2) = e_2$
- $f(3) = e_3$
- $f(4) = e_4$
- $f(5) = e_5$

Adjacency also preserved. Therefore G and G' are said to be isomorphic.

13. What is Subgraph?

A graph G' is said to be a subgraph of a graph G, if all the vertices and all the edges of G' are in G, and each edge of G' has the same end vertices in G' as in G.

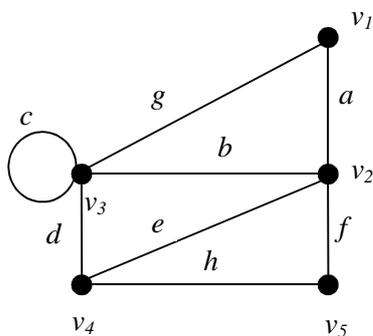


14. Define Walk, Path and Circuit.

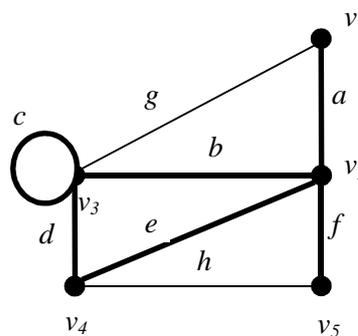
A **walk** is defined as a finite alternating sequence of vertices and edges, beginning and ending with vertices. No edge appears more than once. It is also called as an edge train or a chain.

An open walk in which no vertex appears more than once is called **path**. The number of edges in the path is called **length of a path**.

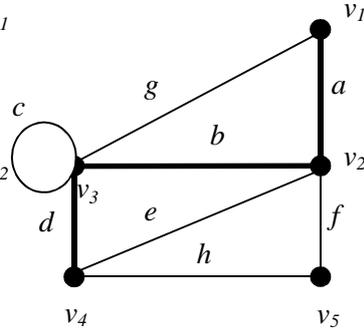
A closed walk in which no vertex (except initial and final vertex) appears more than once is called a circuit. That is, a circuit is a closed, nonintersecting walk.



Graph G:



Open walk



Path of length 3

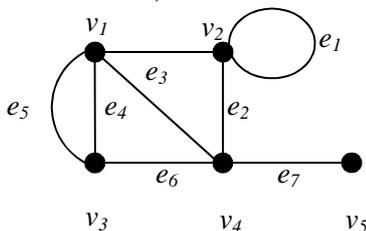
$v_1 a v_2 b v_3 c v_3 d v_4 e v_2 f v_5$ is a walk. v_1 and v_5 are terminals of walk.

$v_1 a v_2 b v_3 d v_4$ is a path. $a v_2 b v_3 c v_3 d v_4 e v_2 f v_5$ is not a path.

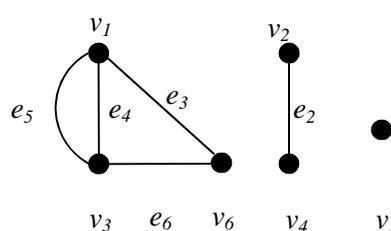
$v_2 b v_3 d v_4 e v_2$ is a circuit.

15. Define connected graph. What is Connectedness?

A graph G is said to be **connected** if there is at least one path between every pair of vertices in G. Otherwise, G is disconnected.



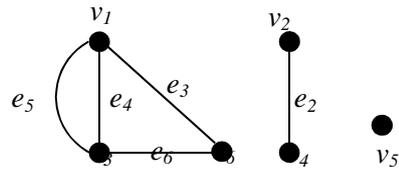
Connected Graph G



Disconnected Graph H

16. Define Components of graph.

A disconnected graph consists of two or more connected graphs. Each of these connected subgraphs is called a component.

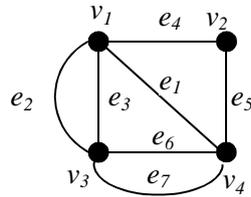


Disconnected Graph H with 3 components

17. Define Euler graph.

A path in a graph G is called Euler path if it includes every edges exactly once. Since the path contains every edge exactly once, it is also called Euler trail / Euler line.

A closed Euler path is called Euler circuit. A graph which contains an Eulerian circuit is called an Eulerian graph.

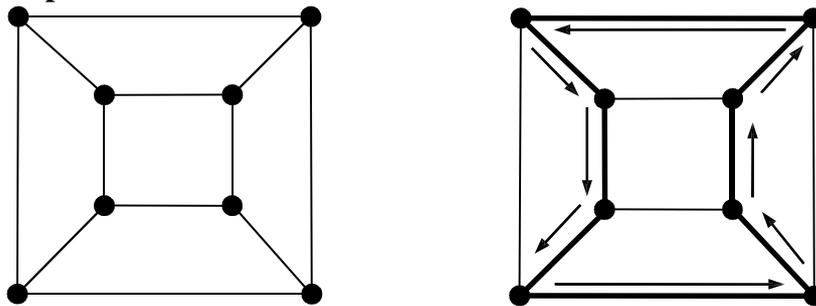


$v_4 e_1 v_1 e_2 v_3 e_3 v_1 e_4 v_2 e_5 v_4 e_6 v_3 e_7 v_4$ is an Euler circuit. So the above graph is Euler graph.

18. Define Hamiltonian circuits and paths

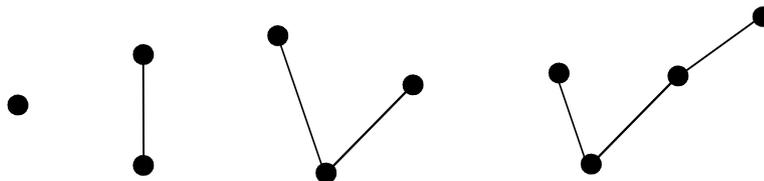
A **Hamiltonian circuit** in a connected graph is defined as a closed walk that traverses every vertex of graph G exactly once except starting and terminal vertex.

Removal of any one edge from a Hamiltonian circuit generates a path. This path is called **Hamiltonian path**.



19. Define Tree

A tree is a connected graph without any circuits. Trees with 1, 2, 3, and 4 vertices are shown in figure.

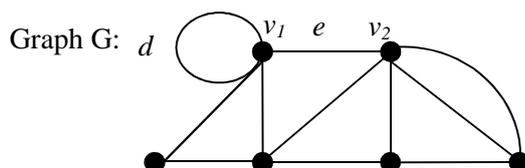


20. List out few Properties of trees.

1. There is one and only one path between every pair of vertices in a tree T.
2. In a graph G there is one and only one path between every pair of vertices, G is a tree.
3. A tree with n vertices has $n-1$ edges.
4. Any connected graph with n vertices has $n-1$ edges is a tree.
5. A graph is a tree if and only if it is minimally connected.
6. A graph G with n vertices has $n-1$ edges and no circuits are connected.

21. What is Distance in a tree?

In a connected graph G, the distance $d(v_i, v_j)$ between two of its vertices v_i and v_j is the length of the shortest path.



a c f h j k
 v_6 b v_3 g v_4 i v_5

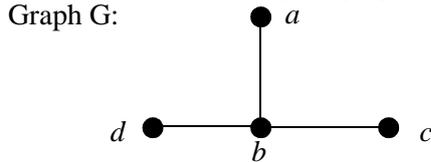
Paths between vertices v_6 and v_2 are (a, e), (a, c, f), (b, c, e), (b, f), (b, g, h), and (b, g, i, k).
 The shortest paths between vertices v_6 and v_2 are (a, e) and (b, f), each of length two.
 Hence $d(v_6, v_2) = 2$

22. Define eccentricity and center.

The eccentricity $E(v)$ of a vertex v in a graph G is the distance from v to the vertex farthest from v in G ; that is,

$$E(v) = \max_{u \in V} (d(v, u))$$

A vertex with minimum eccentricity in graph G is called a center of G



Distance $d(a, b) = 1$, $d(a, c) = 2$, $d(c, b) = 1$, and so on.
 Eccentricity $E(a) = 2$, $E(b) = 1$, $E(c) = 2$, and $E(d) = 2$.
 Center of $G =$ A vertex with minimum eccentricity in graph $G = b$.

23. Define distance metric.

The function $f(x, y)$ of two variables defines the distance between them. These function must satisfy certain requirements. They are

1. Non-negativity: $f(x, y) \geq 0$, and $f(x, y) = 0$ if and only if $x = y$.
2. Symmetry: $f(x, y) = f(y, x)$.
3. Triangle inequality: $f(x, y) \leq f(x, z) + f(z, y)$ for any z .

24. What are the Radius and Diameter in a tree.

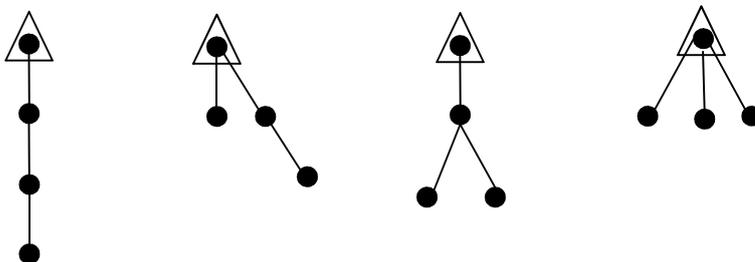
The eccentricity of a center in a tree is defined as the radius of tree.
 The length of the longest path in a tree is called the diameter of tree.

25. Define Rooted tree

A tree in which one vertex (called the root) is distinguished from all the others is called a **rooted tree**.

In general tree means without any root. They are sometimes called as **free trees** (non rooted trees).

The root is enclosed in a small triangle. All rooted trees with four vertices are shown below.



26. Define Rooted binary tree

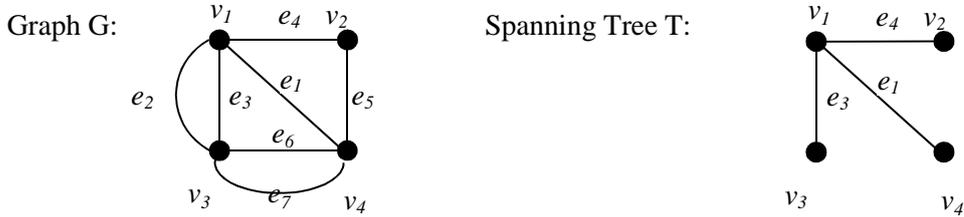
There is exactly one vertex of degree two (root) and each of remaining vertex of degree one or three.

A binary rooted tree is special kind of rooted tree. Thus every binary tree is a rooted tree. A non pendent vertex in a tree is called an internal vertex. Prepared by G. Appasami, Assistant professor, Dr. pauls Engineering College.

UNIT II TREES, CONNECTIVITY & PLANARITY

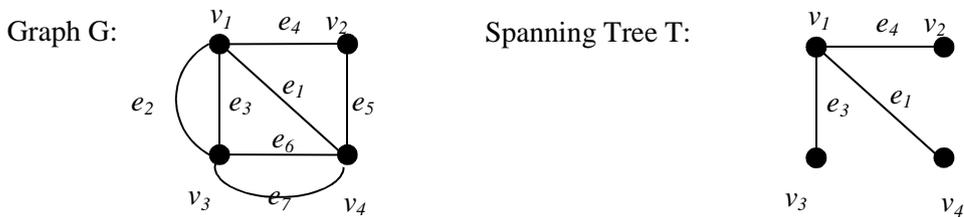
1. Define Spanning trees.

A tree T is said to be a spanning tree of a connected graph G if T is a subgraph of G and T contains all vertices (maximal tree subgraph).



2. Define Branch and chord.

An edge in a spanning tree T is called a *branch* of T . An edge of G is not in a given spanning tree T is called a *chord* (tie or link).

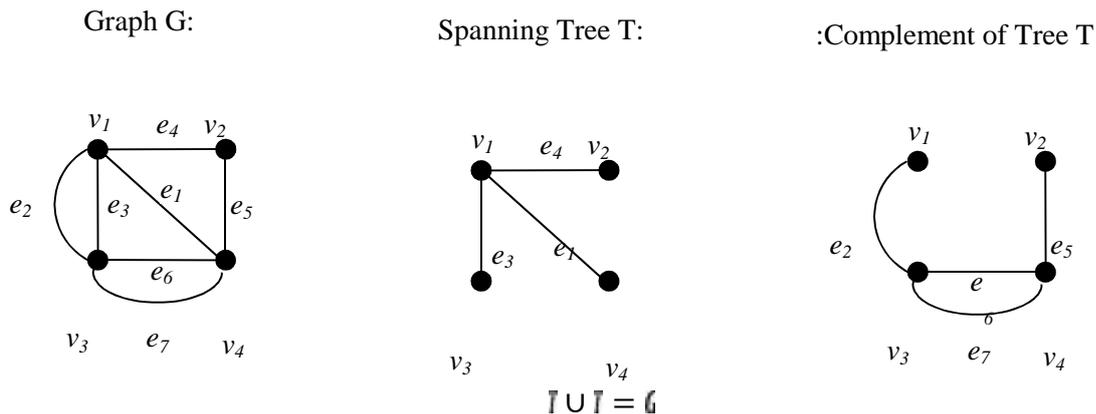


Edge e_1 is a branch of T

Edge e_5 is a chord of T

3. Define complement of tree.

If T is a spanning tree of graph G , then the complement of T of G denoted by \bar{T} is the collection of chords. It also called as *chord set* (tie set or *cotree*) of T



4. Define Rank and Nullity:

A graph G with n number of vertices, e number of edges, and k number of components with the following constraints $n - k \geq 0$ and $e - n + k \geq 0$.

Rank $r = n - k$

Nullity $\mu = e - n + k$ (Nullity also called as *Cyclomatic number* or *first betti number*)

Rank of $G =$ number of branches in any spanning tree of G

Nullity of $G =$ number of chords in G

Rank + Nullity = $e =$ number of edges in G

5. How Fundamental circuits created?

Addition of an edge between any two vertices of a tree creates a circuit. This is because there already exists a path between any two vertices of a tree.

6. Define Spanning trees in a weighted graph

A spanning tree in a graph G is a minimal subgraph connecting all the vertices of G . If G is a weighted graph, then the weight of a spanning tree T of G is defined as the sum of the weights of all the branches in T .

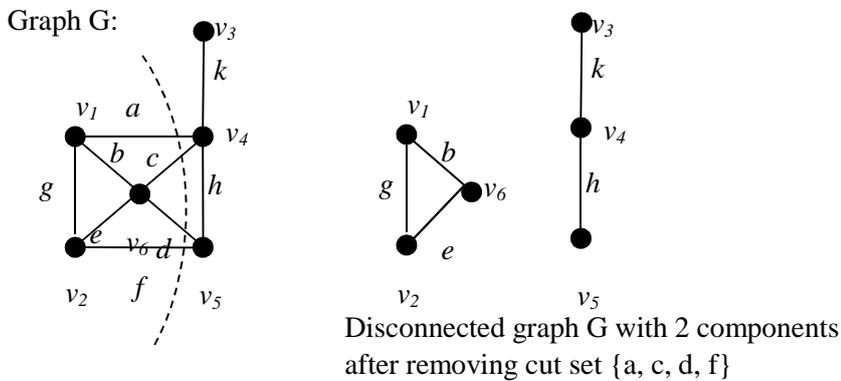
A spanning tree with the smallest weight in a weighted graph is called a *shortest spanning tree* (*shortest-distance spanning tree* or *minimal spanning tree*).

7. Define degree-constrained shortest spanning tree.

A shortest spanning tree T for a weighted connected graph G with a constraint $(d_i) \leq k$ for all vertices in T . for $k=2$, the tree will be Hamiltonian path.

8. Define cut sets and give example.

In a connected graph G , a cut-set is a set of edges whose removal from G leave the graph G disconnected.



Possible cut sets are $\{a, c, d, f\}$, $\{a, b, e, f\}$, $\{a, b, g\}$, $\{d, h, f\}$, $\{k\}$, and so on.
 $\{a, c, h, d\}$ is not a cut set, because its proper subset $\{a, c, h\}$ is a cut set.
 $\{g, h\}$ is not a cut set.

A minimal set of edges in a connected graph whose removal reduces the rank by one is called minimal cut set (simple cut-set or cocycle). Every edge of a tree is a cut set.

9. Write the Properties of cut set

- Every cut-set in a connected graph G must contain at least one branch of every spanning tree of G .
- In a connected graph G , any minimal set of edges containing at least one branch of every spanning tree of G is a cut-set.
- Every circuit has an even number of edges in common with any cut set.

10. Define Fundamental circuits

Adding just one edge to a spanning tree will create a cycle; such a cycle is called a **fundamental cycle (Fundamental circuits)**. There is a distinct fundamental cycle for each edge; thus, there is a one-to-one correspondence between fundamental cycles and edges not in the spanning tree. For a connected graph with V vertices, any spanning tree will have $V - 1$ edges, and thus, a graph of E edges and one of its spanning trees will have $E - V + 1$ fundamental cycles.

11. Define Fundamental cut sets

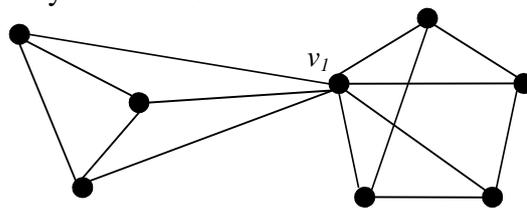
Dual to the notion of a fundamental cycle is the notion of a **fundamental cutset**. By deleting just one edge of the spanning tree, the vertices are partitioned into two disjoint sets. The fundamental cutset is defined as the set of edges that must be removed from the graph G to accomplish the same partition. Thus, each spanning tree defines a set of $V - 1$ fundamental cutsets, one for each edge of the spanning tree.

12. Define edge Connectivity.

Each cut-set of a connected graph G consists of certain number of edges. The number of edges in the smallest cut-set is defined as the **edge Connectivity of G** .

The **edge Connectivity** of a connected graph G is defined as the minimum number of edges whose removal reduces the rank of graph by one.

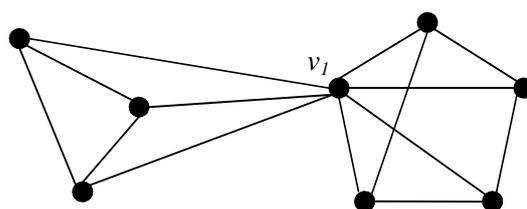
The edge Connectivity of a tree is one.



The edge Connectivity of the above graph G is three.

13. Define vertex Connectivity

The **vertex Connectivity** of a connected graph G is defined as the minimum number of vertices whose removal from G leaves the remaining graph disconnected. The vertex Connectivity of a tree is one.

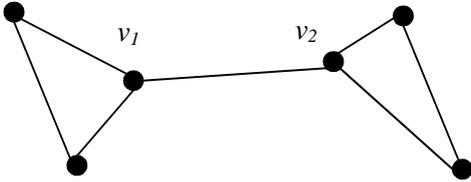


The vertex Connectivity of the above graph G is one.

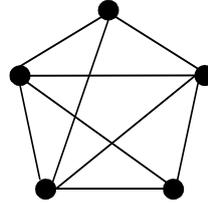
14. Define separable and non-separable graph.

A connected graph is said to be separable graph if its vertex connectivity is one. All other connected graphs are called non-separable graph.

Separable Graph G:

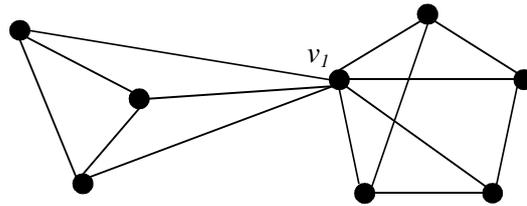


Non-Separable Graph H:



15. Define articulation point.

In a separable graph a vertex whose removal disconnects the graph is called a *cut-vertex*, a *cut-node*, or an *articulation point*.



v_1 is an articulation point.

16. What is Network flows

A **flow network** (also known as a transportation **network**) is a **graph** where each edge has a capacity and each edge receives a **flow**. The amount of **flow** on an edge cannot exceed the capacity of the edge.

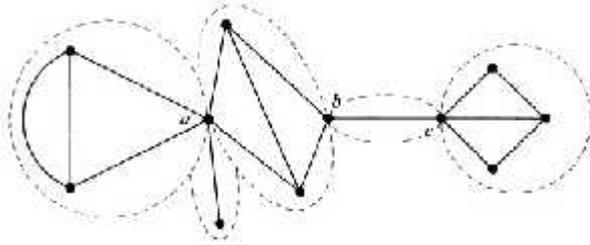
17. Define max-flow and min-cut theorem (equation).

The maximum flow between two vertices a and b in a flow network is equal to the minimum of the capacities of all cut-sets with respect t

The max. flow between two vertices = Min. of the capacities of all cut-sets.

18. Define component (or block) of graph.

A separable graph consists of two or more non separable subgraphs. Each of the largest nonseparable is called a block (or component).



The above graph has 5 blocks.

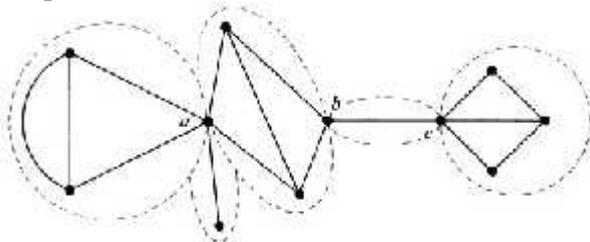
19. Define 1-Isomorphism

A graph G_1 was 1-Isomorphic to graph G_2 if the blocks of G_1 were isomorphic to the blocks of G_2 .

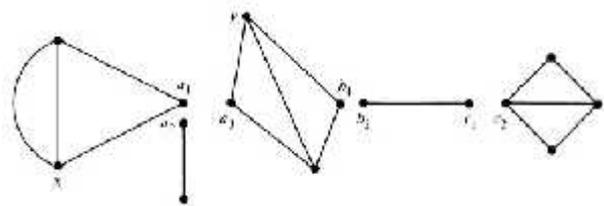
Two graphs G_1 and G_2 are said to be 1-Isomorphic if they become isomorphic to each other under repeated application of the following operation.

Operation 1: "Split" a cut-vertex into two vertices to produce two disjoint subgraphs.

Graph G_1 :



Graph G_2 :



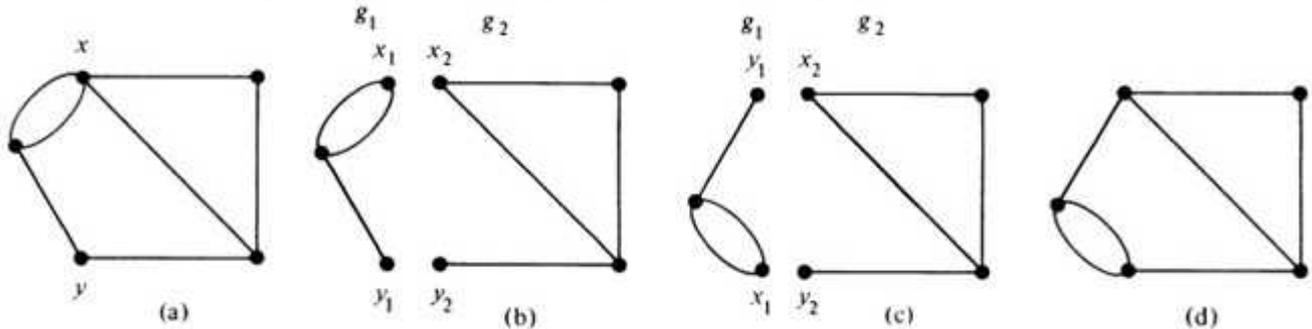
Graph G_1 is 1-Isomorphism with Graph G_2 .

20. Define 2-Isomorphism

Two graphs G_1 and G_2 are said to be **2-Isomorphic** if they become isomorphic after undergoing *operation 1* or *operation 2*, or both operations any number of times.

Operation 1: "Split" a cut-vertex into two vertices to produce two disjoint subgraphs.

Operation 2: "Split" the vertex x into x_1 and x_2 and the vertex y into y_1 and y_2 such that G is split into g_1 and g_2 . Let vertices x_1 and y_1 go with g_1 and vertices x_2 and y_2 go with g_2 . Now rejoin the graphs g_1 and g_2 by merging x_1 with y_2 and x_2 with y_1 .



21. Briefly explain Combinational and geometric graphs

An abstract graph G can be defined as $G = (V, E)$

Where the set V consists of five objects named $a, b, c, d,$ and e , that is, $V = \{ a, b, c, d, e \}$ and the set E consist of seven objects named 1, 2, 3, 4, 5, 6, and 7, that is, $E = \{ 1, 2, 3, 4, 5, 6, 7 \}$, and the

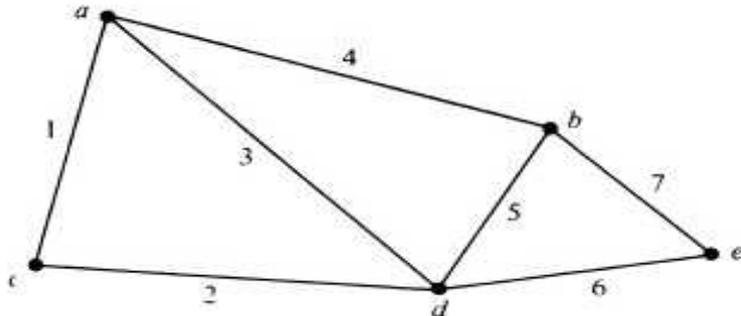
relationship between the two sets is defined by the mapping , which consist of

Type your text

$$= [1 \rightarrow (a, c), 2 \rightarrow (c, d), 3 \rightarrow (a, d), 4 \rightarrow (a, b), 5 \rightarrow (b, d), 6 \rightarrow (d, e), 7 \rightarrow (b, e)].$$

Here the symbol $1 \rightarrow (a, c)$, says that object 1 from set E is mapped onto the pair (a, c) of objects from set V.

This combinatorial abstract object G can also be represented by means of a geometric figure.



The figure is one such geometric representation of this graph G.

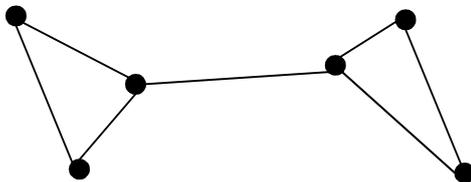
Any graph can be geometrically represented by means of such configuration in three dimensional Euclidian space.

22. Distinguish between Planar and non-planar graphs

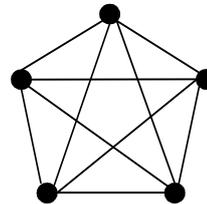
A graph G is said to be *planar* if there exists some geometric representation of G which can be drawn on a plan such that no two of its edges intersect.

A graph that cannot be drawn on a plan without crossover its edges is called *non-planar*.

Planar Graph G:



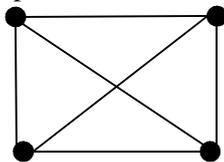
Non-planar Graph H:



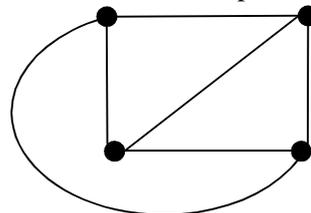
23. Define embedding graph.

A drawing of a geometric representation of a graph on any surface such that no edges intersect is called embedding.

Graph G:

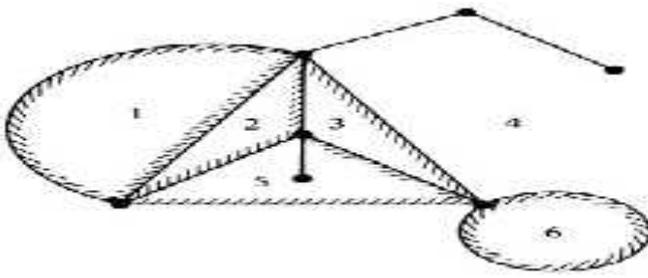


Embedded Graph G:



24. Define region in graph.

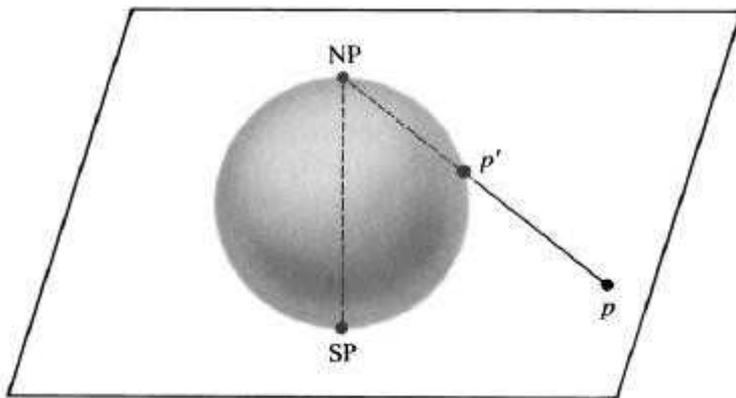
In any planar graph, drawn with no intersections, the edges divide the planes into different **regions (windows, faces, or meshes)**. The regions enclosed by the planar graph are called **interior faces** of the graph. The region surrounding the planar graph is called the **exterior** (or infinite or unbounded) face of the graph. Prepared by G. Appasami, Assistant professor, Dr. pauls Engineering College.



The graph has 6 regions.

25. Why the graph is embedding on sphere.

To eliminate the distinction between finite and infinite regions, a planar graph is often embedded in the surface of sphere. This is done by stereographic projection.



UNIT V GENERATING FUNCTIONS

Define Generating function.

A **generating function** describes an infinite sequence of numbers (a_n) by treating them like the coefficients of a series expansion. The sum of this infinite series is the generating function. Unlike an ordinary series, this formal series is allowed to diverge, meaning that the generating function is not always a true function and the "variable" is actually an indeterminate.

The generating function for $1, 1, 1, 1, 1, 1, 1, 1, 1, \dots$, whose ordinary generating function is

$$\sum_{n=0}^{\infty} x^n = \frac{1}{1-x}$$

The generating function for the geometric sequence $1, a, a^2, a^3, \dots$ for any constant a :

$$\sum_{n=0}^{\infty} a^n x^n = \frac{1}{1-ax}$$

What is Partitions of integer?

Partitioning a positive n into positive summands and seeking the number of such partitions without regard to order is called Partitions of integer.

This number is denoted by $p(n)$. For example

$$P(1) = 1: \quad 1$$

$$P(2) = 2: \quad 2 = 1 + 1$$

$$P(3) = 3: \quad 3 = 2 + 1 = 1 + 1 + 1$$

$$P(4) = 5: \quad 4 = 3 + 1 = 2 + 2 = 2 + 1 + 1 = 1 + 1 + 1 + 1$$

$$P(5) = 7: \quad 5 = 4 + 1 = 3 + 2 = 3 + 1 + 1 = 2 + 2 + 1 = 2 + 1 + 1 + 1 = 1 + 1 + 1 + 1 + 1$$

Define Exponential generating function

For a sequence $a_0, a_1, a_2, a_3, \dots$ of real numbers.

$$E(x) = a_0 + a_1 x + \frac{a_2 x^2}{2!} + \frac{a_3 x^3}{3!} + \dots = \sum_{n=0}^{\infty} \frac{a_n x^n}{n!}$$

is called the exponential generating function for the given sequence.

Define Maclaurin series expansion of e^x and e^{-x} .

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \dots$$

$$e^{-x} = 1 - x + \frac{x^2}{2!} - \frac{x^3}{3!} + \frac{x^4}{4!} - \dots$$

Adding these two series together, we get,

$$e^x + e^{-x} = 2 \left(1 + \frac{x^2}{2!} + \frac{x^4}{4!} + \dots \right)$$

Define Summation operator

$$\sum_{n=0}^{\infty} \frac{x^{2n}}{(2n)!} + \dots$$

Generating function for a sequence $a_0, a_0 + a_1, a_0 + a_1 + a_2, a_0 + a_1 + a_2 + a_3, \dots$

For $() = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + \dots$, consider the function $f(x)/(1-x)$

$$\frac{()}{1-x} = \sum_{n=0}^{\infty} a_n x^n \cdot \sum_{k=0}^{\infty} x^k = \sum_{n=0}^{\infty} (a_0 + a_1 + \dots + a_n) x^n$$

$$= a_0 + (a_0 + a_1)x + (a_0 + a_1 + a_2)x^2 + \dots + (a_0 + a_1 + a_2 + a_3)x^3 + \dots$$
 So $f(x)/(1-x)$ generates the sequence of sums $a_0, a_0 + a_1, a_0 + a_1 + a_2, a_0 + a_1 + a_2 + a_3, \dots$.
 $1/(1-x)$ is called the summation operator.

Recurrence relations

A **recurrence relation** is an equation that recursively defines a sequence or multidimensional array of values, once one or more initial terms are given: each further term of the sequence or array is defined as a function of the preceding terms.

The term **difference equation** sometimes (and for the purposes of this article) refers to a specific type of recurrence relation. However, "difference equation" is frequently used to refer to *any* recurrence relation.

Fibonacci numbers

The recurrence satisfied by the Fibonacci numbers is the archetype of a homogeneous linear recurrence relation with constant coefficients (see below). The Fibonacci sequence is defined using the recurrence

$$F_n = F_{n-1} + F_{n-2}$$

with seed values $F_0 = 0$ and $F_1 = 1$

We obtain the sequence of Fibonacci numbers, which begins

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, ...

First order linear recurrence relation

The general form of First order linear homogeneous recurrence relation can be written as

$a_{n+1} = d a_n$, $n \geq 0$, where d is a constant. The relation is first order since a_{n+1} depends on a_n .

a_0 or a_1 are called boundary conditions.

Second order recurrence relation

Let $k \in \mathbf{Z}^+$ and $C_0 (\neq 0)$, C_1 , $C_2, \dots, C_k (\neq 0)$ be real numbers. If a_n , for $n \geq 0$, is a discrete function, then

$$C_0 a_n + C_1 a_{n-1} + C_2 a_{n-2} + \dots + C_k a_{n-k} = f(n), \quad n \geq k,$$

is a linear recurrence relation (with constant coefficients) of order k . When $f(n) = 0$ for all $n \geq 0$, the relation is called *homogeneous*; otherwise, it is called *nonhomogeneous*.

In this section we shall concentrate on the homogeneous relation of order two:

$$C_0 a_n + C_1 a_{n-1} + C_2 a_{n-2} = 0, \quad n \geq 2.$$

On the basis of our work in Section 10.1, we seek a solution of the form $a_n = cr^n$, where $c \neq 0$ and $r \neq 0$.

Substituting $a_n = cr^n$ into $C_0 a_n + C_1 a_{n-1} + C_2 a_{n-2} = 0$, we obtain

$$C_0 cr^n + C_1 cr^{n-1} + C_2 cr^{n-2} = 0.$$

With $c, r \neq 0$, this becomes $C_0 r^2 + C_1 r + C_2 = 0$, a quadratic equation which is called the *characteristic equation*. The roots r_1, r_2 of this equation determine the following three cases: (a) r_1, r_2 are distinct real numbers; (b) r_1, r_2 form a complex conjugate pair; or (c) r_1, r_2 are real, but $r_1 = r_2$. In all cases, r_1 and r_2 are called the *characteristic roots*.

Non-homogeneous recurrence relations

We now turn to the recurrence relations

$$a_n + C_1 a_{n-1} = f(n), \quad n \geq 1, \quad (1)$$

$$a_n + C_1 a_{n-1} + C_2 a_{n-2} = f(n), \quad n \geq 2, \quad (2)$$

where C_1 and C_2 are constants, $C_1 \neq 0$ in Eq. (1), $C_2 \neq 0$, and $f(n)$ is not identically 0. Although there is no general method for solving all nonhomogeneous relations, for certain functions $f(n)$ we shall find a successful technique.

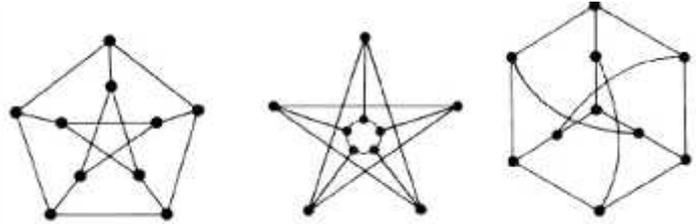
We start with the special case for Eq. (1), when $C_1 = -1$. For the nonhomogeneous relation $a_n - a_{n-1} = f(n)$, we have

$$\begin{aligned} a_1 &= a_0 + f(1) \\ a_2 &= a_1 + f(2) = a_0 + f(1) + f(2) \\ a_3 &= a_2 + f(3) = a_0 + f(1) + f(2) + f(3) \\ &\vdots \\ a_n &= a_{n-1} + f(n) = a_0 + f(1) + \cdots + f(n) = a_0 + \sum_{i=1}^n f(i). \end{aligned}$$

We can solve this type of relation in terms of n , if we can find a suitable summation formula for $\sum_{i=1}^n f(i)$.

CS6702 GRAPH THEORY AND APPLICATIONS
QUESTION BANK
UNIT 1 INTRODUCTION
PART – B

1. Explain various applications of graph.
2. Define the following kn, cn, kn,n, dn, trail, walk, path, circuit with an example.
3. Show that a connected graph G is an Euler graph iff all vertices are even degree.
4. Prove that a simple graph with n vertices and k components can have at most $(n-k)(n-k+1)/2$ edges.



5. Are they isomorphic?
 6. Prove that in a complete graph with n vertices there are $(n-1)/2$ edges-disjoint Hamiltonian circuits, if n is odd number ≥ 3 .
 7. Prove that, there is one and only one path between every pair of vertices in a tree T.
 8. Prove the given statement, “A tree with n vertices has n-1 edges”.
 9. Prove that, any connected graph with n vertices has n-1 edges is a tree.
 10. Show that a graph is a tree if and only if it is minimally connected.
 11. Prove that, a graph G with n vertices has n-1 edges and no circuits are connected.
-

UNIT II TREES, CONNECTIVITY & PLANARITY

PART – B

1. Find the shortest spanning tree for the following graph.

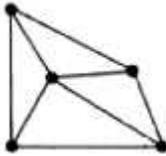
	v_1	v_2	v_3	v_4	v_5	v_6
v_1	–	10	16	11	10	17
v_2	10	–	9.5	∞	∞	19.5
v_3	16	9.5	–	7	∞	12
v_4	11	∞	7	–	8	7
v_5	10	∞	∞	8	–	9
v_6	17	19.5	12	7	9	–

2. Explain 1 - isomorphism and 2 - isomorphism of graphs with your own example.
 3. Prove that a connected graph G with n vertices and e edges has $e-n+2$ regions.
 4. Write all possible spanning tree for K_5 .
 5. Prove that every cut-set in a connected graph G must contain at least one branch of every spanning tree of G.
 6. Prove that the every circuit which has even number of edges in common with any cut-set.
 7. Show that the ring sum of any two cut-sets in a graph is either a third cut set or an edge disjoint union of cut sets.
 8. Explain network flow problem in detail.
 9. If G_1 and G_2 are two 1-isomorphic graphs, the rank of G_1 equals the rank of G_2 and the nullity of G_1 equals the nullity of G_2 , prove this.
 10. Prove that any two graphs are 2-isomorphic if and only if they have circuit correspondence.
-

UNIT III MATRICES, COLOURING AND DIRECTED GRAPH

PART – B

42. Prove that any simple planar graph can be embedded in a plane such that every edge is drawn as a straight line.
43. Show that a connected planar graph with n vertices and e edges has $e-n+2$ regions.
3. Define chromatic polynomial. Find the chromatic polynomial for the following graph.



4. Explain matching and bipartite graph in detail.
 5. Write the observations of minimal covering of a graph.
 6. Prove that the vertices of every planar graph can be properly colored with five colors.
 7. Explain matching in detail.
 8. Prove that a covering g of graph G is minimal iff g contains no path of length three or more.
 9. Illustrate four-color problem.
 10. Explain Euler digraphs in detail.
-

UNIT IV PERMUTATIONS & COMBINATIONS

PART – B

1. Explain the Fundamental principles of counting.
 2. Find the number of ways of ways of arranging the word APPASAMIAP and out of it how many arrangements have all A's together.
 3. Discuss the rules of sum and product with example.
 4. Determine the number of (staircase) paths in the xy -plane from $(2, 1)$ to $(7, 4)$, where each path is made up of individual steps going 1 unit to the right (R) or one unit upward (U).iv.
Find the coefficient of a^5b^2 in the expansion of $(2a - 3b)^7$.
 5. State and prove binomial theorem.
 6. How many times the print statement executed in this program segment?
 7. Discuss the Principle of inclusion and exclusion.
 8. How many integers between 1 and 300 (inc.) are not divisible by at least one of 5, 6, 8?
 9. How 32 bit processors address the content? How many address are possible?
 10. Explain the Arrangements with forbidden positions.
-

```

for i := 1 to 20 do
  for j := 1 to i do
    for k := 1 to j do
      print (i * j * k)

```

UNIT V GENERATING FUNCTIONS

PART – B

1. Explain Generating functions
 2. Find the convolution of the sequences $1, 1, 1, 1, \dots$ and $1, -1, 1, -1, 1, -1$.
 3. Find the number of non negative & positive integer solutions of for $x_1+x_2+x_3+x_4=25$.
 4. Find the coefficient of x^5 in $(1-2x)^7$.
 5. The number of virus affected files in a system is 1000 and increases 250% every 2 hours.
 6. Explain Partitions of integers
 7. Use a recurrence relation to find the number of viruses after one day.
 8. Explain First order homogeneous recurrence relations.
 9. Solve the recurrence relation $a_{n+2}-4a_{n+1}+3a_n=-200$ with $a_0=3000$ and $a_1=3300$.
 10. Solve the Fibonacci relation $F_n = F_{n-1}+F_{n-2}$.
 11. Find the recurrence relation from the sequence $0, 2, 6, 12, 20, 30, 42, \dots$.
 12. Determine $(1+3i)^{10}$.
 13. Discuss Method of generating functions.
-

CS6703 GRID AND CLOUD COMPUTING

UNIT 1 QUESTION BANK

Part - A

1. What is Grid Computing?

Grid computing is a processor architecture that combines computer resources from various domains to reach a main objective. In **grid computing**, the **computers** on the network can work on a task together, thus functioning as a supercomputer.

2. What is QOS?

Grid computing system is the ability to provide the quality of service requirements necessary for the end-user community. QOS provided by the grid like performance, availability, management aspects, business value and flexibility in pricing.

3. What are the derivatives of grid computing?

There are 8 derivatives of grid computing. They are as follows:

- a) Compute grid
- b) Data grid
- c) Science grid
- d) Access grid
- e) Knowledge grid
- f) Cluster grid
- g) Terra grid
- h) Commodity grid

4. What are the features of data grids?

The ability to integrate multiple distributed, heterogeneous and independently managed data sources.

The ability to provide data catching and/or replication mechanisms to minimize network traffic.

The ability to provide necessary data discovery mechanisms, which allow the user to find data based on characteristics of the data.

5. Define – Cloud Computing.

Cloud computing, often referred to as simply “the cloud,” is the delivery of on-demand computing resources—everything from applications to data centers—over the Internet on a pay-for-use basis. Storing and accessing data and programs over the Internet instead of your computer's hard drive

6. What is business on demand?

Business On Demand is not just about utility computing as it has a much broader set of ideas about the transformation of business practices, process transformation, and technology implementations.

The essential characteristics of on-demand businesses are responsiveness to the dynamics of business, adapting to variable cost structures, focusing on core business



competency, and resiliency for consistent availability.

7. What are the facilities provided by virtual organization?

The formation of virtual task forces, or groups, to solve specific problems associated with the virtual organization.

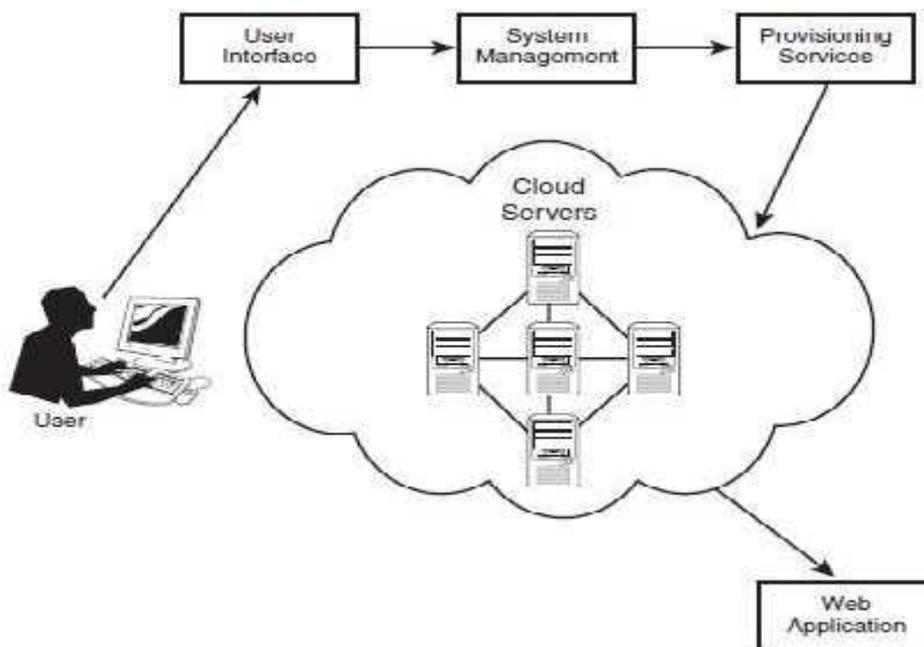
The dynamic provisioning and management capabilities of the resource required meeting the SLA's.

8. What are the properties of Cloud Computing?

There are six key properties of cloud computing: Cloud computing is

- x user-centric
- x task-centric
- x powerful
- x accessible
- x intelligent
- x programmable

9. Sketch the architecture of Cloud.



10. What are the types of Cloud service development?

- x Software as a Service
 - x Platform as a Service
 - x Web Services
 - x On-Demand Computing
-

11. What is meant by scheduler?

Schedulers are types of applications responsible for the management of jobs, such as allocating resources needed for any specific job, partitioning of jobs to schedule parallel execution of tasks, data management, event correlation, and service-level management capabilities.

12. What is meant by resource broker?

Resource broker provides pairing services between the service requester and the service provider. This pairing enables the selection of best available resources from the service provider for the execution of a specific task.

13. What is load balancing?

Load balancing is concerned with the integrating the system in order to avoid processing delays and over-commitment of resources. It involves partitioning of jobs, identifying the resources and queuing the jobs.

14. What is grid infrastructure?

Grid infrastructure forms the core foundation for successful grid applications. This infrastructure is a complex combination of number of capabilities and resources identified for the specific problem and environment being addressed.

15. Define – Distributed Computing.

Distributed computing is a field of **computer** science that studies **distributed** systems. A **distributed** system is a software system in which components located on networked **computers** communicate and coordinate their actions by passing messages. The components interact with each other in order to achieve a common goal.

PART – B

- 1) Explain in detail about virtual organization. (16)
 - 2) Write about the scope of grid computing in business areas. (16)
 - 3) Explain some of the grid application and their usage patterns. (16)
 - 4) Write short notes on. (16)
 - a) Schedulers
 - b) Resource broker
 - c) Load balancing
 - d) Grid portals
 - 5) What are the data and functional requirements of grid computing? (16)
 - 6) Explain briefly about grid infrastructure. (16)
 - 7) Describe in detail about the Technologies for network based systems? (16)
-

GRID AND CLOUD COMPUTING

UNIT 2 QUESTION BANK

Part - A

1. Define – OSGI.

Open Grid Services Architecture (OGSA) is a set of standards defining the way in which information is shared among diverse components of large, heterogeneous grid systems. In this context, a grid system is a scalable wide area network (WAN) that supports resource sharing and distribution. OGSA is a trademark of the Open Grid Forum.

2. Define – OSGA.

The **Open Grid Services Infrastructure** (OGSI) was published by the Global Grid Forum (GGF) as a proposed recommendation in June 2003.^[1] It was intended to provide an infrastructure layer for the Open Grid Services Architecture (OGSA). OGSI takes the statelessness issues (along with others) into account by essentially extending Web services to accommodate grid computing resources that are both transient and stateful.

3. Define – Peer to Peer Computing.

Peer to Peer computing is a relatively new computing discipline in the realm of distributed computing. P2P system defines collaboration among a larger number of individuals and/or organizations, with a limited set of security requirements and a less complex resource-sharing topology.

4. What is Dynamic Accounting System?

DAS provides the following enhanced categories of accounting functionality to the IPG community:

- Allows a grid user to request access to a local resource via the presentation of grid credentials

- Determines and grants the appropriate authorizations for a user to access a local resource without requiring a preexisting account on the resource to govern local authorizations.

4. Define – SOA.

A service-oriented architecture is intended to define loosely coupled and interoperable services/applications, and to define a process for integrating these interoperable components.

6. What are the major goals of OSGA?

- 2 Identify the use cases that can drive the OGSA platform components.
- 3 Identify and define the core OGSA platform components.
- 4 Define hosting and platform specific bindings.
- 5 Define resource models and resource profiles with interoperable solutions.



7. What are the layers available in OGSA architectural organizations?

- a Native platform services and transport mechanisms.
- b OGSA hosting environment.
- c OGSA transport and security.
- d OGSA infrastructure (OGSI).
- e OGSA basic services (meta-OS and domain services)

5. What is meant by grid infrastructure?

Grid infrastructure is a complex combination of a number of capabilities and resources identified for the specific problem and environment being addressed. It forms the core foundations for successful grid applications.

10. List some grid computing toolkits and frameworks?

- x Globus Toolkit
 - i Globus Resource Allocation Manager(GRAM)
 - ii Grid Security Infrastructure(GSI)
 - iii Information Services
 - iv Legion
 - v Condor and Condor-G
 - vi NIMROD
 - vii UNICORE
 - viii NMI

11. Define - GRAM.

GRAM provides resource allocation, process creation, monitoring, and management services. The most common use of GRAM is the remote job submission and control facility. GRAM simplifies the use of remote systems.

11. What is the role of the grid computing organization?

- x Organizations developing grid standards and best practices guidelines.
 - x Organizations developing grid computing toolkits, frameworks and middleware solutions.
 - x Organizations building and using grid - based solutions to solve their computing, data, and network requirements.
 - x Organizations working to adopt grid concepts into commercial products, via utility computing and business on demand computing.
-

12. What are the different layers of grid architecture?

Fabric Layer: Interface to local resources
Connectivity Layer: Manages Communications
Collective Layer: Coordinating Multiple Resources
Application Layer: User Defined Application.

4) What are the fundamental components of SOAP specification?

An envelope that defines a framework for describing message structure.
A set of encoding rules for expressing instances of application defined data types
A convention for representing remote procedure (RPC) and responses.
A set of rules for using SOAP with HTTP.
Message exchange patterns (MEP) such as request-response, one-way and peer-to-peer conversations.

5) Define - SOAP.

SOAP is a simple and lightweight XML-based mechanism for creating structured data packages that can be exchanged between network applications. SOAP provides a simple enveloping mechanism and is proven in being able to work with existing networking services technologies such as HHTTP.SOAP is also flexible and extensible. SOAP is based on the fact that it builds upon the XML info set.

15. Define WSDL.

WSDL is an XML Info set based document, which provides a model and XML format for describe web services. This enables services to be described and enables the client to consume these services in a standard way without knowing much on the lower level protocol exchange binding including SOAP and HTTP. This high level abstraction on the service limits human interaction and enables the automatic generation of proxies for web services, and these proxies can be static or dynamic. It allows both document and RPC - oriented messages.

PART – B

- 8) Write short notes on Open Grid Service Architecture. (16)
 - 9) Explain in detail, the functional requirements of OGSA. (16)
 - 10) Explain Practical & Detailed view of OGSA/OGSI. (16)
 - 11) Explain in detail, OGSA services.(16)
 - 12) Describe about the relation of grid architecture with other distributed technologies.(16)
 - 13) What are the third generation initiatives of grid computing?
 - 14) Discuss briefly about organization building and using grid based solution to solve their computing data and network requirements.
-

GRID AND CLOUD COMPUTING

UNIT 3

Part - A

1. What is the working principle of Cloud Computing?

The cloud is a collection of computers and servers that are publicly accessible via the Internet. This hardware is typically owned and operated by a third party on a consolidated basis in one or more data center locations. The machines can run any combination of operating systems.

2. What is Virtualization?

Virtualization is a foundational element of **cloud computing** and helps deliver on the value of **cloud computing**," Adams said. "**Cloud computing** is the delivery of shared **computing** resources, software or data — as a service and on-demand through the Internet.

3. Define Cloud services with example.

Any web-based application or service offered via cloud computing is called a cloud service. Cloud services can include anything from calendar and contact applications to word processing and presentations.

4. What are the types of Cloud service development?

- x Software as a Service
- x Platform as a Service
- x Infrastructure as a Service

5. Explain cloud provider and cloud broker?

Cloud Provider: Is a company that offers some component of cloud computing typically infrastructure as a service, software as a Service or Platform as a Service. It is something referred as CSP.

Cloud Broker: It is a third party individual or business that act as an intermediary between the purchase of cloud computing service and sellers of that service.

6. Define - Private Cloud.

The *private cloud* is built within the domain of an intranet owned by a single organization. Therefore, they are client owned and managed. Their access is limited to the owning clients and their partners. Their deployment was not meant to sell capacity over the Internet through publicly accessible interfaces. Private clouds give local users a flexible and agile private infrastructure to run service workloads within their administrative domains.

7. Define - Public Cloud.

A *public cloud* is built over the Internet, which can be accessed by any user who has paid for the service. Public clouds are owned by service providers. They are accessed by subscription. Many

companies have built public clouds, namely Google App Engine, Amazon AWS, Microsoft Azure, IBM Blue Cloud, and Salesforce Force.com. These are commercial providers that offer a publicly accessible remote interface for creating and managing VM instances within their proprietary infrastructure.

8. Define - Hybrid Cloud.

A *hybrid cloud* is built with both public and private clouds; Private clouds can also support a *hybrid cloud* model by supplementing local infrastructure with computing capacity from an external public cloud. For example, the *research compute cloud (RC2)* is a private cloud built by IBM.

9. Define anything-as-a-service?

Providing services to the client on the basis on meeting their demands at some pay per use cost such as data storage as a service, network as a service, communication as a service etc. It is generally denoted as anything as a service (XaaS).

10. What is mean by SaaS?

The software as a service refers to browser initiated application software over thousands of paid customer. The SaaS model applies to business process industry application, consumer relationship management (CRM), Enterprise resource Planning (ERP), Human Resources (HR) and collaborative application.

11. What is mean by IaaS?

The Infrastructure as a Service model puts together the infrastructure demanded by the user namely servers, storage, network and the data center fabric. The user can deploy and run on multiple VM's running guest OS on specific application.

12. Explain PaaS?

The Platform as a Service model enables the user to deploy user built applications onto a virtualized cloud platform. It includes middleware, database, development tools and some runtime support such as web2.0 and java. It includes both hardware and software integrated with specific programming interface.

5. List out the advantages of Cloud Computing.

- Lower IT Infrastructure Costs
 - Fewer Maintenance Issues
 - Lower Software Costs
 - Instant Software Updates
 - Increased Computing Power
 - Unlimited Storage Capacity
 - Increased Data Safety
-

- a) Improved Compatibility Between Operating Systems
- b) Improved Document Format Compatibility
- c) Easier Group Collaboration
- d) Universal Access to Documents
- e) Latest Version Availability
- f) Removes the Tether to Specific Devices

6 List out the disadvantages of Cloud Computing.

- a) Requires a Constant Internet Connection
- b) Doesn't Work Well with Low-Speed Connections
- c) Can Be Slow
- d) Features Might Be Limited
- e) Stored Data Might Not Be Secure
- f) If the Cloud Loses Your Data, You're Screwed

7 What is Hypervisor?

A **hypervisor** or virtual machine monitor (VMM) is a piece of computer software, firmware or hardware that creates and runs virtual machines. A computer on which a **hypervisor** is running one or more virtual machines is defined as a host machine. Each virtual machine is called a guest machine.

16. What are the types of hypervisor?

There are two types of hypervisors:

- 6. Type 1 (bare-metal)
- 7. Type 2 (hosted)

Type 1 hypervisors run directly on the system hardware. They are often referred to as a "native" or "bare metal" or "embedded" hypervisors in vendor literature.

Type 2 hypervisors run on a host operating system. When the virtualization movement first began to take off, Type 2 hypervisors were most popular. Administrators could buy the software and install it on a server they already had.

PART – B

11. Write short notes on cloud deployment model. (16)
 12. Explain in detail, categories of cloud. (16)
 13. Explain in detail, pros and cons of cloud. (8)
 14. Explain in detail, different implementation level of virtualization? (16)
 15. Write short notes on OS level virtualization. List the pros and cons of OS level virtualization. (16)
 16. Explain in detail, the virtualization of CPU, Memory and I/O devices. (16)
 17. Write short notes on virtual clusters. (8)
 18. Explain in detail, the virtualization for data center automation. (16)
-

GRID AND CLOUD COMPUTING

UNIT 4

Part -A

1. What is The Globus Toolkit Architecture (GT4)

The Globus Toolkit, started in 1995 with funding from DARPA, is an open middleware library for the grid computing communities. The toolkit addresses common problems and issues related to grid resource discovery, management, communication, security, fault detection, and portability. The library includes a rich set of service implementations.

2. What is GT4 library?

The high-level services and tools, such as MPI, Condor-G, and Nirod/G, are developed by third parties for generalpurpose distributed computing applications. The local services, such as LSF, TCP, Linux, and Condor, are at the bottom level and are fundamental tools supplied by other developers.

3. What is meant by Globus Container ?

The Globus Container provides a basic runtime environment for hosting the web services needed to execute grid jobs.

4. What are the Functional Modules in Globus GT4 Library

- ? x Global Resource Allocation Manager
- x Communication
- x Grid Security Infrastructure
- x Monitory and Discovery Service
- x Health and Status
- x Global Access of Secondary Storage
- x Grid File Transfer

5. What is meant by input splitting ?

For the framework to be able to distribute pieces of the job to multiple machines, it needs to fragment the input into individual pieces, which can in turn be provided as input to the individual distributed tasks. Each fragment of input is called an input split.

6. What are the five categories of Globus Toolkit 4 ?

- Common runtime components
- Security
- Data management
- Information services
- Execution management

7. What are the are the available input formats?

- KeyValueTextInputFormat
- TextInputFormant
- NLineInputFormat
- MultiFileInputFormat
- SequenceFileInputFormat

8. What is meant by HDFS ?

Hadoop comes with a distributed filesystem called HDFS, which stands for Hadoop Distributed Filesystem. HDFS is a filesystem designed for storing very large files with streaming data access patterns, running on clusters of commodity hardware.

9. What is meant by Block

A disk has a block size, which is the minimum amount of data that it can read or write. Filesystems for a single disk build on this by dealing with data in blocks, which are an integral multiple of the disk block size. Filesystem blocks are typically a few kilobytes in size, while disk blocks are normally 512 bytes. HDFS, too, has the concept of a block, but it is a much larger unit—64 MB by default.

10. Differentiate Namenodes and Datanodes

An HDFS cluster has two types of node operating in a master-worker pattern: a namenode (the master) and a number of datanodes (workers). The namenode manages the filesystem namespace. It maintains the filesystem tree and the metadata for all the files and directories in the tree. This information is stored persistently on the local disk in the form of two files: the namespace image and the edit log. The namenode also knows the datanodes on which all the blocks for a given file are located.

11. List the various Hadoop filesystems ?

Local, HDFS, HFTP, HSFTP, WebHDFS.

12. What is meant by FUSE?

Filesystem in Userspace (FUSE) allows filesystems that are implemented in user space to be integrated as a Unix filesystem. Hadoop's Fuse-DFS contrib module allows any Hadoop filesystem (but typically HDFS) to be mounted as a standard filesystem.

13. What is Hadoop File system ?

Hadoop is written in Java, and all Hadoop filesystem interactions are mediated through the Java API. The filesystem shell, for example, is a Java application that uses the Java FileSystem class to provide filesystem operations.

14. How to Reading Data from a Hadoop URL

One of the simplest ways to read a file from a Hadoop filesystem is by using a java.net.URL object to open a stream to read the data from. The general idiom is:

```
InputStream in =
null; try {
in = new URL("hdfs://host/path").openStream(); //
process in} finally {
IOUtils.closeStream(in);
}
```

15. How to write data in Hadoop?

The FileSystem class has a number of methods for creating a file. The simplest is the method that takes a Path object for the file to be created and returns an output stream to write to:

```
public FSDataOutputStream create(Path f) throws IOException
```

16. How are Deleting Datas are Deleted in Hadoop ?

Use the delete() method on FileSystem to permanently remove files or directories:

```
public boolean delete(Path f, boolean recursive) throws IOException
```

If f is a file or an empty directory, then the value of recursive is ignored.

17. Illustrate MapReduce logical data flow

18. What are two types of nodes that control the job execution process?

a jobtracker and a number of tasktrackers controls the job execution process. The jobtracker coordinates all the jobs run on the system by scheduling tasks to run on tasktrackers. Tasktrackers run tasks and send progress reports to the jobtracker, which keeps a record of the overall progress of each job. If a task fails, the jobtracker can reschedule it on a different tasktracker.

19. Illustrate MapReduce data flow with a single reduce task

20. Illustrate MapReduce data flow with multiple reduce tasks

Part -B

1. Explain the Globus Toolkit Architecture (GT4)

2. Explain MapReduce Model in detail

3. Explain Map & Reduce function?

4. Explain HDFS Concepts in detail?

5. Explain Anatomy of a File Read?

6. Explain Anatomy of a File write?

UNIT V
PART A

1. List the challenges in building trust management?
2. What are the security requirements of grid?
3. What are the types of message level security?
4. What is IAM?
5. List the components in IAM architecture provider.
6. What is privacy in cloud?
7. List the important tasks in the management of identities in cloud?
8. What is SD?
9. What is TI?
10. List some potential security issues.
11. What is security assurance condition?
12. Give the steps accomplished in fuzzy inference.
13. Which information are taken into account for calculating site trust worthiness?
14. What are the major authenticated methods?
15. Give the category classifications of authority?
16. What is the role of GSI functional layers?
17. What are the additional protection mechanisms of GSI?
18. Give the various levels of security.
19. Name the Cloud security controls.
20. Give some of the data security issues.
21. List the types of PHRs.

PART B

1. Discuss in detail about the various trust models in grids?
2. Write about Authorization and Delegation in Grids?
3. Explain briefly about Grid Security Infrastructure?
4. Explain briefly about the aspects of data security, provider data and security.
5. Describe in detail about the IAM architecture and its practices in cloud.
6. Write about the various key privacy issues in the cloud?

UNIT I: INTRODUCTION (16 marks)

Part-B

1.Explain all the evolutionary changes in the Age of Internet Computing.

The Age of Internet Computing

High-performance computing (HPC)

High-throughput computing (HTC)

The Platform Evolution

Computer technology has gone through five generations of development, with each generation lasting from 10 to 20 years

High-Performance Computing

The speed of HPC systems has increased from Gflops in the early 1990s to now Pflops in 2010.

High-Throughput Computing

This HTC paradigm pays more attention to high-flux computing. The main application for high-flux computing is in Internet searches and web services by millions or more users simultaneously.

Three New Computing Paradigms

The maturity of *radio-frequency identification (RFID)*, *Global Positioning System (GPS)*, and sensor technologies has triggered the development of the

Internet of Things (IoT).

Computing Paradigm Distinctions

The high-technology community has argued for many years about the precise definitions of centralized computing, parallel computing, distributed computing, and cloud computing.

- **Centralized computing** this is a computing paradigm by which all computer resources are centralized in one physical system. All resources (processors, memory, and storage) are fully shared and tightly coupled within one integrated OS. Many data centers and supercomputers are *centralized systems*, but they are used in parallel, distributed, and cloud computing applications

- **Parallel computing** in parallel computing, all processors are either tightly coupled with centralized shared memory or loosely coupled with distributed memory. Some authors refer to this discipline as *parallel processing*. A computer system capable of parallel computing is commonly known as a *parallel computer*. Programs running in a parallel computer are called *parallel programs*. The process of writing *parallel programs* is often referred to as *parallel programming*

- **Distributed computing** a *distributed system* consists of multiple autonomous computers, each having its own private memory, communicating through a computer network. Information exchange in a distributed system is accomplished through *message passing*. A computer program that runs in a distributed system is known as a *distributed program*. The process of writing distributed programs is referred to as *distributed programming*.

- **Cloud computing** An *Internet cloud* of resources can be either a centralized or a distributed computing system. Or *utility computing* or *service computing* *Ubiquitous computing* refers to computing with pervasive devices at any place and time using wired or wireless communication. *The Internet of Things (IoT)* is a networked connection of everyday objects including computers, sensors, humans, etc.

Distributed System Families

The system efficiency is decided by speed, programming, and energy factors Meeting these goals requires to yield the following design objectives:

- **Efficiency** measures the utilization rate of resources in an execution model by exploiting massive parallelism in HPC.

- **Dependability** measures the reliability and self-management from the chip to the system and application levels.

- **Adaptation in the programming model** measures the ability to support billions of job requests over massive data sets and virtualized cloud resources under various workload and service models.

- **Flexibility in application deployment** measures the ability of distributed systems to run well in both HPC and HTC applications.

2. Explain about Scalable Computing Trends and its New Paradigms. What do you mean by the Internet of Things and Cyber-Physical Systems? Discuss.

Scalable Computing Trends and New Paradigms

Moore's law indicates that processor speed doubles every 18 months.

Gilder's law indicates that network bandwidth has doubled each year in the past.

Degrees of Parallelism

bit-level parallelism (BLP) converts bit-serial processing to word-level processing gradually This led us to the next wave known as *instruction-level parallelism (ILP)* *Data-level parallelism (DLP)* was made popular through *SIMD (single instruction, multiple data)* and vector

machines using vector or array types of instructions. From *chip multiprocessors (CMPs)*, we have been exploring *task-level parallelism (TLP)*.

Innovative Applications

Both HPC and HTC systems desire transparency in many application aspects

Applications of High-Performance and High-Throughput Systems

The Trend toward Utility Computing

These paradigms are composable with QoS and *SLAs (service-level agreements)*.

The Hype Cycle of New Technologies

This cycle shows the expectations for the technology at five different stages. The expectations rise sharply from the trigger period to a high peak of inflated expectations.

The Internet of Things and Cyber-Physical Systems

The Internet of Things

Three communication patterns co-exist: namely H2H (human-to-human), H2T (human-to-thing), and T2T (thing-to-thing)

Cyber-Physical Systems

A *cyber-physical system (CPS)* is the result of interaction between computational processes and the physical world.

3. Write in detail about Clusters of Cooperative Computers.

Clusters of Cooperative Computers

Cluster Architecture

The cluster is connected to the Internet via a virtual private network (VPN) gateway. The gateway IP address locates the cluster. Most clusters have loosely coupled node Computers. All resources of a server node are managed by their own OS.

Single-System Image

Cluster designers desire a *cluster operating system* or some middleware to support SSI at various levels, including the sharing of CPUs, memory, and I/O across all cluster nodes.

Hardware, Software, and Middleware Support

Special cluster middleware supports are needed to create SSI or *high availability (HA)*. Both sequential and parallel applications can run on the cluster, and special parallel environments are needed to facilitate use of the cluster resources.

Major Cluster Design Issues

Unfortunately, a cluster-wide OS for complete resource sharing is not available yet.

Critical Cluster Design Issues and Feasible Implementations

Features Functional Characterization Feasible Implementations

4. Explain in detail about Grid Computing Infrastructures. Discuss about Cloud Computing over the Internet.

Grid Computing Infrastructures

Computational Grids

A *computing grid* offers an infrastructure that couples computers, software/middleware, special instruments, and people and sensors together. The grid is often constructed across LAN, WAN, or Internet backbone networks at a regional, national, or global scale. They can also be viewed as *virtual platforms* to support *virtual organizations*.

Grid Families

Grid technology demands new distributed computing models, software/middleware support, network protocols, and hardware infrastructures. National grid projects are followed by industrial grid platform development by IBM, Microsoft, Sun, HP, Dell, Cisco, EMC, Platform Computing, and others.

Cloud Computing Over the Internet

A cloud is a pool of virtualized computer resources. A cloud can host a variety of different workloads, including batch-style backend jobs and interactive and user-facing applications.”

Internet Clouds

Cloud computing leverages its low cost and simplicity to benefit both users and providers.

The Cloud Landscape

- **Infrastructure as a Service (IaaS)** this model puts together infrastructures demanded by users—namely servers, storage, networks, and the data center fabric. The user can deploy and run on multiple VMs running guest OSes on specific applications. The user does not manage or control the underlying cloud infrastructure, but can specify When to request and release the needed resources.

- **Platform as a Service (PaaS)** This model enables the user to deploy user-built applications onto a virtualized cloud platform. PaaS includes middleware, databases, development tools, and some runtime support such as Web 2.0 and Java. The platform Includes both hardware and software integrated with specific programming interfaces.

- **Software as a Service (SaaS)** this refers to browser-initiated application software over thousands of paid cloud customers. The SaaS model applies to business processes, Industry applications, *consumer relationship management (CRM)*, *enterprise resources Planning (ERP)*, *human resources (HR)*, and collaborative applications.

Internet clouds offer four deployment modes: *private*, *public*, *managed*, and *hybrid* these modes demand different levels of security implications. The following list highlights eight reasons to adapt the cloud for upgraded Internet applications and web services:

1. Desired location in areas with protected space and higher energy efficiency
2. Sharing of peak-load capacity among a large pool of users, improving overall utilization
3. Separation of infrastructure maintenance duties from domain-specific application development
4. Significant reduction in cloud computing cost, compared with traditional computing Paradigms
5. Cloud computing programming and application development
6. Service and data discovery and content/service distribution
7. Privacy, security, copyright, and reliability issues
8. Service agreements, business models, and pricing policies

5. Illustrate Service-Oriented Architecture (SOA).

Service-Oriented Architecture (SOA)

In grids/web services, Java, and CORBA, an entity is, respectively, a service, a Java object, and a CORBA distributed object in a variety of languages. These architectures build on the traditional seven Open Systems Interconnection (OSI) layers that provide the base networking abstractions.

Layered Architecture for Web Services and Grids

The entity interfaces correspond to the *Web Services Description Language (WSDL)*, Java method, and *CORBA interface definition language (IDL)* specifications. These interfaces

are linked with customized, high-level communication systems: SOAP, RMI, and IIOP. These communication systems are built on message-oriented middleware infrastructure such as Web Sphere MQ or *Java Message Service* (JMS) which provide rich functionality and support virtualization of routing, senders, and recipients. In the case of fault tolerance, the features in the *Web Services Reliable Messaging* (WSRM) framework mimic the OSI layer capability modified to match the different abstractions

At the entity levels. Security is a critical capability that either uses or re implements the capabilities seen in concepts such as *Internet Protocol Security* (IPsec) and secure sockets in the OSI layers. The CORBA Trading Service, UDDI (*Universal Description, Discovery, and Integration*), LDAP (*Lightweight Directory Access Protocol*), and ebXML (*Electronic Business using eXtensible Markup Language*)

Web Services and Tools

Both web services and REST systems have very distinct approaches to building reliable **Interoperable systems:** This specification is carried with communicated messages using Simple Object Access Protocol (SOAP). The hosting environment then becomes a universal distributed operating system with fully distributed capability carried by SOAP

Messages: REST can use XML schemas but not those that are part of SOAP; “XML over HTTP” is a popular design choice in this regard.

The Evolution of SOA

Filter services (*fs* in the figure) are used to eliminate unwanted raw data, in order to respond to specific requests from the web, the grid, or web services.

Grids versus Clouds

THE general approach used in workflow, the BPEL Web Service standard, and several important workflow approaches including Pegasus, Taverna, Kepler, Trident, and Swift. May end up building with a *system of systems*: such as a *cloud of clouds*, a *grid of clouds*, or a *cloud of grids*, or *inter-clouds* as a basic SOA architecture.

6. What are the Grid Standards?

Standards bodies that are involved in areas related to grid computing include:

- Global Grid Forum (GGF)
- Organization for the Advancement of Structured Information Standards (OASIS)
- World Wide Web Consortium (W3C)
- Distributed Management Task Force (DMTF)
- Web Services Interoperability Organization (WS-I)

OGSA

The Global Grid Forum has published the Open Grid Service Architecture (OGSA). OGSA defines requirements for these core capabilities and thus provides a general reference architecture for grid computing environments. It identifies the components and functions that are useful if not required for a grid environment.

OGSI

The Global Grid Forum extended the concepts defined in OGSA to define specific interfaces to various services that would implement the functions defined by OGSA. A Grid service is a Web service that conforms to a set of interfaces and behaviours that define how a client interacts

with a Grid service. OGSi provides the Web Service Definition Language (WSDL) definitions for these key interfaces.

OGSA-DAI

The OGSA-DAI (data access and integration) project is concerned with Constructing middleware to assist with access and integration of data from separate data sources via the grid.

GridFTP

GridFTP is a secure and reliable data transfer protocol providing high performance and optimized for wide-area networks that have high bandwidth. GridFTP uses basic Grid security on both control (command) and data channels. Features include multiple data channels for parallel transfers, partial file transfers, third-party transfers, and more.

GridFTP can be used to move files (especially large files) across a network efficiently and reliably.

WSRF

WSRF defines a set of specifications for defining the relationship between Web services and stateful resources. WSRF is a general term that encompasses several related proposed standards that cover:

- Resources
- Resource lifetime
- Resource properties
- Service groups (collections of resources)
- Faults
- Notifications
- Topics

Web services related standards

Standards commonly associate with Web services are

- XML
- WSDL
- SOAP

UNIT II: GRID SERVICES (16 marks)

1. Explain Open Grid Services Architecture (OGSA) in detail?

Open Grid Services Architecture (OGSA)

The OGSA is an open source grid service standard jointly developed by academia and the IT industry under coordination of a working group in the Global Grid Forum (GGF). The standard was specifically developed for the emerging grid and cloud service communities.

OGSA Framework

The OGSA was built on two basic software technologies: the Globus Toolkit widely adopted as a grid technology solution for scientific and technical computing, and web services (WS 2.0) as a popular standards-based framework for business and network applications. The OGSA is intended to support the creation, termination, management, and invocation of stateful, transient grid services via standard interfaces and conventions. The OGSA framework specifies the physical environment, security, infrastructure profile, resource provisioning, virtual domains, and execution environment for various Grid services and API access tools.

OGSA Interfaces

The OGSA is centered on grid services. These services demand special well-defined application interfaces. These interfaces provide resource discovery, dynamic service creation, lifetime management, notification, and manageability. Two key properties of a grid service are transience and statefulness. These properties have significant implications regarding how a grid service is named, discovered, and managed. Being transient means the service can be created and destroyed dynamically; statefulness refers to the fact that one can distinguish one service instance from another.

Grid Service Handle

A GSH is a globally unique name that distinguishes a specific grid service instance from all others. The OGSA employs a “handle-resolution” mechanism for mapping from a GSH to a GSR. The GSH must be globally defined for a particular Instance.

Grid Service Migration

This is a mechanism for creating new services and specifying assertions regarding the lifetime of a service. The OGSA model defines a standard interface, known as a factor, to implement this reference. Any service that is created must address the former services as the reference of later services. Each dynamically created grid service instance is associated with a specified lifetime.

OGSA Security Models

The OGSA supports security enforcement at various levels. The grid works in a heterogeneous distributed environment, which is essentially open to the general public. At the security policy and user levels, we want to apply a service or endpoint policy, resource mapping rules, authorized access of critical resources, and privacy protection. At the Public Key Infrastructure (PKI) service level, the OGSA demands security binding with the security protocol stack and bridging of certificate authorities (CAs), use of multiple trusted intermediaries, and so on. Trust models and secure logging are often practiced in grid platforms.

OGSA Grid Service Interfaces Developed by the OGSA Working Group

Port Type	Operation	Brief Description
Grid service	Find service data	Query a grid service instance, including the handle, reference, primary key, home handle map, interface information, and service-specific information. Extensible support for various query languages.
	Termination time	Set (and get) termination time for grid service instance.
	Destroy	Terminate grid service instance.
Notification source	Subscribe to notification tools	Subscribe to notifications of service events. Allow delivery via third-party messaging services.
Notification sink	Deliver notification	Carry out asynchronous delivery of notification messages.
Registry	Register service	Conduct self-state registration of Grid Service Handles (GSHs).
	Unregister service	Unregister a GSH.
Factory	Create service	Create a new grid service instance.
Handle map	Find by handle	Return the Grid Service Reference (GSR) associated with the GSH.

2. Explain OGSA Services?

The OGSA developed within the OGSA Working Group of the Global Grid Forum, is a service-oriented architecture that aims to define a common, standard, and open architecture for grid-based applications. “Open” refers to both the process to develop standards and the standards themselves. In OGSA, everything from registries, to computational tasks, to data resources is considered a service.

These extensible set of services are the building blocks of an OGSA-based grid. OGSA is intended to:

- Facilitate use and management of resources across distributed, heterogeneous Environments
- Deliver seamless QoS

- Define open, published interfaces in order to provide interoperability of diverse resources
 - Exploit industry-standard integration technologies
 - Develop standards that achieve interoperability
 - Integrate, virtualize, and manage services and resources in a distributed, heterogeneous environment
 - Deliver functionality as loosely coupled, interacting services aligned with industry accepted web service standards.

OGSI, developed by the Global Grid Forum, gives a formal and technical specification of a grid service. Grid service interfaces correspond to *portTypes* in WSDL. The set of *portTypes* supported by a grid service, along with some additional information relating to versioning, are specified in the grid service's *serviceType*, a WSDL extensibility element defined by OGSA. The interfaces address discovery, dynamic service creation, lifetime management, notification, and manageability; whereas the conventions address naming and upgradeability. Grid service implementations can target native platform facilities for integration with, and of, existing IT infrastructures.



Figure: OGSA Architecture

OGSA services are summarized as follows:

- **Infrastructure Services** Refer to a set of common functionalities, such as naming, typically required by higher level services.
- **Execution Management Services** Concerned with issues such as starting and managing tasks, including placement, provisioning, and life-cycle management. Tasks may range from simple jobs to complex workflows or composite services.
- **Data Management Services** Provide functionality to move data to where it is needed, maintain replicated copies, run queries and updates, and transform data into new formats. These services must handle issues such as data consistency, persistency, and integrity. An OGSA data service is a web service that implements one or more of the base data interfaces to enable access to, and management of, data resources in a distributed environment. The three base interfaces, *Data Access*, *Data Factory*, and *Data Management*, define basic operations for representing, accessing, creating, and managing data.
- **Resource Management Services** Provide management capabilities for grid resources: management of the resources themselves, management of the resources as grid components, and management of the OGSA infrastructure. For example, resources can be monitored, reserved, deployed, and configured as needed to meet application QoS requirements. It also requires an information model and data model (representation) of the grid resources and services.

- **Security Services** Facilitate the enforcement of security-related policies within a (virtual) organization, and supports safe resource sharing. Authentication, authorization, and integrity assurance are essential functionalities provided by these services.
- **Information Services** Provide efficient production of, and access to, information about the grid and its constituent resources. The term “information” refers to dynamic data or events used for status monitoring; relatively static data used for discovery; and any data that is logged. Troubleshooting is just one of the possible uses for information provided by these services.
- **Self-Management Services** Support service-level attainment for a set of services (or resources), with as much automation as possible, to reduce the costs and complexity of managing the system. These services are essential in addressing the increasing complexity of owning and operating an IT infrastructure.

3.WHAT IS OGSA/OGSI? A PRACTICAL VIEW

It is called an *architecture* because it is mainly about describing and building a well-defined set of interfaces from which systems can be built, based on open standards such as WSDL.

The objectives of OGSA are:

- Manage resources across distributed heterogeneous platforms.
- Support QoS-oriented Service Level Agreements (SLAs). The topology of grids is often complex; the interactions between/among grid resources are almost invariably dynamic. It is critical that the grid provide robust services such as authorization, access control, and delegation.
- Provide a common base for autonomic management. A grid can contain a plethora of resources, along with an abundance of combinations of resource configurations, conceivable resource-to-resource interactions, and a litany of changing state and failure modes. Intelligent self-regulation and autonomic management of these resources is highly desirable.
- Define open, published interfaces and protocols for the interoperability of diverse resources. OGSA is an open standard managed by a standards body. Exploit industry standard integration technologies and leverage existing solutions where appropriate. The foundation of OGSA is rooted in Web services, for example, SOAP and WSDL, are a major part of this specification.

OGSA’s companion OGSI document consists of specifications on how work is managed, distributed, and how service providers and grid services are described. WSDL provides a simple method of describing and advertising the Web services that support the grid’s application. A set of services based on open and stable protocols can hide the complexity of service requests by users or by other elements of a grid. Grid services enable *virtualization*; virtualization, in turn, can transform computing into a ubiquitous infrastructure. OGSA relies on the definition of grid services in WSDL, which, as noted, defines, for this context, the *operations names, parameters,* and their *types* for grid service access .

It is an open and standards-based solution. This implies that, in the future, grid services can be built that are compatible with the OGSI standard, even though they may be based on a variety of different languages and platforms. The UDDI registry and WSIL document are used to locate grid services. The transport protocol SOAP is used to connect data and applications for accessing grid services.

The interfaces of grid services address discovery, dynamic service-instance creation, lifetime management, notification, and manageability; the conventions of Grid services address naming and upgrading issues. The standard interface of a grid service includes multiple bindings and implementations. OGSA also provides a grid security mechanism to ensure that all the communications between services are secure.

. A grid service capability could be comprised of computational resources, storage resources, networks, programs, databases, and so on. A grid service implements one or more interfaces, where each interface defines a set of method operations that is invoked by constructing a method call through, method signature adaptation using SOAP.

There are two fundamental requirements for describing Web services based on the OGSi:

The ability to describe interface inheritance—a basic concept with most of the distributed object systems. The ability to describe additional information elements with the interface definitions.

Table 4.6 Proposed OGSA grid service interfaces*

Port type	Operation	Description
GridService	FindServiceData	Query a variety of information about the grid service instance, including basic introspection information (handle, reference, primary key, home handle map; terms to be defined), richer per-interface information, and service-specific information (e.g., service instances known to a registry). Extensible support for various query languages.
	SetTerminationTime	Set (and get) termination time for grid service instance.
	Destroy	Terminate grid service instance.
Notification-Source	SubscribeToNotificationTopic	Subscribe to notifications of service-related events, based on message type and interest statement. Allows for delivery via third-party messaging services.
Notification-Sink	DeliverNotification	Carry out asynchronous delivery of notification messages.
Registry	RegisterService	Conduct self-state registration of grid service handles.
	UnregisterService	Deregister a grid service handle.
Factory	CreateService	Create new grid service instance.
Handle Map	FindByHandle	Return grid service reference currently associated with supplied grid service handle.

4. What is OGSA/OGSI? A More Detailed View

Introduction

The OGSA integrates key grid technologies with Web services mechanisms to create a distributed system framework based on the OGSi. A *grid service instance* is a service that conforms to a set of conventions, expressed as WSDL interfaces, extensions, and behaviours, for such purposes as lifetime management, discovery of characteristics, and notification. Grid services provide for the controlled management of the distributed and often long-lived state that is commonly required in sophisticated distributed applications. OGSi also introduces standard factory and registration interfaces for creating and discovering grid services.

OGSi defines a component model that extends WSDL and XML schema definition to incorporate the concepts of

- Stateful Web services
- Extension of Web services interfaces
- Asynchronous notification of state change
- References to instances of services
- Collections of service instances
- Service state data that augment the constraint capabilities of XML schema definition

The OGSi specifies (1) how grid service instances are named and referenced; (2) the base, common interfaces that all grid services implement; and (3) the additional interfaces and behaviours associated with factories and service groups.

Setting the Context

GGF calls OGSi the “base for OGSA.” Specifically, there is a relationship between OGSi and distributed object systems and also a relationship between OGSi and the existing Web services framework.

Relationship to Distributed Object Systems.

A given grid service implementation is an addressable and potentially stateful instance that implements one or more interfaces described by WSDL portTypes. Grid service factories can be used to create instances implementing a given set of portType(s). Each grid service instance has a notion of identity with respect to the other instances in the distributed grid. Each instance can be characterized as state coupled with behaviour published through type-specific operations.

Grid service instances are made accessible to client applications through the use of a grid service handle and a grid service reference (GSR). A client application can use a grid service reference to send requests, represented by the operations defined in the portType(s) of the target service description directly to the specific instance at the specified network-attached service endpoint identified by the grid service reference.

Client-Side Programming Patterns.

OGSi exploits an important component of the Web services framework: the use of WSDL to describe multiple protocol bindings, encoding styles, messaging styles, and so on, for a given Web service. The Web Services Invocation Framework (WSIF) and Java API for XML RPC (JAX-RPC) are among the many examples of infrastructure software that provide this capability.

Various tools can take the WSDL description of the Web service and generate interface definitions in a wide range of programming-language-specific constructs.

A *proxy* provides a client-side representation of remote service instance’s interface. Proxy behaviors specific to a particular encoding and network protocol are encapsulated in a *protocol-specific (binding-specific) stub*. This includes both application-specific services and common infrastructure services that are defined by OGSA.

Client Use of Grid Service Handles and References.

A grid service handle (GSH) can be thought of as a permanent network pointer to a particular grid service instance. The client resolves a GSH into a GSR by invoking a HandleResolver grid service instance identified by some out-of-band mechanism. The HandleResolver may have the GSR stored in a local cache. The HandleResolver may need to invoke another HandleResolver to resolve the GSH.

Relationship to Hosting Environment.

OGSi does not dictate a particular service-provider-side implementation architecture. A container implementation may provide a range of functionality beyond simple argument demarshaling.

The Grid Service

The purpose of the OGSi document is to specify the (standardized) interfaces and behaviours that define a *grid service*. In brief, a grid service is a WSDL-defined service that conforms to a set of conventions relating to its interface definitions and behaviours.

The OGSi document expands upon this brief statement by

- Introducing a set of WSDL conventions that one uses in the grid service specification; these conventions have been incorporated in WSDL 1.2.
- Defining *service data* that provide a standard way for representing and querying metadata and state data from a service instance
- Introducing a series of core properties of grid service, including:

- Defining grid service description and grid service instance, as organizing principles for their extension and their use
- Defining how OGSi models time
- Defining the grid service handle and grid service reference constructs that are used to refer to grid service instances
- Defining a common approach for conveying fault information from operations.
- This approach defines a base XML schema definition and associated semantics for WSDL fault messages to support a common interpretation; the approach simply defines the base format for fault messages, without modifying the WSDL fault message model.
- Defining the life cycle of a grid service instance

WSDL Extensions and Conventions

It uses WSDL as the mechanism to describe the public interfaces of grid services. er, WSDL 1.1 is deficient in two critical areas: lack of interface (portType) extension and the inability to describe additional information elements on a portType.

Service Data

The approach to *stateful* Web services introduced in OGSi identified the need for a common mechanism to expose a service instance's state data to service requestors for query, update, and change notification. The GGF is endeavouring to introduce this concept to the broader Web services community.

Service data can be exposed for read, update, or subscription purposes. Since WSDL defines operations and messages for portTypes, the declared state of a service must be externally accessed only through service operations defined as part of the service interface. To avoid the need to define service Data-specific operations for each service Data element, the grid service portType provides base operations for manipulating service Data elements by name. Elements of the publicly available state exposed by the service's interface.

Motivation and Comparison to JavaBean Properties.

The OGSi specification introduces the serviceData concept to provide a flexible, properties-style approach to accessing state data of a Web service. The OGSi specification has chosen not to require getXXX and setXXX WSDL Operations for each serviceData element.

Extending portType with serviceData

ServiceData defines a new portType child element named serviceData, used to define serviceData elements, or SDEs, associated with that portType. These serviceData element definitions are referred to as serviceData declarations, or SDDs.

ServiceDataValues

Each service instance is associated with a collection of serviceData elements: those serviceData elements defined within the various portTypes that form the service's interface, and also, potentially, additional service-Data elements added at runtime. OGSi calls the set of serviceData elements associated with a service instance its "serviceData set."

Each service instance must have a "logical" XML document, with a root element of serviceDataValues that contains the serviceData element values. An example of a serviceDataValues element was given above.

SDE Aggregation within a portType Interface Hierarchy

WSDL 1.2 has introduced the notion of multiple portType extension, and one can model that construct within the GWSDL namespace. A portType can extend zero or more other portTypes. There is no direct relationship between a wsdl: service and the portTypes supported by the service modeled in the WSDL syntax."

Dynamic serviceData Elements

The grid service portType illustrates the use of dynamic SDEs. This contains a serviceData element named “serviceDataName” that lists the serviceData elements currently defined.

Core Grid Service Properties

This subsection discusses a number of properties and concepts common to all grid services.

Service Description and Service Instance

One can distinguish in OGSi between the *description* of a grid service and an *instance* of a grid service:

- A *grid service description* describes how a client interacts with service instances. This description is independent of any particular instance. Within a WSDL document, the grid service description is embodied in the most derived porttype of the instance, along with its associated porttypes bindings, messages, and types definitions.
- A grid service description may be simultaneously used by any number of *grid service instances*, each of which
 - Embodies some state with which the service description describes how to interact
 - Has one or more grid service handles
 - Has one or more grid service references to it

A service description is used primarily for two purposes.

First, as a description of a service interface, it can be used by tooling to automatically generate client interface proxies, server skeletons, and so forth.

Second, it can be used for discovery, for example, to find a service instance that implements a particular service description, or to find a factory that can create instances with a particular service description.

The service description is meant to capture both interface syntax and semantics. *Interface syntax* is described by WSDL portTypes. *Semantics* may be inferred through the name assigned to the portType.

Modeling Time in OGSi: The need arises at various points throughout this specification to represent time that is meaningful to multiple parties in the distributed Grid. Clients need to negotiate service instance lifetimes with services, and multiple services may need a common understanding of time in order for clients to be able to manage their simultaneous use and interaction.

5. What are the various OGSA Services?

- Handle Resolution
- Virtual Organization Creation and Management
- Service Groups and Discovery Services
- Choreography, Orchestrations and Workflow
- Transactions
- Metering Service
- Rating Service
- Accounting Service
- Billing and Payment Service
- Installation, Deployment, and Provisioning
- Distributed Logging
- Messaging and Queuing

- Event
- Policy and Agreements
- Base Data Services
- Other Data Services
- Discovery Services
- Job Agreement Service
- Reservation Agreement Service
- Data Access Agreement Service
- Queuing Service
- Open Grid Services Infrastructure
- Common Management Model

6. Explain Data intensive grid service models?

Applications in the grid are normally grouped into two categories: computation-intensive and data-intensive. The grid system must be specially designed to discover, transfer, and manipulate these massive data sets. Transferring massive data sets is a time-consuming task. Efficient data management demands low-cost storage and high-speed data movement.

Data Replication and Unified Namespace

This data access method is also known as caching, which is often applied to enhance data efficiency in a grid environment. By replicating the same data blocks and scattering them in multiple regions of a grid, users can access the same data with locality of references. Furthermore, the replicas of the same data set can be a backup for one another. Some key data will not be lost in case of failures. The increase in storage requirements and network bandwidth may cause additional problems.

Replication strategies determine when and where to create a replica of the data. The factors to consider include data demand, network conditions, and transfer cost. The strategies of replication can be classified into method types: dynamic and static. Dynamic strategies can adjust locations and number of data replicas according to changes in conditions.

The most common replication strategies include preserving locality, minimizing update costs, and maximizing profits.

Grid Data Access Models

Multiple participants may want to share the same data collection. To retrieve any piece of data, we need a grid with a unique global namespace. There are four access models for organizing a data grid

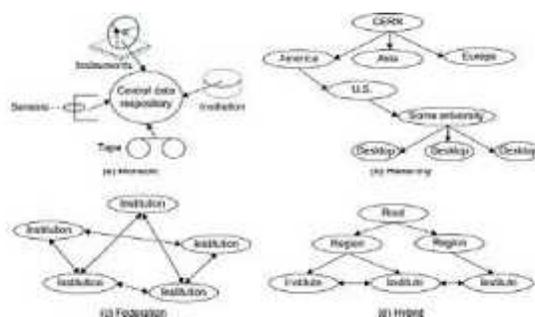


Figure: Four architectural models for building a data grid.

Monadic model: This is a centralized data repository model. All the data is saved in a central data repository. When users want to access some data they have to submit requests directly to

the central repository. No data is replicated for preserving data locality. This model is the simplest to implement for a small grid. For a large grid, this model is not efficient in terms of performance and reliability. Data replication is permitted in this model only when fault tolerance is demanded.

Hierarchical model: The hierarchical model, is suitable for building a large data grid which has only one large data access directory. The data may be transferred from the source to a second-level center. Then some data in the regional center is transferred to the third-level center. After being forwarded several times, specific data objects are accessed directly by users. Generally speaking, a higher-level data center has a wider coverage area. It provides higher bandwidth for access than a lower-level data center. KI security services are easier to implement in this hierarchical data access model.

Federation model: This data access model is better suited for designing a data grid with multiple sources of data supplies. Sometimes this model is also known as a mesh model. The data sources are distributed to many different locations. Although the data is shared, the data items are still owned and controlled by their original owners. According to predefined access policies, only authenticated users are authorized to request data from any data source. This mesh model may cost the most when the number of grid institutions becomes very large.

Hybrid model: This data access model combines the best features of the hierarchical and mesh models. Traditional data transfer technology, such as FTP, applies for networks with lower bandwidth. Network links in a data grid often have fairly high bandwidth, and other data transfer models are exploited by high-speed data transfer tools such as GridFTP developed with the Globus library. The cost of the hybrid model can be traded off between the two extreme models for hierarchical and mesh-connected grids.

Parallel versus Striped Data Transfers

Compared with traditional FTP data transfer, *parallel data transfer* opens multiple data streams for passing subdivided segments of a file simultaneously.

. In *striped data transfer*, a data object is partitioned into a number of sections, and each section is placed in an individual site in a data grid. When a user requests this piece of data, a data stream is created for each site, and all the sections of data objects are transferred simultaneously. Striped data transfer can utilize the bandwidths of multiple sites more efficiently to speed up data transfer

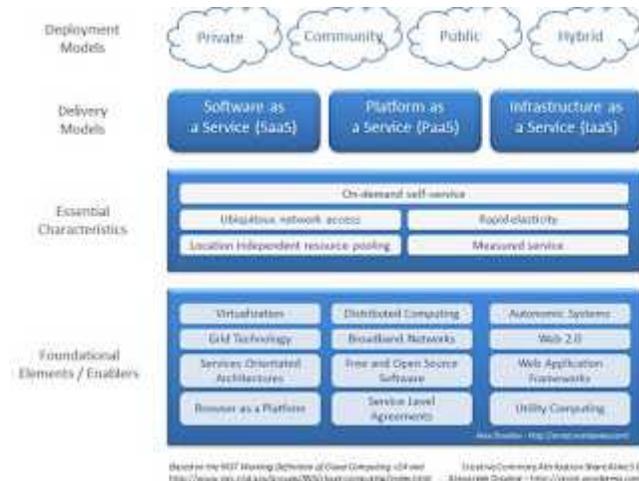
UNIT III: VIRTUALIZATION (16 marks)

1. Explain NIST Cloud Computing Architecture?

The NIST definition “Cloud computing”

- “... a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.”

- The NIST definition also identifies
 - 5 essential characteristics
 - 3 service models
 - 4 deployment models



4 deployment models

- **Public:** Accessible, via the Internet, to anyone who pays
 - Owned by service providers; e.g., Google App Engine, Amazon Web Services, Force.com.
- **Community:** Shared by two or more organizations with joint interests, such as colleges within a university
- **Private:** Accessible via an intranet to the members of the owning organization
 - Can be built using open source software such as CloudStack or OpenStack
 - Example of private cloud: NASA’s cloud for climate modeling
- **Hybrid**
 - A private cloud might buy computing resources from a public cloud.

3 service models

- Cloud Software as a Service (SaaS)
 - Use provider’s applications over a network
- Cloud Platform as a Service (PaaS)
 - Deploy customer-created applications to a cloud
- Cloud Infrastructure as a Service (IaaS)
 - Rent processing, storage, network capacity, and other fundamental computing resources

5 essential characteristics

- On-demand self-service: consumers can acquire the necessary computational resources without having to interact with human service providers.
- Ubiquitous network access: cloud features don’t require special devices – laptops, mobile phones, etc. are generally supported.
- Resource pooling: cloud resources are pooled to serve many customers “... using a multi-tenant model, with different physical and virtual resources...”
- Rapid elasticity: resources can be allocated and de-allocated quickly as needed.
- Measured service: resource use is measured and monitored; charges are made based on usage and service type (e.g., storage, CPU cycles, etc.)

2. Explain Cloud Design Objectives?

Cloud Design Objectives

The following list highlights six design objectives for cloud computing:

- **Shifting computing from desktops to data centers** Computer processing, storage, and software delivery is shifted away from desktops and local servers and toward data centers over the Internet.
- **Service provisioning and cloud economics** Providers supply cloud services by signing SLAs with consumers and end users. The services must be efficient in terms of computing, storage, and power consumption. Pricing is based on a pay-as-you-go policy.
- **Scalability in performance** the cloud platforms and software and infrastructure services must be able to scale in performance as the number of users increases.
- **Data privacy protection** Can you trust data centers to handle your private data and Records? This concern must be addressed to make clouds successful as trusted services.
- **High quality of cloud services** The QoS of cloud computing must be standardized to make clouds interoperable among multiple providers.
- **New standards and interfaces** this refers to solving the data lock-in problem associated with data centers or cloud providers. Universally accepted APIs and access Protocols are needed to provide high portability and flexibility of virtualized applications.

3. Explain Infrastructure, platform, software As An service?

Cloud computing delivers infrastructure, platform, and software (application) as services, which are made available as subscription-based services in a pay-as-you-go model to consumers. The services provided over the cloud can be generally categorized into three different service models: namely IaaS, Platform as a Service (PaaS), and Software as a Service (SaaS). All three models allow users to access services over the Internet, relying entirely on the infrastructures of cloud service providers. These models are offered based on various SLAs between providers and users. In a broad sense, the SLA for cloud computing is addressed in terms of service availability, performance, and data protection and security.

Infrastructure as a Service

This model allows users to use virtualized IT resources for computing, storage, and networking. In short, the service is performed by rented cloud infrastructure. The user can deploy and run his applications over his chosen OS environment. The user does not manage or control the underlying cloud infrastructure, but has control over the OS, storage, deployed applications, and possibly select networking components. This IaaS model encompasses *storage as a service*, *compute instances as a service*, and *communication as a service*. The *Virtual Private Cloud (VPC)* in Example 4.1 shows how to provide Amazon EC2 clusters and S3 storage to multiple users. Many startup cloud providers have appeared in recent years. GoGrid, FlexiScale, and Aneka are good examples.

Platform as a Service (PaaS)

To be able to develop, deploy, and manage the execution of applications using provisioned resources demands a cloud platform with the proper software environment. Such a platform includes operating system and runtime library support. This has triggered the creation of the PaaS model to enable users to develop and deploy their user applications.

The platform cloud is an integrated computer system consisting of both hardware and software infrastructure. The user application can be developed on this virtualized cloud platform using some programming languages and software tools supported by the provider (e.g., Java, Python, .NET). The user does not manage the underlying cloud infrastructure. The cloud provider supports user application development and testing on a well-defined service platform.

This PaaS model enables a collaborated software development platform for users from different parts of the world.

Software as a Service (SaaS)

This refers to browser-initiated application software over thousands of cloud customers. Services and tools offered by PaaS are utilized in construction of applications and management of their deployment on resources offered by IaaS providers. The SaaS model provides software applications as a service. As a result, on the customer side, there is no upfront investment in servers or software licensing. On the provider side, costs are kept rather low, compared with conventional hosting of user applications. Customer data is stored in the cloud that is either vendor proprietary or publicly hosted to support PaaS and IaaS. The best examples of SaaS services include Google Gmail and docs, Microsoft SharePoint, and the CRM software from Salesforce.com. They are all very successful in promoting their own business or are used by thousands of small businesses in their day-to-day operations. Providers such as Google and Microsoft offer integrated IaaS and PaaS services, whereas others such as Amazon and GoGrid offer pure IaaS services and expect third-party PaaS providers such as Manjrasoft to offer application development and deployment services on top of their infrastructure services.

4. Explain Implementation levels of virtualization in detail

A traditional computer runs with a host operating system specially tailored for its hardware architecture. After virtualization, different user applications managed by their own operating systems (guest OS) can run on the same hardware, independent of the host OS. This is often done by adding additional software, called a virtualization layer. This virtualization layer is known as hypervisor or virtual machine monitor (VMM). The VMs are shown in the upper boxes, where applications run with their own guest OS over the virtualized CPU, memory, and I/O resources. The main function of the software layer for virtualization is to virtualize the physical hardware of a host machine into virtual resources to be used by the VMs, exclusively. This can be implemented at various operational levels, as we will discuss shortly. The Virtualization software creates the abstraction of VMs by interposing a virtualization layer at various levels of a computer system. Common virtualization layers include the *Instruction set architecture (ISA)* level, hardware level, operating system level, and library support level, and application level.

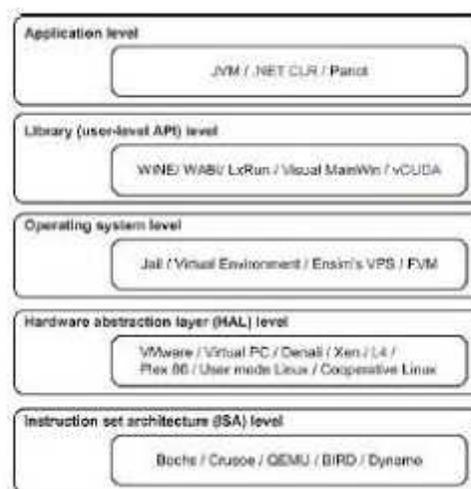


Figure: Virtualization ranging from hardware to applications in five abstraction levels

Instruction Set Architecture Level

At the ISA level, virtualization is performed by emulating a given ISA by the ISA of the host machine. For example, MIPS binary code can run on an x86-based host machine with the help of ISA emulation. The basic emulation method is through code interpretation. An interpreter program interprets the source instructions to target instructions one by one. One source instruction may require tens or hundreds of native target instructions to perform its function. This approach translates basic blocks of dynamic source instructions to target instructions. The basic blocks can also be extended to program traces or super blocks to increase translation efficiency. Instruction set emulation requires binary translation and optimization. A virtual instruction set architecture (V-ISA) thus requires adding a processor-specific software translation layer to the compiler.

Hardware Abstraction Level

Hardware-level virtualization is performed right on top of the bare hardware. On the one hand, this approach generates a virtual hardware environment for a VM. On the other hand, the process manages the underlying hardware through virtualization. The idea is to virtualize a computer's resources, such as its processors, memory, and I/O devices. The intention is to upgrade the hardware utilization rate by multiple users concurrently. The idea was implemented in the IBM VM/370 in the 1960s. More recently, the Xen hypervisor has been applied to virtualize x86-based machines to run Linux or other guest OS.

Operating System Level

This refers to an abstraction layer between traditional OS and user applications. OS-level virtualization creates isolated containers on a single physical server and the OS instances to utilize the hardware and software in data centers. The containers behave like real servers. OS-level virtualization is commonly used in creating virtual hosting environments to allocate hardware resources among a large number of mutually distrusting users.

Library Support Level

Most applications use APIs exported by user-level libraries rather than using lengthy system calls by the OS. Since most systems provide well-documented APIs, such an interface becomes another candidate for virtualization. Virtualization with library interfaces is possible by controlling the communication link between applications and the rest of a system through API hooks. The software tool WINE has implemented this approach to support Windows applications on top of UNIX hosts.

User-Application Level

Virtualization at the application level virtualizes an application as a VM. On a traditional OS, an application often runs as a process. Therefore, application-level virtualization is also known as process-level virtualization. The most popular approach is to deploy high level language (HLL) VMs. In this scenario, the virtualization layer sits as an application program on top of the operating system, and the layer exports an abstraction of a VM that can run programs written and compiled to a particular abstract machine definition. Any program written in the HLL and compiled for this VM will be able to run on it. The Microsoft .NET CLR and Java Virtual Machine (JVM) are two good examples of this class of VM.

Relative Merits of Different Approaches

The column headings correspond to four technical merits. "Higher Performance" and "Application Flexibility" are self-explanatory. "Implementation Complexity" implies the cost to implement that particular virtualization level. "Application Isolation" refers to the effort required to isolate resources committed to different VMs. Each row corresponds to a particular level of virtualization.

Level of Implementation	Higher Performance	Application Flexibility	Implementation Complexity	Application Isolation
ISA	X	XXXXX	XXX	XXX
Hardware-level virtualization	XXXXX	XXX	XXXXX	XXXX
OS-level virtualization	XXXXX	XX	XXX	XX
Runtime library support	XXX	XX	XX	XX
User application level	XX	XX	XXXXX	XXXXX

5. Explain Virtualization of CPU, Memory, and I/O Devices?

Hardware Support for Virtualization

Modern operating systems and processors permit multiple processes to run simultaneously. If there is no protection mechanism in a processor, all instructions from different processes will access the hardware directly and cause a system crash. Therefore, all processors have at least two modes, user mode and supervisor mode, to ensure controlled access of critical hardware. Instructions running in supervisor mode are called privileged instructions. Other instructions are unprivileged instructions. The

VMware Workstation is a VM software suite for x86 and x86-64 computers. This software suite allows users to set up multiple x86 and x86-64 virtual computers and to use one or more of these VMs simultaneously with the host operating system. The VMware Workstation assumes the host-based virtualization. Xen is a hypervisor for use in IA-32, x86-64, Itanium, and PowerPC 970 hosts.

CPU Virtualization

A VM is a duplicate of an existing computer system in which a majority of the VM instructions are executed on the host processor in native mode. Thus, unprivileged instructions of VMs run directly on the host machine for higher efficiency. Other critical instructions should be handled carefully for correctness and stability. The critical instructions are divided into three categories: privileged instructions, control-sensitive instructions, and behavior-sensitive instructions. Privileged instructions execute in a privileged mode and will be trapped if executed outside this mode. Control-sensitive instructions attempt to change the configuration of resources used. Behavior-sensitive instructions have different behaviors depending on the configuration of resources, including the load and store operations over the virtual memory.

The VMM acts as a unified mediator for hardware access from different VMs to guarantee the correctness and stability of the whole system.

Hardware-Assisted CPU Virtualization

This technique attempts to simplify virtualization because full or paravirtualization is complicated. Intel and AMD add an additional mode called privilege mode level to x86 processors.

Memory Virtualization

Virtual memory virtualization is similar to the virtual memory support provided by modern operating systems. All modern x86 CPUs include a memory management unit (MMU) and a translation lookaside buffer (TLB) to optimize virtual memory performance. A two-stage mapping process should be maintained by the guest OS and the VMM, respectively: virtual memory to physical memory and physical memory to machine memory. The guest OS continues to control the mapping of virtual addresses to the physical memory addresses of VMs. The MMU already handles virtual-to-physical translations as defined by the OS. Then the physical memory addresses are translated to machine addresses using another set of page tables defined by the hypervisor. Processors use TLB hardware to map the virtual memory directly to the machine memory to avoid the two levels of translation on every access. When the guest OS changes the

virtual memory to a physical memory mapping, the VMM updates the shadow page tables to enable a direct lookup.

I/O Virtualization

I/O virtualization involves managing the routing of I/O requests between virtual devices and the shared physical hardware. At the time of this writing, there are three ways to implement I/O virtualization: full device emulation, para-virtualization, and direct I/O.

Full device emulation is the first approach for I/O virtualization. Generally, this approach emulates well-known, real-world devices. All the functions of a device or bus infrastructure, such as device enumeration, identification, interrupts, and DMA, are replicated in software. This software is located in the VMM and acts as a virtual device. The I/O access requests of the guest OS are trapped in the VMM which interacts with the I/O devices.

The para-virtualization method of I/O virtualization is typically used in Xen. It is also known as the split driver model consisting of a frontend driver and a backend driver. The frontend driver is running in Domain U and the backend driver is running in Domain 0. They interact with each other via a block of shared memory. The frontend driver manages the I/O requests of the guest OSes and the backend driver is responsible for managing the real I/O devices and multiplexing the I/O data of different VMs.

Virtualization in Multi-Core Processors

Virtualizing a multi-core processor is relatively more complicated than virtualizing a uniprocessor. Though multicore processors are claimed to have higher performance by integrating multiple processor cores in a single chip, multi-core virtualization has raised some new challenges to computer architects, compiler constructors, system designers, and application programmers. There are mainly two difficulties: Application programs must be parallelized to use all cores fully, and software must explicitly assign tasks to the cores, which is a very complex problem.

Physical versus Virtual Processor Cores

This technique alleviates the burden and inefficiency of managing hardware resources by software. It is located under the ISA and remains unmodified by the operating system or VMM (hypervisor).

Virtual Hierarchy

The emerging many-core chip multiprocessors (CMPs) provides a new computing landscape. To optimize for space-shared workloads, they propose using virtual hierarchies to overlay a coherence and caching hierarchy onto a physical processor. Unlike a fixed physical hierarchy, a virtual hierarchy can adapt to fit how the work is space shared for improved performance and performance isolation.

6. Explain Live VM Migration Steps and Performance Effects?

In a cluster built with mixed nodes of host and guest systems, the normal method of operation is to run everything on the physical machine. When a VM fails, its role could be replaced by another VM on a different node, as long as they both run with the same guest OS. The advantage is enhanced failover flexibility.

The migration copies the VM state file from the storage area to the host machine.

There are four ways to manage a virtual cluster.

First, you can use a guest-based manager, by which the cluster manager resides on a guest system. The host-based manager supervises the guest systems and can restart the guest system on another physical machine.

These two cluster management systems are either guest-only or host-only, but they do not mix.

A third way to manage a virtual cluster is to use an independent cluster manager on both the host and guest systems. This will make infrastructure management more complex.

Finally, you can use an integrated cluster on the guest and host systems. VM can be in one of the following four states.

An **inactive** state is defined by the virtualization platform, under which the VM is not enabled.

An **active** state refers to a VM that has been instantiated at the virtualization platform to perform a real task.

A **paused** state corresponds to a VM that has been instantiated but disabled to process a task or paused in **awaiting** state.

Live migration of a VM consists of the following six steps:

Steps 0 and 1: **Start migration**. This step makes preparations for the migration, including determining the migrating VM and the destination host. Although users could manually make a VM migrate to an appointed host, in most circumstances, the migration is automatically started by strategies such as load balancing and server consolidation.

Steps 2: **Transfer memory**. Since the whole execution state of the VM is stored in memory, sending the VM's memory to the destination node ensures continuity of the service provided by the VM. All of the memory data is transferred in the first round, and then the migration controller recopies the memory data which is changed in the last round. These steps keep iterating until the dirty portion of the memory is small enough to handle the final copy. Although precopying memory is performed iteratively, the execution of programs is not obviously interrupted.

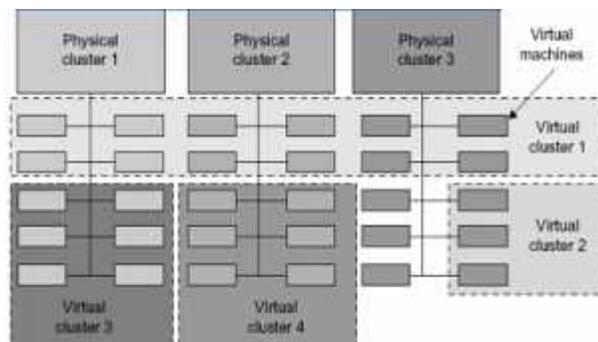
Step 3: **Suspend** the VM and copy the last portion of the data. The migrating VM's execution is suspended when the last round's memory data is transferred. Other nonmemory data such as CPU and network states should be sent as well. During this step, the VM is stopped and its applications will no longer run. This "service unavailable" time is called the "downtime" of migration, which should be as short as possible so that it can be negligible to users.

Steps 4 and 5: **Commit and activate** the new host. After all the needed data is copied, on the destination host, the VM reloads the states and recovers the execution of programs in it, and the service provided by this VM continues. Then the network connection is redirected to the new VM and the dependency to the source host is cleared. The whole migration process finishes by removing the original VM from the source host.

7. Explain Virtualization for data center automation in detail?

A physical cluster is a collection of servers (physical machines) interconnected by a physical network such as a LAN.

Virtual clusters are built with VMs installed at distributed servers from one or more physical clusters. The VMs in a virtual cluster are interconnected logically by a virtual network across several physical networks. Each virtual cluster is formed with physical machines or a VM hosted by multiple physical clusters. The virtual cluster boundaries are shown as distinct boundaries.



The provisioning of VMs to a virtual cluster is done dynamically to have the following Interesting properties:

- The virtual cluster nodes can be either physical or virtual machines. Multiple VMs running with different OSES can be deployed on the same physical node.
- A VM runs with a guest OS, which is often different from the host OS, that manages the resources in the physical machine, where the VM is implemented.
- The purpose of using VMs is to consolidate multiple functionalities on the same server. This will greatly enhance server utilization and application flexibility.
- VMs can be colonized (replicated) in multiple servers for the purpose of promoting distributed parallelism, fault tolerance, and disaster recovery.
- The size (number of nodes) of a virtual cluster can grow or shrink dynamically, similar to the way an overlay network varies in size in a peer-to-peer (P2P) network.
- The failure of any physical nodes may disable some VMs installed on the failing nodes. But the failure of VMs will not pull down the host system.

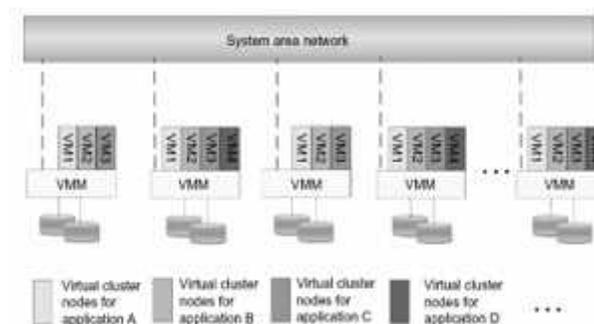
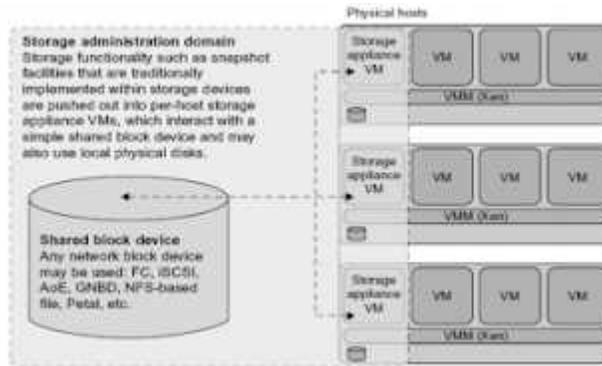


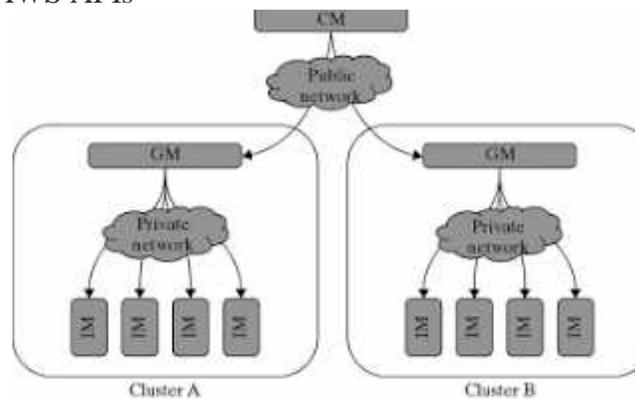
Figure: A Virtual Clusters based on Application Partitioning: Parallax Providing Virtual Disks to Clients VMs from a Large Common Shared Physical Disk.



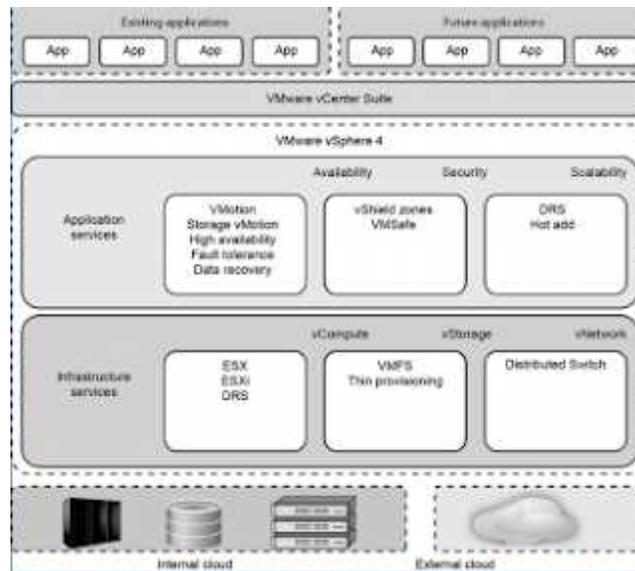
Cloud OS for Building Private Clouds (VI: Virtual Infrastructure, EC2: Elastic Compute Cloud).

Manager/ OS, Platforms, License	Resources Being Virtualized, Web Link	Client API, Language	Hypervisors Used	Public Cloud Interface	Special Features
Nimbus Linux, Apache v2	VM creation, virtual cluster, www.nimbusproject.org/	EC2 WS, WS/F, CLI	Xen, KVM	EC2	Virtual networks
Eucalyptus Linux, BSD	Virtual networking (Example 3.12 and 3.13), www.eucalyptus.com/	EC2 WS, CLI	Xen, KVM	EC2	Virtual networks
OpenNebula Linux, Apache v2	Management of VM, host, virtual network, and scheduling tools, www.opennebula.org/	XML/RPC, CLI, Java	Xen, KVM	EC2, Elastic Host	Virtual networks, dynamic provisioning
vSphere 4 Linux, Windows, proprietary	Virtualizing OS for data centers (Example 3.13), www.vmware.com/products/vsphere/ [90]	CLI, GUI, Petal, WS	VMware ESX, ESX	VMware vCloud partners	Data protection, vStorage, VMFS, DRM, HA

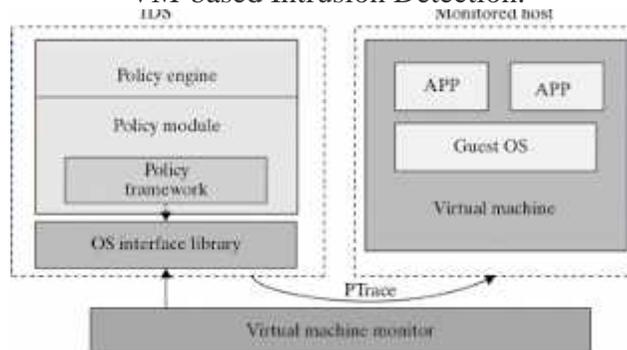
Eucalyptus: An Open-Source OS for Setting Up and Managing Private Clouds (IaaS)
Three Resource Managers: CM (Cloud Manager), GM (Group Manager), and IM (Instance Manager) Works like AWS APIs



VMware vSphere 4 – A Commercial Cloud OS.



VM-based Intrusion Detection.



Techniques for establishing trusted zones for virtual cluster insulation and VM isolation

1. UNIT IV: PROGRAMMING MODEL(16 marks)

1. Explain the Globus Toolkit Architecture (GT4)

The Globus Toolkit is an open middleware library for the grid computing communities. These open source software libraries support many operational grids and their applications on an international basis. The toolkit addresses common problems and issues related to grid resource discovery, management, communication, security, fault detection, and portability. The software itself provides a variety of components and capabilities. The library includes a rich set of service implementations. The implemented software supports grid infrastructure management, provides tools for building new web services in Java, C, and Python, builds a powerful standard-based.

Security infrastructure and client APIs (in different languages), and offers comprehensive command-line programs for accessing various grid services. The Globus Toolkit was initially motivated by a desire to remove obstacles that prevent seamless collaboration, and thus sharing

of resources and services, in scientific and engineering applications. The shared resources can be computers, storage, data, services, networks, science instruments (e.g., sensors), and so on.

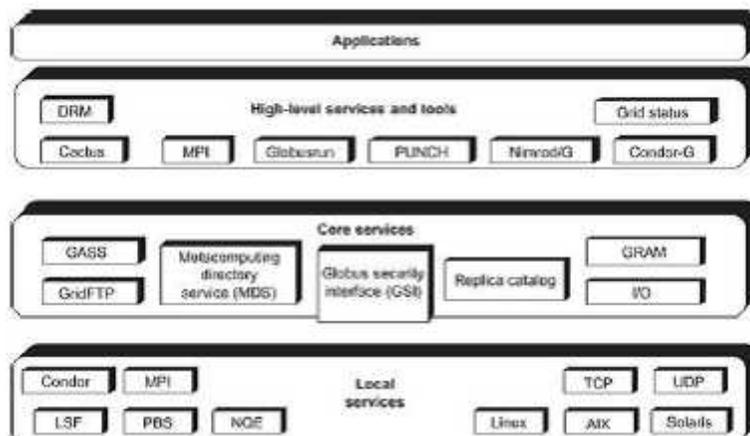


Figure: Globus Toolkit GT4 supports distributed and cluster computing services

The GT4 Library

The GT4 Library GT4 offers the middle-level core services in grid applications. The high-level services and tools, such as MPI, Condor-G, and Nirod/G, are developed by third parties for general purpose distributed computing applications. The local services, such as LSF, TCP, Linux, and Condor, are at the bottom level and are fundamental tools supplied by other developers.

Globus Job Workflow

A typical job execution sequence proceeds as follows: The user delegates his credentials to a delegation service. The user submits a job request to GRAM with the delegation identifier as a parameter. GRAM parses the request, retrieves the user proxy certificate from the delegation service, and then acts on behalf of the user. GRAM sends a transfer request to the RFT, which applies GridFTP to bring in the necessary files.

GRAM invokes a local scheduler via a GRAM adapter and the SEG initiates a set of user jobs. The local scheduler reports the job state to the SEG. Once the job is complete, GRAM uses RFT and GridFTP to stage out the resultant files.

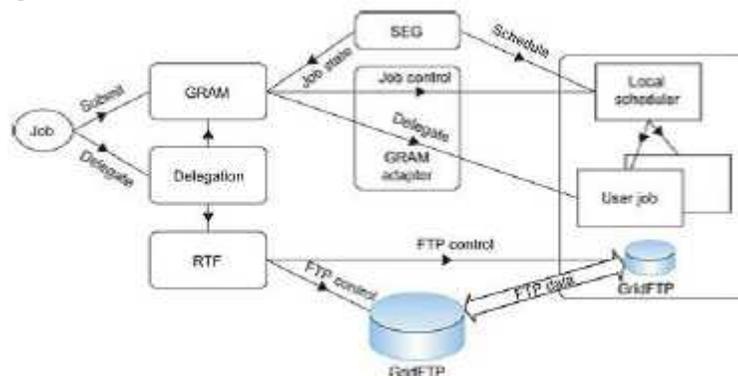


Figure: Globus job workflow among interactive functional modules.

Client-Globus Interactions

There are strong interactions between provider programs and user code. GT4 makes heavy use of industry-standard web service protocols and mechanisms in service

Description, discovery, access, authentication, authorization, and the like. GT4 makes extensive use of Java, C, and Python to write user code. Web service mechanisms define specific interfaces for grid computing. Web services provide flexible, extensible, and widely adopted XML-based interfaces.

These demand computational, communication, data, and storage resources. We must enable a range of end-user tools that provide the higher-level capabilities needed in specific user applications. Developers can use these services and libraries to build simple and complex systems quickly.

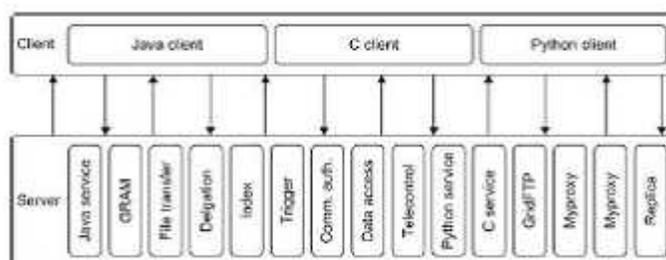
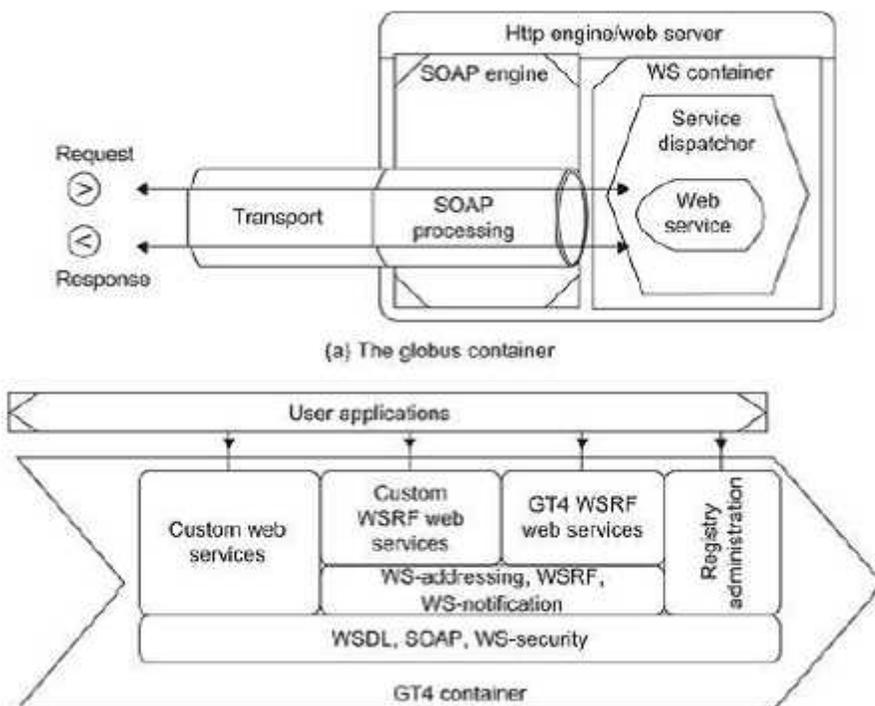


Figure: Client and GT4 server interactions; vertical boxes correspond to service programs and horizontal boxes represent the user codes.

The horizontal boxes in the client domain denote custom applications and/or third-party tools that access GT4 services. The toolkit programs provide a set of useful infrastructure services.

Three containers are used to host user-developed services written in Java, Python, and C, respectively. These containers provide implementations of security, management, discovery, state management, and other mechanisms frequently required when building services.



2. Explain MapReduce Model in detail

The model is based on two distinct steps for an application:

- **Map**: An initial ingestion and transformation step, in which individual input records can be processed in parallel.

- **Reduce**: An aggregation or summarization step, in which all associated records must be processed together by a single entity.

The core concept of MapReduce in Hadoop is that input may be split into logical chunks, and each chunk may be initially processed independently, by a map task. The results of these individual processing chunks can be physically partitioned into distinct sets, which are then sorted. Each sorted chunk is passed to a reduce task.

A map task may run on any compute node in the cluster, and multiple map tasks may be running in parallel across the cluster. The map task is responsible for transforming the input records into key/value pairs. The output of all of the maps will be partitioned, and each partition will be sorted. There will be one partition for each reduce task. Each partition's sorted keys and the values associated with the keys are then processed by the reduce task. There may be multiple reduce tasks running in parallel on the cluster.

The application developer needs to provide only four items to the Hadoop framework: the class that will read the input records and transform them into one key/value pair per record, a map method, a reduce method, and a class that will transform the key/value pairs that the reduce method outputs into output records.

My first MapReduce application was a specialized web crawler. This crawler received as input large sets of media URLs that were to have their content fetched and processed. The media items were large, and fetching them had a significant cost in time and resources.

The job had several steps:

1. Ingest the URLs and their associated metadata.
2. Normalize the URLs.
3. Eliminate duplicate URLs.
4. Filter the URLs against a set of exclusion and inclusion filters.
5. Filter the URLs against a do not fetch list.
6. Filter the URLs against a recently seen set.
7. Fetch the URLs.
8. Fingerprint the content items.
9. Update the recently seen set.
10. Prepare the work list for the next application.

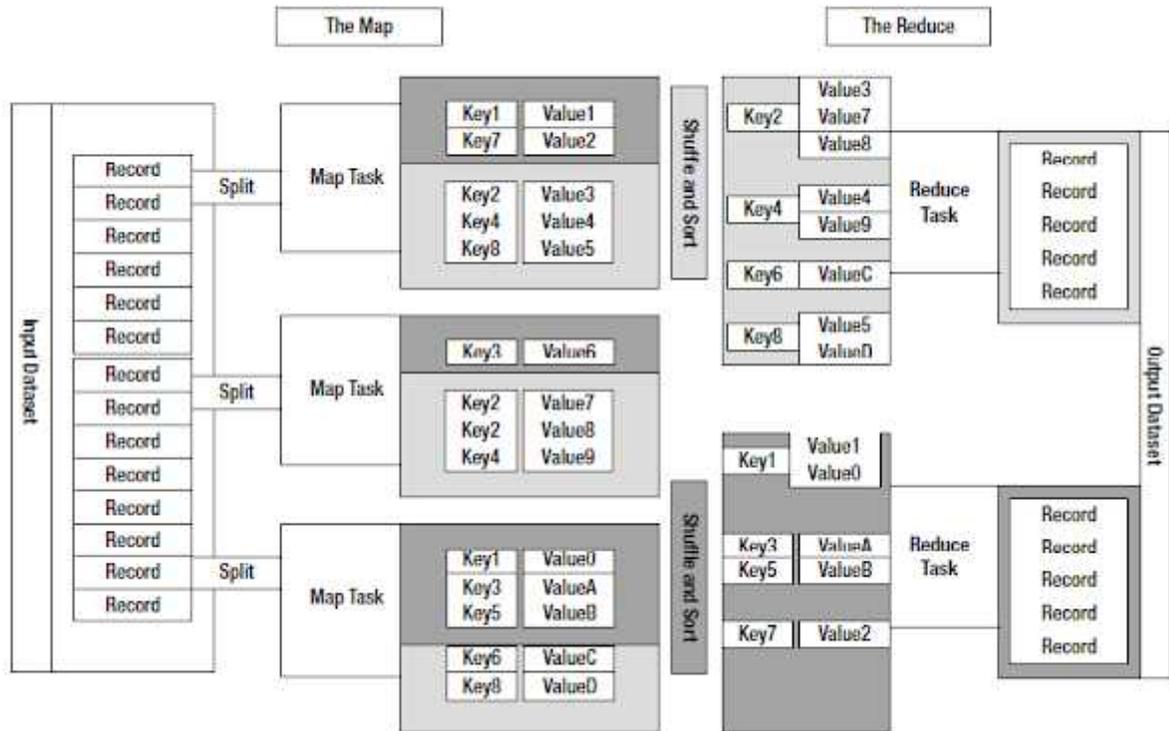


Figure: *The MapReduce model*

Introducing Hadoop

Hadoop is the Apache Software Foundation top-level project that holds the various Hadoop subprojects that graduated from the Apache Incubator. The Hadoop project provides and supports the development of open source software that supplies a framework for the development of highly scalable distributed computing applications. The Hadoop framework handles the processing details, leaving developers free to focus on application logic. The introduction on the Hadoop project web page states:

The Apache Hadoop project develops open-source software for reliable, scalable, distributed computing, including:

Hadoop Core, our flagship sub-project, provides a distributed filesystem (HDFS) and support for the MapReduce distributed computing metaphor.

HBase builds on Hadoop Core to provide a scalable, distributed database.

Pig is a high-level data-flow language and execution framework for parallel computation. It is built on top of Hadoop Core.

ZooKeeper is a highly available and reliable coordination system. Distributed applications use ZooKeeper to store and mediate updates for critical shared state.

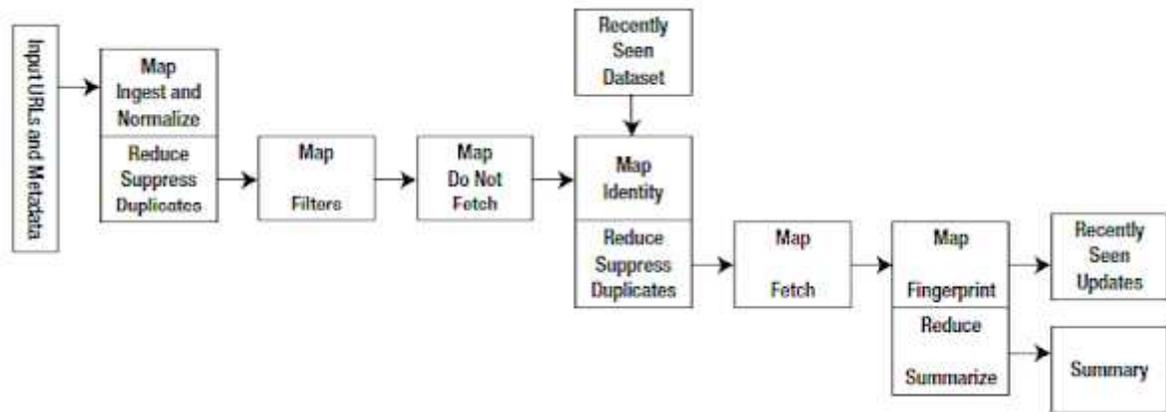
Hive is a data warehouse infrastructure built on Hadoop Core that provides data summarization, adhoc querying and analysis of datasets.

The Hadoop Core project provides the basic services for building a cloud computing environment with commodity hardware, and the APIs for developing software that will run on that cloud.

The two fundamental pieces of Hadoop Core are the MapReduce framework, the cloud computing environment, and the Hadoop Distributed File System (HDFS).

The Hadoop Core MapReduce framework requires a shared file system. This shared file system does not need to be a system-level file system, as long as there is a distributed file system plug-in available to the framework.

The Hadoop Core framework comes with plug-ins for HDFS, CloudStore, and S3. Users are also free to use any distributed file system that is visible as a system-mounted file system, such as Network File System (NFS), Global File System (GFS), or Lustre.



The Hadoop Distributed File System (HDFS)MapReduce environment provides the user with a sophisticated framework to manage the execution of map and reduce tasks across a cluster of machines.

The user is required to tell the framework the following:

- The location(s) in the distributed file system of the job input
- The location(s) in the distributed file system for the job output
- The input format
- The output format
- The class containing the map function
- Optionally, the class containing the reduce function
- The JAR file(s) containing the map and reduce functions and any support classes

The final output will be moved to the output directory, and the job status will be reported to the user. MapReduce is oriented around key/value pairs. The framework will convert each record of input into a key/value pair, and each pair will be input to the map function once. The map output is a set of key/value pairs—nominally one pair that is the transformed input pair. The map output pairs are grouped and sorted by key. The reduce function is called one time for each key, in sort sequence, with the key and the set of values that share that key. The reduce method may output an arbitrary number of key/value pairs, which are written to the output files in the job output directory. If the reduce output keys are unchanged from the reduce input keys, the final output will be sorted. The framework provides two processes that handle the management of MapReduce jobs:

- TaskTracker manages the execution of individual map and reduce tasks on a compute node in the cluster.
- JobTracker accepts job submissions, provides job monitoring and control, and manages the distribution of tasks to the TaskTracker nodes.

The JobTracker is a single point of failure, and the JobTracker will work around the failure of individual TaskTracker processes.

The Hadoop Distributed File System

HDFS is a file system that is designed for use for MapReduce jobs that read input in large chunks of input, process it, and write potentially large chunks of output. HDFS does not handle random access particularly well. For reliability, file data is simply mirrored to multiple storage nodes. This is referred to as *replication* in the Hadoop community. As long as at least one replica of a data chunk is available, the consumer of that data will not know of storage server failures.

HDFS services are provided by two processes:

- NameNode handles management of the file system metadata, and provides management and control services.
- DataNode provides block storage and retrieval services.

There will be one NameNode process in an HDFS file system, and this is a single point of failure. Hadoop Core provides recovery and automatic backup of the NameNode, but no hot failover services. There will be multiple DataNode processes within the cluster, with typically one DataNode process per storage node in a cluster.

3. Explain Map & Reduce function?

A Simple Map Function: IdentityMapper

The Hadoop framework provides a very simple map function, called IdentityMapper. It is used in jobs that only need to reduce the input, and not transform the raw input. All map functions must implement the Mapper interface, which guarantees that the map function will always be called with a key. The key is an instance of a WritableComparable object, a value that is an instance of a Writable object, an output object, and a reporter.

IdentityMapper.java

```
package org.apache.hadoop.mapred.lib;
import java.io.IOException;
import org.apache.hadoop.mapred.Mapper;
import org.apache.hadoop.mapred.OutputCollector;
import org.apache.hadoop.mapred.Reporter;
import org.apache.hadoop.mapred.MapReduceBase;
/** Implements the identity function, mapping inputs directly to outputs. */
public class IdentityMapper<K, V>
extends MapReduceBase implements Mapper<K, V, K, V> {
/** The identify function. Input key/value pair is written directly to
* output.*/
public void map(K key, V val,
OutputCollector<K, V> output, Reporter reporter)
throws IOException {
output.collect(key, val);
}
}
```

A Simple Reduce Function: IdentityReducer

The Hadoop framework calls the reduce function one time for each unique key. The framework provides the key and the set of values that share that key.

IdentityReducer.java

```
package org.apache.hadoop.mapred.lib;
```

```

import java.io.IOException;
import java.util.Iterator;
import org.apache.hadoop.mapred.Reducer;
import org.apache.hadoop.mapred.OutputCollector;
import org.apache.hadoop.mapred.Reporter;
import org.apache.hadoop.mapred.MapReduceBase;
/** Performs no reduction, writing all input values directly to the output. */
public class IdentityReducer<K, V>
extends MapReduceBase implements Reducer<K, V, K, V> {
Chapter 2 THE BASICS OF A MAPREDUCE JOB 35
/** Writes all keys and values directly to output. */
public void reduce(K key, Iterator<V> values,
OutputCollector<K, V> output, Reporter reporter)
throws IOException {
while (values.hasNext()) {
output.collect(key, values.next());
}
}
}

```

If you require the output of your job to be sorted, the reducer function must pass the key objects to the `output.collect()` method unchanged. The reduce phase is, however, free to output any number of records, including zero records, with the same key and different values.

4. Explain HDFS Concepts in detail?

Blocks

A disk has a block size, which is the minimum amount of data that it can read or write. Filesystem blocks are typically a few kilobytes in size, while disk blocks are normally 512 bytes. HDFS has the concept of a block, but it is a much larger unit—64 MB by default. Files in HDFS are broken into block-sized chunks, which are stored as independent units. Unlike a filesystem for a single disk, a file in HDFS that is smaller than a single block does not occupy a full block's worth of underlying storage.

Simplicity is something to strive for in all systems, but is especially important for a distributed system in which the failure modes are so varied. The storage subsystem deals with blocks, simplifying storage management and eliminating metadata concerns.

Namenodes and Datanodes

An HDFS cluster has two types of node operating in a master-worker pattern: a *namenode* (the master) and a number of *datanodes* (workers). The namenode manages the filesystem namespace. It maintains the filesystem tree and the metadata for all the files and directories in the tree.

The namenode also knows the datanodes on which all the blocks for a given file are located, however, it does not store block locations persistently, since this information is reconstructed from datanodes when the system starts.

A *client* accesses the filesystem on behalf of the user by communicating with the namenode and datanodes. Datanodes are the workhorses of the filesystem. Hadoop can be configured so that the namenode writes its persistent state to multiple filesystems. These writes are synchronous and atomic. The usual configuration choice is to write to local disk as well as a remote NFS mount.

It is also possible to run a *secondary namenode*, which despite its name does not act as a namenode. Its main role is to periodically merge the namespace image with the edit log to prevent the edit log from becoming too large. The secondary namenode usually runs on a separate physical machine, since it requires plenty of CPU and as much memory as the namenode to perform the merge. It keeps a copy of the merged namespace image, which can be used in the event of the namenode failing.

HDFS Federation

The namenode keeps a reference to every file and block in the filesystem in memory, which means that on very large clusters with many files, memory becomes the limiting factor for scaling.

HDFS Federation, introduced in the 0.23 release series, allows a cluster to scale by adding namenodes, each of which manages a portion of the filesystem namespace. For example, one namenode might manage all the files rooted under */user*, say, and a second

Namenode might handle files under */share*. Under federation, each namenode manages a *namespace volume*, which is made up of the metadata for the namespace, and a *block pool* containing all the blocks for the files in the namespace. Namespace volumes are independent of each other, which means namenodes do not communicate with one another, and furthermore the failure of one namenode does not affect the availability of the namespaces managed by other namenodes.

Block pool storage is *not* partitioned, however, so datanodes register with each namenode in the cluster and store blocks from multiple block pools.

HDFS High-Availability

The combination of replicating namenode metadata on multiple filesystems, and using the secondary namenode to create checkpoints protects against data loss, but does not provide high-availability of the filesystem. The namenode is still a *single point of failure* (SPOF), since if it did fail, all clients—including MapReduce jobs—would be unable to read, write, or list files, because the namenode is the sole repository of the metadata and the file-to-block mapping. In such an event the whole Hadoop system would effectively be out of service until a new namenode could be brought online. In the event of the failure of the active namenode, the standby takes over its duties to continue servicing client requests without a significant interruption.

A few architectural changes are needed to allow this to happen:

- The namenodes must use highly-available shared storage to share the edit log.

When a standby namenode comes up it reads up to the end of the shared edit log to synchronize its state with the active namenode, and then continues to read new entries as they are written by the active namenode.

- Datanodes must send block reports to both namenodes since the block mappings are stored in a namenode's memory, and not on disk.

- Clients must be configured to handle namenode failover, which uses a mechanism that is transparent to users.

If the active namenode fails, then the standby can take over very quickly since it has the latest state available in memory: both the latest edit log entries, and an up-to-date block mapping. The actual observed failover time will be longer in practice (around a minute or so), since the system needs to be conservative in deciding that the active namenode has failed.

Failover and fencing

The transition from the active namenode to the standby is managed by a new entity in the system called the *failover controller*. Failover controllers are pluggable, but the first implementation uses ZooKeeper to ensure that only one namenode is active. Each namenode

runs a lightweight failover controller process whose job it is to monitor its namenode for failures and trigger a failover should a namenode fail.

Failover may also be initiated manually by an administrator, in the case of routine maintenance, for example.

In the case of an ungraceful failover, however, it is impossible to be sure that the failed namenode has stopped running. The HA implementation goes to great lengths to ensure that the previously active namenode is prevented from doing any damage and causing corruption—a method known as *fencing*. The system employs a range of fencing mechanisms, including killing the namenode’s process, revoking its access to the shared storage directory, and disabling its network port via a remote management command. As a last resort, the previously active namenode can be fenced with a technique rather graphically known as *STONITH*, or “shoot the other node in the head”, which uses a specialized power distribution unit to forcibly power down the host machine. Client failover is handled transparently by the client library. The simplest implementation uses client-side configuration to control failover. The HDFS URI uses a logical hostname which is mapped to a pair of namenode addresses, and the client library tries each namenode address until the operation succeeds.

5. Explain Anatomy of a File Read?

The client opens the file it wishes to read by calling `open ()` on the `FileSystem` object, which for HDFS is an instance of `DistributedFileSystem`. `DistributedFileSystem` calls the namenode, using RPC, to determine the locations of the blocks for the first few blocks in the file. The namenode returns the addresses of the datanodes that have a copy of that block.

If the client is itself a datanode, then it will read from the local datanode, if it hosts a copy of the block. The `DistributedFileSystem` returns an `FSDDataInputStream` to the client for it to read data from. `FSDDataInputStream` in turn wraps a `DFSInputStream`, which manages the datanode and namenode I/O.

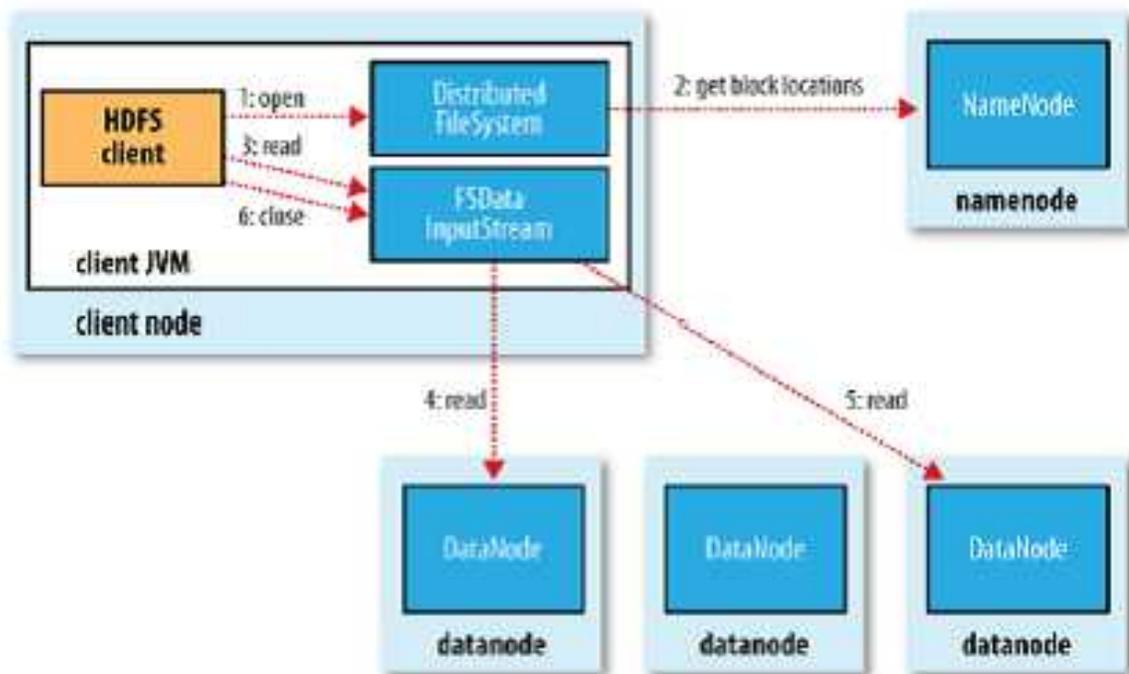


Figure: A client reading data from HDFS

The client then calls `read ()` on the stream. `DFSInputStream`, which has stored the datanode addresses for the first few blocks in the file, then connects to the first (closest) datanode for the first block in the file. Data is streamed from the datanode back to the client, which calls `read ()` repeatedly on the stream. When the end of the block is reached, `DFSInputStream` will close the connection to the datanode, then find the best datanode for the next block. This happens transparently to the client, which from its point of view is just reading a continuous stream.

Blocks are read in order with the `DFSInputStream` opening new connections to datanodes as the client reads through the stream. It will also call the namenode to retrieve the datanode locations for the next batch of blocks as needed. When the client has finished reading, it calls `close ()` on the `FSDDataInputStream`.

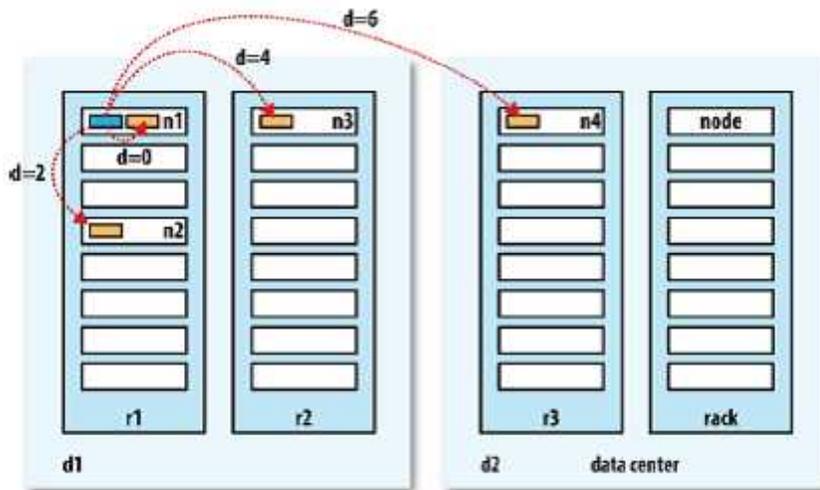


Figure: *Network distance in Hadoop*

During reading, if the `DFSInputStream` encounters an error while communicating with a datanode, then it will try the next closest one for that block. It will also remember datanodes that have failed so that it doesn't needlessly retry them for later blocks.

The `DFSInputStream` also verifies checksums for the data transferred to it from the datanode. If a corrupted block is found, it is reported to the namenode before the `DFSInput Stream` attempts to read a replica of the block from another datanode. One important aspect of this design is that the client contacts datanodes directly to retrieve data and is guided by the namenode to the best datanode for each block. This design allows HDFS to scale to a large number of concurrent clients, since the data traffic is spread across all the datanodes in the cluster.

6. Explain Anatomy of a File write?

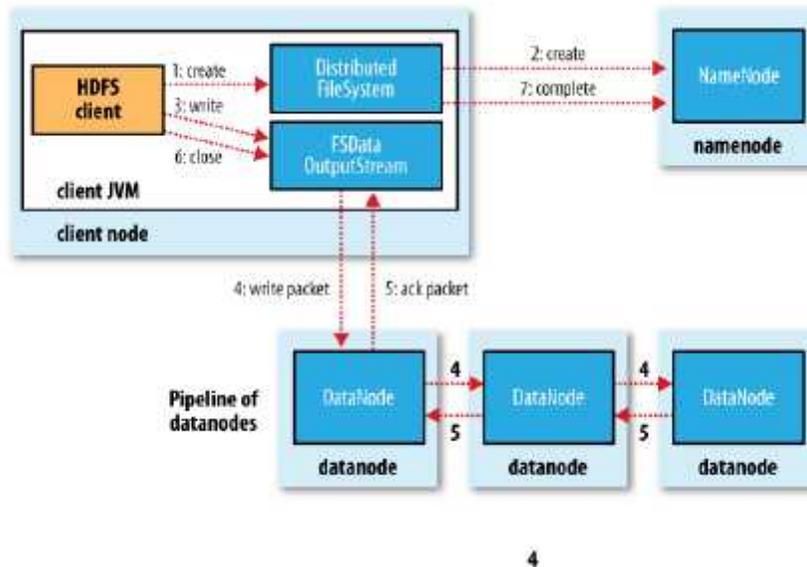


Figure: A client writing data to HDFS

The DistributedFileSystem returns an FSDDataOutputStream for the client to start writing data to. Just as in the read case, FSDDataOutputStream wraps a DFSOutput Stream, which handles communication with the datanodes and namenode. As the client writes data (step 3), DFSOutputStream splits it into packets, which it writes to an internal queue, called the *data queue*. The data queue is consumed by the Data Streamer, whose responsibility it is to ask the namenode to allocate new blocks by picking a list of suitable datanodes to store the replicas. The list of datanodes forms a pipeline—we'll assume the replication level is three, so there are three nodes in the pipeline. The DataStreamer streams the packets to the first datanode in the pipeline, which stores the packet and forwards it to the second datanode in the pipeline. Similarly, the second datanode stores the packet and forwards it to the third (and last) datanode in the pipeline (step 4). DFSOutputStream also maintains an internal queue of packets that are waiting to be acknowledged by datanodes, called the *ack queue*. A packet is removed from the ack queue only when it has been acknowledged by all the datanodes in the pipeline (step 5). If a datanode fails while data is being written to it, then the following actions are taken, which are transparent to the client writing the data.

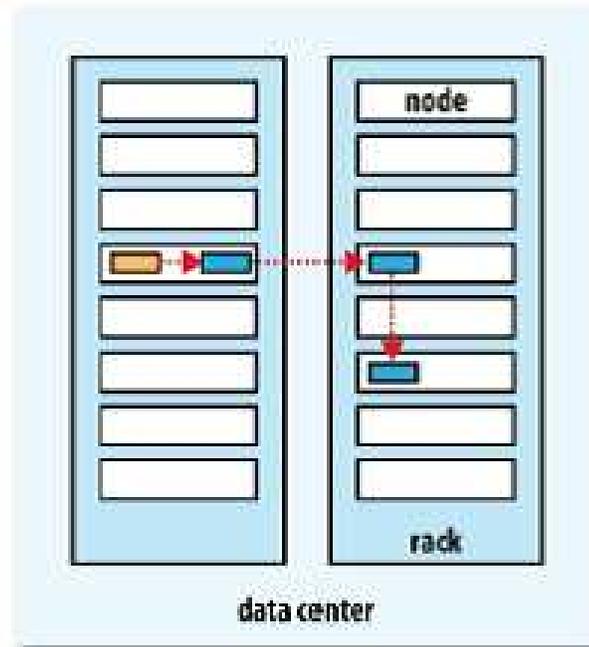


Figure: A typical replica pipeline

First the pipeline is closed, and any packets in the ack queue are added to the front of the data queue so that datanodes that are downstream from the failed node will not miss any packets. The current block on the good datanodes is given a new identity, which is communicated to the namenode, so that the partial block on the failed datanode will be deleted if the failed. Datanode recovers later on. The failed datanode is removed from the pipeline and the remainder of the block's data is written to the two good datanodes in the pipeline. The namenode notices that the block is under-replicated, and it arranges for a further replica to be created on another node. Subsequent blocks are then treated as normal. It's possible, but unlikely, that multiple datanodes fail while a block is being written. As long as `dfs.replication.min` replicas (default one) are written, the write will succeed, and the block will be asynchronously replicated across the cluster until its target replication factor is reached.

When the client has finished writing data, it calls `close ()` on the stream (step 6). This action flushes all the remaining packets to the datanode pipeline and waits for acknowledgments before contacting the namenode to signal that the file is complete (step 7). The namenode already knows which blocks the file is made up.

CS 6704 RESOURECE MANAGEMENT AND TECHNIQUES

UNIT – 1

LINEAR PROGRAMMING

PART – A

1. What is operation research?

There are many definitions of operations research. According to one such definition “Operation research is the application of scientific methods to complex problems arising from operations involving large systems of men, machines, materials and money in Industry, business, government and defence.

2. What are the phases of an operations research study?

- (i) Problem formulation
- (ii) Construction of a mathematical model
- (iii) Controlling and updating
- (iv) Testing the model and its solution

3. Define a feasible solution.

(AU 2016)

Any solution to a LPP which satisfies the non-negativity restrictions of the LPP is called it’s feasible solution.

4. Define optimal solution.

(AU

2016)

Any feasible solution which optimizes (minimizes or maximizes) the objective function is called its optimal solution.

5. What is the difference between feasible solution and basic feasible solution?

The solution of m basic variables when each of the $(n-m)$ non-basic variables is set to Zero is called basic solution.

A basic solution in which all the basic variables are 0 is called a basic feasible solution.

6. Define unbounded solution.

If the value of the objective functions z can be increased or decreased indefinitely. Such Solutions are called unbounded solutions.

7. What are the two forms of a LPP?

The two forms of LPP are (i) standard form and (ii) Canonical form

8. What do you mean by standard form of LPP?

In standard form, irrespective of the objective function namely maximize or minimize, all the constraints are expressed as equations, also right hand side constants are non- negative i.e., all the variables are non-negative.

9. What do you mean by canonical form of LPP?

In canonical form, if the objective function is of maximization, then all the constraints Other than non-negativity conditions are ‘ \leq ’ type. Similarly, if the objective function is minimization, all the constraints are ‘ \geq ’ type.

10. What are the limitations of LPP?

- (i) For large problems having many limitations and constraints, the computational difficulties are enormous even when computers are used.
- (ii) Many times it is not possible to express both the objective function and constraints in linear form.
- (iii) The solution variables may have any values. Sometimes the solution variables are restricted to take only integer values.
- (iv) This method does not take only integer values.

11. Define Deterministic model.

Deterministic model is a model which does not take uncertainty into account.

12. What is an example of Descriptive model?

An opinion poll, any survey.

13. Define the term Iconic model with example.

This is a physical, or pictorial representation of various aspects of a system.

Example: Toy, Miniature model of a building, scaled up model of a cell in biology etc.

14. What are slack variables?

The non-negative variable which added to LHS of the constraint to convert the inequality ‘ \leq ’, into an equation is called the slack variable.

$$\sum_{j=1}^n a_{ij}x_j + s_i = b_i \quad i=1,2,\dots,m \quad \text{Where } s_i \text{ are called the slack variables.}$$

15. What are surplus variables?

$$\sum_{j=1}^n a_{ij}x_j - s_i = b_i \quad i=1,2,\dots,m$$

The non-negative variable which is removed from LHS of the constraint to convert the inequality ‘ \geq ’, into an equation is called the surplus variable.

16. What is meant by decision variable?

While making mathematical modeling of operation research problems, the variables which are used and the value of which gives the solution are the decision variables.

17. Define artificial variable.

Any non-negative variable which is introduced in the constraint in order to get the initial basic feasible solution is called artificial variable.

18. What are the methods used to solve an LPP involving artificial variables?

- (i) Big M method or penalty method
- (ii) Two-phase simplex method

19. What is degeneracy?

A solution is degenerate if one or more basic variables vanished.

20. Explain the importance of the LPP.

In LPP, all the decision variables were allowed to take any non-negative real values as it is quite possible and appropriate to have fractional values in many solutions and which are meaningless in the content of the actual decision problem. This is the main reason where LPP is so important for marginal decisions.

PART- B

1. a. Explain the scope of Operation Research.
(8)
- b. List the phases of OR and explain them.
(8)
2. a. Explain classification of models
(8)
- b. What is an iconic model in the study of operations research?
(8)
3. a. A manufacturer of packing material manufactures two types of packing tins, Round and flat. Major production facilities involved are cutting and joining. The cutting department can process 300 round tins or 500 flat tins per hour. The joining department can process 500 round tins or 300 flat tins per hour. The contribution towards profit for a round tin is the same as that of a flat tin. Formulate a linear

programming problem for maximum contribution.

(8)

- b. A company makes two kinds of leather belts A and B. Each belt of Type-A requires twice as much time as a belt of Type-B. If all belts were of type-B, the company could make 1000 per day. The supply of leather is sufficient for only 800 belts per day. Availability of buckles per day for A and B are respectively 400 and 700. The respective profits per belt are Rs.4 and Rs.3. Formulate this as a LPP and solve it graphically.

(8)

4. a. A firm manufactures two products A and B on which the profits earned per unit are Rs.3 and Rs.4 respectively. Each product is processed on two machines M_1 and M_2 . Product A requires one minute of processing time on M_1 and two minutes on M_2 while B requires one minute on M_1 and one minute on M_2 . Machine M_1 is available for not more than 7 hours 30 minutes while machine M_2 is available for 10 hours during any working day. Find the number of units of products A and B to be manufactured to get maximum profit. Formulate the above as a LPP and solve by graphical method.

(8)

- b. A company produces 2 types of hats. Every hat A require twice as much labor time as the second hat be. If the company produces only hat B then it can produce a total of 500 hats a day. The market limits daily sales of the hat A and hat B to 150 and 250 hats. The profit on hat A and hat B are Rs.8 and Rs.5 respectively. Solve graphically to get the optimal solution.

(8)

5. Solve the following linear programming problem using graphical method. (AU 2016)

$$\text{Maximize } z = 100x_1 + 80x_2$$

$$\text{Subject to the constraints: } 5x_1 + 10x_2 \leq 50$$

$$8x_1 + 2x_2 \leq 16$$

$$3x_1 - 2x_2 \leq 6; \quad x_1, x_2 \geq 0$$

(16)

5. a. Use Graphical method to solve the following LPP

$$\text{Maximize } z = 15x_1 + 10x_2$$

$$\text{Subject to the constraints: } 4x_1 + 6x_2 \leq 360$$

$$3x_1 + 0x_2 \leq 180$$

$$0x_1 + 5x_2 \leq 200; \quad x_1, x_2 \geq 0$$

(8)

- b. Solve the following LPP by the graphical method $\text{Max } Z = 3x_1 + 2x_2$

$$\text{Subject to the constraints: } -2x_1 + x_2 \leq 1$$

$$x_1 \leq 2$$

$$x_1 + x_2 \leq 3; \quad x_1, x_2 \geq 0$$

(8)

6. a. Solve the following LPP by the graphical method

$$\text{Minimize } Z = 3x_1 + 5x_2$$

$$\text{Subject to the constraints: } -3x_1 + 4x_2 \leq 12$$

$$x_1 \leq 4$$

$$2x_1 - x_2 \leq -2$$

$$x_2 \leq 2$$

$$2x_1 + 3x_2 = 12 \text{ and } x_1, x_2 \geq 0$$

(8)

- b. Apply graphical method to find non-negative values of x_1 and x_2 which minimize $z = 10x_1 + 25x_2$ subject to $x_1 + x_2 = 50$, $x_1 \geq 20$ and $x_2 \leq 40$.

7. a. Describe simplex method for solving linear programming problem.

(8)

- b. Express the following LPP in the canonical form, Maximize $z = 3x_1 + x_2$
 Subject to $x_1 + 2x_2 = 5$
 $3x_1 + 5x_2 = 6$ and $x_1, x_2 \geq 0$

(8)

8. a. Use simplex method to solve the following LPP, Maximize $z = 4x_1 + 10x_2$

Subject to the constraints $2x_1 + x_2 = 50$

$$2x_1 + 5x_2 = 100$$

$$2x_1 + 3x_2 = 90 \text{ and } x_1, x_2 \geq 0$$

(8)

- b. Solve the following problem by simplex method, Minimize $z = x_1 - 3x_2 + 2x_3$

Subject to the constraints $3x_1 - x_2 + 2x_3 = 7$

$$-2x_1 + 4x_2 = 12$$

$$-4x_1 + 3x_2 + 8x_3 = 10 \text{ and } x_1, x_2, x_3 \geq 0$$

(8)

10. Solve the following LPP by simplex method.

(AU)

2016)

Maximize $z = 4x_1 + x_2 + 3x_3 + 5x_4$

Subject to the constraints $4x_1 - 6x_2 - 5x_3 + 4x_4 = -20$

$$3x_1 - 2x_2 + 4x_3 + x_4 = 10$$

$$8x_1 - 3x_2 + 3x_3 + 2x_4 = 20 \text{ and } x_1, x_2, x_3, x_4 \geq 0$$

(16)

9. a. Solve by simplex method: Maximize $z = 3x_1 + 5x_2 + 4x_3$

Subject to the constraints $2x_1 + 3x_2 = 8$

$$2x_2 + 5x_3 = 10$$

$$3x_1 + 2x_2 + 4x_3 = 15 \text{ and } x_1, x_2, x_3 \geq 0$$

(8)

- b. A manufacturer is engaged in producing 2 products X and Y, the contribution margin being Rs.15 and Rs.45 respectively. A unit of product X requires 1 unit of facility A and 0.5 unit of facility B. A unit of product Y requires 1.6 units of facility A, 2.0 units of facility B and 1 unit of raw material C. The availability of total facility A and B and raw material C during a particular time period are 240, 162 and 50 units respectively. Find out the product mix which will maximize the contribution margin, by simplex method.

(8)

10. a. Use Two phase simplex method to Maximize $z = 5x_1 - 4x_2 + 3x_3$

Subject to the constraints $2x_1 + x_2 - 6x_3 = 20$

$$6x_1 + 5x_2 + 10x_3 = 76$$

$$8x_1 - 3x_2 + 6x_3 = 50 \text{ and } x_1, x_2, x_3 \geq 0$$

(8)

b. Use Two phase simplex method to Maximize $z = -4x_1 - 3x_2 - 9x_3$

Subject to the constraints $2x_1 + 4x_2 + 6x_3 = 15$

$$6x_1 + x_2 + 6x_3 = 12 \text{ and } x_1, x_2, x_3 \geq 0$$

(8)

11. a. Use Artificial variable technique to solve the following LPP,

$$\text{Maximize } = x_1 + 2x_2 + 3x_3 - x_4$$

$$\text{Subject to the constraints } x_1 + 2x_2 + 3x_3 = 15$$

$$2x_1 + x_2 + 5x_3 = 20$$

$$x_1 + 2x_2 + x_3 + x_4 = 10 \text{ and } x_1, x_2, x_3, x_4 \geq 0$$

(8)

b. Solve the following LPP by Big-M method. Minimize $z = 5x_1 - 6x_2 - 7x_3$

$$\text{Subject to: } x_1 + 5x_2 - 3x_3 = 15$$

$$5x_1 - 6x_2 + 10x_3 = 20$$

$$x_1 + x_2 + x_3 = 5 \text{ and } x_1, x_2, x_3 \geq 0$$

(8)

12. a. What is Sensitivity analysis in an LPP? Discuss its significance fully.

(8)

b. Discuss the effect of (i) variation of b_i (ii) variation of C_j (iii) variation of a_{ir}

(8)

13. a. Consider the LPP, Max $Z = 5x_1 + 12x_2 + 4x_3$

$$\text{Subject to } x_1 + 2x_2 = 5$$

$$5x_1 - x_2 + 2x_3 = 2 \text{ and } x_1, x_2, x_3 \geq 0$$

(i) Solve the LPP

(ii) Discuss the effect of changing the requirement vector from $\begin{pmatrix} 5 \\ 2 \end{pmatrix}$ to $\begin{pmatrix} 7 \\ 2 \end{pmatrix}$, on the optimal solution.

(8)

b. Consider the LPP, Max $Z = 5x_1 + 3x_2$

$$\text{Subject to } 3x_1 + 5x_2 = 15$$

$$5x_1 + 6x_2 = 10 \text{ and } x_1, x_2 \geq 0$$

(i) Solve the LPP

(ii) Find how far the component C_1 of C can be increased without affecting the optimality of the solution.

(8)

UNIT-II DUALITY AND NETWORKS PART-A

1. Define dual of LPP.

For every LPP there is a unique LPP associated with it involving the same data closely related optimal solution. The original problem is then called the primal problem while the other is called its dual problem.

2. What are the characteristics of a primal and dual problem?

(AU

2016)

There are situations in which the value of a variable in a LP problem can be positive, negative or zero. These are called unrestricted variables. An unrestricted variable x_i can be converted in to a non negative variables by using the substitution $x_i = x_i' - x_i''$, $x_i' > 0, x_i'' > 0$

3. State the optimality condition in dual simplex method.

In dual simplex method, if all $Z_j - C_j \geq 0$ and also all $X_{Bi} \geq 0$ then the current

solution is an optimum feasible solution.

4. What do you mean by shadow pricing?

(AU 2016)

If either the primal or the dual problem has a finite optimal solution, then the other problem also has finite optimal solution and the values of the objective functions are equal. i.e., $\text{Max } Z = \text{Min } W$. The solution of the other problem can be read from the $(z_j - c_j)$ row below the columns of slack, surplus variables. The values of the dual variables are called shadow prices.

5. What is the difference between regular simplex method and dual simplex method?

In regular simplex method we first determine the entering variable and then the leaving variable while in the case of dual simplex method we first determine the leaving variable and then the entering variable.

6. What do you understand by transportation problem?

T.P is a special class of LPP in which we transport the commodity (single product) from the source to a destination in such a way that the total transportation problem is minimum.

7. list any three approaches used with T.P for determining the starting solution.

- (i) North west corner rule
- (ii) Least cost method (or) Matrix minima method
- (iii) Vogel's approximation method

8. What do you mean by degeneracy in a T.P?

If the number of occupied cells in a $m \times n$ T.P is less than $m+n-1$ then it's called a degeneracy in a T.P.

9. What do you mean by an unbalanced T.P?

Any T.P is said to be unbalanced if

$$\sum_{i=1}^m a_i \neq \sum_{j=1}^n b_j$$

i.e, if the total supply is not equal to the total demand.

10. How do you convert the unbalanced T.P into a balanced one?

The unbalanced T.P can be converted into a balanced one by adding a dummy row (source) with cost zero and the excess demand is entered as a rim requirement if total supply < total demand. On the other hand the total supply > total demand, we introduce a dummy column (destination) with cost zero 0 and the excess supply is entered as a rim requirement for the dummy destination.

11. What is an assignment problem?

The problem of assigning the number of jobs to equal number of facilities (machines or persons or destinations) at minimum cost or maximum profit is called an assignment problem.

12. What do you mean by an unbalanced assignment problem?

If the number of rows is not equal to the number of columns in the cost matrix of the assignment problem or if the cost matrix of the given assignment problem is not a square matrix, then the given assignment problem is said to be unbalanced.

13. State the difference between the T.P and A.P.

The major difference between T.P and A.P are,

- (i) The cost matrix in T.P is not necessarily a square matrix, whereas in A.P, it is a

square matrix.

- (ii) supply and demand at any source at any destination may be positive quantity a_i, b_j in T.P whereas in A.P it will be 1 i.e., $a_i = b_j = 1$

14. What is the objective of the travelling salesman problem?

The objective of the travelling salesman problem is that the salesman has to visit Various cities, not visiting twice to the same place and return to the starting place by spending minimum transportation cost.

15. State the necessary and sufficient condition for a transportation problem to have a solution.

(AU 2016)

Shortest – Path problem which the objective is to find the shortest distance and the corresponding path from a given source node to a given destination node in a given distance network.

16. What is the name of the method used in getting the optimum assignment?

Hungarian Method

PART-B

1. Find the maximum of $Z = 6x + 8y$ subject to $5x + 2y \leq 20$, $x + 2y \leq 10$, $x, y \geq 0$ by (16) solving its dual problem.

2. Solve by Dual simplex method the following LPP (16)

$$\begin{aligned} \text{Minimize } Z &= 5x_1 + 6x_2 \\ \text{Subject to the constraints} \\ x_1 + x_2 &\leq 2, \\ 4x_1 + x_2 &\leq 4 \\ x_1, x_2 &\geq 0 \end{aligned}$$

3. Use dual simplex method to solve the LPP (16)

$$\begin{aligned} \text{Maximize } Z &= -2x_1 - x_3 \\ \text{Subject to the constraints} \\ x_1 + x_2 - x_3 &\leq 5, \\ x_1 - 2x_2 + 4x_3 &\leq 8, \\ x_1, x_2, x_3 &\geq 0 \end{aligned}$$

4. Use dual simplex method to solve the LPP. (AU 2016)

$$\begin{aligned} \text{Maximize } Z &= -3x_1 - 2x_2 \\ \text{Subject to the constraints} \\ x_1 + x_2 &\leq 1, \\ x_1 + x_2 &\leq 7, \\ x_1 + 2x_2 &\leq 7 \\ x_2 &\leq 3 \\ \text{and } x_1, x_2 &\geq 0 \end{aligned}$$

- 4.(a) Given the linear programming problem solve by duality (8)

$$\begin{aligned} \text{Maximize } z &= 3x_1 + 5x_2 \\ \text{subject to : } x_1 + x_2 &\leq 1, 2x_1 + 3x_2 \leq 1 \text{ and } x_1, x_2 \geq 0 \end{aligned}$$

- (b) Consider the L.P.P by duality : Maximize $z = -x_1 + 2x_2 - x_3$
(8)

$$\begin{aligned} \text{Subject to the constraints: } & 3x_1 + x_2 - x_3 = 10 \\ & -x_1 + 4x_2 + x_3 = 6 \\ & x_2 + x_3 = 4 \text{ and } x_1, x_2, x_3 \geq 0 \end{aligned}$$

- 5.(a) Determine an initial basic feasible solution to the following transportation problem using NWCR
(8)

	D ₁	D ₂	D ₃	D ₄	Supply
O ₁	6	4	1	5	14
O ₂	8	9	2	7	16
O ₃	4	3	6	2	5
Required	6	10	15	4	35

- (b) Obtain an initial basic feasible solution to the following TP using Matrix minima method
(8)

	D ₁	D ₂	D ₃	D ₄	Supply
O ₁	1	2	3	4	6
O ₂	4	3	2	0	8
O ₃	0	2	2	1	10
Required	4	6	8	6	24

6. Obtain an initial basic feasible solution to the following TP using VAM **(16)**

	D ₁	D ₂	D ₃	D ₄	Supply
O ₁	11	13	17	14	250
O ₂	16	18	14	10	300
O ₃	21	24	13	10	400
Demand	200	225	275	250	950

7. Solve the following transportation problem starting with the initial solution obtained by VAM
(16)

	D ₁	D ₂	D ₃	D ₄	Supply
O ₁	2	2	2	1	3
O ₂	10	8	5	4	7
O ₃	7	6	6	8	5
Required	4	3	4	4	15

8. Consider the problem of assigning four sales persons to four different sales regions as shown in the following table such that the total sales is maximized. (AU

2016)(16)

		Sales region			
		1	2	3	4
Salesman	1	10	22	12	14
	2	16	18	22	10
	3	24	20	12	18
	4	16	14	24	20

The cell entries represent annual sales figures in lakhs of rupees. Find the optional allocation of the sales persons to different regions.

8. A company has 3 plants A,B, and C , three warehouses X,Y,Z. A number of units (16)

available at the plants is 60,70,80 and the demand at X,Y,Z are 50,80,80 respectively. The unit cost of the transportation is given in the following table

	X	Y	Z
A	8	7	3
B	3	8	9
C	11	3	5

Find the allocation so that the total transportation cost is minimum.

9. A company has 4 machines to do 3 jobs. Each job can be assigned to one and only one machine. The cost of each job on each machine is given below. Determine the (16)

job assignments which will minimize the total cost.

18	24	28	32
8	13	17	18
10	15	19	22

- 10(a) Rent Car is developing a replacement plan for its car fleet for a 4-year planning horizon that starts January 1, 2001, and terminates December 31, 2004. At the start of each year, a decision is made as to whether a car should be kept in operation or replaced. A car must be in service a minimum of 1 year and a maximum of 3 years. The following table provides the replacement cost as a function of the year a car is acquired and the number of years in operation. (8)

Equipment acquired at start of	Replacement cost(\$) for given years in operation		
	1	2	3
2001	4000	5400	9800
2002	4300	6200	8700
2003	4800	7100	--

2004	490	--	--
------	-----	----	----

(b) Assuming that a

car must be kept in service at least 2 years, with a maximum service life of 4 years. The planning horizon is from the start of 2001 to the end of 2005. The following table provides the necessary data (8)

Year acquired	Replacement cost(\$) for given years in operation		
	1	2	3
2001	3800	4100	6800
2002	4000	4800	7000
2003	4200	5300	7200
2004	4800	5700	--
2005	5300	--	--

UNIT-III INTEGER DYNAMIC PROGRAMMING PART-A

1. What do you mean by integer programming problem?

An LPP in which some or all of the variables in the optimal solution are restricted to assume non – negative integer values is called an integer programming problem.

2. Define a pure integer programming problem.

In a LPP if all the variables in the optimal solution are restricted to assume non-negative integer values then it is called a pure IPP.

3. Mention some important applications of integer programming problem. (AU 2016)

Capital budgeting, Construction scheduling, Routing and shipping schedule, Capacity expansion.

4. Define a mixed integer programming problem.

In a LPP if only some of the variables in the optimal solution are restricted to assume non – negative integer values, while the remaining variables are free to take any non-negative values then it is called a mixed IPP.

5. Differentiate between pure and mixed IPP.

In a pure IPP all the variables in the optimal solution are restricted to assume non-negative integer values. Whereas in mixed IPP, only some of the variables in the optimal solutions are restricted to assume non-negative integer values.

6. What are the methods used in solving IPP (AU 2016)

There are two methods namely

1. Cutting methods (Gomory's cutting plane method)
2. Search method (Branch and Bound typing)

7. Explain Gomorian constraint (or) Fractional Cut constraint.

A new constraint introduced to the problem such that the new set of feasible solution includes all the original feasible integer solution but does not include the optimum non-integer solution initially found. This new constraint is called Gomorian constraint (or) Fractional Cut constraint

7. Where is branch and bound method used?

This method is an enumeration method which is used when all feasible integer points are not enumerated.

8. What is dynamic programming?

Many decision making problems involve a process that takes place in multiple stages in such a way that in each stage the process is dependent and the strategy chosen. Such type of problems are called dynamic programming problem.

9. Define the terms in dynamic programming : stage, state ,state variables

Stage : A stage may be defined as a portion of the problem that possess a set of mutually exclusive alternatives from which the best alternative is to be selected.

State : States are various possible conditions in which the system may find itself at that stage of the problem

State Variables : The current situation of the system at a stage is described by a set of variables called state variables.

10. Give a few applications of DPP.

(i) It is used to determine the optimal combination of advertising media (TV, radio, newspapers) and the frequency of advertising.

(ii) It has been used to determine the inventory level and for formulating the inventory recording.

11. State Bellman's principle of optimality.

It states that "an optimal policy has the property that whatever be the initial state and the initial decisions , the remaining decisions must constitute an optimal policy for the state resulting from the first decision".

12. What are the advantages of Dynamic programming?

The decision making process consist of selecting a combination of plans from a large number alternative combinations, which also need a lot of computational work, where too much time is involved. Also the number of combinations is very large. These drawbacks can be avoided by using DPP as it divides a given problem in sub-problems or stages. Only one stage is considered at a time and various feasible combinations are eliminated by reducing the volume of computations.

13. Write the any two need of dynamic programming ?

(i) All the decisions of a combination are specified

(ii) The number of combinations is so large.

14. Write the any two characteristics of dynamic programming problems?

(i) The problem can be divided in to stages, with a policy decision required at each stage.

(ii) The effect of the policy decision at each stage is to transform the current state into a state associated with the next stage.

15. Define forward computational procedure.

If the dynamic programming problem is solved by using the recursive equation starting from the first through the last stage, i.e., obtaining the sequence $f_1, f_2, f_3, \dots, f_n$ of the optimal solutions. This computation is called the forward computational procedure.

PART B

1. Find an optimum integer solution to the following LPP

(16)

Maximize $Z = x_1 + 2x_2$

Subject to the constraints

$2x_2 \leq 7,$

$x_1 + x_2 \leq 7$

$$2x_2 = 11$$

$x_1, x_2 \geq 0$, x_1, x_2 are integers

2. Solve the following integer programming problem (16)

$$\text{Maximize } Z = 2x_1 + 20x_2 - 10x_3$$

Subject to the constraints

$$2x_1 + 20x_2 + 4x_3 = 15,$$

$$6x_1 + 20x_2 + 4x_3 = 20,$$

$x_1, x_2, x_3 \geq 0$ and are integers

3. Solve the following LPP.

(AU 2016)(16)

$$\text{Minimize } Z = -2x_1 - 3x_2$$

Subject to the constraints

$$2x_1 + 2x_2 = 7$$

$$x_1 \leq 2$$

$$x_2 \leq 2$$

and $x_1, x_2 \geq 0$ and are integer

4. A manufacturer of baby-dolls makes two types of dolls, doll X and doll Y. Processing of these two dolls is done on two machines, A and B. Doll X requires two hours on machine A and six hours on machine B. Doll Y requires five hours on machine A and also five hours on machine B. There are sixteen hours of time per day available on machine A and thirty hours on machine B. The profit gained on both the dolls is same, i.e., one rupee per doll. What should be the daily production of each of the two dolls?

(a) Set up and solve the I.P.P

(b) If the optimal solution is not integer valued, use the Gomory technique to derive the optimal solution.

(16)

5. Using Gomory's cutting plane method

(16)

$$\text{Maximize } Z = 2x_1 + 2x_2$$

Subject to the constraints

$$5x_1 + 3x_2 = 8,$$

$$2x_1 + 4x_2 = 8$$

$x_1, x_2 \geq 0$ and all are integers

6. Solve the following mixed integer programming problem by using Gomory's cutting plane method

(16)

$$\text{Maximize } Z = x_1 + x_2$$

Subject to the constraints

$$3x_1 + 2x_2 = 5,$$

$$x_2 \leq 2$$

$x_1, x_2 \geq 0$, x_1 is an integer

7. Solve the following mixed integer programming problem:

(16)

$$\text{Maximize } Z = x_1 + x_2$$

Subject to the constraints

$$2x_1 + 5x_2 = 16$$

$$6x_1 + 5x_2 = 30$$

$x_2 \geq 0, x_1$, non negative integers

8. Solve the following mixed integer programming problem:
(16)

$$\begin{aligned} & \text{Minimize } Z = x_1 - 3x_2 \\ & \text{Subject to the constraints} \\ & \quad x_1 + x_2 \leq 5 \\ & \quad -2x_1 + 4x_2 \leq 11 \\ & \quad x_1, x_2 \geq 0 \text{ and } x_2 \text{ is an integer.} \end{aligned}$$

9. Solve the following mixed integer programming problem:
(16)

$$\begin{aligned} & \text{Minimize } Z = 10x_1 + 9x_2 \\ & \text{Subject to the constraints} \\ & \quad x_1 \leq 8 \\ & \quad x_2 \leq 10 \\ & \quad 5x_1 + 3x_2 \leq 45 \\ & \quad x_1, x_2 \geq 0 \text{ and } x_1 \text{ is an integer.} \end{aligned}$$

10. Solve the following all integer programming problem using the Branch and bound method. (16)

$$\begin{aligned} & \text{Minimize } Z = 3x_1 + 2.5x_2 \\ & \text{Subject to the constraints} \\ & \quad x_1 + 2x_2 \leq 20, \\ & \quad 3x_1 + 2x_2 \leq 50 \\ & \quad \text{and } x_1, x_2 \text{ are nonnegative integers} \end{aligned}$$

11. Use Branch and Bound technique to solve the following
(16)

$$\begin{aligned} & \text{Maximize } Z = x_1 + 4x_2 \\ & \text{Subject to the constraints} \\ & \quad 2x_1 + 4x_2 \leq 7, \\ & \quad 5x_1 + 3x_2 \leq 15 \\ & \quad x_1, x_2 \geq 0 \text{ and are integers} \end{aligned}$$

12. Use Branch and Bound technique to solve the following (16)

$$\begin{aligned} & \text{Maximize } Z = 2x_1 + 2x_2 \\ & \text{Subject to the constraints} \\ & \quad 5x_1 + 3x_2 \leq 8, \\ & \quad x_1 + 2x_2 \leq 4 \\ & \quad x_1, x_2 \geq 0 \text{ and integers} \end{aligned}$$

13. Use Branch and Bound technique to solve the following (16)

$$\begin{aligned} & \text{Maximize } Z = 3x_1 + 4x_2 \\ & \text{Subject to the constraints} \\ & \quad 7x_1 + 16x_2 \leq 52, \\ & \quad 3x_1 - 2x_2 \leq 18 \\ & \quad x_1, x_2 \geq 0 \text{ and integers} \end{aligned}$$

14. Use dynamic programming to solve the following LPP
(16)

$$\begin{aligned} & \text{Maximize } Z = x_1 + 9x_2 \\ & \text{Subject to the constraints} \\ & \quad 2x_1 + x_2 \leq 25 \\ & \quad x_2 \leq 11, \end{aligned}$$

$$x_1, x_2 \geq 0$$

15. A student has to take examinations in three courses A, B and C. He has three days available for study. He feels it would be best to denote a whole day to the study of the same course. So that he may study a course for one day, two days or three days or not at all. His estimates of grades he may get by study are as follows

Course / study days	A	B	C
0	0	1	0
1	1	1	1
2	1	3	3
3	3	4	3

- How should he plan to study so that he maximizes the sum of his grades. (AU 2016)(16)

16. The owner of a chain of four grocery stores has purchased six crates of fresh strawberries. The estimated probability distribution of potential sales of the strawberries before spoilage differ among four stores. The following table gives the estimated total expected profit at each store when various number of crates are allocated to it. For administrative reasons, the owner does not wish to split crates between stores. However, he is willing to distribute zero crates to any of his stores. Find the allocation of six crates to four stores so as to maximize the expected profit (16)

Number of Crates	Stores			
	1	2	3	4
0	0	0	0	0
1	4	2	6	2
2	6	4	8	3
3	7	6	8	4
4	7	8	8	4
5	7	9	8	4
6	7	10	8	4

17. Solve the following LPP using dynamic programming principles: (16)

$$\begin{aligned} &\text{Maximize } Z = 2x_1 + 5x_2 \\ &\text{Subject to the constraints} \\ &2x_1 + x_2 \leq 43 \\ &x_2 \leq 46, \end{aligned}$$

$$x_1, x_2 = 0$$

UNIT – IV
CLASSICAL OPTIMIZATION THEORY
PART -A

1. Write the formula for Newton Raphson method

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} \quad (\text{or}) \quad f'(x_k) = \frac{f(x_k)}{x_k - x_{k+1}}$$

2. Which methods use in Equality constraints

Two methods use in equality constraints

- (i) Jacobean method
- (ii) Lagrangean method

3. Define constrained derivatives method (or) Jacobean method

Minimize $z=f(X)$

Subject to $g(X)=0$, where $X=(x_1, x_2, \dots, x_n)$ and $g=(g_1, g_2, \dots, g_m)^T$

The function $f(X)$ and $g(X), i=1, 2, \dots, m$. are twice continuously differentiable.

4. Define Jacobean matrix

If $X=(Y, Z)$, such that $Y=(y_1, y_2, \dots, y_m)$, $Z=(z_1, z_2, \dots, z_{n-m})$. the vectors Y and Z are called

dependent and independent variables.

$$\nabla f(Y, Z) = (\nabla_Y f, \nabla_Z f)$$

$$\nabla g(Y, Z) = (\nabla_Y g, \nabla_Z g) \quad \text{Define } J_{m \times n} = \nabla_Y g = \begin{pmatrix} \nabla_Y g_1 \\ \cdot \\ \cdot \\ \cdot \\ \nabla_Y g_m \end{pmatrix} \text{ is called the Jacobean}$$

matrix

5. Define Control matrix

If $X=(Y, Z)$, such that $Y=(y_1, y_2, \dots, y_m)$, $Z=(z_1, z_2, \dots, z_{n-m})$. the vectors Y and Z are called

dependent and independent variables.

$$\nabla f(Y, Z) = (\nabla_Y f, \nabla_Z f)$$

$$\nabla g(Y, Z) = (\nabla_Y g, \nabla_Z g)$$

Define $C_{m \times n-m} = \nabla_z g = \begin{pmatrix} \nabla_z g_1 \\ \cdot \\ \cdot \\ \cdot \\ \nabla_z g_m \end{pmatrix}$ is called the Control matrix

6. Define sensitivity analysis in the Jacobean method

The Jacobean method can be used to study the effect of small changes in the right hand side of the constraints on the optimal value of f. what is the effect of changing $g_i(X)=0$ to $g_i(X)=\partial g_i$ on the optimal value of f. this type of investigation is called sensitivity analysis

7. Define sensitivity coefficients

$$\frac{\partial f}{\partial g} = \nabla_{y_0} f J^{-1}$$

The effect of small changes ∂g on the optimum value of f can be studied by evaluating the rate of change of f with respect to g. these rates are usually referred to as sensitivity coefficients

8. Define Lagrangean method

In the Jacobean method, let the vector λ represent the sensitivity coefficients

$$\text{(ie) } \frac{\partial f}{\partial g} = \nabla_{y_0} J^{-1} = \lambda$$

$$\partial f - \lambda \partial g = 0$$

The resulting equations together with the constrained equations $g(X)=0$ yields the feasible values of X and λ that satisfy the necessary condition for stationary points, this procedure defines the Lagrangean method for identifying the stationary points of optimization with equality constraints.

9. Define Lagrangean function

Let $L(X, \lambda) = f(X) - \lambda g(X)$

The function L is called lagrangean function and parameters λ is called Lagrange multipliers, by definition these multipliers have the same interpretation as the sensitivity coefficients of the Jacobean method

10. Write down the lagrangian method for Khun-Tucker method for following non linear programming with inequality constraints (AU 2016)

Max $Z=f(x)$, $x=(x_1, x_2, \dots, x_n)$ subject to $g(x) < 0$

The inequality constraints may be covered into equation by using non negative slack Variables.

Let $S_i^2 (\geq 0)$ be the slack quantity added to i^{th} constraint $g(x) = 0$ and define

$$S = (S_1, S_2, \dots, S_m)$$

M= total number of inequality constraints

11. Which method use in inequality constraints

Kuhn tucker conditions only use in inequality constraints

1. 12. Sufficiency of Kuhn Tucker Conditions

- 1) $\nabla_x f(X^*) \quad \quad \quad \} \nabla_x g(X^*) \leq 0$
- 2) $[\nabla_x f(X^*) \quad \quad \quad \} \nabla_x g(X^*)] X^* = 0$
- 3) $X^* \geq 0$
- 4) $g(X^*) \leq b$
- 5) $\}^*(g(X^*) - b) = 0$
- 6) $\}^* \geq 0$

13. Define maximization type objective function.

The stationary points will give the maximum objective function value if the sign of each of the last (n-m) principal minor determinants of the bordered Hessian matrix is the same as that of $(-1)^{m+1}$, ending with the $(2m+1)^{\text{th}}$ principal minor determinant.

14. Define minimization type objective function.

The stationary points will give the minimum objective function value if the sign of each of the last (n-m) principal minor determinants of the bordered Hessian matrix is the same as that of $(-1)^m$, ending with the $(2m+1)^{\text{th}}$ principal minor determinant.

15. Examine $f(x) = 6x^5 - 4x^3 + 10$ for extreme points (AU 2016)

$$f(x) = 6x^5 - 4x^3 + 10$$

$$f'(x) = 30x^4 - 12x^2$$

$$f'(x) = 0 \text{ implies } x = 0, x^2 = 2/5$$

PART - B

1. Consider the linear program, by Jacobean method

(AU

2016) (16)

$$\begin{aligned} \text{Maximize } & z = 2x_1 + 3x_2 \\ \text{subject to } & x_1 + x_2 + x_3 = 5 \\ & x_1 + x_2 + x_4 = 3 \\ & x_1, x_2, x_3, x_4 \geq 0 \end{aligned}$$

2.(a) Consider the linear program,

$$\begin{aligned} \text{Maximize } & f(X) = x_1^2 + 2x_2^2 + 10x_3^2 + 5x_1x_2 \\ \text{subject to } & g_1(X) = x_1 + x_2^2 + 3x_2x_3 - 5 = 0 \\ & g_2(X) = x_1^2 + 5x_1x_2 + x_3^2 - 7 = 0 \end{aligned}$$

Apply Jacobean method to find $\partial f(X)$ in the neighborhood of feasible point (1,1,1),

Assume that this neighborhood is specified by $\partial g_1 = -.01, \partial g_2 = -.02, \text{ and } \partial x_1 = .01$ (8)

(b) Consider the linear program,

$$\begin{aligned} \text{Maximize } f(X) &= x_1^2 + x_2^2 + x_3^2 + x_4^2 \\ \text{subject to } g_1(X) &= x_1 + 2x_2 + 3x_3 - 5x_4 - 10 = 0 \\ g_2(X) &= x_1 + 2x_2 + 5x_3 + 6x_4 - 15 = 0 \end{aligned}$$

Show that by selecting x_3 and x_4 as independent variables, the Jacobean method fails to provide a solution and state the reason

(8)

3.(a) Consider the linear program, by Lagrangean method

(8)

$$\begin{aligned} \text{Maximize } f(X) &= x_1^2 + x_2^2 + x_3^2 \\ \text{subject to } g_1(X) &= x_1 + x_2 + 3x_3 - 2 = 0 \\ g_2(X) &= 5x_1 + 2x_2 + x_3 - 5 = 0 \end{aligned}$$

(b) Solve the following linear programming problem, by Jacobean and Lagrangean methods

(8)

$$\begin{aligned} \text{Maximize } f(X) &= 5x_1 + 3x_2 \\ \text{subject to } g_1(X) &= x_1 + 2x_2 + x_3 - 6 = 0 \\ g_2(X) &= 3x_1 + x_2 + x_4 - 9 = 0 \\ x_1, x_2, x_3, x_4 &\geq 0 \end{aligned}$$

4.(a) Find the optimal solution to the problem

(8)

$$\begin{aligned} \text{Maximize } f(X) &= x_1^2 + 2x_2^2 + 10x_3^2 \\ \text{subject to } g_1(X) &= x_1 + x_2^2 + x_3 - 5 = 0 \\ g_2(X) &= x_1 + 5x_2 + x_3 - 7 = 0 \end{aligned}$$

Suppose that $g_1(X)=.01$ and $g_2(X)=.02$. Find the corresponding change in the optimal Value of $f(X)$

(8)

(b) Solve the following nonlinear programming problem using Lagrangean method

(8)

$$\begin{aligned} \text{Maximize } z &= 4x_1 - 0.02x_1^2 + x_2 - 0.02x_2^2 \\ x_1 + 2x_2 &= 120 \\ x_1, \text{ and } x_2 &\geq 0 \end{aligned}$$

5.(a) Solve the following nonlinear programming problem using Lagrangean method

(8)

$$\begin{aligned} \text{Maximize } z &= 2x_1^2 - 3x_2^2 + 18x_2 \\ 2x_1 + x_2 &= 8 \\ x_1, \text{ and } x_2 &\geq 0 \end{aligned}$$

(b) Solve the following nonlinear programming problem using Lagrangean method (8)

$$\begin{aligned} \text{Maximize } z &= x_1^2 + 2x_2^2 + x_3^2 \\ 2x_1 + x_2 + 2x_3 &= 30 \\ x_1, x_2 \text{ and } x_3 &\geq 0 \end{aligned}$$

6.(a) Solve the following nonlinear programming problem using Lagrangean method (8)

$$\begin{aligned} \text{Maximize } z &= x_1^2 + 2x_2^2 + 1.5x_3^2 \\ 2x_1 + 2x_2 + 3x_3 - 30 &= 0 \\ 3x_1 - 4x_2 + 4x_3 - 20 &= 0 \\ x_1, x_2 \text{ and } x_3 &\geq 0 \end{aligned}$$

(b) Solve the following nonlinear programming problem using Lagrangean multipliers Method

(8)

$$\begin{aligned} \text{Maximize } z &= 4x_1^2 + 2x_2^2 + x_3^2 - 4x_1x_2 \\ \text{subject to } x_1 + x_2 + x_3 &= 15 \\ 2x_1 - x_2 + 2x_3 &= 20 \\ x_1, x_2 \text{ and } x_3 &\geq 0 \end{aligned}$$

7.(a) Solve the following nonlinear programming problem using Lagrangean multipliers Method

(8)

$$\begin{aligned} \text{Maximize } z &= x_1^2 + x_2^2 + x_3^2 \\ \text{subject to } x_1 + x_2 + 3x_3 &= 2 \\ 5x_1 + 2x_2 + x_3 &= 5 \\ x_1, x_2 \text{ and } x_3 &\geq 0 \end{aligned}$$

(b) Obtain the necessary condition for the optimum solution of the following nonlinear Programming problem (NLLP)

(8)

$$\begin{aligned} \text{Minimize } z &= 2x_1^2 - 24x_1 + 2x_2^2 - 8x_2 + 2x_3^2 - 12x_3 + 200 \\ \text{subject to the constraints } x_1 + x_2 + x_3 &= 11 \\ x_1, x_2, x_3 &\geq 0 \end{aligned}$$

8.(a) Obtain the necessary condition for the optimum solution of the following nonlinear Programming problem (NLLP)

(8)

$$\begin{aligned} \text{Maximize } z &= x_1^2 + x_2^2 + 5x_3^2 \\ \text{subject to the constraint s } x_1 + x_2 + 3x_3 &= 2 \\ 5x_1 + x_2 + x_3 &= 5 \\ x_1, x_2, x_3 &\geq 0 \end{aligned}$$

(b) Solve the nonlinear Programming problem (NLLP) using the method of lagrangian Multipliers

(8)

$$\begin{aligned} \text{Optimize } z &= 4x_1^2 + 2x_2^2 + x_3^2 - 4x_1x_2 \\ \text{subject to the constraints } x_1 + x_2 + x_3 &= 15 \\ 2x_1 - x_2 + 2x_3 &= 20 \\ x_1, x_2, x_3 &\geq 0 \end{aligned}$$

9.(a) Solve the nonlinear Programming problem (NLLP) using the method of lagrangian Multipliers

(8)

$$\begin{aligned} \text{Minimize } z &= 6x_1^2 + 5x_2^2 \\ \text{subject to the constraints } x_1 + 5x_2 &= 3 \\ x_1, x_2 &\geq 0 \end{aligned}$$

(b) Solve the nonlinear Programming problem (NLLP) using the method of lagrangian Multipliers

(8)

$$\begin{aligned} \text{Minimize } f(x_1, x_2) &= 3x_1^2 + 2x_1x_2 + 6x_1 + 2x_2 \\ \text{subject to the constraints } 2x_1 - x_2 &= 4 \\ x_1, x_2, x_3 &\geq 0 \end{aligned}$$

10.(a) Solve the nonlinear Programming problem (NLLP) using the method of lagrangian Multipliers

(8)

$$\begin{aligned} \text{Minimize } z &= 2x_1^2 + x_2^2 + 3x_3^2 + 10x_1 + 8x_2 + 6x_3 - 100 \\ \text{subject to the constraints } x_1 + x_2 + x_3 &= 20 \\ x_1, x_2, x_3 &\geq 0 \end{aligned}$$

(b) Solve the following nonlinear programming problem using Kuhn-Tucker conditions

(8)

$$\begin{aligned} \text{Maximize } z &= 3x_1^2 + 14x_1x_2 - 8x_2^2 \\ \text{subject to } 3x_1 + 6x_2 &\leq 72 \\ x_1, x_2 &\geq 0 \end{aligned}$$

11.(a) Solve the following nonlinear programming problem using Kuhn-Tucker conditions

(8)

$$\begin{aligned} \text{Maximize } z &= x_1^2 + x_1x_2 - 2x_2^2 \\ \text{subject to } 4x_1 + 2x_2 &\leq 24 \\ 5x_1 + 10x_2 &\leq 30 \\ x_1, x_2 &\geq 0 \end{aligned}$$

(b) Solve the following nonlinear programming problem using Kuhn-Tucker conditions

(8)

$$\begin{aligned} \text{Maximize } & z = x_1^2 + x_1x_2 - 2x_2^2 \\ \text{subject to } & 4x_1 + 2x_2 \leq 24 \\ & x_1, x_2 \geq 0 \end{aligned}$$

12.(a) Solve the following nonlinear programming problem using Kuhn-Tucker conditions

(8)

$$\begin{aligned} \text{Maximize } & z = 8x_1 + 10x_2 - x_1^2 - x_2^2 \\ \text{subject to } & 3x_1 + 2x_2 \leq 6 \\ & x_1, x_2 \geq 0 \end{aligned}$$

(b) Solve the following nonlinear programming problem using Kuhn-Tucker conditions

(8)

$$\begin{aligned} \text{Maximize } & f(\mathbf{X}) = x_1^3 - x_2^2 + x_1x_3^3 \\ \text{subject to } & x_1 + x_2^2 + x_3 = 5 \\ & 5x_1^2 - x_2^2 - x_3 \geq 2 \\ & x_1, x_2, x_3 \geq 0 \end{aligned}$$

13.(a) Solve the following nonlinear programming problem using Kuhn-Tucker conditions

(8)

$$\begin{aligned} \text{Minimize } & f(\mathbf{X}) = x_1^4 + x_2^2 + 5x_1x_2x_3 \\ \text{subject to } & x_1^2 - x_2^2 + x_3^3 \leq 10 \\ & x_1^3 + x_2^2 + 4x_3^3 \geq 20 \\ & x_1, x_2, x_3 \geq 0 \end{aligned}$$

(b) Use Kuhn Tucker conditions to solve the following NLPP

(8)

$$\begin{aligned} \text{Maximize } & z = 8x_1 + 10x_2 - x_1^2 - x_2^2 \\ \text{subject to the constraints } & 3x_1 + 2x_2 \leq 6 \\ & x_1, x_2 \geq 0 \end{aligned}$$

14. Use Kuhn Tucker conditions to solve the following NLPP

(AU

2016)(16)

$$\begin{aligned} \text{Minimize } & z = x_1^2 + x_2^2 + x_3^2 \\ \text{subject to the constraints } & g_1(\mathbf{X}) = 2x_1 + x_2 - 5 \leq 0 \\ & g_2(\mathbf{X}) = x_1 + x_2 - 2 \leq 0 \\ & g_3(\mathbf{X}) = 1 - x_1 \leq 0 \\ & g_4(\mathbf{X}) = 2 - x_2 \leq 0 \\ & g_5(\mathbf{X}) = -x_3 \leq 0 \end{aligned}$$

15 .(a) Use Kuhn Tucker conditions to solve the following NLPP

(8)

$$\begin{aligned} \text{Minimize } z &= 2x_1 + 3x_2 - x_1^2 - 2x_2^2 \\ \text{subject to the constraints } x_1 + 3x_2 &\leq 6 \\ 2x_1 + x_2 &\leq 4 \\ x_1, x_2 &\geq 0 \end{aligned}$$

(b) Use Kuhn Tucker conditions to solve the following NLPP (8)

$$\begin{aligned} \text{Maximize } z &= 3x_1 + x_2 \\ \text{subject to the constraints } x_1^2 + 3x_2^2 &\leq 5 \\ x_1 - x_2 &\leq 1 \\ x_1, x_2 &\geq 0 \end{aligned}$$

UNIT-V

OBJECT SCHEDULING

PART-A

1. Define activity?

This is a task or job of work, which takes or consumes time and resources. For example, “Build a wall”, Dig foundations for a building”, “Verify the names of debtors in a sales ledger”, etc.

An activity is represented in a network by an arrow. The tail of the arrow indicates where the task begins and the head where the task ends. The arrow points from left to right and it is not drawn to scale.

2. Define event?

This is a point in time and indicates the “start” or “finish” of an activity or activities. An event is represented in a network by a circle or node .

3. Define dummy activity?

This is an activity, which does not consume time or resources. It issued to merely show clear, logical dependencies between activities so as not to violets the rules for drawing networks. It is represented in a network by dotted arrow thus.

4. Define Network?

This is the combination of activities, dummy activities and events in logical sequence according to the rules for drawing networks.

5. State the rules in drawing a NETWORK DIAGRAM?

i. Node 1 represents the ‘start’ of the project. An arc or arcs should lead from node 1 to represent such activity that has no predecessor. A network has only one “start” node.

ii. A node (called the “finish” node) representing the completion of the project should be included in the network. A network has only one “finish” node.

iii. Number the nodes in the network so that the node representing the completion of an activity always has a larger number than the node representing the beginning of an activity.

(There may be more than one numbering scheme)

iv. An activity should not be represented by more than one arc in the network

v. Two nodes can be connected by at most one arc. That is activities should not share the same “start” node (tail event) and “finish” node (head event).

vi. Every activity must have one preceding event (tail event) and one succeeding event(head event).

6. State the FULKERSON’S RULE?

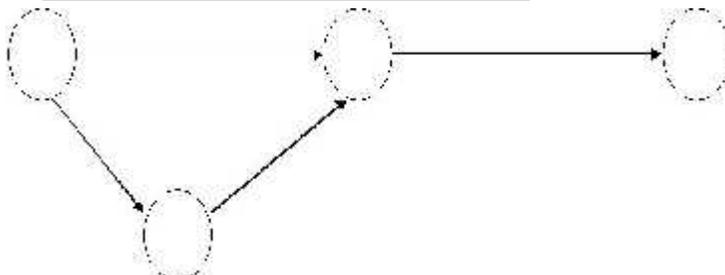
Numbering the Events (Fulkerson's Rule)

1. The initial event which has all outgoing arrows with no incoming arrow is numbered "1".
2. Delete all the arrows coming out from node "1". This will convert some more nodes into initial events. Number these events as 2, 3, 4,
3. Delete all the arrows going out from these numbered events to create more initial events. Assign the next numbers to these events.
4. Continue until the final or terminal node, which has all arrows coming in with no arrow going out is numbered.
5. The Galaxy plc is to buy a small business, Tiny Ltd. The whole procedure involves four activities:
 - A. Develop a list of sources for financing;
 - B. Analyse the financial records of Tiny Ltd;
 - C. Develop a business plan (sales projections, cash flow projections, etc.);
 - D. Submit a proposal to a lending institution.

The precedence relationship of these four activities is described as in the Table below.

Construct the network diagram.

Activity	Immediate Predecessor
A	-
B	-
C	B
D	A, C



7. Define critical path with example?

The critical path of a network gives the shortest time in which the whole project can be completed. It is the chain of activities with the longest duration times. Certain activities are critical to the on-line completion of the project. For example, if you are planning a dinner party and spend all your time setting the table and forget to put the main dish in the oven, the dinner will be late. This also means that other activities have float or slack time (time to spare) available such as laying the table

From the network diagram, the paths are:

1. A through C to D ($4 + 6 + 3 = 13$)
2. A through B to D ($4 + 7 + 3 = 14$)

The longer of this is A through C to D. Therefore, the critical path is: A through C to D

8. Define EARLY TIMES?

The Early Start Time (EST) is the earliest time at which the event corresponding to node i can occur. Calculations begin from the 'start' node and move to the 'finish' node. At each node a number is computed representing the Earliest Start Time (EST).

9. Define LATE TIMES?

The Late Completion Time (LCT) for node i represented by LCT_i is the latest time at which the event corresponding to node i can occur without delaying the completing of the project.

Calculations begin from the 'finish' node and move to the 'start' node. At each node a number is computed representing the Latest Completion Time (LCT) the corresponding event.

10 .Define FLOAT?

Float (spare time or slack time) is the amount of time a path of activities could be delayed without delaying the overall project. A float can only be associated with activities, which are non-critical. By definition, activities on the critical path cannot have a float (spare time). In other words, the float for an activity is the difference between the maximum time available for that activity and the duration of that activity. The float for an activity with given by:

Float = Latest Start Time – Earliest Start Time (F= LST– EST) or (LCT-ECT)

11.What is independent float time?

The *independent float time* of an *activity* is the amount by which the duration of an activity could be extended without affecting the total project time, the time available for subsequent activities or the time available for the preceding activities. = [Free Float_{ij} – (Slack of event i)] or ZERO, whichever is higher. Also EST of following activity – LFT of preceding activity – Duration of current activity or Zero, whichever is higher.

12. What is interfering float time?

The *interfering float time* is the part of total float which causes a reduction in the float of successor activities. It is that portion of the activity float which cannot be consumed without affecting adversely the float of the subsequent activity or activities. = LFT – (EST of following activity) or ZERO, whichever is higher.

13. What are the benefits of PERT?

PERT is useful because it provides the following information:

- Expected project completion time.
- Probability of completion before a specified date.
- The critical path activities that directly impact the completion time.
- The activities that have slack time and that can lend resources to critical path activities.
- Activity starts and end dates.

14. What are the steps in the pert planning process?

PERT planning involves the following steps:

- Identify the specific activities and milestones.
- Determine the proper sequence of the activities.
- Construct a network diagram.
- Estimate the time required for each activity.
- Determine the *critical path*.
- Update the PERT chart as the project progresses.

15. What do you mean by project crashing?

Project Crashing: There are usually compelling reasons to complete the project earlier than the originally estimated duration of critical path computed on the normal basis of a new project.

Direct Cost: This is the cost of the materials, equipment and labour required to perform the activity. When the time duration is reduced the project direct cost increases.

Activity Cost Slope = $(C_c - N_c) \div (N_t - C_t)$

Where, C_c = Crash Cost = Direct cost that is anticipated in completing

an

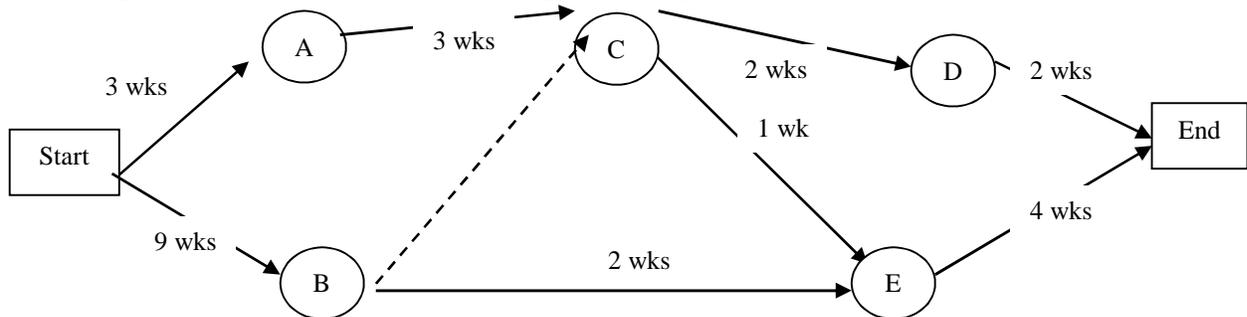
activity within crash time. N_c = Normal Cost = This is the lowest possible direct cost required to complete an activity N_t = Normal Time = Min.

time

required to complete an activity at normal cost. C_t = Crash Time = Min. time required to complete an activity.

16. For this ADM,

- Find the critical path
- Find the duration of the critical path
- How many non-critical paths are there?
- Find the duration of all non-critical paths
- If the duration of Activity C-E changes to 2, what is the effect on the project?
- What activity / activities must be completed before Activity C-D begins?
- If management tells you to complete the project two weeks early, what is the project float? Does the critical path change?



- The critical path is START-B, B-E, E-END
- Duration of critical path = $9 + 2 + 4 = 15$
- Total Number of Paths = 5 (Did you find all 5?)
- Non-critical paths and durations
 - START-A, A-C, C-D, D-END, Duration: $3 + 3 + 2 + 2 = 10$
 - START-A, A-C, C-E, E-END, Duration: $3 + 3 + 1 + 4 = 11$
 - START-B, B-C, C-E, E-END, Duration: $9 + (\text{dummy}) + 1 + 4 = 14$
 - START-B, B-C, C-D, D-END, Duration: $9 + (\text{dummy}) + 2 + 2 = 13$
- If C-E changes from 1 to 2, the critical path(s) and durations would be:
 - START-B, B-E, E-END, Duration: 15
 - START-B, B-C, C-E, E-END, Duration: 15

Yes, you can have more than one critical path. What is the effect on the project? The project is riskier. Always check for a second or more critical path whenever you answer a question that changes the critical path.

- START-A, A-C, and START-B. Remember, B-C is a dummy, not an activity.
- The project float is -2 and the critical path would not change. The question is about PROJECT float. Project float compares the project length to an external due date.

17. what are the steps in drawing CPM/PRET?

Steps for drawing CPM/PERT network:

- Analyze & breakup of the entire project into smaller systems i.e. specific activities and/or events.
- Determine the interdependence & sequence of those activities.
- Draw a network diagram.
- Estimate the completion time, cost, etc. for each activity.
- Identify the critical path (longest path through the network).

6.Update the CPM/PERT diagram as the project progresses.

18. Define critical path
2016)

(AU

Path connecting the first initial node to the very last terminal node of longest duration in any project network is called the critical path

19. Define probability Estimate?

Probability Estimate: It is used to calculate the probability of completing the time within given duration (*Using Normal Distribution*):

$$Z = (T_1 - T_{cp}) / \sigma_t$$

Where, Z = Standard Normal Variate
T₁ = Duration in which we wish to complete the project
T_{cp} = Duration on critical path

σ_t = Standard Deviation of the earliest finish of network = Square root of **sum of variance of all activity durations of critical path**, where
Variance Distribution (σ_t^2) = $[(t_p - t_o)/6]^2$

20. Distinction between PERT and CPM

	PERT		CPM
1.	PERT is used for non-repetitive jobs like planning the assembly of the space.	1.	CPM is used for repetitive job like building a house
2.	It is a probabilistic model.	2.	It is a deterministic model.
3.	It is event-oriented as the results of analysis are expressed in terms of events or distinct points in time indicative of progress.	3.	It is activity-oriented as the result or calculations are considered in terms of activities or operations of the project.
4.	It is applied mainly for planning and scheduling research programmes.	4.	It is applied mainly for construction and business problems.
5.	PERT incorporates statistical analysis and thereby determines the probabilities concerning the time by which each activity or entire project would be completed	5.	CPM does not incorporate statistical analysis in determining time estimates, because time is precise and known.
6.	PERT serves as useful control device as it assists management in controlling a project by calling attention to such delays	6.	It is difficult to use CPM as a control device for the simple reason that one must repeat the entire evaluation of the project each time the changes are introduced into the network

PART-B

1. a. Depict the following dependency relationships by means of network diagrams. (The Alphabets stands for activities)

1. A and B control F; B and C control G.
2. A and B control F; B controls G while C controls G and H.
3. A controls F and G; B controls G while C controls G and H.
4. A controls F and G; B and C control G with H depending upon C.
5. F and G are controlled by A, G and H are controlled by B with H controlled by B and C.

6. A controls F, G and H; B controls G and H with H controlled by C. (8)

b. Develop a network based on the following information;

(8)

<u>Activity</u>	<u>Immediate predecessors</u>
A	-
B	-
C	A
D	B
E	C,D
F	D
G	E

2. a. Construct the project network comprised of activities A to L with the following precedence relationships:

- A,B and C, the first activities of the project can be executed concurrently
- A & B precede D
- B precedes E,F,H
- F and C precede G
- E and H precede I & J
- C,D,F and J precede K
- K precede L
- I, G, and L are terminal activities of the project.

(8)

b. Construct the project network comprised of activities A to P that satisfies the following precedence relationships:

- A,B and C, the first activities of the project can be executed concurrently
- D,E and F follow A
- I and G follow both B and D
- H follows both C & G
- K and L follow I
- J succeeds both E and H
- M and N succeed F, but cannot start until both E and H are completed.
- O succeeds both M and I
- P succeeds J,L and O
- K,N and P are the terminal activities of the project.

(8)

3. a. Construct the project network comprised of activities A to P that satisfies the following precedence relationships:

- A,B and C, the first activities of the project can be executed concurrently
- D,E and F follow A
- I and G follow both B and D
- H follows both C & G
- K and L follow I
- J succeeds both E and H
- M and N succeed F, but cannot start until both E and H are completed.
- O succeeds both M and I
- P succeeds J,L and O
- K,N and P are the terminal activities of the project.

(8)

b. A project consists of a series of tasks labeled A, B, ..., H, I with the following relationships
 (W<X, Y means X and Y cannot start until W is completed; X, Y<W means W cannot start until both X and Y are completed). With this notation construct the network diagram having the following constraints:

$$A < D, E; B, D < F; C < G; B < H; F, G < I$$

(8)

4. a. The footing of a building can be completed in four consecutive sections. The activities for each section include (1) digging, (2) placing steel, and (3) pouring concrete. The digging of one section cannot start until that of the preceding section has been completed. The same restriction applies placing steel & pouring concrete. Develop the project network.

b. A project schedule has the following characteristics: (8)
 Times (week)

1-3	1	5-7	8
2-4	1	6-8	1
3-4	1	7-8	2
3-5	6	8-9	1
4-9	5	8-10	8
		9-10	7

- (i) Construct the PERT network
- (ii) Compute E and L for each event;
Float for each activity; and
- (iii) Find critical path and its duration.

5. a. Given is the following information regarding a project:

Activity	A	B	C	D	E	F	G	H	I	J	K	L
Dependence	-	-	-	AB	B	B	FC	B	EH	EH	CDFJ	K
Duration (days)	3	4	2	5	1	3	6	4	4	2	1	5

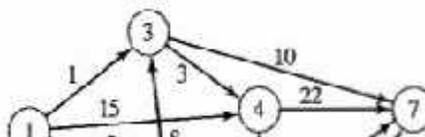
- i) Draw the Network Diagram and identify the Critical Path and Project Duration.
- ii) Find the three types of float (viz. Total, Free and Independent) for each activity. (8)

b. The utility data for a network are given below. Determine the total, free, independent and Interfering floats and identify the critical path.

Activity:	0-1	1-2	1-3	2-4	2-5	3-4	3-6	4-7	5-7	6-7
Duration:	2	8	10	6	3	3	7	5	2	8

(8)

6. a. For the network given below, compute E and L for each event & determine the total, free, independent and interfering floats and identify the critical path.



(8)

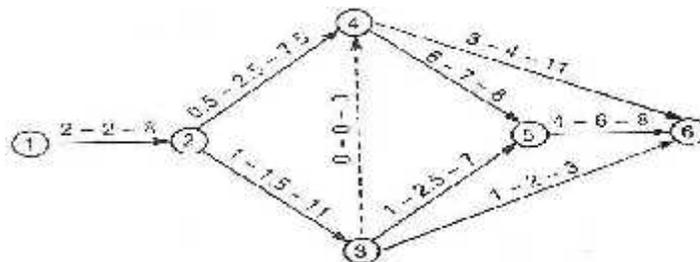
b. The following table gives the activities in a construction project and the time duration of each activity:

Activity	Preceding activity	Normal Time (Days)
A	-	16
B	-	20
C	A	8
D	A	10
E	B, C	6
F	D, E	12

- (i) Draw the activity network of the project.
(ii) Find critical path.
(iii) Find the total float and free-float for each activity.

(8)

7.a. Consider the network shown below. The three time estimates for the activities are given along the arrows. Determine the critical path. What is the probability that the project will be completed in 20 days? (8)



b. Consider the schedule of activities and related information as given below, for the construction of a plant: (8)

Activity	Expected Time (Months)	Variance	Expected Cost (Millions of Rs.)
1-2	4	1	5
2-3	2	1	3
3-6	3	1	4
2-4	6	2	9
1-5	2	1	2
5-6	5	1	12
4-6	9	5	20

5-7	7	8	7
7-8	10	16	14
6-8	1	1	4

Assuming that the cost and time required for one activity is independent of the time and cost of any other activity are expected to follow normal distribution.

Draw a network based on the above data and calculate:

- Critical path
- Expected cost of construction of the plant.
- Expected time required to build the plant.
- The standard deviation of the expected time.

8. a A project consists of seven activities and the time estimates of the activities are furnished as under:

Activity	Optimistic Days	Most likely Days	Pessimistic Days
1-2	4	10	16
1-3	3	6	9
1-4	4	7	16
2-5	5	5	5
3-5	8	11	32
4-6	4	10	16
5-6	2	5	8

- Draw the network diagram.
- Identify the critical path and its duration.
(8)
- What is the probability that project will be completed in 5 days earlier than the critical path duration?
- What project duration will provide 95% confidence level of completion ?

b. The time estimate (in weeks) for the activities of a PERT network is given below:

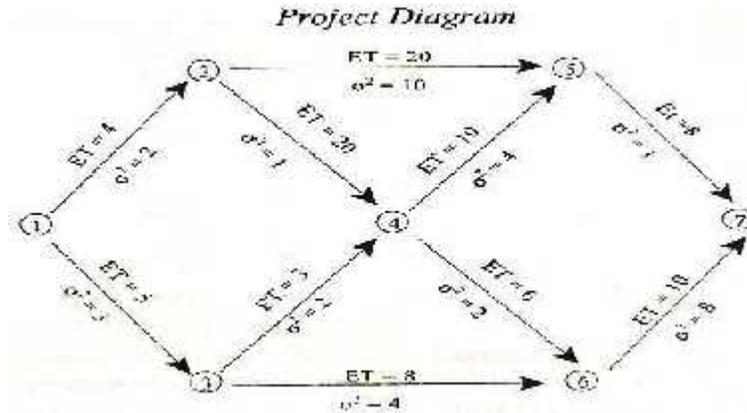
Activity	t_o	t_m	t_p
1-2	1	1	7
1-3	1	4	7
1-4	2	2	8
2-5	1	1	1
3-5	2	5	14
4-6	2	5	8
5-6	3	6	15

- Draw the project network and identify all the paths through it. **(8)**
- Determine the expected project length.
- Calculate the standard deviation and variance of the project length.
- What is the probability that the project will be completed.
 - at least 4 weeks earlier than expected time?
 - no more that 4 weeks later than expected time?
- If the project due date is 19 weeks, what is the probability of not meeting the due date?
What should be the scheduled completion time for the probability of completion to be 90%?

90%?

9. a .Given the following project network, determine:

1. Earliest expected completion time for each event
2. Latest allowable completion time for each event
3. Slack time for each event
4. Critical Path
5. The probability that project will be completed on schedule, if scheduled completion time is 38



(8)

9. b. A small project is composed of seven activities, whose time estimates are listed below. Activities are identified by their beginning (i) and (j) node number.

Activity (i-j)	Estimated durations (in days)		
	Optimistic	Most Likely	Pessimistic
1-2	2	2	14
1-3	2	8	14
1-4	4	4	16
2-5	2	2	2
3-5	4	10	28
4-6	4	10	16

- (a) Draw the project network.
 (b) Find the expected duration and variance for each activity. What is the expected project length. (8)

- 10.a. A project consists of the following activities, whose time estimates are given against each as under:

Activity	Estimated duration (weeks)		
	Optimistic	Most likely	Pessimistic
1-2	3	6	15
1-3	2	5	14
1-4	6	12	30
2-5	2	5	8
2-6	5	11	17
3-6	3	6	15
4-7	3	9	27
5-7	1	4	7
6-7	4	19	28

Required :

- (i) Draw the project net work.
- (ii) Find the expected duration and variance of each activity.
- (iii) Determine the critical path and the expected project duration.
- (iv) What is the probability that the project will be completed in 38 weeks? **(8)**

b. An Engineering Project has the following activities, whose time estimates are listed below:

Activity (i-j)	Estimated Duration (in months)		
	Optimistic	Most Likely	Pessimistic
1-2	2	2	14
1-3	2	8	14
1-4	4	4	16
2-5	2	2	2
3-5	4	10	28
4-6	4	10	16
5-6	6	12	30

- (a) Draw the project network and find the critical path.
- (b) Find the expected duration and variance for each activity. What is the expected project length?
- (c) Calculate the variance and standard deviation of the project length.
- (d) What is the probability that the project will be completed at least eight months earlier than expected time?
- (e) If the project due date is 38 months, what is the probability of not meeting the due date?

Given:

Given:	z	0.50	0.67	1.00	1.33	2.00
	P	0.3085	0.2514	0.1587	0.0918	0.0228

11.a. A civil engineering firm has to bid for the construction of a dam. The activities and their time estimates are given below:

Activity	Optimistic	Most likely	Pessimistic
1-2	14	17	25
2-3	14	18	21
2-4	13	15	18
2-8	16	19	28
3-4 (dummy)	0	0	0
3-5	15	18	27
4-6	13	17	21
5-7 (dummy)	0	0	0
5-9	14	18	20
6-7 (dummy)	0	0	0
6-8 (dummy)	0	0	0
7-9	16	20	41

8-9	14	16	22
-----	----	----	----

The policy of the firm with respect to submitting bids is to bid the minimum amount that will provide a 95% of probability of at best breaking-even. The fixed costs for the project are eight lakhs and the variable costs are 9000 every day spent working on the project. The duration is in days and the costs are in rupees.

What amount should the firm bid under this policy? **(8)**

b. The optimistic, most likely and pessimistic times of the activities of a project are given below. Activity 40-50 must not start before 22 days, while activity 70-90 must end by 35 days. The scheduled completion time of the project is 46 days. Draw the network and **(8)** determine the critical path. What is the probability of completing the project in scheduled time?

Activity	$t_o-t_m-t_p$	Activity	$t_o-t_m-t_p$
10-20	4-8-12	50-70	3-6-9
20-30	1-4-7	50-80	4-6-8
20-40	8-12-16	60-100	4-6-8
30-50	3-5-7	70-90	4-8-12
40-50	0-0-0	80-90	2-5-8
40-60	3-6-9	90-100	4-10-16

12. Consider the project summarized in the following table:

Activity	Immediate Predecessor(s)	Duration(weeks)		
		a	m	b
A	-	4	4	10
B	-	1	2	9
C	-	2	5	14
D	A	1	4	7
E	A	1	2	3
F	A	1	5	9
G	B,C	1	2	9
H	C	4	4	4
I	D	2	2	8
J	E,G	6	7	8

- Construct the project network.
- Find the expected duration and the variance of each activity.
- Find the critical path and the expected project completion time.
- What is the probability of completing the project on or before 35 weeks? **(AU 2016)**

13.a. The Madras Construction Company is bidding on a contract to install a line of microwave towers. It has identified the following activities, along with their expected time, predecessor restrictions, and worker requirements:

Activity	Duration, Weeks	Predecessor	Crew size, workers
A	4	None	4
B	7	None	2
C	3	A	2
D	3	A	4
E	2	B	3
F	2	B	3
G	2	D,E	3

H	3	F,G	4
---	---	-----	---

The contract specifies that the project must be completed in 14 weeks. This company will assign a fixed number of workers to the project for its entire duration, and so it would like to ensure that the minimum number of workers is assigned and that the project will be completed in 14 weeks. Find a schedule which will do this. (8)

b. A company had planned its operations as follows:

Activity:	1-2	2-4	1-3	3-4	1-4	2-5	4-7	3-6	5-7	6-8	7-8
Duration (Days):	7	8	8	6	6	16	19	24	9	7	8

(i) Draw the network and find the critical paths.

(ii) After 15 days of working, the following progress is noted:

(a) Activities 1-2, 1-3 and 1-4 completed as per original schedule.

(b) Activity 2-4 is in progress and will be completed in 4 more days.

(c) Activity 3-6 is in progress and will need 17 more days to complete.

(d) The staff at activity 3-6 is specialized. They are directed to complete 3-6 and undertake an activity 6-7, which will require 7 days. This rearrangement arose due to a modification in a specialization.

(e) Activity 6-8 will be completed in 4 days instead of the originally planned 7 days.

(f) There is no change in the other activities.

Update the network diagram after 15 days of start of work based on the assumption given above. Indicate the revised critical paths along with their duration. (8)

14. A project consists of activities from A to J as shown in the following table. The immediate predecessor(s) and the duration in weeks of each of the activities are given in the same table. Draw the project network and find the critical path and corresponding project completion time. Also find the total float as well as free float for each of the non critical activities.

(AU 2016) (16)

Activity	Immediate Predecessor(s)	Duration(weeks)
A	-	4
B	-	3
C	A,B	2
D	A,B	5
E	B	6
F	C	4
G	D	3
H	F,G	7
I	F,G	4
J	E,H	2

15.a. Office Automation, Inc., has developed a proposal for introducing a new computerized office system that will improve word processing and interoffice communications for a particular company. Contained in the proposal is a list of activities that must be accomplished to complete the new office system project. Information about the activities is shown below.

Activity	Description	Immediate predecessor	Time (weeks)		Cost (\$000's)	
			Normal	Crash	Normal	Crash
A	Plan needs	-	10	8	30	70
B	Order equipment	A	8	6	120	150
C	Install equipment	B	10	7	100	160
D	Setup training lab	A	7	6	40	50
E	Conduct training	D	10	8	50	75
F	Test system	C, E	3	3	60	60

- Show the network for the project.
- Develop an activity schedule for the project.
- What are the critical path activities, and what is the expected project completion time?
- Assume that the company wishes to complete the project in 6 months or 26 weeks. What crashing decisions would be recommended to meet the desired completion time at the least possible cost? Work through the network, and attempt to make the crashing decisions by inspection.
- Develop an activity schedule for the crashed project.
- What is the added project cost to meet the 6-month completion time? **(8)**

- b.** The data shown in the following table relates to a contract being undertaken. There are also site costs of £500 per day.

You are required to:

- calculate and state the time for completion on a normal basis;
- calculate and state the critical path on this basis, and the cost;
- calculate and state the cost of completion in the shortest possible time. **(8)**

Activity	Completion time (days)	Cost of activity (£1,000)	Possible reduction time (days)	Extra cost for reduction (£/day)
A(1-2)	5	6	1	300
B(1-3)	8	10	2	200
C(1-4)	15	17	4	700
D(2-3)	4	5	1	400
E(2-5)	12	15	3	200
F(3-4)	6	8	2	200
G(4-5)	7	9	1	400
H(4-6)	11	13	3	300
I(4-7)	10	12	2	600
J(5-6)	8	14	2	300
K(6-8)	9	25	3	100
L(7-8)	10	13	2	500

IT 6801 – SERVICE ORIENTED ARCHITECTURE

TWO MARKS QUESTION AND ANSWER

UNIT – I

1. What is XML ?

Extensible markup language. It offer a standard, flexible and inherently extensible data format, XML significantly reduces the burden of deploying the many technologies needed to ensure the success of Web services.

2. Define XML attributes

- XML elements can have attributes in the start tag, just like HTML.
- Attributes are used to provide additional information about elements.
- Attributes cannot contain multiple values (child elements can)
 - Attributes are not easily expandable (for future changes)

3. Write the main difference between XML and HTML

Main Difference between XML and HTML XML was designed to carry data.

XML is not a replacement for HTML.

XML and HTML were designed with different goals:

XML was designed to describe data and to focus on what data is. HTML was designed to display data and to focus on how data looks.

HTML is about displaying information, while XML is about describing information

4. What is meant by a XML namespace?

XML Namespaces provide a method to avoid element name conflicts. When using prefixes in XML, a so-called **namespace** for the prefix must be defined. The namespace is defined by the **xmlns attribute** in the start tag of an element. The namespace declaration has the following syntax. **xmlns:prefix="URI"**.

```
<root> <h:table xmlns:h="http://www.w3.org/TR/html4/"> <h:tr>
<h:td>Apples</h:td>
<h:td>Bananas</h:td>
</h:tr> </h:table>
<f:table xmlns:f="http://www.w3schools.com/furniture">
<f:name>African Coffee Table</f:name> <f:width>80</f:width>
<f:length>120</f:length> </f:table> </root>
```

5. What is XML namespace?

XML allows document authors to create custom elements.

- This extensibility can result in naming collisions (i.e. different elements that have the same name) among elements in an XML document.

An XML namespace is a collection of element and attribute names. Each namespace has a unique name that provides a means for document authors to unambiguously refer to elements with the same name (i.e. prevent collisions).

6. What is the purpose of namespace?

XML Namespaces provide a method to avoid element name conflicts. In XML, element names are defined by the developer. This often results in a conflict when trying to mix XML documents from different XML applications.

7. What are all the Transformation techniques?

- XSLT - it is an XML- based languages used to transform XML documents into others format such as HTML for web display.
- XLINK - highlighting that element or taking the user directly to that point in the document.
- XPATH - xpath gets its name from its use of a payh notation to navigate through the hierarchical tree structure of an XML document
- XQUERY - it is W3C initiative to define a standard set of constructs for querying & searching XML document.

8. What is XSLT?

- XSLT stands for XSL Transformations
- XSLT is the most important part of XSL
- XSLT transforms an XML document into another XML document
- XSLT uses XPath to navigate in XML documents.

XSLT is a W3C Recommendation

9. Define the term DTD.

A Document Type Definition (DTD) defines the legal building blocks of an XML document. It defines the document structure with a list of legal elements and attributes.

10. List two types of DTD declaration

DTD is stands for Document Type Definition which is used to structure the XML document. The type of DTD are as follows i) Internal Declaration ii) External Declaration.

11. How to declare DTD attributes?

An attribute declaration has the following syntax:

```
<!ATTLIST element-name attribute-name attribute-type
default-value>
```

DTD example:

```
<!ATTLIST payment type CDATA
```

```
"check">
```

XML example:

```
<payment type="check" />
```

12. What is XML schema?

An XML schema is itself an XML document. It provides more detail about the kind of data that can appear as part of an XML document.

13. What is the purpose of XML schema? (APR/MAY 2013)

- The schemas are more specific and provide the support for data types.
- The schema is aware of namespace
- The XML Schema is written in XML itself and has a large number of built-in and derived types.
- The xml schema is the W3C recommendation. Hence it is supported by various XML validator and XML Processors.

14. What are the disadvantages of schema?

- The XML schema is complex to design and hard to learn
- The XML document cannot be if the corresponding schema file is absent.
- Maintaining the schema for large and complex operations sometimes slows down the processing of XML document.

15. Explain DTD for XML Schemas.

- XML documents are processed by applications

- Applications have assumptions about XML documents •
DTDs allow to formalize some of these constraints.

16. List some browsers that support XML and XSL

Mozilla Firefox

As of version 1.0.2, Firefox has support for XML and XSLT (and CSS).

Mozilla: Mozilla includes Expat for XML parsing and has support to display XML + CSS. Mozilla also has some support for Namespaces. Mozilla is available with an XSLT implementation.

Netscape: As of version 8, Netscape uses the Mozilla engine, and therefore it has the same XML / XSLT support as Mozilla.

Opera: As of version 9, Opera has support for XML and XSLT (and CSS). Version 8 supports only XML + CSS. Internet Explorer: As of version 6, Internet Explorer supports XML, Namespaces, CSS, XSLT, and XPath. Version 5 is NOT compatible with the official W3C XSL Recommendation.

17. What is XML presentation technique?

XML presentation technologies provide a modular way to deliver and display content to a variety of devices. There are different presentation technologies used in XML to display the content. Eg: CSS

18. List some of presentation technologies.

Presentation technologies provide a modular way to deliver and display content to a variety of devices. i) CSS ii) XSL iii) XFORMS iv) XHTML

19. Explain XSLT.

XSLT (eXtensible Stylesheet Language Transformations) is the recommended style sheet language for XML.

XSLT is far more sophisticated than CSS. With XSLT you can add/remove elements and attributes to or from the output file. You can also rearrange and sort elements, perform tests and make decisions about which elements to hide and display, and a lot more.

XSLT uses XPath to find information in an XML document.

20. Define XML attribute,

XML Attributes Must be Quoted

Attribute values must always be quoted. Either single or double quotes can be used. For a person's gender, the person element can be written like this:

<person gender="female"> or like

this: <person gender='female'>

If the attribute value itself contains double quotes you can use single quotes, like in this example: <gangster name='George "Shotgun" Ziegler'> or you can use character entities: <gangster name="George "Shotgun" Ziegler">

UNIT – II

1. Write about DOM.

DOM is W3c supported standard application programming interface(API) that provides a platform and language- neutral interface to allow developers to programmatically access and modify the content and structure documents.

2. What is SAX?

SAX is an example of a grass- roots development effort to provide a simple; Java based API for processing XML.

3. What are the levels of DOM?

DOM provides a platform and language- neutral interface to allow developers to programmatically access and modify the content and structure documents. It has Level 0, Level 1, Level 2, Level 3

4. Compare CSS and XSL.

CSS can be used with HTML. But XSL can't be used in HTML Both can be used in XML

CSS is not a transformation language but XSL.

5. What does XML-Signature elements provide?

The XML-Signature elements provides message integrity and authentication information about the originator of the message.

6. Give the basic structure of the XML signature.

```

<Signature>
  <SignedInfo>
    <CanonicalizationMethod />
    <SignatureMethod />
    <Reference>
      <Transforms>
      <DigestMethod>
      <DigestValue>
    </Reference>
  </SignedInfo>
</Signature>

```

```

        <Reference />
    </SignedInfo>
    <SignatureValue/>
    <KeyInfo/>
    <Object/>
</Signature>

```

7. What is DOM?

The Document Object Model (DOM) is the model that describes how all elements in an HTML page, like input fields, images, paragraphs etc., are related to the topmost structure: the document itself. By calling the element by its proper DOM name, we can influence it.

8. What are the 2 traditional ways of assigning event handlers in DOM .

1) Via HTML, using attributes 2) Via scripting

9. How to add Nodes in DOM Tree.

- Nodes can also be added to the DOM. You've already seen how attribute nodes can be created and applied to an element so let's look at adding element and text nodes within the document tree (without using the innerHTML property).
- The first step is to create a node object of the type you want using one of document.createElement(), document.createAttribute() or document.createTextNode(). For attributes, however, you'll probably just want to create an element node and assign it attributes directly.

10. What are the types of nodes in DOM Tree?

Element nodes, as we've seen, correspond to individual tags or tag pairs in the HTML code. They can have child nodes, which may be other elements or text nodes.

Text nodes represent content, or character data. They will have a parent node and possibly sibling nodes, but they cannot have child nodes.

Attribute nodes are a special case. They are not considered a part of the document tree - they do not have a parent, children or siblings. Instead, they are used to allow access to an element node's attributes.

11. What is Window Object in DOM?

The window object represents an open window in a browser. If a document contains frames (or tags), the browser creates one window object for the HTML document, and one additional window object for each frame. Some of the window object properties are:

- closed
- document
- frames
- history.

12. Mention any 3 XML Parsers.

- SAX (Simple API for XML) Parser
- DOM (Document Object Model) Parser and
- XSLT (XML Style Sheet) Parsers.

13. What is the purpose of the XML DTD.

The purpose of a DTD is to define the structure of an XML document. It defines the structure with a list of legal elements:

14. What is XSL Programming?

XSL (XML Stylesheet) Programming is the Next Generation of the CSS (Cascading Style Sheet Programming). In CSS, users can use certain style tags which the browsers can understand and are predefined by W3 consortium. XSL takes this to one step ahead and users can define any tags in the XML file. XML sheets can help in showing the same data in different formats.

15. What is XSLT?

- XSLT stands for XSL Transformations
- XSLT is the most important part of XSL
- XSLT transforms an XML document into another XML document
- XSLT uses XPath to navigate in XML documents
- XSLT is a W3C Recommendation.

16. What are Java Servlets?

Servlets are Java technology's answer to CGI programming. They are programs that run on a Web server and build Web pages.

17. What are the uses of XML?

XML is used in many aspects of web development, often to simplify data storage and sharing.

18. What are the various features of XML?

- Security
- Portability
- Scalability

- Reliability

19. List out the advantages of XML.

- XML files are human - readable
- Widespread industry support
- Relational Databases
- XML support technologies
- More meaningful searches

20. What is XML declaration?

It identifies the version of the XML specification to which the document conforms.

Example:

```
<?xml version="1.0"?>
```

An XML declaration can also include an

- Encoding Declaration
- Stand-alone Document Declaration

UNIT – III

1. What is Service Oriented Architecture?

Service oriented architecture is essentially a collection of services. These services communicate with each other. The communication can involve either simple data passing or it could involve two or more services coordinating some activity.

2. Define Contemporary SOA.

Contemporary SOA represents an architecture that promotes service orientation through the use of web services.

3. List out some characteristics of Contemporary SOA.

Some of the characteristics of contemporary SOA are:-

- i. Contemporary SOA is at the core of the service oriented platform.
- ii. Contemporary SOA increases quality of service.
- iii. Contemporary SOA is fundamentally autonomous.
- iv. Contemporary SOA is based on open standards.
- v. Contemporary SOA supports vendor diversity.
- vi. Contemporary SOA fosters intrinsic interoperability.
- vii. Contemporary SOA promotes discovery.
- viii. Contemporary SOA promotes federation.
- ix. Contemporary SOA promotes architectural composability.
- x. Contemporary SOA fosters inherent reusability.

4. What are the benefits of SOA?

The benefits of SOA are:

- i. Improved integration and intrinsic interoperability
- ii. Inherent reuse
- iii. Streamlined architectures and solutions
- iv. Leveraging the legacy investment
- v. Establishing standardized XML data representation
- vi. Focused investment on communications infrastructure
- vii. “Best-of-breed” alternatives
- viii. Organizational agility

5. What are the common pitfalls of adopting SOA?

The common pitfalls of adopting SOA are:

- i. Building service oriented architectures like traditional distributed architectures
- ii. Not standardizing SOA
- iii. Not creating a transition plan
- iv. Not starting with an XML foundation architecture
- v. Not understanding SOA performance requirements
- vi. Not understanding web services security
- vii. Not keeping in touch with product platforms and standards development.

6. What is Architecture?

Architecture refers a systematic arrangement of computerized automation technological solutions.

7. What is application architecture?

Application architecture is a template for all others which specifically explained the technology, boundaries, rules, limitations, and design characteristics that apply to all solutions based on this template.

8. What is Single-tier client-server architecture?

Single-tier client-server architecture is an environment in which bulky mainframe back-ends server served the thin clients.

9. List out the primary characteristics of the two tier client server architecture?

The primary characteristics of the two tier client server architectures is given below which is compared to SOA

- i. Application logic
- ii. Application processing

- iii. Technology
- iv. Security
- v. Administration

10. What is multi-tier client-server architectures?

Multi-tier architecture (often referred to as n-tier architecture) is a client-server architecture in which the presentation, the application processing, and the data management are logically separate processes.

11. What are the issues that are raised in the client-server and the distributed Internet architecture?

The issues that are raised in the client-server and the distributed Internet architecture comparisons are discussed in a comparison between multi-tier client-server and SOA.

- i. Application logic
- ii. Application processing
- iii. Technology
- iv. Security
- v. Administration

12. List some of the characteristics of Application Service layer.

- i. Expose functionality within a specific processing context
- ii. Draw upon available resources within a given platform
- iii. Solution – agnostic
- iv. Generic and reusable
- v. Achieve point-to-point integration with other application services
- vi. Inconsistent in terms of the interface granularity they expose
- vii. Mixture of custom-developed and third-party purchased services

13. Write down the steps for composing SOA.

Step 1: Choose service layers
 Step 2: Position core standards
 Step 3: Choose SOA extensions

14. What are the design characteristics required to facilitate interoperability in contemporary SOA?

The design characteristics required to facilitate interoperability are:

- i. Standardization
- ii. Scalability
- iii. Behavioral predictability
- iv. Reliability

15. Write down the layers of abstraction identified for SOA.

The three layers of abstraction identified for SOA are:

- i. the application service layer
- ii. the business service layer
- iii. the orchestration service layer

16. What are the Types of Architecture:

• Application architecture • Enterprise architecture • Service-oriented architecture.

17. Define Application architecture .

Application architecture is to an application development team what a blueprint is to a team of construction workers.

18. Define Enterprise architecture .

Enterprise architectures often contain a long-term vision of how the organization plans to evolve its technology and environments. For example, the goal of phasing out an outdated technology platform may be established in this specification.

19. Logic components of automation logic/ SOA.

Fundamental parts of the framework 1. SOAP messages 2. Web service operations 3. Web services 4. Activities

20. Define messages.

Messages = units of communication (A message represents the data required to complete some or all parts of a unit of work).

21. Define operations.

Operations = units of work (An operation represents the logic required to process messages in order to complete a unit of work.)

22. Define services.

Services = units of processing logic (A service represents a logically grouped set of operations capable of performing related units of work.)

23. Define processes.

Processes = units of automation logic. (A process contains the business rules that determine which service operations are used to complete a unit of automation.

24. What are the layers of abstraction.

The three layers of abstraction we identified for SOA are: *) the application service layer *) the business service layer *) the orchestration service layer.

25. Define Application service layer.

While application services are responsible for representing technology and application logic, the business service layer introduces a service concerned solely with representing business logic, called the business service.

26. Define Business service layer.

Business services are the lifeblood of contemporary SOA. They are responsible for expressing business logic through service-orientation and bring the representation of corporate business models into the Web services arena.

UNIT IV

1. Expand UDDI.

UDDI stands for Universal Description Discovery and Integration.

2. What is the use of RPC?

Client-server remote procedure call (RPC) connection is used for remote communication between components residing on client workstations and servers.

3. Write down the advantage of RPC?

Advantages of RPC are:

- v. Better load balancing:
More evenly distributed processing (e.g., application logic distributed between several servers)
- vi. More scalable:
Only servers experiencing high demand need be upgraded
- vii. Multiple concurrent requests are processed

4. Write down the disadvantages of RPC?

Disadvantages of RPC are:

- In heavily loaded network
- i. More distributed processing necessitates more data exchanges
- ii. Difficult to program and test due to increased complexity

5. Define the definition element?

The definition element is the root element in WSDL. It defines the name of the web service and specifies the namespace that would be used in the WSDL document.

6. Describe the message element.

The <message> element describes the data being exchanged between the web service providers and consumers. The <message> element assigns the message a name and contains one or more part child elements that each are assigned a type.

7. Define the binding element.

The binding element begins the concrete portion of the service definition, to assign a communications protocol that can be used to access and interact with the WSDL. The binding construct contains one or more operation elements.

8. List out the elements in the WSDL document structure.

Element	Defines
<types>	The data types used by the web service

<message>	The messages used by the web service
<portType>	The operations performed by the web service
<binding>	The communication protocols used by the web service
<service>	The service location used by the web service

9. What is Web Services?

A web service is used to implement architecture according to service oriented architecture (SOA) concepts. The basic unit of communication is message.

10. What are the basic parts comprised in the web services framework?

The basic parts comprised in the web services framework are:

- i. one or more architectures
- ii. technologies
- iii. concepts
- iv. models
- v. sub-frameworks

11. List out the characteristics of web services framework.

The various characteristics of web services framework are:

- i. An abstract (vendor-neutral) existence defined by standards organizations and implemented by (proprietary) technology platforms.
- ii. Core building blocks that include web services, service descriptions, and messages.
- iii. A communication agreement centered around service descriptions based on WSDL.
- iv. A messaging framework comprised of SOAP technology and concepts.
- v. A service description registration and discovery architecture sometimes realized through UDDI.
- vi. A well-defined architecture that supports messaging patterns and compositions.
- vii. A second generation of web services extensions (also known as the WS-* specifications) continually broadening its underlying feature-set.

12. Write down the advantage of web services.

The various advantages of web services are:

- | | |
|------|-------------------------------------|
| i. | Flexible |
| ii. | Adaptable |
| iii. | Promotes interoperability |
| iv. | Reduces complexity by encapsulation |
| v. | Enables just-in-time integration |

13. Give the classification of web services design.

The different classification of web services design is:

- i. Temporary classification (service roles)

- ii. Permanent classification (service models)

14. What is the service provider?

The service provider is used to identify the organization (or individual) responsible for actually providing the web service. It simply referred as the service being invoked.

15. What is service requestor?

Service requestor is a processing logic unit capable of issuing a request message that can be understood by the service provider.

16. What are service descriptions?

A WSDL service description explains how the service description document itself is organized. It is also known as WSDL service definition or just WSDL definition.

17. What are the categories of service description?

Service description is divided into two categories

- i. Abstract description
- ii. Concrete description

18. What does abstract description establish?

An abstract description establishes the interface characteristics of the web service without any reference to the technology used to host or enable a web service to transmit messages.

19. What are the parts that comprise an abstract description?

The three main parts that comprise an abstract description are

- i. Port type
- ii. Operation
- iii. Message

20. What does port type in abstract description provide?

Port type provides a high-level view of the service interface by sorting the messages a service can process into groups of functions.

21. What is metadata?

Metadata provides information about the service.

22. What is the use of SOAP?

The Simple Object Access Protocol (SOAP) is used to define a standard message format

which is used for communication between services running on different operating systems.

23. List out some of the characteristics of SOAP messaging framework.

SOAP messaging framework has the following three characteristics that are

- i. Extensible
- ii. Interoperable
- iii. Independent

24. What are the parts of SOAP message?

SOAP message consists of the three parts:

- SOAP envelope
- SOAP header (optional)
- SOAP body
- SOAP fault

25. List out messaging styles offered by SOAP.

- i. RPC (Remote Procedure Call) style
- ii. Document – style

26. Sketch the anatomy of a SOAP message.

```
<?xml version="1.0"?>
<soap:Envelope
  xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
  soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">
  <soap:Header>
  .....
  </soap:Header>

  <soap:Body>
  .....
  <soap:Fault>
  .....
  </soap:Fault>
  </soap:Body>
</soap:Envelope>
```

27. What is SOAP node?

The programs that use services to transmit and receive SOAP messages are referred to as SOAP nodes.

28. What is called the SOAP message path?

The route taken by the message is called the SOAP message path. The set of SOAP nodes through which the SOAP message passes, including the initial sender, the ultimate receiver and one or more intermediaries are called the SOAP message path.

29. Define Message Exchange Pattern.

Message Exchange Pattern (MEP) defines the way that SOAP messages are exchanged between the web service requester and web service provider. It represents a set of templates.

30. List out some primitive MEPs.

A common set of primitive MEPs are listed below

- i. Request-response
- ii. Fire-and-forget
- iii. Complex MEPs

31. What is Publish-and-subscribe pattern?

Publish-and-subscribe pattern is an asynchronous MEP in which publisher sends messages to all interested subscribers.

32. What is coordination?

Coordination is the act of one entity (known as the coordinator) disseminating information to a number of participants for coordinating the activities of the web services that are part of a business process.

33. What does the style attribute of soap:binding element define?

The style attribute of the soap:binding element defines whether the SOAP messages used to support an operation are to be formatted.

34. List out the format supported by the style attribute of the soap:binding element.

- i. Document – style messages
- ii. RPC – style messages

35. What does the soap :body element define?

The soap:body element defines the data type system to be used SOAP processors, via the use attribute. The use attribute can be set to “encoding” or “literal”.

36. What is the use of import element?

The import element is used to import parts of the WSDL definition as well as XSD schemas.

37. What is the use of the documentation element?

The documentation element is used to add descriptive, human-readable annotations within a WSDL definition.

38. What is SOAP?

SOAP is an XML-based messaging protocol. It defines a set of rules for structuring messages that can be used for simple one-way messaging but is particularly useful for performing RPC-style (Remote Procedure Call) request-response dialogues.

39. Give the structure of a SOAP message.

A SOAP message is encoded as an XML document consisting of

- an <Envelope> element, which contains
 - an optional <Header> element, and
 - a mandatory <Body> element.
 - the <Fault> element, contained within the <Body> is used for reporting errors.

40. What is the Envelope element?

The SOAP <Envelope> is the root element in every SOAP message, and contains two child elements

- i. an optional <Header>
- ii. a mandatory <Body>

41. What is the use of Header element?

The SOAP <Header> is used to pass application related information that is to be processed by SOAP nodes along the message path.

42. Give the skeleton for the Envelope element?

The root Envelope constructs hosting Header and Body constructs.

```
<Envelope xmlns="http://schemas.xmlsoap/soap/envelope/">
  <Header> .....</Header>
  <Body>.....</Body>
</Envelope>
```

43. What is the Fault element?

The SOAP <Fault> is a sub-element of the SOAP body, which is used for reporting errors. It is used to carry error and status information within a SOAP message.

44. What is WS-choreography?

Web service choreography (WS-Choreography) is a XML based business process modeling language that describes collaboration protocols of cooperating web service participants, in which services act as peers, and interactions may be long lived and stateful.

45. How will you define the participant in WS-Choreography?

```
<participantType name="Buyer">
  <description type="documentation">
    Buyer Participant
  </description>
  <roleType typeRef="tns:BuyerRole"/>
</participantType>
```

46. How will you declare the relationship between the roles in WS-Choreography?

```
<relationshipType name="ncname">
  <role type="qname" behavior="list of ncname"?/>
  <role type="qname" behavior="list of ncname"?/>
</relationshipType>
```

UNIT V

1. What is Service oriented analysis?

The service oriented analysis is the process of determining how business automation requirements can be represented through service orientation.

2. What are the goals needed for performing a service-oriented analysis?

The overall goals of performing a service-oriented analysis are as follows:

- i. Define a preliminary set of service operation candidates
- ii. Group service operation candidates into logical contexts. These contexts represent service candidates.
- iii. Define preliminary service boundaries so that they do not overlap with any existing or planned services.
- iv. Identify encapsulated logic with reuse potential.
- v. Ensure that the context of encapsulated logic is appropriate for its intended use.
- vi. Define any known preliminary composition models.

3. Give the step-by-step process in the service oriented analysis.

Step 1: Define business automation requirements

Step 2: Identify existing automation systems

Step 3: Model candidate services

4. What is Service modeling?

Service modeling is a process of identifying candidate service operation and then grouping them into a logical context.

5. What is Business-centric SOA?

Business-centric SOA is the process of introducing service oriented principles into the domain of business analysis.

6. What is the use of service candidates?

The service candidate is used to distinguish a conceptualized service from an actual implemented service.

7. What is the key service orientation principles applied to the service candidate?

- i. Reusability
- ii. Autonomy
- iii. Statelessness
- iv. Discoverability

8. What is service oriented design?

Service oriented design phase is a process that transforms previously modeled service candidates into physical service designs.

9. Give the overall goals for performing a service oriented design.

The overall goals of performing a service oriented design are as follows:

- i. Determine the core set of architectural extensions.
- ii. Set the boundaries of the architecture.
- iii. Identify required design standards.
- iv. Define abstract service interface designs.
- v. Identify potential service compositions.
- vi. Assess support for service orientation principles.
- vii. Explore support for characteristics of contemporary SOA.

10. What does abstract definition contain?

The abstract definition contains a series of parts that include

- i. Types
- ii. Message
- iii. Port type (or interface)

11. What does concrete definition comprised of?

The concrete definition is comprised of

- i. Binding parts
- ii. Service parts

12. What are the steps needed to design the Entity-centric business service?

- Step 1: Review existing services
- Step 2: Define the message schema types
- Step 3: Derive an abstract service interface
- Step 4: Apply principles of service orientation
- Step 5: Standardize and refine the service interface
- Step 6: Extend the service design
- Step 7: Identify required processing

13. List out the SOA principles supported by Application service design.

- i. Reusability
- ii. Autonomy
- iii. Statelessness
- iv. Discoverability

14. Write down the steps for Task-centric business service design.

- Step 1: Define workflow logic
- Step 2: Derive initial interface
- Step 3: Apply principles of service orientation
- Step 4: Standardize service interface
- Step 5: Identify required processing

15. Give the architecture components of J2EE to SOA.

- i. Java Server Pages (JSPs)
- ii. Struts
- iii. Java Servlets
- iv. Enterprise JavaBeans (EJBs)

16. What is JAX-WS?

JAX-WS is a technology for building web services using XML. In JAX-WS, a web service operation invocation is represented by an XML-based protocol such as SOAP.

17. Expand SEI.

SEI stands for

- Service Endpoint Interface or
- Service Endpoint Implementation

18. What is SEI?

SEI is a java interface or class that declares the methods that a client can invoke on the service.

19. Expand JAXB and JAXR.

JAXB stands for Java Architecture for XML Binding (JAXB)

JAXR stands for Java API for XML Registries (JAXR)

20. What is JAXB?

Java Architecture for XML binding API (JAXB) provides a means of generating Java classes from XSD schemas and further abstracting XML-level development.

21. Give the general steps to use the JAXB API.

The general steps to use the JAXB API are:

- i. Bind the schema
- ii. Unmarshal
- iii. Marshal

22. Write down the advantages of JAXB.

It simplifies access to an XML document from a Java program.

It uses memory efficiently.

It is flexible.

It allows transportation from one XML document to another.

23. What is JAXR?

The Java API for XML Registries (JAXR) provides a uniform and standard Java API for accessing various kinds of XML registries.

24. What are the components of JAXR?

- i. JAXR client
- ii. JAXR provider

25. Write down the packages that are implemented by JAXR.

- i. javax.xml.registry
- ii. javax.xml.registry.infomodel

26. What are the tasks involved in managing registry data?

- i. Getting authorization from the registry
- ii. Creating an organization
- iii. Adding classifications
- iv. Adding services and service binding to an organization
- v. Publishing a specification concept
- vi. Removing data from the registry

27. Expand JAX-RPC and WSIT.

JAX-RPC stands for Java API for XML based RPC.

WSIT stands for Web Services Interoperability Technologies.

28. What is the use of JAX-RPC?

JAX-RPC is used for building and deploying SOAP+WSDL web services clients and endpoints. It enables clients to invoke web services developed across heterogeneous platforms.

29. What are the benefits of JAX-RPC?

- i. Portable and interoperable web services
- ii. Ease of development of web service endpoints and clients
- iii. Increased developer productivity
- iv. Support for open standards: XML, SOAP, WSDL
- v. Standard API developed under Java Community Process (JCP)
- vi. Support for tools
- vii. RPC programming model with support for attachments
- viii. Support for SOAP message processing model and extensions
- ix. Secure web services
- x. Extensible type mapping

30. Expand WS-BPEL.

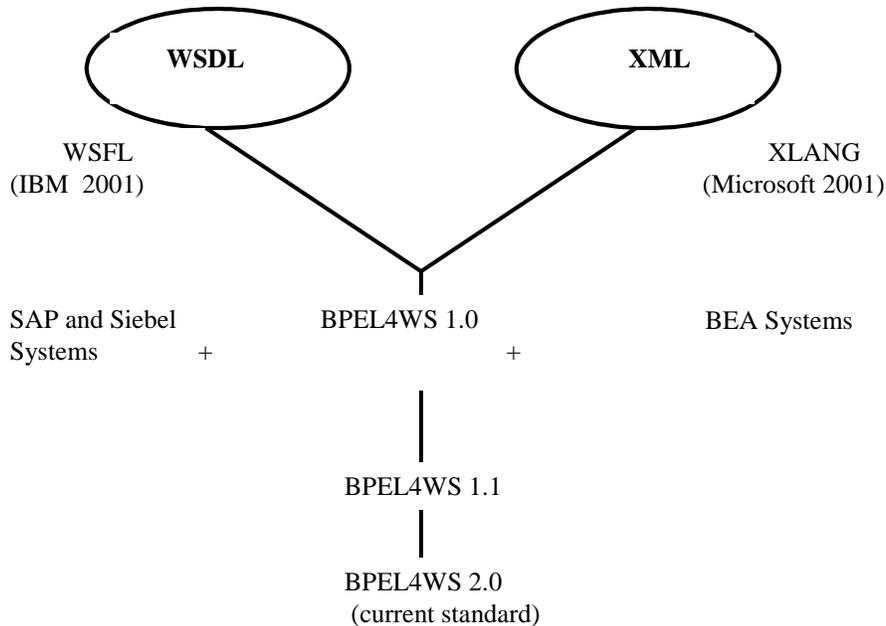
WS-BPEL stands for Web Services Business Process Execution Language.

31. What is WS-BPEL?

WS-BPEL is an XML based language (ie., it is described by a grammar) enabling users to

describe business process activities as Web Services and define how they can be connected to accomplish specific tasks.

32. Draw the WS-BPEL family tree.



33. Give the overview of WS-Coordination.

WS-Coordination is a framework for coordinating distributed activities

- Coordinator
 - Activation service for creating coordination instance
 - Registration service for registering participating application
 - Additional protocol specific service
- Set of coordination protocols

34. What is the use of CoordinationContext element?

The CoordinationContext is used to carry information about active coordination to participants

- Information inside context is coordination protocol specific
- Context format is not mandated by the standard
- Typically passed is SOAP headers

35. What is WS-Policy?

WS-Policy defines a framework for allowing web services to express their constraints and requirements in relation to security, processing, or message content.

36. What is the goal of WS-Policy?

WS-Policy provides the mechanisms needed to enable web services application to specify

policies.

37. Give the specifications of WS-Policy framework.

The WS-Policy framework is comprised of the following three specifications:

- WS-Policy
- WS-PolicyAssertions
- WS-PolicyAttachments

38. What is WS-Security?

WS-Security is known as Web Services Security is a flexible extensible framework to SOAP to apply security to web services.

39. Why is WS-Security needed?

The WS-Security is used to implement

- Message-level security measures
 - Protect message contents during transport and during processing by service intermediaries.
- Authentication and authorization control
 - Protect service provides from malicious requestors.

40. Give the specifications of WS-Security framework.

The WS-Security framework is comprised of the following specifications:

- WS-Security
- XML-Encryption
- XML-Signature

42. Give the syntax of WS-Security element.

<Envelope>

<Header>

.....

<wsse:Security actor="..." mustUnderstand="...">

.....

</wsse:Security>

</Header>

<Body>

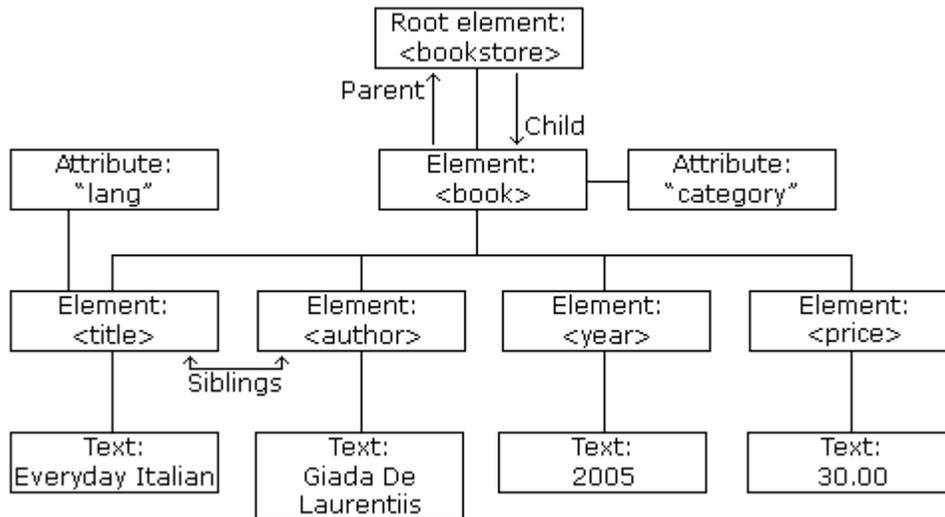
.....

</Body>

</Envelope>

UNIT - I

1. Draw the XML Tree Structure or XML Document structures with style sheets or well formed and Valid document



```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/css" href="xmlstyle.css"?>
<bookstore>
  <book category="cooking">
    <title lang="en">Everyday Italian</title>
    <author>Giada De Laurentiis</author>
    <year>2005</year>
    <price>30.00</price>
  </book>
  <book category="children">
    <title lang="en">Harry Potter</title>
    <author>J K. Rowling</author>
    <year>2005</year>
    <price>29.99</price>
  </book>
  <book category="web">
    <title lang="en">Learning XML</title>
    <author>Erik T. Ray</author>
    <year>2003</year>
    <price>39.95</price>
  </book>
</bookstore>
```

The style sheet, "xmlstyle.css", may look like:

```
bookstore
{ display: block }
title{ display: block;
font-family: arial;
color: #008000;
```

```

font-weight: 600;
font-size: 22;
text-align: center }
author
{ display: block;
font-family: arial;
color: #000080;
font-weight: 400;
font-size: 20 }
year
{ display: block;
list-style-type: decimal;
font-family: arial;
color: #000000;
font-weight: 400;
font-size: 18 }
price
{ display: block;
list-style-type: square;
font-family: arial;
color: #0000ff;
font-weight: 200;
font-size: 14 }

```

2. Explain Namespaces in XML.

- XML namespaces are used for providing uniquely named elements and attributes in an XML document. They are defined in a W3C recommendation. An XML instance may contain element or attribute names from more than one XML vocabulary.
- XML Namespaces provide a method to avoid element name conflicts.

```

<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
<html>
<body>
<h2>My CD Collection</h2>
<table border="1">
<tr>
<th style="text-align:left">Title</th>
<th style="text-align:left">Artist</th>
</tr>
<xsl:for-each select="catalog/cd">
<tr>
<td><xsl:value-of select="title"/></td>
<td><xsl:value-of select="artist"/></td>
</tr>

```

```

    </xsl:for-each>
  </table>
</body>
</html>
</xsl:template>

</xsl:stylesheet>

```

2. 3. Elucidate the XML DTD.

The purpose of a DTD is to define the structure of an XML document. It defines the structure with a list of legal elements:

The Internal DTD:

```

<?xml version="1.0"?>
<!DOCTYPE employees[
<!ELEMENT employees(employee+)>
<!ELEMENT employee(name,position,age,sex,status,address,city,state,zip,phone)?>
<!ELEMENT name (#PCDATA)>
<!ELEMENT position (#PCDATA)>
<!ELEMENT age (#PCDATA)>
<!ELEMENT sex (#PCDATA)>
<!ELEMENT status (#PCDATA)>
<!ATTLIST address (#PCDATA)>
<!ELEMENT city (#PCDATA)>
<!ELEMENT state (#PCDATA)>
<!ELEMENT zip (#PCDATA)>
<!ELEMENT phone (#PCDATA)>
]>
<employee>
  <name>john</name>
  <position>manager</position>
  <age>25</age>
  <sex>male</sex>
  <status>unmarried</status>
  <address>Cauvery nagar</address>
  <city>karur</ciy>
  <state>Tamilnadu</state>
  <zip>639101</zip>
  <phone>978595785</phone>
</employee>
<employee>
  <name>ram</name>
  <position>programmer</position>
  <age>23</age>
  <sex>male</sex>
  <status>unmarried</status>
  <address>Cauvery nagar</address>
  <city>karur</ciy>

```

```

    <state>Tamilnadu</state>
    <zip>639101</zip>
    <phone>788595785</phone>
</employee>

```

```

    <employee>
    <name>ranjith</name>
    <position>teamleader</position>
    <age>26</age>
    <sex>male</sex>
    <status>unmarried</status>
    <address>Cauvery nagar</address>
    <city>karur</ciy>
    <state>Tamilnadu</state>
    <zip>639102</zip>
    <phone>788785785</phone>
</employee>
</employees>

```

**The External DTD:
Employee.xml**

```

<?xml version="1.0"?>

<!DOCTYPE employees SYSTEM "employees.dtd">

<employees>
<employee>
    <name>john</name>
    <position>manager</position>
    <age>25</age>
    <sex>male</sex>
    <status>unmarried</status>
    <address>Cauvery nagar</address>
    <city>karur</ciy>
    <state>Tamilnadu</state>
    <zip>639101</zip>
    <phone>978595785</phone>
</employee>
<employee>
    <name>ram</name>
    <position>programmer</position>
    <age>23</age>
    <sex>male</sex>
    <status>unmarried</status>
    <address>Cauvery nagar</address>
    <city>karur</ciy>
    <state>Tamilnadu</state>
    <zip>639101</zip>
    <phone>788595785</phone>

```

```
</employee>
```

```

<employee>
  <name>ranjith</name>
  <position>teamleader</position>
  <age>26</age>
  <sex>male</sex>
  <status>unmarried</status>
  <address>Cauvery nagar</address>
  <city>karur</ciy>
  <state>Tamilnadu</state>
  <zip>639102</zip>
  <phone>788785785</phone>

```

```
</employee>
```

employees.dtd:

```

<!ELEMENT employees(employee+)>
<!ELEMENT employee(name,position,age,sex,status,address,city,state,zip,phone)?>
<!ELEMENT name (#PCDATA)>
<!ELEMENT position (#PCDATA)>
<!ELEMENT age (#PCDATA)>
<!ELEMENT sex (#PCDATA)>
<!ELEMENT status (#PCDATA)>
<!ATTLIST address (#PCDATA)>
<!ELEMENT city (#PCDATA)>
<!ELEMENT state (#PCDATA)>
<!ELEMENT zip (#PCDATA)>
<!ELEMENT phone (#PCDATA)>

```

4. Explain XML – SCHEMA

- An XML Schema describes the structure of an XML document. The XML Schema language is also referred to as XML Schema Definition (XSD).
- The purpose of an XML Schema is to define the legal building blocks of an XML document: the elements and attributes that can appear in a document. the number of (and order of) child elements. data types for elements and attributes.

```

<?xml version="1.0" encoding="UTF-8" ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:simpleType name="stringtype">
    <xs:restriction base="xs:string"/>
  </xs:simpleType>

  <xs:simpleType name="inttype">
    <xs:restriction base="xs:positiveInteger"/>
  </xs:simpleType>

  <xs:simpleType name="dectype">

```

```

<xs:restriction base="xs:decimal"/>
</xs:simpleType>

<xs:complexType name="shiptotype">
  <xs:sequence>
    <xs:element name="name" type="stringtype"/>
    <xs:element name="address" type="stringtype"/>
    <xs:element name="city" type="stringtype"/>
    <xs:element name="country" type="stringtype"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="itemtype">
  <xs:sequence>
    <xs:element name="title" type="stringtype"/>
    <xs:element name="note" type="stringtype" minOccurs="0"/>
    <xs:element name="quantity" type="inttype"/>
    <xs:element name="price" type="dectype"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="shipordertype">
  <xs:sequence>
    <xs:element name="orderperson" type="stringtype"/>
    <xs:element name="shipto" type="shiptotype"/>
    <xs:element name="item" maxOccurs="unbounded" type="itemtype"/>
  </xs:sequence>
  <xs:attribute name="orderid" type="orderidtype" use="required"/>
</xs:complexType>
<xs:element name="shiporder" type="shipordertype"/>
</xs:schema>

```

5. Difference between DTD and XSD.

No.	DTD	XSD
1)	DTD stands for Document Type Definition.	XSD stands for XML Schema Definition.
2)	DTDs are derived from SGML syntax.	XSDs are written in XML.
3)	DTD doesn't support datatypes.	XSD supports datatypes for elements and attributes.
4)	DTD doesn't support namespace.	XSD supports namespace.
5)	DTD doesn't define order for child elements.	XSD defines order for child elements.
6)	DTD is not extensible.	XSD is extensible.

7)	DTD is not simple to learn.	XSD is simple to learn because you don't need to learn new language.
8)	DTD provides less control on XML structure.	XSD provides more control on XML structure.

6. Explain X – Files (X – Path, X-Link, X-Pointer)

X-Path:

- XPath is a syntax for defining parts of an XML document. XPath uses path expressions to navigate in XML documents. XPath contains a library of standard functions. XPath is a major element in XSLT and in XQuery. XPath is a W3C recommendation.

```
<?xml version="1.0" encoding="UTF-8"?>
<bookstore>
<book category="cooking">
  <title lang="en">Everyday Italian</title>
  <author>Giada De Laurentiis</author>
  <year>2005</year>
  <price>30.00</price>
</book>
<book category="children">
  <title lang="en">Harry Potter</title>
  <author>J K. Rowling</author>
  <year>2005</year>
  <price>29.99</price>
</book>
<book category="web">
  <title lang="en">XQuery Kick Start</title>
  <author>James McGovern</author>
  <author>Per Bothner</author>
  <author>Kurt Cagle</author>
  <author>James Linn</author>
  <author>Vaidyanathan Nagarajan</author>
  <year>2003</year>
  <price>49.99</price>
</book>
<book category="web">
  <title lang="en">Learning XML</title>
  <author>Erik T. Ray</author>
  <year>2003</year>
  <price>39.95</price>
</book>
</bookstore>
```

In the table below we have listed some XPath expressions and the result of the expressions:

XPath Expression	Result
/bookstore/book[1]	Selects the first book element that is the child of the bookstore element
/bookstore/book[last()]	Selects the last book element that is the child of the bookstore element
/bookstore/book[last()-1]	Selects the last but one book element that is the child of the bookstore element
/bookstore/book[position()<3]	Selects the first two book elements that are children of the bookstore element
/bookstore/book[price>35.00]	Selects all the book elements of the bookstore element that have a price element with a value greater than 35.00
/bookstore/book[price>35.00]/title	Selects all the title elements of the book elements of the bookstore element that have a price element with a value

X-Pointers

XPointer is a system for addressing components of XML based internet media

```
<?xml version = "1.0" encoding = "utf-8" ?>
  <books>
    <book author = "ram" id = "java">
      <picture url = "http://www.books.com/java.jpg" />
      <description> the java is a language is use in 4 billion products.</description>
    </book>
    <book author = "alen musk" id = "php">
      <picture url = "http://www.books.com/php.jpg" />
      <description> the php is a language is use to develop web sites.</description>
    </book>
  </books>
```

3. *Linking XML Document*

Linking the entire document(as with xlink) XPointer allows you to link to specific parts of the document.To link to a specific part of a page, add a number sign(#) and an XPointer expression after the url in the xlink:href attributes.

Xlink:href attribute would look like this:

```
xlink:href = "http://www.books.com/bookdata.xml#xpointer(id('java'))"
```

Shorthand for above XPointer statement look like this `xlink:href = "http://www.books.com/bookdata.xml#java"`

X-Link:

XLink is used to create hyperlinks in XML documents.

- XLink is used to create hyperlinks within XML documents.
- Any element in an XML document can behave as a link.
- With XLink, the links can be defined outside the linked files.
- XLink is a W3C Recommendation.

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<bookstore xmlns:xlink="http://www.w3.org/1999/xlink">
```

```
<book title="Harry Potter">
```

```
<description
```

```
xlink:type="simple"
```

```
xlink:href="/images/HPotter.gif"
```

```
xlink:show="new">
```

As his fifth year at Hogwarts School of Witchcraft and Wizardry approaches, 15-year-old Harry Potter is.....

```
</description>
```

```
</book>
```

```
<book title="XQuery Kick Start">
```

```
<description
```

```
xlink:type="simple"
```

```
xlink:href="/images/XQuery.gif"
```

```
xlink:show="new">
```

XQuery Kick Start delivers a concise introduction to the XQuery standard.....

```
</description>
```

```
</book>
```

```
</bookstore>
```

Example explained:

- The XLink namespace is declared at the top of the document (`xmlns:xlink="http://www.w3.org/1999/xlink"`)
- The `xlink:type="simple"` creates a simple "HTML-like" link
- The `xlink:href` attribute specifies the URL to link to (in this case - an image)

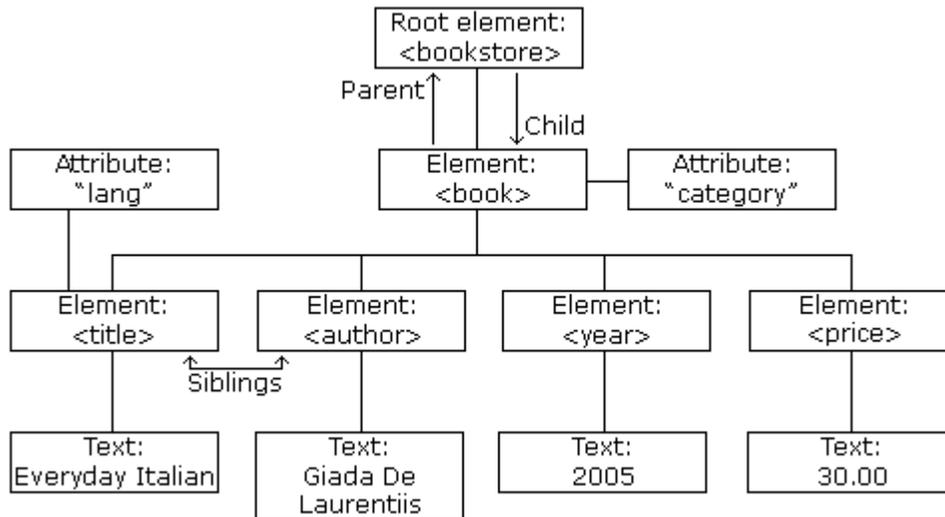
- The xlink:show="new" specifies that the link should open in a new window

UNIT – II

1. Explain the DOM (DOCUMENT OBJECT MODEL)

The DOM defines a standard for accessing and manipulating documents.

"The W3C Document Object Model (DOM) is a platform and language-neutral interface that allows programs and scripts to dynamically access and update the content, structure, and style of a document."



DOM Levels:

DOM Level 1

It allows traversal of an XML document as well as the manipulation of the content in that document

DOM Level 2

Extend level 1 with additional features such as namespace support, events, Traversals and ranges,

DOM Level3

The DOM Level 3 specification contains five different specifications: The DOM3 Core, Load and Save, Validation, Events, and XPath

2. Explain XML Parser using DOM.

Following are the steps used while parsing a document using DOM Parser.

- Import XML-related packages.
- Create a DocumentBuilder
- Create a Document from a file or stream
- Extract the root element
- Examine attributes
- Examine sub-elements

Here is the input xml file we need to parse:

```

<?xml version="1.0"?>
<company>
  <staff id="1001">
    <firstname>yong</firstname>
    <lastname>mook kim</lastname>
    <nickname>mkyong</nickname>
    <salary>100000</salary>
  </staff>
  <staff id="2001">
    <firstname>low</firstname>
    <lastname>yin fong</lastname>
    <nickname>fong fong</nickname>
    <salary>200000</salary>
  </staff>
</company>

```

DomParserDemo.java

```

package com.mkyong seo;

import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.DocumentBuilder;
import org.w3c.dom.Document;
import org.w3c.dom.NodeList;
import org.w3c.dom.Node;
import org.w3c.dom.Element;
import java.io.File;

public class ReadXMLFile {

    public static void main(String argv[]) {

        try {

            File fXmlFile = new File("/Users/mkyong/staff.xml");
            DocumentBuilderFactory dbFactory = DocumentBuilderFactory.newInstance();
            DocumentBuilder dBuilder = dbFactory.newDocumentBuilder();
            Document doc = dBuilder.parse(fXmlFile);

            doc.getDocumentElement().normalize();

            System.out.println("Root element : " + doc.getDocumentElement().getNodeName());

            NodeList nList = doc.getElementsByTagName("staff");

```

```

System.out.println("-----");

for (int temp = 0; temp < nList.getLength(); temp++) {

    Node nNode = nList.item(temp);

    System.out.println("\nCurrent Element : " + nNode.getNodeName());

    if (nNode.getNodeType() == Node.ELEMENT_NODE) {

        Element eElement = (Element) nNode;

        System.out.println("Staff id : " + eElement.getAttribute("id"));
        System.out.println("First Name : " +
eElement.getElementsByTagName("firstname"));
        System.out.println("Last Name : " +
eElement.getElementsByTagName("lastname"));
        System.out.println("Nick Name : " +
eElement.getElementsByTagName("nickname"));
        System.out.println("Salary : " +
eElement.getElementsByTagName("salary"));

    }

} catch (Exception e) {
    e.printStackTrace();
}
}
}

```

This would produce the following result:

Root element :company

Current Element :staff

Staff id : 1001

First Name : yong

Last Name : mook kim

Nick Name : mkyong

Salary : 100000

Current Element :staff

Staff id : 2001

First Name : low

Last Name : yin fong
 Nick Name : fong fong
 Salary : 200000

3. Elucidate XML parser using SAX:

Java SAX Parser provides API to parse XML documents. SAX Parser is different from DOM parser because it doesn't load complete XML into memory and read xml document sequentially.

SAX callback methods :

- **startDocument() and endDocument()** – Method called at the start and end of an XML document.
- **startElement() and endElement()** – Method called at the start and end of a document element.
- **characters()** – Method called with the text contents in between the start and end tags of an XML document element.

Steps

1. **Create the SAX parser and parse the XML file:** In this step we will take one factory instance from **SAXParserFactory** to parse the xml file this factory instance in turns give us instance of parser using the **parse()** method will parse the Xml file.
2. **Event Handling:** when Sax Parser starts the parsing whenever it finds the start or end tag it will invoke the corresponding event handling method which is **public void startElement (...)** and **public void end Element (...)**.
3. **Register the events:** The class extends the Default Handler class to listen for callback events and we register this handler to sax Parser to notify us for call back event

4. XML file

Create a simple XML file

```
<?xml version="1.0"?>
<company>
  <staff>
    <firstname>yong</firstname>
    <lastname>mook kim</lastname>
    <nickname>mkyong</nickname>
    <salary>100000</salary>
  </staff>
  <staff>
    <firstname>low</firstname>
```

```

        <lastname>yin fong</lastname>
        <nickname>fong fong</nickname>
        <salary>200000</salary>
    </staff>
</company>

```

5. Java file

Use SAX parser to parse the XML file

```

import javax.xml.parsers.SAXParser;
import javax.xml.parsers.SAXParserFactory;
import org.xml.sax.Attributes;
import org.xml.sax.SAXException;
import org.xml.sax.helpers.DefaultHandler;
public class ReadXMLFile {
    public static void main(String argv[]) {
        try {

            SAXParserFactory factory = SAXParserFactory.newInstance();
            SAXParser saxParser = factory.newSAXParser();

            DefaultHandler handler = new DefaultHandler() {
                boolean bfname = false;
                boolean blname = false;
                boolean bname = false;
                boolean bsalary = false;

                public void startElement(String uri, String localName,String qName,
                    Attributes attributes) throws SAXException {

                    System.out.println("Start Element :" + qName);

                    if (qName.equalsIgnoreCase("FIRSTNAME")) {
                        bfname = true;
                    }

                    if (qName.equalsIgnoreCase("LASTNAME")) {
                        blname = true;
                    }

                    if (qName.equalsIgnoreCase("NICKNAME")) {
                        bname = true;
                    }

                    if (qName.equalsIgnoreCase("SALARY")) {
                        bsalary = true;
                    }
                }
            };

```

```

    }

    public void endElement(String uri, String localName,
        String qName) throws SAXException {

        System.out.println("End Element : " + qName);

    }

    public void characters(char ch[], int start, int length) throws SAXException {

        if (bfname) {
            System.out.println("First Name : " + new String(ch, start, length));
            bfname = false;
        }

        if (blname) {
            System.out.println("Last Name : " + new String(ch, start, length));
            blname = false;
        }

        if (bname) {
            System.out.println("Nick Name : " + new String(ch, start, length));
            bname = false;
        }

        if (bsalary) {
            System.out.println("Salary : " + new String(ch, start, length));
            bsalary = false;
        }

    }

};

saxParser.parse("c:\\file.xml", handler);

} catch (Exception e) {
    e.printStackTrace();
}

}

}

```

Result:

```

Start Element :company
Start Element :staff
Start Element :firstname
First Name : yong
End Element :firstname
Start Element :lastname
Last Name : mook kim
End Element :lastname
Start Element :nickname
Nick Name : mkyong
End Element :nickname
Start Element :salary
Salary : 100000
End Element :salary
End Element :staff
Start Element :staff
Start Element :firstname
First Name : low
End Element :firstname
Start Element :lastname
Last Name : yin fong
End Element :lastname
Start Element :nickname
Nick Name : fong fong
End Element :nickname
Start Element :salary
Salary : 200000
End Element :salary
End Element :staff
End Element :company

```

3. Elucidate the Transforming XML with XSL.

XSL Technologies:

- XSL Transformation Language
- XSL Formatting Object Language

XSL Transformation Language

XSLT is used to convert an XML document to another format (HTML file, PDF document, or other format).

Transform the following XML document ("cdcatalog.xml") into XHTML:

```

<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="cdcatalog.xsl"?>
<catalog>
  <cd>

```

```

<title>Empire Burlesque</title>
<artist>Bob Dylan</artist>
<country>USA</country>
<company>Columbia</company>
<price>10.90</price>
<year>1985</year>
</cd>

```

.

```
</catalog>
```

Create an XSL Style Sheet ("cdcatalog.xsl") with a transformation template:

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<xsl:stylesheet version="1.0"
```

```
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
```

```
<xsl:template match="/">
```

```
<html>
```

```
<body>
```

```
<h2>My CD Collection</h2>
```

```
<table border="1">
```

```
<tr bgcolor="#9acd32">
```

```
<th>Title</th>
```

```
<th>Artist</th>
```

```
</tr>
```

```
<xsl:for-each select="catalog/cd">
```

```
<tr>
```

```
<td><xsl:value-of select="title"/></td>
```

```
<td><xsl:value-of select="artist"/></td>
```

```
</tr>
```

```
</xsl:for-each>
```

```
</table>
```

```
</body>
```

```
</html>
```

```
</xsl:template>
```

```
</xsl:stylesheet>
```

4. Elaborate the XSL Formatting Object Language

- ✓ XSL Formatting Objects is itself an XML-based markup language that lets you specify in great detail the pagination, layout, and styling information that will be applied to your content.
- ✓ Page Masters and Page Sequences
- ✓ A sequence of pages follows a page master.
- ✓ A page master defines the basic layout of a page and the regions on the page.

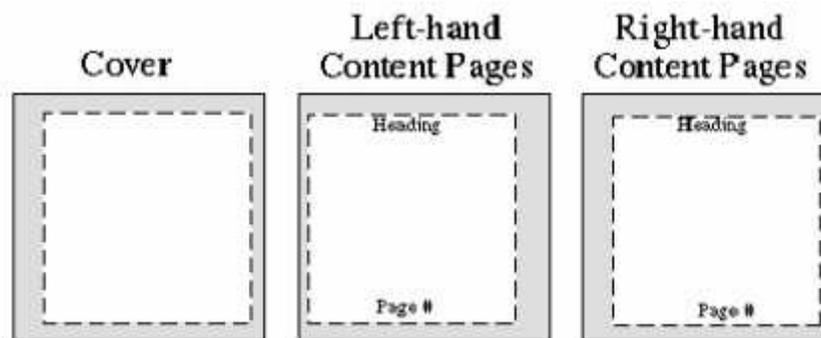
There is a "simple page master" built-in with the following regions:

region-before - The header of the page.

- region-after - The footer of the page.
- region-start - The "left margin" region.
- region-end - The "right margin" region.
- region-body - The "right margin" region.

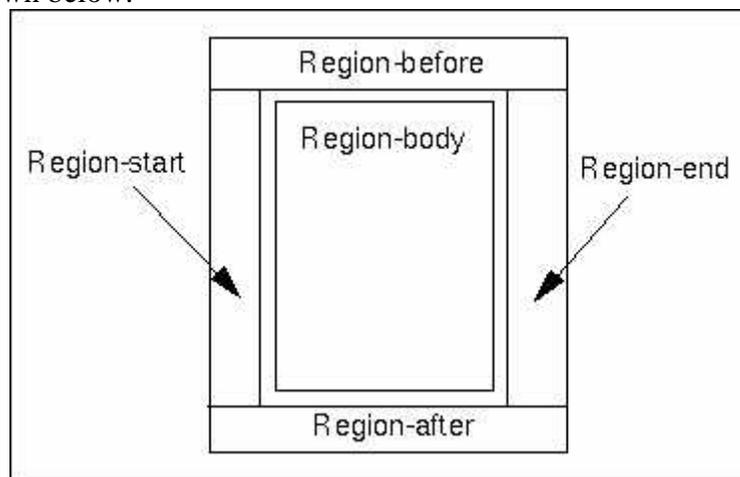
- **Page Layouts:**

After the FO document's beginning `<fo:root>` tag, we have to describe what kinds of pages our document can have. Our document will have three kinds of pages shown in the diagram below. To accommodate the stapling area, the cover page and right-hand pages will have more margin space at the left. The content pages will also have a region for a header and footer.



- Let's start out by specifying the page widths and heights and margins. The units below are all in centimeters, but you may use any of the CSS units, such as px (pixel), pt (point), em, in, mm, etc. Each of these specifications is called a simple-page-master and must be given a master-name so you can refer to it later.
- `<fo:layout-master-set>`
- `<fo:simple-page-master master-name="cover"`
- `page-height="12cm"`
- `page-width="12cm"`
- `margin-top="0.5cm"`
- `margin-bottom="0.5cm"`
- `margin-left="1cm"`
- `margin-right="0.5cm">`
- `</fo:simple-page-master>`
-
- `<fo:simple-page-master master-name="leftPage"`
- `page-height="12cm"`
- `page-width="12cm"`
- `margin-left="0.5cm"`
- `margin-right="1cm"`
- `margin-top="0.5cm"`
- `margin-bottom="0.5cm">`
- `</fo:simple-page-master>`
-
- `<fo:simple-page-master master-name="rightPage"`
- `page-height="12cm"`

- `page-width="12cm"`
- `margin-left="1cm"`
- `margin-right="0.5cm"`
- `margin-top="0.5cm"`
- `margin-bottom="0.5cm">`
- `</fo:simple-page-master>`
- `<!-- more info will go here -->`
- `</fo:layout-master-set>`
- The margins are areas which will not contain any printed output.
- **The Content Area**
- All of the printing occurs within the dotted lines in the diagram above. This is the page content area (officially called the page-reference-area), which can be divided into five regions as shown below.



- **Region Dimensions**
- The cover page doesn't need a header or footer, so we need only specify information for the region-body by adding the information shown in bold below.
- `<fo:simple-page-master master-name="cover"`
- `page-height="12cm"`
- `page-width="12cm"`
- `margin-top="0.5cm"`
- `margin-bottom="0.5cm"`
- `margin-left="1cm"`
- `margin-right="0.5cm">`
- **`<fo:region-body`**
- **`margin-top="3cm" />`**
- `</fo:simple-page-master>`
- The left and right pages will have a header and footer, so we must specify the extent of the region-before and region-after.
- `<fo:simple-page-master master-name="leftPage"`
- `page-height="12cm"`
- `page-width="12cm"`
- `margin-left="0.5cm"`
- `margin-right="1cm"`
- `margin-top="0.5cm"`

- `margin-bottom="0.5cm">`
- `<fo:region-before extent="1cm"/>`
- `<fo:region-after extent="1cm"/>`
- `<fo:region-body`
- `margin-top="1.1cm"`
- `margin-bottom="1.1cm" />`
- `</fo:simple-page-master>`
-
- `<fo:simple-page-master master-name="rightPage"`
- `page-height="12cm"`
- `page-width="12cm"`
- `margin-left="1cm"`
- `margin-right="0.5cm"`
- `margin-top="0.5cm"`
- `margin-bottom="0.5cm">`
- `<fo:region-before extent="1cm"/>`
- `<fo:region-after extent="1cm"/>`
- `<fo:region-body`
- `margin-top="1.1cm"`
- `margin-bottom="1.1cm" />`
- `</fo:simple-page-master>`

5. Elucidate Modelling database in XML.

- Review the database schema.
- Construct the desired XML document.
- Define a schema for the XML document.
- Create the JAXB binding schema.
- Generate the JAXB classes based on the schema.
- Develop a Data Access Object (DAO).
- Develop a servlet for HTTP access.

UNIT – III

12. List out some characteristics of Contemporary SOA.

Some of the characteristics of contemporary SOA are:-

- i. Contemporary SOA is at the core of the service oriented platform.
- ii. Contemporary SOA increases quality of service.
- iii. Contemporary SOA is fundamentally autonomous.
- iv. Contemporary SOA is based on open standards.
- v. Contemporary SOA supports vendor diversity.
- vi. Contemporary SOA fosters intrinsic interoperability.
- vii. Contemporary SOA promotes discovery.
- viii. Contemporary SOA promotes federation.
- ix. Contemporary SOA promotes architectural composability.
- x. Contemporary SOA fosters inherent reusability.

13. What are the benefits of SOA?

The benefits of SOA are:

- i. Improved integration and intrinsic interoperability
- ii. Inherent reuse
- iii. Streamlined architectures and solutions
- iv. Leveraging the legacy investment
- v. Establishing standardized XML data representation
- vi. Focused investment on communications infrastructure
- vii. “Best-of-breed” alternatives
- viii. Organizational agility

14. What are the common pitfalls of adopting SOA?

The common pitfalls of adopting SOA are:

- i. Building service oriented architectures like traditional distributed architectures
- ii. Not standardizing SOA
- iii. Not creating a transition plan
- iv. Not starting with an XML foundation architecture
- v. Not understanding SOA performance requirements
- vi. Not understanding web services security
- vii. Not keeping in touch with product platforms and standards development.

4. Comparing SOA with client server architecture and distributed internet architectures.

What is Architecture?

- ✓ Application architecture.
- ✓ Enterprise architecture.
- ✓ SOA vs. Client server architecture.
- ✓ Client server architecture.

Characteristics. The primary characteristics of the two tier client server architectures is given below which is compared to SOA

- ✓ Application logic
- ✓ Application processing
- ✓ Technology
- ✓ Security
- ✓ Administration

5. Explain the Service Layers in detail.

Service layer abstraction:

- ✓ Problems solved by layering services.
- ✓ Layer of abstraction.

Application Service Layer:

- ✓ Characteristics.
- ✓ Hybrid application services.
- ✓ Application integration services.
- ✓ Proxy service.

**Business Service Layer
Orchestration Service Layer.**

UNIT – IV

1. Explain Web Services in detail.

Service Roles:

- ✓ Service Provider.
- ✓ Service Requestor.
- ✓ Service Intermediaries.

Service Models:

- ✓ Business Service Model.
- ✓ Utility service Model.
- ✓ Controller Service Model.

2. Elaborate the WSDL in detail.

- ✓ Service endpoints and service descriptions.
- ✓ Abstract description.
- ✓ Concrete description.
- ✓ Metadata and Service contracts.
- ✓ Semantic descriptions.
- ✓ Service description advertisement and discovery.

3. Elucidate the Messaging with SOAP.

Soap Message Format:

- ✓ Basic Structure.
- ✓ Message Styles.
- ✓ Attachments.
- ✓ Faults.

Nodes:

- ✓ Node Types.
- ✓ SOAP Intermediaries.
- ✓ Message Paths.
- ✓

4. Explain UDDI with its registry.

Universal Description Discovery and Integration (UDDI):

- ✓ UDDI Data model.
- ✓ Information in UDDI Registry.
- ✓ UDDI nodes, Registries, and affiliated registries.
- ✓ UDDI Registry APIs.

5. Discuss the Message Exchange patterns.

Definitions.

Primitive MEPs

Request – Response

Fire- and – forget

Complex MEPS

MEPs and SOAP

MEPs and WSDL

Request – Response operation

Solicit – Response operation

One way operation

Notification operation

MEPs and SOA

6. Describe Orchestration and Choreography.

Orchestration:

Business protocols and process definition

Process services and partner services

Basic activities and structured activities

Sequence, flows and links

Orchestration and activities

Orchestration and co -ordination

Orchestration and SOA

Choreography:

Collaboration.

Roles and participants.

Relationships and channels.

Interaction and work units.

Reusability, composability and modularity.

Orchestration and choreography.

UNIT – V

1. Discuss in detail about service modeling.

Service versus service candidate

Process description

Decompose the business process

Identify business service operation candidates

Abstract orchestration logic

Create business service candidate.

Refine and apply principles of services orientation

Identify candidate service composition

Revise business service operation grouping

Analyze application processing requirements

Identify application service operation candidate

Create application service candidate

Revise candidate service composition

Revise application service operation grouping.

2. Write in detail about Service Oriented Design.

Introduction to service oriented design

Objectives of service oriented design
 Service oriented design process
 Prerequisites.

3. Write in detail about SOAP with examples.

The envelope element
 The header element
 The body element
 The fault element.

4. Explain about SOA Composition Guidelines.

Steps to composition SOA
 Steps to composing SOA
 Consideration for choosing service layers
 Consideration for positioning core SOA standards
 Consideration for choosing SOA extensions

5. Discuss in detail about SOA Support with J2EE and its API's.

- 1) Platform overview
 - 1) Primitive SOA support
 - 2) Support for service orientation principles
 - 3) Contemporary SOA support.

6. Discuss in detail about SOA Support with .NET.

Platform overview
 Primitive SOA support
 Support for service orientation principles
 Contemporary SOA support.

7. Discuss in detail about the WS – BPEL with code snippets.

WS-BPEL language basics
 A brief history of BPEL 4 WS and WS-BPEL
 Prerequisites
 The process element
 The partner links and partner link element
 The partner link type element
 The variables element

8. Explain about WS-Coordination with code example.

The coordination context element
 The identifier and expires elements
 The coordination type element
 The registration service element
 Designating the WS-Business activity coordination type
 Designating the WS-Atomic transaction coordination type.

9. Explain about WS-Policy with code example.

The policy element and common policy assertion

- The exactly one element
- The all element
- The usage attributes
- The preference attributes
- The policy reference element
- The policy URIs attributes
- The policy Attachment element
- Additional types of policy assertions.

10. Explain about WS-Security with code example.

- The security element (ws-security)
- The username token, username and password elements (WS-security)
- The binary security token element (WS-security)
- The security token reference element (WS-security)

Information Retrieval

UNIT – I

INTRODUCTION

-PART – A

QUESTIONS AND ANSWERS

1. Define information retrieval.(nov/dec 2016)

Information retrieval (IR) is finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections (usually stored on computers).

2. What are the applications of IR?

- Indexing
- Ranked retrieval
- Web search
- Query processing

3. Give the historical view of Information Retrieval.

- Boolean model, statistics of language (1950's)
- Vector space model, probabilistic indexing, relevance feedback (1960's)
- Probabilistic querying (1970's)
- Fuzzy set/logic, evidential reasoning (1980's)
- Regression, neural nets, inference networks, latent semantic indexing, TREC (1990's)

4. What are the components of IR?(nov/dec 2016)

- The document subsystem
- The indexing subsystem
- The vocabulary subsystem
- The searching subsystem
- The ser-system interface
- The matching subsystem

5. How to AI applied in IR systems?(nov/dec 2016)

Four main roles investigated

- Information characterisation
- Search formulation in information seeking
- System Integration

- ☐ Support functions

6. How to introduce AI into IR systems?

- ☐ User simply enters a query, suggests what needs to be done, and the system executes the query to return results.
- ☐ First signs of AI. System actually starts suggesting improvements to user.
- ☐ Full Automation. User queries are entered and the rest is done by the system.

7. What are the areas of AI for information retrieval?

- ☐ Natural language processing
- ☐ Knowledge representation
- ☐ Machine learning
- ☐ Computer Vision
- ☐ Reasoning under uncertainty
- ☐ Cognitive theory

8. Give the functions of information retrieval system.

- ☐ To identify the information(sources) relevant to the areas of interest of the target users community
- ☐ To analyze the contents of the sources(documents)
- ☐ To represent the contents of the analyzed sources in a way that will be suitable for matching user's queries
- ☐ To analyze user's queries and to represent them in a form that will be suitable for matching with the database
- ☐ To match the search statement with the stored database
- ☐ To retrieve the information that is relevant
- ☐ To make necessary adjustments in the system based on feedback form the users.

9. List the issues in information retrieval system.

- ☐ Assisting the user in clarifying and analyzing the problem and determining information needs.
- ☐ Knowing how people use and process information.
- ☐ Assembling a package of information that enables group the user to come closer to a solution of his problem.
- ☐ Knowledge representation.
- ☐ Procedures for processing knowledge/information.
- ☐ The human-computer interface.
- ☐ Designing integrated workbench systems.

- ❑ Designing user-enhanced information systems.
- ❑ System evaluation.

10. What are some open source search frameworks?

- ❑ Google Search API
- ❑ Apache Lucene
- ❑ blekko API
- ❑ Carrot2
- ❑ Egothor
- ❑ Nutch

11. Define relevance.

Relevance appears to be a subjective quality, unique between the individual and a given document supporting the assumption that relevance can only be judged by the information user. Subjectivity and fluidity make it difficult to use as measuring tool for system performance.

12. What is meant by stemming?

Stemming is techniques used to find out the root/stem of a word. Used to improve effectiveness of IR and text mining. Stemming usually refers to a crude heuristic process that chops off the ends of words in the hope of achieving this goal correctly most of the time, and often includes the removal of derivational affixes.

13. Define indexing & document indexing.

Association of descriptors (keywords, concepts, metadata) to documents in view of future retrieval.

Document indexing is the process of associating or tagging documents with different “search” terms. Assign to each document (respectively query) a descriptor represented with a set of features, usually weighted keywords, derived from the document (respectively query) content.

14. Discuss the impact of IR on the web.

The impacts of information retrieval on the web are influenced in the following areas.

- ❑ Web Document Collection
- ❑ Search Engine Optimization
- ❑ Variants of Keyword Stuffing
- ❑ DNS cloaking: Switch IP address
- ❑ Size of the Web
- ❑ Sampling URLs
- ❑ Random Queries and Searches

15. List Information retrieval models.(nov/dec 2016)

- Boolean model
- Vector space model
- Statistical language model

16. Define web search and web search engine.

Web search is often not informational -- it might be navigational (give me the url of the site I ant to reach) or transactional (show me sites where I can perform a certain transaction, e.g. shop, download a file, or find a map).

Web search engines crawl the Web, downloading and indexing pages in order to allow full-text search. There are many general purpose search engines; unfortunately none of them come close to indexing the entire Web. There are also thousands of specialized search services that index specific content or specific sites.

17. What are the components of search engine?

Generally there are three basic components of a search engine as listed below:

1. Web Crawler
2. Database
3. Search Interfaces

18. Define web crawler.

This is the part of the search engine which combs through the pages on the internet and gathers the information for the search engine. It is also known as spider or bots. It is a software component that traverses the web to gather information.

19. What are search engine processes?

Indexing Process

- Text acquisition
- Text transformation
- Index creation

Query Process

- User interaction
- Ranking
- Evaluation

20. How to characterize the web?

Web can be characterized by three forms

- Search engines -AltaVista
- Web directories -Yahoo
- Hyperlink search-Web Glimpse

21. What are the challenges of web?

- Distributed data
- Volatile data
- Large volume
- Unstructured and redundant data
- Data quality
- Heterogeneous data

PART – B

QUESTIONS AND ANSWERS

1. Write about history of Information Retrieval.

Early keyword-based engines ca. 1995-1997

Altavista, Excite, Infoseek, Inktomi, Lycos

1998+: Link-based ranking pioneered by Google

Blew away all early engines save Inktomi

2005+: Google gains search share, dominating in Europe and very strong in North America

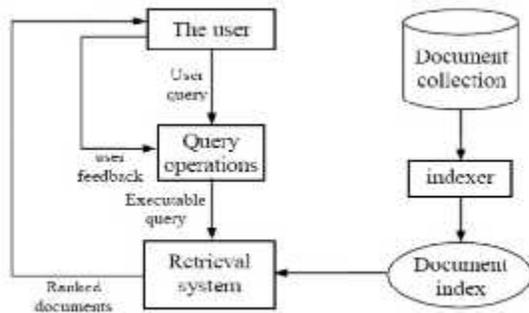
2009: Yahoo! and Microsoft propose combined paid search offering

2. Explain the Information Retrieval. (nov/dec 2016)

IR helps users find information that matches their information needs expressed as queries.

Historically, IR is about document retrieval, emphasizing document as the basic unit.

– Finding documents relevant to user queries.



Architecture of Information Retrieval

IR Queries

- Keyword queries
- Boolean queries (using AND, OR, NOT)
- Phrase queries
- Proximity queries
- Full document queries
- Natural language questions

IR Models

– Boolean model

- Vector space model
- Statistical language model

3. Discuss the influence of AI in Information Retrieval.

Areas of AI for IR

- ☐ Natural language processing
- ☐ Knowledge representation
 - ❖ Expert systems
 - ❖ Ex: Logical formalisms, conceptual graphs, etc
- ☐ Machine learning
 - ❖ Short term: over a single session
 - ❖ Long term: over multiple searches by multiple users
- ☐ Computer Vision
 - ❖ Ex: OCR
- ☐ Reasoning under uncertainty
 - ❖ Ex: Dempster-Shafer, Bayesian networks, probability theory, etc
- ☐ Cognitive theory
 - ❖ Ex: User modelling

AI applied to IR

- ☐ Four main roles investigated
 - ❖ Information characterisation
 - ❖ Search formulation in information seeking
 - ❖ System Integration
 - ❖ Support functions
- ☐ AI has a very valuable contribution to make
 - ❖ Specialised systems where domain is controlled, well-integrated and understood
 - ❖ Support functions
 - ❖ Case-based reasoning and dialogue functions
 - ❖ Integrated functions

4. Explain in detail about Search Engine.

Search Engine is in the field of IR .Searching authors, titles and subjects in library card catalogs or computers. Document classification and categorization, user interfaces, data visualization, filtering

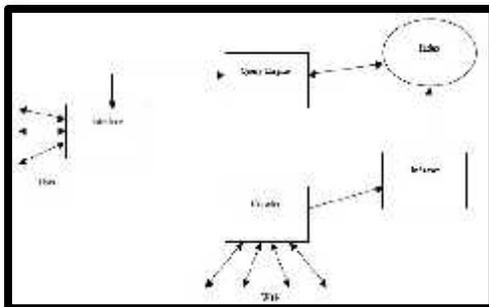
Types of Search Engines

- ☐ Search by Keywords (e.g. AltaVista, Excite, Google, and Northern Light)
- ☐ Search by categories (e.g. Yahoo!)
- ☐ Specialize in other languages (e.g. Chinese Yahoo! and Yahoo! Japan)
- ☐ Interview simulation (e.g. Ask Jeeves!)

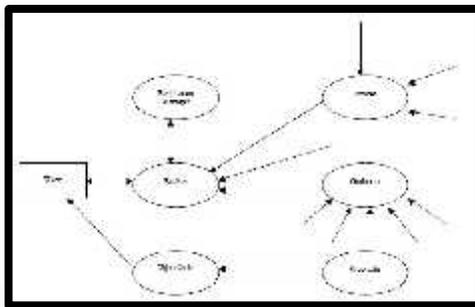


Search Engine Architectures

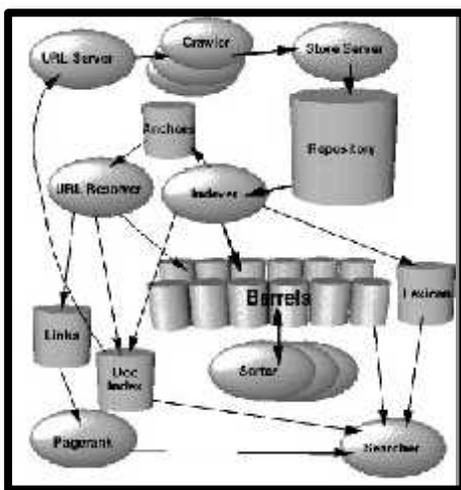
- ❖ AltaVista
- ❖ Harvest
- ❖ Google



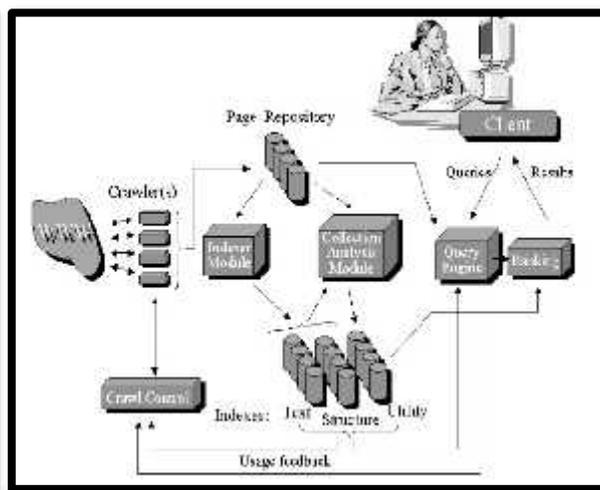
AltaVista Architecture



Harvest Architecture



Google Architecture



Modern Search Engine

5. Discuss web information retrieval system.

Web Search Engine Evolution

- ☐ Web Search 1.0 –Traditional Text Retrieval
- ☐ Web Search 2.0 –Page-level Relevance Ranking
- ☐ Next Generation Web Search

Web Analysis and Its Relationship to IR

- ☐ Goals of Web analysis:
 - ❖ Improve and personalize search results relevance
 - ❖ Identify trends
- ☐ Classify Web analysis:
 - ❖ **Web content analysis**
 - ❖ **Web structure analysis**
 - ❖ **Web usage analysis**
- ☐ Searching the Web
- ☐ Analyzing the Link Structure of Web Pages
- ☐ Web Content Analysis

Trends in Information Retrieval

Faceted search

- ❖ Allows users to explore by filtering available information
- ❖ **Facet** :Defines properties or characteristics of a class of objects

Social search

New phenomenon facilitated by recent Web technologies: **collaborative social search, guided participation**

UNIT II
INFORMATION RETRIEVAL
PART – A
QUESTIONS AND ANSWERS

1. What are the three classic models in information retrieval system?

1. Boolean model
2. Vector Space model
3. Probabilistic model

2. What is the basis for boolean model?

Simple model based on set theory and Boolean algebra

- Documents are sets of terms
- Queries are specified as Boolean expressions on terms

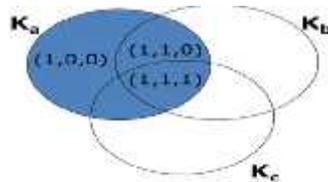
3. How can we represent the queries in boolean model?

Queries specified as boolean expressions

- Precise semantics
- Neat formalism
- $q = k_a \cap (k_b \cup k_c)$

4. Definition of boolean model?

Index term weight variables all are binary



- $w_{ij} \in \{0,1\}$
- Query $q = k_a \cap (k_b \cup k_c)$
- $\text{sim}(q_i, d_j) = 1$, i.e. doc's are relevant

0, otherwise i.e. doc's are not relevant

$q = k_a \square (k_b \square \square k_c)$, can be written as disjunctive normal

form, $\text{vec}(q_{\text{dnf}}) = (1,1,1) \vee (1,1,0) \vee (1,0,0)$

5. What are the advantages of Boolean model?

- Clean Formalism
- Easy to implement
- Intuitive concept
- Still it is dominant model for document database systems

6. What are the disadvantages of Boolean model?

Exact matching may retrieve too few or too many documents

- Difficult to rank output, some documents are more important than others
- Hard to translate a query into a Boolean expression
- All terms are equally weighted
- More like data retrieval than information retrieval
- No notion for partial matching

7. Define the Vector Model

This model recognizes that the Use of binary weights is too limiting and proposes a framework in which partial matching is possible.

- Non-binary weights provide consideration for partial matches
- These term weights are used to compute a degree of similarity **between a query and each document**
- Ranked set of documents provides for better matching
 - $w_{i,j} \geq 0$ associated with the pair (k_i, d_j)
 - $\text{vec}(d_j) = (w_{1,j}, w_{2,j}, \dots, w_{t,j})$ $\vec{d}_j = (w_{1,j}, w_{2,j}, \dots, w_{t,j})$
 - $w_{i,q} \geq 0$ associated with the pair (k_i, q)
 - $\text{vec}(q) = (w_{1,q}, w_{2,q}, \dots, w_{t,q})$ $\vec{q} = (w_{1,q}, w_{2,q}, \dots, w_{t,q})$

■ t- total no. Of index terms in the collection

$$\blacksquare \text{ Sim}(d_j, q) = \frac{\sum_{i=1}^t w_{i,j} \times w_{i,q}}{\sqrt{\sum_{i=1}^t w_{i,j}^2} \times \sqrt{\sum_{i=1}^t w_{i,q}^2}}$$

8. What are the advantages of Vector Model?

- Simple model based on linear algebra
- Term weights not binary
- Allows computing a continuous degree of similarity between queries and documents
- Allows ranking documents according to their possible relevance
- Allows partial matching
- Allows efficient implementation for large document collections

8. What are the disadvantages of Vector Model?

- Index terms are assumed to be mutually independent
- Search keywords must precisely match document terms
- Long documents are poorly represented
- The order in which the terms appear in the document is lost in the vector space representation
- Weighting is intuitive, but not very formal

9. What are the Parameters in calculating a weight for a document term or query term?

Term Frequency (tf): Term Frequency is the number of times a term i appears in document j (tf_{ij})

– Document Frequency (df): Number of documents a term i appears in, (df_i).

– Inverse Document Frequency (idf): A discriminating measure for a term i in collection, i.e., how discriminating term i is. ($idf_i = \log_{10}(n / df_i)$, where n is the number of document

10. How can you calculate tf and idf in vector model?

- The normalized frequency (term factor) $f_{i,j}$ is,

$f_{i,j} = \text{freq}_{i,j} / \max_l \text{freq}_{l,j}$; if k_i not appear in d_j then $f_{i,j} = 0$;

- Inverse document frequency (idf)

is $\text{idf}_i = \log(N/n_i)$ or $\text{idf}_i = \log(D/\text{df}_i)$

Where

- N- total no.of documents in the collection
- n_i – no.of documents in which the index terms k_i appears
- $\text{freq}_{i,j}$ – frequency of the term k_i in the document d_j
- \max_l – maximum over all terms frequencies

2016) 11.How do you calculate the term weighting in document and Query term weight ?(nov/dec

- Term weighting is,
- $w_{i,j} = \text{tf} * \text{idf}_i$ i.e. $w_{i,j} = f_{i,j} * \log(N/n_i)$
- Query term weight is,
- $w_{i,q} = (0.5 + 0.5 * \text{freq}_{i,q} / \max_l \text{freq}_{l,q}) * \log(N/n_i)$

12.Write the cosine similarity function for vector space model:-

Cosine $\frac{Q \cdot D}{|Q| \cdot |D|}$

$$\therefore \text{Sim}(Q, D) = \frac{\sum_i w_{Q,j} w_{i,j}}{\sqrt{\sum_j w_{Q,j}^2} \sqrt{\sum_i w_{i,j}^2}}$$

13.Define Probabilistic model or Binary Independence Retrieval :-

The Objective of Probabilistic model is to capture the IR problem using a probabilistic framework

Given a user query, there is an ideal answer set

- Querying as specification of the properties of this ideal answer set
- Definition

□
W
eig
ht
var
iab
les
all
are
bin
ary

,
i.e.
wi,
j
 \in
0,1
}
an
d
wi,
q
 \in
0,1
}

□
q-
a
qu
ery
is
a
su
bs
et
of
ind
ex
ter
ms

- R – set of doc's known (initial guess) to be relevant
- \bar{R} – the complement of R , i.e. the set of non-relevant doc's
- $P(R|d_j)$ – probability of d_j relevant to q
- $P(\bar{R}|d_j)$ - probability of d_j non-relevant to q
- $\text{sim}(d_j, q) = P(R|d_j) / P(\bar{R}|d_j)$

14. What are the Fundamental assumptions for probabilistic principle?

- q - user query, d_j – doc in the collections
- Model assumes, relevance depends on the query and the doc representation only
 - R – ideal answer set, relevant to the query
 - \bar{R} - ideal answer set, non-relevant to the query
 - Similarity to the query ratio is, i.e. probabilistic ranking computed as
 - Ratio = $P(d_j \text{ relevant-to } q) / P(d_j \text{ non-relevant-to } q)$
 - The rank minimizes the probability of the erroneous judgment

15. How can you find similarity between doc and query in probabilistic principle Using Bayes' rule?

$$\text{sim}(d_j, q) = P(d_j|R) \times P(R) / P(d_j|\bar{R}) \times P(\bar{R})$$

where

$P(d_j|R)$ - probability of randomly selecting the document d_j from the set R of relevant documents

- $P(R)$ - probability of randomly selecting the document from the entire collection is relevant

→

The meaning of $P(d_j|R)$ and $P(R)$ are analogous and complementary

- Since $P(R)$ and $P(\bar{R})$ are same for all doc's in the collection, then we write,

→

$$\text{sim}(d_j, q) \sim P(d_j|R) / P(d_j|\bar{R})$$

16. Write the advantages and disadvantages of probabilistic model:

□ Advantages

- Doc's are ranked in decreasing order of their probability of relevant
- Disadvantages
 - Need to guess the initial separation of doc's into relevant and non-relevant sets
 - All weights are binary
 - The adoption of the independence assumption for index terms
 - need to guess initial estimates for $P(k_i | R)$
 - method does not take into account tf and idf factors

17. Why the Classic IR might lead to poor retrieval?

- The user information need is more related to concepts and ideas than to index terms but in classic IR
- Unrelated documents might be included in the answer set
- Relevant documents that do not contain at least one index term are not retrieved
- Reasoning: retrieval based on index terms is vague and noisy

18. Definitions Latent Semantic Indexing Model:-

- o Let t be the total number of index terms
- o Let N be the number of documents
- o Let $\text{vec}(M) = M_{ij}$ be a term-document matrix with t -rows and N -columns
- o To each element of this matrix is assigned a weight w_{ij} associated with the pair $[k_i, d_j]$
- o The weight w_{ij} can be based on a tf-idf weighting scheme

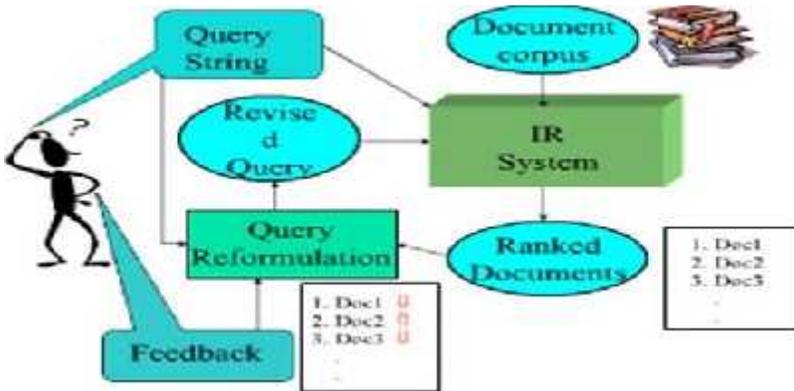
19. Write the advantages of Latent Semantic Indexing Model?

- Latent semantic indexing provides an interesting conceptualization of the IR problem
- It an efficient indexing scheme for the documents in the collection
- It provides,
 - Elimination of noise
 - Removal of redundancy

20. Define Relevance feedback model:-(**nov/dec 2016**)

After initial retrieval results are presented allow the user to provide feedback on the relevance of one or more of the retrieved documents. use this feedback information to reformulate the query and produce new results based on reformulated query. Thus allows more interactive multi pass process.

21. Draw the flow diagram for relevance feedback query processing model: (nov/dec 2016)



22. Write the types of queries:

There are 4 type of queries such as Structured queries, Pattern matching queries, Boolean queries, Context Queries

23. Give short notes for User Relevance Feedback:

It is the most popular query formulation strategy. In a relevance feedback cycle, the user presented with a list of the retrieved documents. Then examine them, marks those which are relevant

Only to 10 (or 20) ranked documents are examined

- o Selecting important terms, or expression, attached to the documents
- o Enhancing the important of these terms in a new query formulation
- o The new query will be

□ 1. Moved towards the relevant documents, 2. Away from the non-relevant ones.

24. What are the two basic approaches in User Relevance Feedback for query processing?

- 1) Query expansion- Expand queries with the vector model
- 2) Term reweighting –
 - i) Reweight query terms with the probabilistic model
 - ii) Reweight query terms with a variant of the probabilistic model

25. What are the Advantages of User Relevance Feedback method?

- It shields the user from the details of the query reformulation process because all the user has to provide is a relevance judgement on documents
- It breaks down the whole searching task into a sequence of small steps which are easier to grasp
- It provides a controlled process designed to emphasize some terms (relevant ones) and de-emphasize others (non-relevant ones)

26. What are the three classic and similar ways to calculate the modified query q_m ?

$$\text{Standard_Rocchio: } \vec{q}_m = \alpha \vec{q} + \frac{\beta}{|D_r|} \sum_{d_j \in D_r} \vec{d}_j - \frac{\gamma}{|D_n|} \sum_{d_i \in D_n} \vec{d}_i$$

$$\text{Ide_Regular: } \vec{q}_m = \alpha \vec{q} + \beta \sum_{d_j \in D_r} \vec{d}_j - \gamma \sum_{d_i \in D_n} \vec{d}_i$$

$$\text{Ide_Der_Hi: } \vec{q}_m = \alpha \vec{q} + \beta \sum_{d_j \in D_r} \vec{d}_j - \gamma \max_{d_i \in D_n} (\vec{d}_i)$$

27. What are the advantages and disadvantages of query processing?

Advantages :

It is simple:

1) The fact that the modified term weights are computed directly from the set of retrieved documents

2) It gives good results:

Observed experimentally and are due to the fact that the modified query vector does reflect a portion of the intended query semantics

Disadvantages

No optimality

PART – B

QUESTIONS AND ANSWERS

1. Explain in detail about vector-space retrieval models with an example:-
 - o Use of binary weights is too limiting
 - o Non-binary weights provide consideration for partial matches
 - o These term weights are used to compute a degree of similarity between a query and each document
 - o Ranked set of documents provides for better matching
 - o Define:
 - $w_{i,j} \geq 0$ associated with the pair (k_i, d_j)
 - $\text{vec}(d_j) = (w_{1,j}, w_{2,j}, \dots, w_{t,j})$
 - $W_{i,q} \geq 0$ associated with the pair (k_i, q)
 - $\text{vec}(q) = (w_{1,q}, w_{2,q}, \dots, w_{t,q})$
 - t - total no. Of index terms in the collection
 - o Use of binary weights is too limiting
 - o Non-binary weights provide consideration for partial matches
 - o These term weights are used to compute a degree of similarity between a query and each document
 - o Ranked set of documents provides for better matching
 - o Define:
 - $w_{i,j} \geq 0$ associated with the pair (k_i, d_j)
 - $\text{vec}(d_j) = (w_{1,j}, w_{2,j}, \dots, w_{t,j})$
 - $W_{i,q} \geq 0$ associated with the pair (k_i, q)
 - $\text{vec}(q) = (w_{1,q}, w_{2,q}, \dots, w_{t,q})$
 - t - total no. Of index terms in the collection
 - o Definition
 - N - total no.of documents in the collection
 - n_i – no.of documents in which the index terms k_i appears
 - $\text{freq}_{i,j}$ – frequency of the term k_i in the document d_j
 - \max_l – maximum over all terms frequencies
 - The normalized frequency (term factor) $f_{i,j}$ is,
 - **$f_{i,j} = \text{freq}_{i,j} / \max_l \text{freq}_{l,j}$** ; if k_i not appear in d_j then $f_{i,j} = 0$;
 - Inverse document frequency (idf)
 - **$\text{idf}_i = \log(N/n_i)$ or $\text{idf}_i = \log(D/\text{df}_i)$**
 - Term weighting is,
 - **$w_{i,j} = \text{tf} * \text{idf}_i$ i.e. $w_{i,j} = f_{i,j} * \log(N/n_i)$**
 - Query term weight is,

$$\blacksquare \quad w_{i,q} = (0.5 + 0.5 * \text{freq}_{i,q} / \text{maxfreq}_{l,q}) * \log(N/n_i)$$

2. Explain about Boolean model for IR:

Simple model based on set theory and Boolean algebra

Documents are sets of terms

Queries are Boolean expressions on terms

Historically the most common model

Library OPACs

Dialog system

Many web search engines, too

Queries specified as boolean expressions

Precise semantics

3. Neat formalism

4. $q = k_a \cup (k_b \cap k_c)$

- Terms are either present or absent. Thus, $w_{ij} \in \{0,1\}$
- There are three connectives used: and, or, not
- **D**: set of words (indexing terms) present in a document
 - each term is either present (1) or absent (0)
- **Q**: A Boolean expression
 - terms are index terms
 - operators are AND, OR, and NOT
- **F**: Boolean algebra over sets of terms and sets of documents
- **R**: a document is predicted as relevant to a query expression if it satisfies the query expression

((text \cup information) \cup retrieval) \cap theory)
- Each query term specifies a set of documents containing the term
- AND (\cap): the intersection of two sets
- OR (\cup): the union of two sets
- NOT ($\bar{}$): set inverse, or really set difference
- Definition
 - Index term weight variables all are binary
 - $w_{ij} \in \{0,1\}$
 - Query $q = k_a \cup (k_b \cap k_c)$
 - $\text{sim}(q_i, d_j) = 1$, i.e. doc's are relevant
0, otherwise i.e. doc's are not relevant
- $q = k_a \cup (k_b \cap k_c)$, can be written as disjunctive normal form,

$$\text{vec}(q_{dnf}) = (1,1,1) \vee (1,1,0) \vee (1,0,0)$$

3.Explain about Probabilistic IR:

- o Assuming independence index terms,

$$\text{sim}(d_j, q) \sim \frac{P(k_i | R)}{P(k_i | R) + P(\bar{k}_i | R)} * \frac{P(\bar{k}_i | R)}{P(k_i | R) + P(\bar{k}_i | R)}$$

o $P(k_i | R)$: probability that the index term k_i is present in a document randomly selected from the set R of relevant documents

o Taking logarithms, recalling that $P(k_i | R) + P(\bar{k}_i | R) = 1$

$$\text{sim}(d_j, q) \sim \log \frac{P(k_i | R)}{P(k_i | R) + P(\bar{k}_i | R)} * \frac{P(\bar{k}_i | R)}{P(k_i | R) + P(\bar{k}_i | R)}$$

$$\text{sim}(d_j, q) \sim \sum_{i=1}^t w_{i,q} * w_{i,j} * (\log \frac{P(k_i | R)}{1 - P(k_i | R)} + \log \frac{1 - P(k_i | R)}{P(k_i | R)})$$

o Which is a key expression for ranking computation in the probabilistic model
o =>Improving the Initial Ranking

4. Explain about Inverted indices, efficient processing with sparse vectors

5. Explain about Latent Semantic Indexing method:

□ Definitions

- Let t be the total number of index terms
- Let N be the number of documents
- Let $\text{vec}(M) = M_{ij}$ be a term-document matrix with t -rows and N -columns
- To each element M_{ij} of this matrix is assigned a weight w_{ij} associated with the pair $[k_i, d_j]$
- The weight w_{ij} can be based on a tf-idf weighting scheme, like Vector model

□ The matrix $\text{vec}(M)$ can be decomposed into 3 matrices (singular value decomposition) as follows:

- $(M_{ij}) = (K) (S) (D)^t$
- (K) is the matrix of eigenvectors derived from the term-term correlation matrix given by $(M)(M)^t$
- $(D)^t$ is the matrix of eigenvectors derived from the transpose of the doc-doc matrix given by $(M)^t(M)$
- (S) is an $r \times r$ diagonal matrix of singular values

Where, $r = \min(t, N)$ that is, the rank of (M_{ij})

In the matrix (S), select only the s largest singular values

- Keep the corresponding columns in (K) and (D)^t i.e. The remaining singular values of the S are deleted.
- The resultant matrix is called (M)_s and is given by

$$M_s = K_s S_s D^t$$

where $s, s < r$, is the dimensionality of a reduced concept space

- The parameter, s should be
 - large enough to allow fitting all the structure in the real data
 - small enough to allow filter out the non-relevant representational details (i.e. based on index-term representation)

6. Give brief notes about user Relevance feedback method and how it is used in query expansion: It is the most popular query formulation strategy

In a relevance feedback cycle,

- The user presented with a list of the retrieved documents
- Then examine them, marks those which are relevant
- Only to 10 (or 20) ranked documents are examined
- Selecting important terms, or expression, attached to the documents
- Enhancing the important of these terms in a new query formulation
- The new query will be
 - Moved towards the relevant documents
 - Away from the non-relevant ones

□ Two basic approaches are,

- Query expansion
- Term reweighting

7. Write the advantages and disadvantages for classic models which are used in IR and discriminate their techniques:

- a. Boolean model ,vector model , Probabilistic IR advantage and disadvantages
- b. Techniques

8. Write the formal characterization of IR Models:

- ☐ Ranking algorithms are at the core of IR systems
- ☐ A ranking algorithm operates according to basic premises regarding notation of the relevance
- ☐ We should state clearly what exactly an IR Model is
 - “An IR Model is a quadruple $[D, Q, f, R(q_i, d_j)]$ ”
 - Where,
 - ☐ D – a set composed of logical views for the documents in the collection
 - ☐ Q – a set composed of logical views for the user information needs – queries
 - ☐ f – a framework for modeling doc representations, queries and their relationships
 - ☐ $R(q_i, d_j)$ – a ranking function, $q_i \in Q$ and $d_j \in D$, ranking based on q_i
- ☐ To build the model
 - To represent the document and user information need
 - From these to form a framework in which they can be modeled
 - This framework used for constructing ranking function

9. Sort and rank the documents in descending order according to the similarity values:

Suppose we query an IR system for the query "**gold silver truck**"

The database collection consists of three documents ($D = 3$) with the following content, D_1 : "Shipment of gold damaged in a fire"

- D_2 : "Delivery of silver arrived in a silver truck"
- D_3 : "Shipment of gold arrived in a truck" Answer:

TERM VECTOR MODEL BASED ON $w_{ij} = tf_{ij} \cdot IDF_j$											
Query, Q: "gold silver truck"											
D ₁ : "Shipment of gold damaged in a fire"											
D ₂ : "Delivery of silver arrived in a silver truck"											
D ₃ : "Shipment of gold arrived in a truck"											
D = 3; IDF = log(D/df _j)											
		Counts, tf _{ij}						Weights, w _{ij} = tf _{ij} * IDF _j			
Terms	Q	D ₁	D ₂	D ₃	df	D/df _j	IDF _j	Q	D ₁	D ₂	D ₃
a	0	1	1	1	3	3/3 = 1	1	0	0	0	0
arrived	0	0	1	1	2	3/2 = 1.5	0.1761	0	0	0.1761	0.1761
carried	0	1	0	0	1	3/1 = 3	0.4771	0	0.4771	0	0
delivery	0	0	1	0	1	3/1 = 3	0.4771	0	0	0.4771	0
fire	0	1	0	0	1	3/1 = 3	0.4771	0	0.4771	0	0
gold	1	1	0	1	2	3/2 = 1.5	0.1761	0.1761	0.1761	0	0.1761
in	0	1	1	1	3	3/3 = 1	1	0	0	0	0
of	0	1	1	1	3	3/3 = 1	1	0	0	0	0
silver	1	0	2	0	1	3/1 = 3	0.4771	0.4771	0	0.9542	0
shipment	0	1	0	1	2	3/2 = 1.5	0.1761	0	0.1761	0	0.1761
truck	1	1	1	1	3	3/3 = 1	1	0.1761	0.1761	0.1761	0.1761

$$D_1 = \sqrt{0.4771^2 + 0.4771^2 + 0.1761^2 + 0.1761^2} = 0.5973 = 0.598$$

$$D_2 = \sqrt{0.1761^2 + 0.4771^2 + 0.9542^2 + 0.1761^2} = 1.000 = 1.000$$

$$D_3 = \sqrt{0.1761^2 + 0.1761^2 + 0.1761^2 + 0.1761^2} = 0.7071 = 0.707$$

$$= \frac{1}{\sqrt{4}}$$

$$Q = \sqrt{0.1761^2 + 0.4771^2 + 0.1761^2} = 0.536 = 0.536$$

$$= \frac{1}{\sqrt{4}}$$

$$Q \cdot D_1 = 0.1761 \cdot 0.1761 = 0.0310$$

$$Q \cdot D_2 = 0.1761 \cdot 0.9542 = 0.1761 \cdot 0.1761 = 0.1862$$

$$Q \cdot D_3 = 0.1761 \cdot 0.1761 = 0.1761 \cdot 0.1761 = 0.0620$$

$$\therefore Q \cdot D = \sum_{j=1}^n w_{Qj} \cdot w_{Dj}$$

$$\text{Cosine } \theta_{r,1} = \frac{Q \cdot D_1}{|Q| |D_1|} = \frac{0.0310}{0.5382 \cdot 0.7192} = 0.0801$$

$$\text{Cosine } \theta_{r,2} = \frac{Q \cdot D_2}{|Q| |D_2|} = \frac{0.4862}{0.5382 \cdot 1.3955} = 0.8246$$

$$\text{Cosine } \theta_{r,3} = \frac{Q \cdot D_3}{|Q| |D_3|} = \frac{0.0820}{0.5382 \cdot 0.3322} = 0.3271$$

$$\therefore \text{Cosine } \theta_{r,i} = \text{Sim}(Q, D_i)$$

$$\therefore \text{Sim}(Q, D_i) = \frac{\sum_j w_{i,j} \cdot w_{r,j}}{\sqrt{\sum_j w_{i,j}^2} \cdot \sqrt{\sum_j w_{r,j}^2}}$$

- Finally we sort and rank the documents in descending order according to the similarity values

□ Rank	1:	Doc	2	=	0.8246
	Rank	2:	Doc	=	0.3271
	Rank	3:	Doc	=	0.0801

UNIT-III**WEB SEARCH-LINK ANALYSIS AND SPECIALIZED SEARCH****PART – A****QUESTIONS AND ANSWERS****1. Define web search engine?**

A **web search engine** is a software system that is designed to search for information on the World Wide Web. The search results are generally presented in a line of results often referred to as search engine results pages (SERPs).

2. What are the Practical Issues in the Web?

Security Commercial transactions over the Internet are not yet a completely safe procedure Privacy Frequently, people are willing to exchange information as long as it does not become public Copyright and patent rights It is far from clear how the wide spread of data on the Web affects copyright and patent laws in the various countries Scanning, optical character recognition (OCR), and cross-language retrieval

3. What are the Main challenges posed by Web?

data-centric: related to the data itself

- distributed data
- high percentage of volatile data
- large volume of data
- unstructured and redundant data
- quality of data
- heterogeneous data

interaction-centric: related to the users and their interactions

- expressing a query
- interpreting results

User key challenge

- to conceive a good query

System key challenge

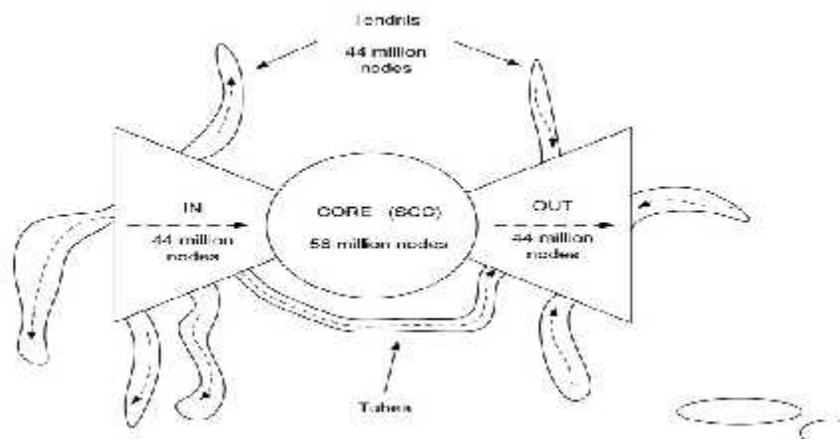
- to do a fast search and return relevant answers, even to poorly formulated queries
- Structure of the Web Graph

4. How Web can be viewed as a graph?

- where the nodes represent individual pages

□ the edges represent links between pages

5. Draw bow-tie structure of the Web?



6. Define power law?

Power Law: function that is invariant to scale changes

$$f(x) = \frac{a}{x^\alpha} \quad \text{with} \quad \alpha > 0$$

Depending on value of α , moments of distribution will be finite or not

$\alpha \leq 2$: average and all higher-order moments are infinite

$2 < \alpha \leq 3$: mean exists, but variance and higher-order moments are infinite

7. Define Logarithmic Normal distribution

$$p(x) = \frac{1}{x\sigma\sqrt{2\pi}} e^{-(\ln x - \mu)^2 / 2\sigma^2}$$

where

- x is the document size
- average size: $\mu = 9.357$ (in a sample)
- standard deviation: $\sigma = 1.318$ (in a sample)

8. Define Pareto distribution

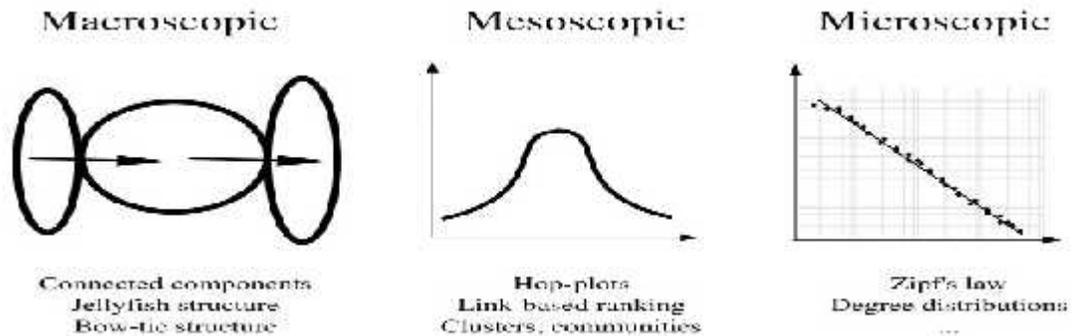
$$p(x) = \frac{\alpha k^\alpha}{x^{1+\alpha}}$$

where

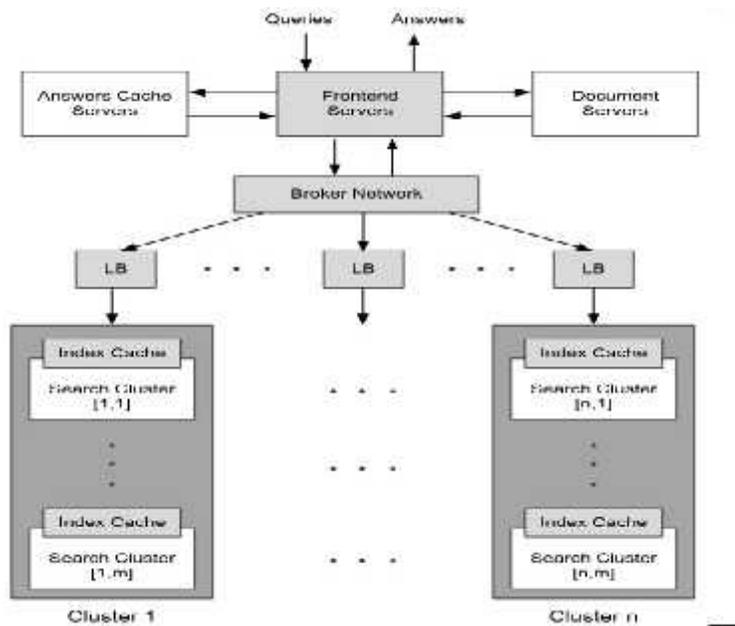
- x is measured in bytes
- k and α are parameters of the distribution

9. What are the levels of link analysis?

- microscopic level: related to the statistical properties of links and individual nodes
- mesoscopic level: related to the properties of areas or regions of the Web
- macroscopic level: related to the structure of the Web at large



10. Draw the Cluster-based Architecture?



11. Define a Web Crawler?

A Web Crawler is a software for downloading pages from the Web.

12. What are the Cycle of a Web crawling process?

- The crawler start downloading a set of seed pages, that are parsed and scanned for new links

- ❑ The links to pages that have not yet been downloaded are added to a central queue for download later
- ❑ Next, the crawler selects a new page for download and the process is repeated until a stop criterion is met.

13. List the Applications of a Web Crawler?

- ❑ create an index covering broad topics (general Web search)
- ❑ create an index covering specific topics (vertical Web search)
- ❑ archive content (Web archival)
- ❑ analyze Web sites for extracting aggregate statistics (Web characterization)
- ❑ keep copies or replicate Web sites (Web mirroring)
- ❑ Web site analysis

14. What are the Types of Web search based on crawling?(nov/dec 2016)

- ❑ General Web search: done by large search engines
- ❑ Vertical Web search: the set of target pages is delimited by a topic, a country or a language

15. What is the Main problem of focused crawling?

To predict the relevance of a page before downloading the page

16. What are the basic rules for Web crawler operation are?

- ❑ A Web crawler must identify itself as such, and must not pretend to be a regular Web user
- ❑ A Web crawler must obey the robots exclusion protocol (robots.txt)
- ❑ A Web crawler must keep a low bandwidth usage in a given Web site.

17. What are the Indexing Issues?

- Availability and speed
 - Most search engines will cache the page being referenced.
- Multiple search terms
 - OR: separate searches concatenated
 - AND: intersection of searches computed.
 - Regular expressions not typically handled.
- Parsing
 - Must be able to handle malformed HTML, partial documents

18. Why compression need?

- ❑ Use less disk space (saves money)

- ☐ Keep more stuff in memory (increases speed)
- ☐ Increase speed of transferring data from disk to memory (increases speed)
 - ☐ [read compressed data and decompress] is faster than [read uncompressed data]
- ☐ Premise: Decompression algorithms are fast
 - ☐ True of the decompression algorithms we use
- ☐ In most cases, retrieval system runs faster on compressed postings lists than on uncompressed postings lists.

19. Define Lossless vs. lossy compression

- ☐ Lossless compression: All information is preserved.
 - ☐ What we mostly do in IR.
- ☐ Lossy compression: Discard some information

PART – B

QUESTIONS AND ANSWERS

1. Briefly explain web search architectures?

Search Engine refers to a huge database of internet resources such as web pages, newsgroups, programs, images etc. It helps to locate information on World Wide Web. User can search for any information by passing query in form of keywords or phrase. It then searches for relevant information in its database and return to the user.

Search Engine Components

Generally there are three basic components of a search engine as listed below:

1. Web Crawler
2. Database
3. Search Interfaces

Web crawler

It is also known as **spider** or **bots**. It is a software component that traverses the web to gather information.

Database

All the information on the web is stored in database. It consists of huge web resources.

Search Interfaces

This component is an interface between user and the database. It helps the user to search through the database.

Search Engine Working

Web crawler, database and the search interface are the major component of a search engine that actually makes search engine to work. Search engines make use of Boolean expression AND, OR, NOT to restrict and widen the results of a search. Following are the steps that are performed by the search engine:

- ❑ The search engine looks for the keyword in the index for predefined database instead of going directly to the web to search for the keyword.
- ❑ It then uses software to search for the information in the database. This software component is known as web crawler.

- Once web crawler finds the pages, the search engine then shows the relevant web pages as a result. These retrieved web pages generally include title of page, size of text portion, first several sentences etc.

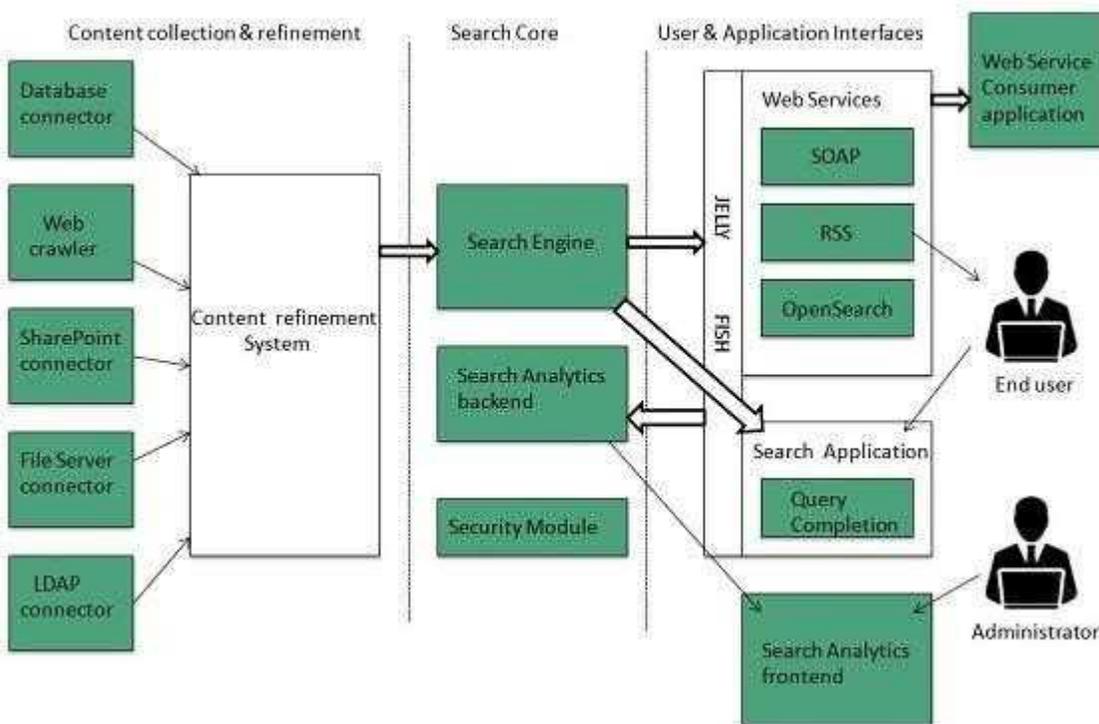
These search criteria may vary from one search engine to the other. The retrieved information is ranked according to various factors such as frequency of keywords, relevancy of information, links etc.

- User can click on any of the search results to open it.

Architecture

The search engine architecture comprises of the three basic layers listed below:

- Content collection and refinement.
- Search core
- User and application interfaces



Search Engine Processing

Indexing Process

Indexing process comprises of the following three tasks:

- ☐ Text acquisition
- ☐ Text transformation
- ☐ Index creation

Text acquisition

It identifies and stores documents for indexing.

Text Transformation

It transforms document into index terms or features.

Index Creation

It takes index terms created by text transformations and create data structures to suport fast searching.

Query Process

Query process comprises of the following three tasks:

- ☐ User interaction
- ☐ Ranking
- ☐ Evaluation

User interaction

It supportst creation and refinement of user query and displays the results.

Ranking

It uses query and indexes to create ranked list of documents.

Evaluation

It monitors and measures the effectiveness and efficiency. It is done offline.

Examples

Following are the several search engines available today:

Search Engine	Description
Google	It was originally called BackRub . It is the most popular search engine globally.
Bing	It was launched in 2009 by Microsoft . It is the latest web-based search engine that

	also delivers Yahoo's results.
Ask	It was launched in 1996 and was originally known as Ask Jeeves . It includes support for match, dictionary, and conversation question.
AltaVista	It was launched by Digital Equipment Corporation in 1995. Since 2003, it is powered by Yahoo technology.
AOL.Search	It is powered by Google.
LYCOS	It is top 5 internet portal and 13th largest online property according to Media Matrix.
Alexa	It is subsidiary of Amazon and used for providing website traffic information.

2. Explain crawling and types of crawling?(nov/dec 2016)

Crawler

Identifies and acquires documents for search engine

Many types – web, enterprise, desktop

Web crawlers follow *links* to find documents

Must efficiently find huge numbers of web pages (*coverage*) and keep them up-to-date (*freshness*)

Single site crawlers for *site search*

Topical or *focused* crawlers for verticalsearch

Document crawlers for enterprise and desktop search

Follow links and scan directories

Web crawlers

Starts with a set of *seeds*, which are a set of URLs given to it as parameters

Seeds are added to a URL request queue

Crawler starts fetching pages from the request queue

Downloaded pages are parsed to find link tags that might contain other useful URLs to fetch

New URLs added to the crawler's request queue, or *frontier*

Continue until no more new URLs or disk full

Explain each types in details.

3.Explain XML retrieval?(nov/dec 2016)

Document-oriented XML retrieval

❑ Document vs. data- centric XML retrieval (recall)

❑ Focused retrieval

❑ Structured documents

- Structured document (text) retrieval
 - XML query languages
 - XML element retrieval
 - (A bit about) user aspects
- Explain the above in details.

4. Describe index compression techniques?

Uncompressed indexes are large

- It might be useful for some modern devices to support information retrieval techniques that would not be able to do with uncompressed indexes

Types of Compression

■ Lossy

- Compression that involves the removal of data.

■ Loseless

Compression that involves no removal of data.

■ A lossy compression scheme

- Static Index Pruning

■ Loseless compression

- Elias Codes
- n-s encoding
- Golomb encoding
- Variable Byte Encoding (vByte)
- Fixed Binary Codewords
- CPSS-Tree

5. Write a detail note on how to measure size of web?

How fast does it index

Number of documents/hour
(Average document size)

How fast does it search

Latency as a function of index size

Expressiveness of query language

Ability to express complex information needs
Speed on complex queries

Uncluttered UI

Is it free?

All of the preceding criteria are *measurable*: we can quantify speed/size
we can make expressiveness precise

The key measure: user happiness

What is this?

Speed of response/size of index are factors

But blindingly fast, useless answers won't make a user happy

Need a way of quantifying user happiness

Issue: who is the user we are trying to make happy?

Depends on the setting

Web engine:

User finds what s/he wants and returns to the engine

Can measure rate of return users

User completes task – search as a means, not end

See Russell <http://dmrussell.googlepages.com/JCDL-talk-June-2007-short.pdf>

eCommerce site: user finds what s/he wants and buys

Is it the end-user, or the eCommerce site, whose happiness we measure?

Measure time to purchase, or fraction of searchers who become buyers?

6. Explain all in details. Write a note on search engine optimization/spam?

Motives

Commercial, political, religious, lobbies

Promotion funded by advertising budget

Operators

Contractors (Search Engine Optimizers) for lobbies, companies

Web masters

Hosting services

Forums

E.g., Web master world (www.webmasterworld.com)

Search engine specific tricks

Discussions about academic papers

More spam techniques

Doorway pages

Pages optimized for a single keyword that re-direct to the real target page

Link spamming

Mutual admiration societies, hidden links, awards – more on these later

Domain flooding: numerous domains that point or re-direct to a target page

Robots

Fake query stream – rank checking programs

“Curve-fit” ranking programs of search engines

Millions of submissions via Add-Url

UNIT IV**WEB SEARCH – LINK ANALYSIS AND SPECIALIZED SEARCH****PART – A****QUESTIONS AND ANSWERS**

1. What is the use of Link analysis?

Link analysis which is used to efficiently identify web communities, based on the structure of the web graph.

2. Define link spam:

Link spam are links between pages that are specifically set up to take advantage of link-based ranking algorithms such as Google's Page Rank (PR). Links added to a web page for the purpose of spam indexing

3. Write any one of the link analysis technique:

Our first technique for link analysis assigns to every node in the web graph a numerical score between 0 and 1, known as its Page Rank. The Page Rank of a node will depend on the link structure of the web graph. Given a query, a web search engine computes a composite score for each web page that combines hundreds of features such as cosine similarity and term proximity, together with the PageRank score. This composite score is used to provide a ranked list of results for the query.

4. How can we assign a page Rank score to each node of the graph?

In assigning a Page Rank score to each node of the web graph, we use the teleport operation in two ways: (1) When at a node with no out-links, the surfer invokes the teleport operation. (2) At any node that has outgoing links, the surfer invokes the teleport operation with probability α and the standard random walk with probability $1 - \alpha$, where $0 < \alpha < 1$ is a fixed parameter chosen in advance. Typically, α might be 0.1.

5. How the web pages will be scored based on queries?

For a given a query, every web page is assigned two scores.

One is called its **hub** score and the other its **authority** score. For any query, we compute two ranked lists of results rather than one. The ranking of one list is induced by the hub scores and that of the other by the authority scores.

6. Explain authority with an example:

Authority: The pages that will emerge with high authority scores.

Example: In this approach stems from a particular insight into the creation of web pages, that there are two primary kinds of web pages useful as results for broad-topic searches. By a broad topic search we mean an informational query such as "I wish to learn about leukemia". There are authoritative sources of information on the topic; in this case, the National Cancer Institute's page on leukemia would be such a page. We will call such pages authorities; in the computation we are about to describe, they are the pages that will emerge with high authority scores.

7. Explain hub with an example:-

Hub: These hub pages are the pages that will emerge with high hub scores

On the other hand, there are many pages on the Web that are hand-compiled lists of links to authoritative web pages on a specific topic. These hub pages are not in themselves authoritative sources of topic-specific information, but rather compilations that someone with an interest in the topic has spent time putting together. The approach we will take, then, is to use these hub pages to discover the authority pages. In the computation we now develop, these hub pages are the pages that will emerge with high hub scores

8. What is relevance at information retrieval?

how well a retrieved document or set of documents meets the information need of the user. Relevance may include concerns such as timeliness, authority or novelty of the result.

9. Define ranking for web?

When the user gives a query, the index is consulted to get the documents most relevant to the query. The relevant documents are then ranked according to their degree of relevance, importance etc.

10. Difficulties in Evaluating IR Systems?

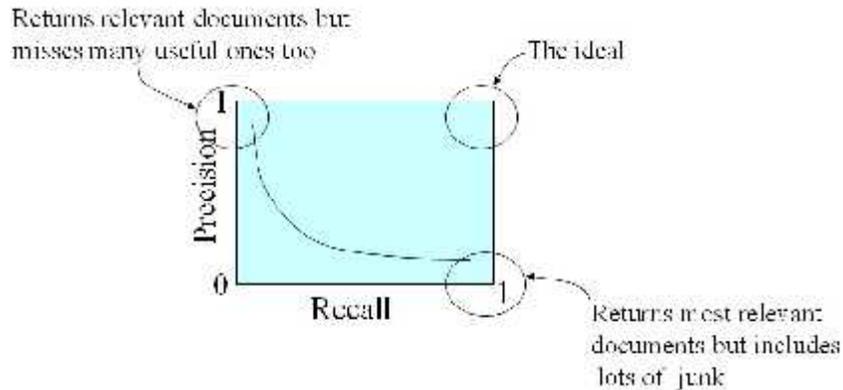
- Effectiveness is related to the **relevancy** of retrieved items.
- Relevancy is not typically binary but continuous.
- Even if relevancy is binary, it can be a difficult judgment to make.
- Relevancy, from a human standpoint, is:
 - Subjective: Depends upon a specific user's judgment.
 - Situational: Relates to user's current needs.
 - Cognitive: Depends on human perception and behavior.
 - Dynamic: Changes over time.

11. Define Precision and Recall?

$$\text{recall} \square \frac{\text{Number of relevant documents retrieved}}{\text{Total number of relevant documents}}$$

$$\text{precision} = \frac{\text{Number of relevant documents retrieved}}{\text{Total number of documents retrieved}}$$

12. Draw the Trade-off between Recall and Precision?



13. What is MapReduce?

The MapReduce algorithm contains two important tasks, namely Map and Reduce. Map takes a set of data and converts it into another set of data, where individual elements are broken down into tuples (key/value pairs). Secondly, reduce task, which takes the output from a map as an input and combines those data tuples into a smaller set of tuples.

14. Define Hadoop?

Hadoop is an open-source framework that allows to store and process big data in a distributed environment across clusters of computers using simple programming models. It is designed to scale up from single servers to thousands of machines, each offering local computation and storage.

15. What are the factors affecting the performance of CLIR systems.

Limited size of Dictionary, Query translation/transliteration performance

16. What are the Challenges in CLIR. (Cross lingual Information Retrieval)

Translation ambiguity, Phrase identification and translation, Translate/transliterate a term, Transliteration errors, Dictionary coverage, Font, Morphological analysis, Out-of-Vocabulary (OOV) problem

17. Define Snippets.

Snippets are short fragments of text extracted from the document content or its metadata. They may be static or query based. In static snippet it always the first 50 words of the document per the content

of its description. In query based snippet is one selectively extracted on the basis of its relation to the searcher's query.

18. List the advantages of invisible web content.

- ❑ Specialized content focus large amounts of information focused on an exact subject.
- ❑ Contains information that might not be available on the visible web.
- ❑ Allows a user to find a precise answer to a specific question.
- ❑ Allows a user to find WebPages from a specific date or time.

20. What is collaborative filtering?(nov/dec 2016)

It is a method of making automatic predictions about the interests of a single user by collecting preferences or taste information from many users.

21. What do you mean by item based collaborative filtering?

Item based CF is a model based approach which produces recommendations based on the relationship between items inferred from the rating matrix. The assumption behind this approach is that users will prefer items that are similar to other items they like.

22. What are problem of user based CF?

The two main problems of user based CF are that the whole user databases has to be kept in memory and that expensive similarity computation between the active user and all other users in the database has to be performed.

23. Define user based collaborative filtering.

User based collaborative filtering algorithms work off the premise that if a user A has a similar profile to another user B, then A is more likely to prefer things that it prefers when compared with a user chosen at random.

PART – B QUESTIONS AND ANSWERS

1. Explain about link analysis:
 - Meta-search Engines.
 - HTML structures & Feature Weighting.

Two methods of Link analysis:

- ❑ Page Rank
- ❑ HITS-hyperlink Induced Topic Search

Three levels of link analysis:

- ❑ Microscopic level
- ❑ Mesoscopic level

- ☐ Macroscopic level

Limitations of Link analysis:

- ☐ Meta tags/invisible text
- ☐ Pay-for-place
- ☐ Stability
- ☐ Topic Drift
- ☐ Convent evolution

2. Explain about HITS algorithms(nov/dec 2016)

- ☐ Hypertext induced Topic selection is a link analysis method developed by John Kleinberg in 1999 using Hub and Authority scores.

Two sets of inter-related pages:

- ☐ Hub Pages-good lists of links on a subject
- ☐ Authority pages-occur recurrently on good hubs for the subjects.

The HITS algorithm

$H(x) \leftarrow a(y)$

$A(x) \leftarrow h(y)$

3. Explain CLIR.(nov/dec 2016)

- ☐ CLIR-Cross Lingual Retrieval
- ☐ Dictionary-based Query Translation
- ☐ Document Translation approach
- ☐ Interlingua based Approach
- ☐ Pseudo-Relevance Feedback (PRF)for CLIR
- ☐ Challenges in CLIR

UNIT V
DOCUMENT TEXT MINING
PART – A
QUESTIONS AND ANSWERS

1. Define Information filtering.(nov/dec 2016)

Information filtering delivers to users only the information that is relevant to them, filtering out all irrelevant new data items

2. Differentiate information filtering and information retrieval

Information retrieval is about fulfilling immediate queries from a library of information available. Example : you have a deal store containing 100 deals and a query comes from a user. You show the deals that are relevant to that query.

Information Filtering is about processing a stream of information to match your static set of likes, tastes and preferences. Example: a clipper service which reads all the news articles published today and serves you content that is relevant to you based on your likes and interests.

3. State some applications of Information retrieval

- ❑ Automatic delivery of news/alerts
- ❑ Online display advertising
- ❑ Publish/subscribe systems

4. What is Relevance Feedback?

Feedback given by the user about the relevance of the documents in the initial set of results.

5. Define text mining

The discovery by computer of new, previously unknown information, by automatically extracting information from a usually large amount of different unstructured textual resources.

6. Differentiate Text Mining vs. Data Mining ,web mining, information retrieval

In Text Mining, patterns are extracted from natural language text rather than databases

Text Mining vs • Web Mining – In Text Mining, the input is free unstructured text, whilst web sources are structured.

Text Mining vs • Information Retrieval (Information Access) – No genuinely new information is found. – The desired information merely coexists with other valid pieces of information.

7. Name any two Document Clustering methods.

- ❑ K-Means clustering.
- ❑ Agglomerative hierarchical clustering.

8. What is Text Preprocessing?

Text pre-processing is an essential part of any NLP system, since the characters, words, and sentences identified at this stage are the fundamental units passed to all further processing stages, from analysis and tagging components, such as morphological analyzers and part-of-speech taggers, through applications, such as information retrieval and machine translation systems.

9. Define classification

Classification is a data mining function that assigns items in a collection to target categories or classes. The goal of classification is to accurately predict the target class for each case in the data.

10. Define clustering

Clustering is a process of partitioning a set of data (or objects) into a set of meaningful sub-classes, called clusters. Help users understand the natural grouping or structure in a data set. Used either as a stand-alone tool to get insight into data distribution or as a preprocessing step for other algorithms.

11. Define naivesbayesclassifiers(nov/dec 2016)

In machine learning, naive Bayes classifiers are a family of simple probabilistic classifiers based on applying Bayes' theorem with strong (naive) independence assumptions between the features.

12. What is decision tree?

A decision tree is a structure that includes a root node, branches, and leaf nodes. Each internal node denotes a test on an attribute, each branch denotes the outcome of a test, and each leaf node holds a class label. The topmost node in the tree is the root node.

13. Define Agglomerative hierarchical clustering

Agglomerative hierarchical clustering is a bottom-up clustering method where clusters have sub-clusters, which in turn have sub-clusters, etc. The classic example of this is species taxonomy. Gene expression data might also exhibit this hierarchical quality (e.g. neurotransmitter gene families). Agglomerative hierarchical clustering starts with every single object (gene or sample) in a single cluster. Then, in each successive iteration, it agglomerates (merges) the closest pair of clusters by satisfying some similarity criteria, until all of the data is in one cluster.

14. Define expectation–maximization (EM).(nov/dec 2016)

Expectation–maximization (EM) algorithm is an iterative method for finding maximum likelihood or maximum a posteriori (MAP) estimates of parameters in statistical models, where the model depends on unobserved latent variables.

15. What is supervised learning?

In supervised learning both input and output are provided. The network then processes the inputs and compares its resulting output against the desired outputs. Errors are then propagated back through the systems causing the system to adjust the weights which control the network.

16. What is unsupervised learning?

Unsupervised learning is a type of machine learning algorithm used to draw inferences from datasets consisting of input data without labeled responses. The most common unsupervised learning method is cluster analysis.

17. What is dendrogram?

A dendrogram is a tree diagram frequently used to illustrate the arrangement of the clusters produced by hierarchical clustering. Dendrograms are often used in computational biology to illustrate the clustering of genes or samples, sometimes on top of heatmaps.

PART – B

QUESTIONS AND ANSWERS

1. Explain naive Bayes classifiers with an example.(nov/dec 2016)

Ans.

In general all of Machine Learning Algorithms need to be trained for supervised learning tasks like classification, prediction etc. or for unsupervised learning tasks like clustering.

By training it means to train them on particular inputs so that later on we may test them for unknown inputs (which they have never seen before) for which they may classify or predict etc (in case of supervised learning) based on their learning. This is what most of the Machine Learning techniques like Neural Networks, SVM, Bayesian etc. are based upon.

So in a general Machine Learning project basically you have to divide your input set to a Development Set (Training Set + Dev-Test Set) & a Test Set (or Evaluation set). Remember your basic objective would be that your system learns and classifies new inputs which they have never seen before in either Dev set or test set.

The test set typically has the same format as the training set. However, it is very important that the

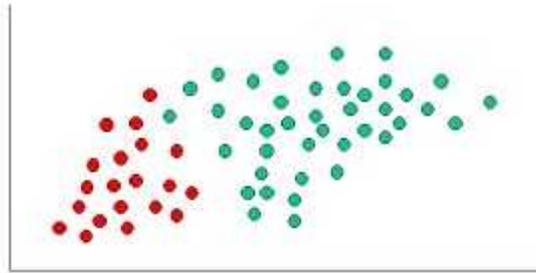
test set be distinct from the training corpus: if we simply reused the training set as the test set, then a model that simply memorized its input, without learning how to generalize to new examples, would receive misleadingly high scores.

In general, for an example, 70% can be training set cases. Also remember to partition the original set into the training and test sets *randomly*.

To demonstrate the concept of Naïve Bayes Classification, consider the example given below:

Naive Bayes Classifier Introductory Overview

The Naive Bayes Classifier technique is based on the so-called Bayesian theorem and is particularly suited when the dimensionality of the inputs is high. Despite its simplicity, Naive Bayes can often outperform more sophisticated classification methods.



To demonstrate the concept of Naïve Bayes Classification, consider the example displayed in the illustration above. As indicated, the objects can be classified as either GREEN or RED. Our task is to classify new cases as they arrive, i.e., decide to which class label they belong, based on the currently existing objects.

Since there are twice as many GREEN objects as RED, it is reasonable to believe that a new case (which hasn't been observed yet) is twice as likely to have membership GREEN rather than RED. In the Bayesian analysis, this belief is known as the prior probability. Prior probabilities are based on previous experience, in this case the percentage of GREEN and RED objects, and often used to predict outcomes before they actually happen.

Thus, we can write:

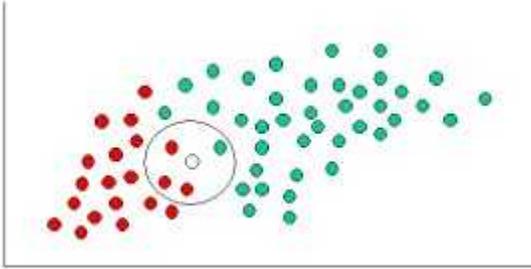
$$\text{Prior probability for GREEN} \propto \frac{\text{Number of GREEN objects}}{\text{Total number of objects}}$$

$$\text{Prior probability for RED} \propto \frac{\text{Number of RED objects}}{\text{Total number of objects}}$$

Since there is a total of 60 objects, 40 of which are GREEN and 20 RED, our prior probabilities for class membership are:

$$\text{Prior probability for GREEN} \propto \frac{40}{60}$$

$$\text{Prior probability for RED} \propto \frac{20}{60}$$



Having formulated our prior probability, we are now ready to classify a new object (WHITE circle). Since the objects are well clustered, it is reasonable to assume that the more GREEN (or RED) objects in the vicinity of X, the more likely that the new cases belong to that particular color. To measure this likelihood, we draw a circle around X which encompasses a number (to be chosen a priori) of points irrespective of their class labels. Then we calculate the number of points in the circle belonging to each class label. From this we calculate the likelihood:

$$\text{Likelihood of } X \text{ given GREEN} \propto \frac{\text{Number of GREEN in the vicinity of } X}{\text{Total number of GREEN cases}}$$

$$\text{Likelihood of } X \text{ given RED} \propto \frac{\text{Number of RED in the vicinity of } X}{\text{Total number of RED cases}}$$

From the illustration above, it is clear that Likelihood of X given GREEN is smaller than Likelihood of X given RED, since the circle encompasses 1 GREEN object and 3 RED ones. Thus:

$$\text{Probability of } X \text{ given GREEN} \propto \frac{1}{40}$$

$$\text{Probability of } X \text{ given RED} \propto \frac{3}{20}$$

Although the prior probabilities indicate that X may belong to GREEN (given that there are twice as many GREEN compared to RED) the likelihood indicates otherwise; that the class membership of X is RED (given that there are more RED objects in the vicinity of X than GREEN). In the Bayesian analysis, the final classification is produced by combining both sources of information, i.e., the prior and the likelihood, to form a posterior probability using the so-called Bayes' rule (named after Rev. Thomas Bayes 1702-1761).

Posterior probability of X being GREEN \propto

Prior probability of GREEN \times *Likelihood of X given GREEN*

$$= \frac{4}{6} \times \frac{1}{40} = \frac{1}{60}$$

Posterior probability of X being RED \propto

Prior probability of RED \times *Likelihood of X given RED*

$$= \frac{2}{6} \times \frac{3}{20} = \frac{1}{20}$$

Finally, we classify X as RED since its class membership achieves the largest posterior probability.

Note. The above probabilities are not normalized. However, this does not affect the classification outcome since their normalizing constants are the same.

As indicated, the objects can be classified as either GREEN or RED. Our task is to classify new cases as they arrive, i.e., decide to which class label they belong, based on the currently existing objects.

Since there are twice as many GREEN objects as RED, it is reasonable to believe that a new case (which hasn't been observed yet) is twice as likely to have membership GREEN rather than RED. In the Bayesian analysis, this belief is known as the prior probability. Prior probabilities are based on previous experience, in this case the percentage of GREEN and RED objects, and often used to predict outcomes before they actually happen.

Thus, we can write:

Prior Probability of GREEN: number of GREEN objects / total number of objects

Prior Probability of RED: number of RED objects / total number of objects

Since there is a total of 60 objects, 40 of which are GREEN and 20 RED, our prior probabilities for class membership are:

Prior Probability for GREEN: 40 / 60

Prior Probability for RED: 20 / 60

Having formulated our prior probability, we are now ready to classify a new object (WHITE circle in the diagram below). Since the objects are well clustered, it is reasonable to assume that the more GREEN (or RED) objects in the vicinity of X, the more likely that the new cases belong to that particular color. To measure this likelihood, we draw a circle around X which encompasses a number (to be chosen a priori) of points irrespective of their class labels. Then we calculate the number of points in the circle belonging to each class label

2. Explain decision tree algorithm with example.

Ans. Very simply, ID3 builds a **decision tree** from a fixed set of **examples**. ... The leaf nodes of the **decision tree** contain the class name whereas a non-leaf node is a **decision** node. The **decision** node is an attribute test with each branch (to another **decision tree**) being a possible value of the attribute.

DecisionTreeAlgorithmID3

Algorithm: Generate_decision_tree. Generate a decision tree from the training tuples of data partition D .

Input:

- Data partition, D , which is a set of training tuples and their associated class labels;
- *attribute_list*, the set of candidate attributes;
- *Attribute_selection_method*, a procedure to determine the splitting criterion that “best” partitions the data tuples into individual classes. This criterion consists of a *splitting_attribute* and, possibly, either a *split_point* or *splitting_subset*.

Output: A decision tree

Method:

- (1) create a node N ;
- (2) if tuples in D are all of the same class, C then
- (3) return N as a leaf node labeled with the class C ;
- (4) if *attribute_list* is empty then
- (5) return N as a leaf node labeled with the majority class in D , // majority voting
- (6) apply *Attribute_selection_method*(D , *attribute_list*) to find the “best” *splitting_criterion*;
- (7) label node N with *splitting_criterion*;
- (8) if *splitting_attribute* is discrete-valued and
 multiway splits allowed then // not restricted to binary trees
- (9) *attribute_list* \leftarrow *attribute_list* - *splitting_attribute*; // remove *splitting_attribute*
- (10) for each outcome j of *splitting_criterion*
 // partition the tuples and grow subtrees for each partition
- (11) let D_j be the set of data tuples in D satisfying the outcome j ; // a partition
- (12) if D_j is empty then
- (13) attach a leaf labeled with the majority class in D to node N ;
- (14) else attach the node returned by *Generate_decision_tree*(D_j , *attribute_list*) to node N ;
- endifor
- (15) return N ;

3. Explain Agglomerative clustering with example.

Ans. In the agglomerative hierarchical approach, we start by defining each data point to be a cluster and combine existing clusters at each step. Here are four different methods for doing this:

1. Single Linkage: In *single linkage*, we define the distance between two clusters to be the minimum distance between any single data point in the first cluster and any single data point in the second cluster. On the basis of this definition of distance between clusters, at each stage of the process we combine the two clusters that have the smallest single linkage distance.

2. Complete Linkage: In *complete linkage*, we define the distance between two clusters to be the maximum distance between any single data point in the first cluster and any single data point in the second cluster. On the basis of this definition of distance between clusters, at each stage of the process we combine the two clusters that have the smallest complete linkage distance.

3. Average Linkage: In *average linkage*, we define the distance between two clusters to be the average distance between data points in the first cluster and data points in the second cluster. On the basis of this definition of distance between clusters, at each stage of the process we combine the two clusters that have the smallest average linkage distance.

4. Centroid Method: In *centroid method*, the distance between two clusters is the distance between the two mean vectors of the clusters. At each stage of the process we combine the two clusters that have the smallest centroid distance.

5. Ward's Method: This method does not directly define a measure of distance between two points or clusters. It is an ANOVA based approach. At each stage, those two clusters merge, which provides the smallest increase in the combined error sum of squares from one-way univariate ANOVAs that can be done for each variable with groups defined by the clusters at that stage of the process

4. Explain K-means algorithm with example.

Ans. Clustering is the process of partitioning a group of data points into a small number of clusters. For instance, the items in a supermarket are clustered in categories (butter, cheese and milk are grouped in dairy products). Of course this is a qualitative kind of partitioning. A quantitative approach would be to measure certain features of the products, say percentage of milk and others, and products with high percentage of milk would be grouped together. In general, we have n data points $\mathbf{x}_i, i=1 \dots n$ that have to be partitioned in k clusters. The goal is to assign a cluster to each data point. K-means is a clustering method that aims to find the positions $\mu_i, i=1 \dots k$ of the clusters that minimize the *distance* from the data points to the cluster. K-means clustering solves

$$\operatorname{argmin}_{i=1 \dots k} \sum_{\mathbf{x} \in c_i} d(\mathbf{x}, \mu_i) = \operatorname{argmin}_{i=1 \dots k} \sum_{\mathbf{x} \in c_i} \|\mathbf{x} - \mu_i\|^2$$

where c_i is the set of points that belong to cluster i . The K-means clustering uses the square of the Euclidean distance $d(\mathbf{x}, \mu_i) = \|\mathbf{x} - \mu_i\|^2$. This problem is not trivial (in fact it is NP-hard), so the K-means algorithm only hopes to find the global minimum, possibly getting stuck in a different solution.

5. What is the expectation maximization algorithm? Give its applications. (nov/dec 2016)

Ans. The **expectation maximization algorithm** is a natural generalization of maximum likelihood estimation to the incomplete data case. In particular, **expectation maximization** attempts to find the parameters that maximize the log probability $\log P(\mathbf{x}; \theta)$ of the observed data.

In statistics, an **expectation–maximization (EM) algorithm** is an iterative method to find maximum likelihood or maximum a posteriori (MAP) estimates of parameters in statistical models, where the model depends on unobserved latent variables. The EM iteration alternates between performing an expectation (E) step, which creates a function for the expectation of the log-likelihood evaluated using the current estimate for the parameters, and a maximization (M) step, which computes parameters maximizing the expected log-likelihood found on the E step. These parameter-estimates are then used to determine the distribution of the latent variables in the next E step.

Given the statistical model which generates a set of observed data, a set of unobserved latent data or missing values, and a vector of unknown parameters, along with a likelihood function, the maximum likelihood estimate (MLE) of the unknown parameters is determined by the marginal likelihood of the observed data. However, this quantity is often intractable (e.g. if is a sequence of events, so that the number of values grows exponentially with the sequence length, making the exact calculation of the sum extremely difficult).

The EM algorithm seeks to find the MLE of the marginal likelihood by iteratively applying these two steps:

Expectation step (E step): Calculate the expected value of the log likelihood function, with respect to the conditional distribution of given under the current estimate of the parameters:

Maximization step (M step): Find the parameter that maximizes this quantity:

The typical models to which EM is applied uses as a latent variable indicating membership in one of a set of groups:

1. The observed data points may be discrete (taking values in a finite or countably infinite set) or continuous (taking values in an uncountably infinite set). Associated with each data point may be a vector of observations.
2. The missing values (aka latent variables) are discrete, drawn from a fixed number of values, and with one latent variable per observed unit.
3. The parameters are continuous, and are of two kinds: Parameters that are associated with all data points, and those associated with a specific value of a latent variable (i.e., associated with all data points which corresponding latent variable has that value).

However, it is possible to apply EM to other sorts of models. The motive is as follows. If the value of the parameter is known, usually the value of the latent variables can be found by maximizing the log-likelihood over all possible values of z , either simply by iterating over z or through an algorithm such as the Viterbi algorithm for hidden Markov models. Conversely, if we know the value of the latent variables z , we can find an estimate of the parameters θ fairly easily, typically by simply grouping the observed data points according to the value of the associated latent variable and averaging the values, or some function of the values, of the points in each group. This suggests an iterative algorithm, in the case where both are unknown:

1. First, initialize the parameters to some random values.
2. Compute the probability of each possible value of z given θ .
3. Then, use the just-computed values of z to compute a better estimate for the parameters θ .
4. Iterate steps 2 and 3 until convergence.

The algorithm as just described monotonically approaches a local minimum of the cost function.

