# SNS COLLEGE OF TECHNOLOGY

**COIMBATORE - 35**

**DEPARTMENT OF INFORMATION TECHNOLOGY (UG)**

**2 MARKS Q & A**

## UNIT I: LIST

## Part-A

**1. Write down the definition of data structures?**

A data structure is a mathematical or logical way of organizing data in the memory that consider not only the items stored but also the relationship to each other and also it is characterized by accessing functions.

**2. What are the two basic types of Data structures?**

1. Primitive Data structure: Eg., int,char,float

2. Non Primitive Data Structure: i. Linear Data structure (The data items are arranged in a order): Eg., Lists Stacks Queues ii. Non linear Data structure (The data items are arranged in not an order): Eg., Trees Graphs

**3. Define Algorithm?**

Algorithm is a solution to a problem independent of programming language. It consist of set of finite steps which, when carried out for a given set of inputs, produce the Corresponding output and terminates in a finite time.

**4. Define Program?**

Set of instructions to find the solution to a problem. It is expressed in a programming language in an explicit and unambiguous manner.

**5. What are the features of an efficient algorithm?**

- Free of ambiguity
- Efficient in execution time
- Concise and compact
- Completeness
- Definiteness
- Finiteness

**6. What are the four basic Operations of Data structures?**

- Traversing

- Searching
- Inserting
- Deleting

**7. List down any four applications of data structures?**

- Compiler design,
- Operating System,
- Database Management system,
- Network analysis

**8. What is meant by an abstract data type (ADT)?**

An ADT is a set of operation. A useful tool for specifying the logical properties of data type is the abstract data type. ADT refers to the basic mathematical concept that defines the data type. Eg. Objects such as list, set and graph along their operations can be viewed as ADT's.

**9. What are the operations of ADT?**

Union, Intersection, size, complement and find are the various operations of ADT.

**10. What is meant by list ADT?**

List ADT is a sequential storage structure. General list of the form A1, A2, A3…..,An and the size of the list is 'n'. Any element in the list at the position I is defined to be ai,ai+1 the successor of ai and ai-1 is the predecessor of ai.

**12. What are the various operations done under list ADT?**

- Print list
- Insert
- Make empty
- Remove
- Next
- Previous
- Find kth

**13. What are the two parts of ADT?**

- Value definition
- Operator definition

**14. What is a pointer?**

Pointer is a variable, which stores the address of the next element in the list. Pointer is basically a number.

**15. What is an array?**

Array may be defined abstractly as a finite ordered set of homogenous elements. Finite means there is a specific number of elements in the array.

**16. What are the different ways to implement list?**

- Simple array implementation of list
- Linked list implementation of list

**17. Write down the uses of array?**

1) The array is a powerful tool that is widely used in computing.

2) Arrays provide us with a very simple efficient way of referring to and perform computations on collections of data that share some common attribute.

3) Arrays can be used to build and simulate finite state automata.

**18. What are differences between array and list?**

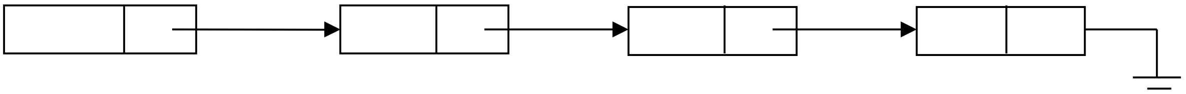
| Array | List |
|---|---|
| An array is an ordered set which consists of a fixed number of objects. | A list is an ordered set, consisting of a variable number of elements. |
| No deletion or insertion operations are performed on array. | It can performs both insertion and deletion operations. |

**19. What is a linked list?**

Linked list is a kind of series of data structures, which are not necessarily adjacent in memory. Each structure contains the element and a pointer to a record containing its successor.

**20. Define singly linked list with neat diagram.**

A singly linked list is a collection of nodes each node is a structure it consisting of an element and a pointer to a structure containing its successor, which is called a next pointer. The last cell's next pointer points to NULL specified by



**21. List the advantages and disadvantages of singly linked list.**

Advantages:

1. Easy insertion and deletion.

2. Less time consumption.

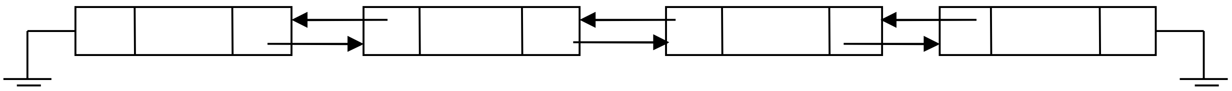
Disadvantages:

1. Data not present in a linear manner.
2. Insertion and deletion from the front of the list is difficult without the use of header node.

**22. Define doubly linked list with neat diagram.**

Doubly linked list is a collection of nodes where each node is a structure containing the following fields

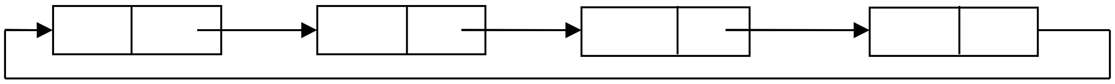
1. Pointer to the previous node.
2. Data.
3. Pointer to the next node.



**23. Define circularly linked list with neat diagram.**

Circularly linked list is a collection of nodes, where each node is a structure containing the element and a pointer to a structure containing its successor.

The pointer field of the last node points to the address of the first node. Thus the linked list becomes circular.



**24. Write the difference between Singly and doubly linked list.**

| Singly Linked List | Doubly Linked List |
|---|---|
| It is a collection of nodes and each node is having one data field and next link field | It is a collection of nodes and each node is having one data field one previous link field and one next link field |
| The elements can be accessed using next link | The elements can be accessed using both previous link as well as next link |
| No extra field is required hence; Node takes less memory in SLL. | One field is required to store previous Link Hence, node takes memory in DLL. |
| Less efficient access to elements | More efficient access to elements. |

**25. Write the difference between doubly and circularly linked list.**

| Doubly Linked List | Circularly Linked List |
|---|---|

| | |
|---|---|
| If the pointer to next node is Null, it specifies the last node. | There is no first and last node. |
| Last node's next field is always Null | Last nodes next field points to the address of the first node. |
| Every node has three fields one is Pointer to the pervious node, next is data and the third is pointer to the next node | It can be a singly linked list and doubly linked list. |

**26. What is the need for the header?**

Header of the linked list is the first element in the list and it stores the number of elements in the list. It points to the first data element of the list.

**27. List three Applications of linked list?**

- Polynomial ADT
- Radix sort
- Multi lists

**28. Write the difference between cursor and pointer implementation of singly linked list.**

| **CURSOR IMPLEMENTATION** | **POINTER IMPLEMENTATION** |
|---|---|
| Global array is maintained for storing the data elements .The corresponding index value for each data item represents its address. | The data are stored in a collection of structures. Each structure contains data and a pointer to the next structure |
| Collection of empty cells is maintained in the form of a array from which cells can be allocated and also cells can be Returned after use. | A new structure can be obtained from the systems global memory by a call to a *malloc()* and released by a call to *free()*. |

## UNIT II – STACK AND QUEUE

**1. Define stack ADT with example**.

A stack is a list with a restriction that insertions and deletions can be performed in only one position namely the end of the list called the top. it follows the principles of LIFO(Last In First Out). e.g.: undo statement in text editor. Pile of bills in a hotel & plates in a tray

**2. What is the purpose of top and pop?**

Top operation examines the element in the top of the list and returns its value. Pop operation deletes the element at the top of the stack and decrements the top of the stack pointer by one.

**3. State the disadvantages of linked list implementation of stack.**

1. Calls to malloc and free functions are expensive.
2. Using pointers is expensive.

**4. State the applications of stack.**

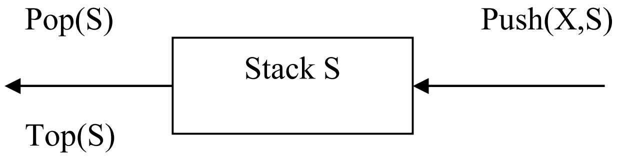
1. Balancing parentheses.
2. Postfix Expression.
    i. Infix to postfix conversion
3. Function calls.

**5. State the operations on stack. Define them and give the diagrammatic representation**.

**Push:** Push is performed by inserting at the top of stack.

**Pop:** pop deletes the most recently inserted element.

**Top:** top operation examines the element at the top of the stack and returns its value.



**6. Write the algorithm for balancing symbols.**

```
1. Make an empty stack.
2. Read characters until end of file.
3. If the character is an opening symbol, then push it onto the stack.
4. If it is a closing symbol
        Then
                If the stack is empty
                Report an error
        Otherwise pop the stack
5. If the symbol popped is not the corresponding openinig symbol
```

```
        Then
                Report an error
    6. If the stack is not empty at the end of file
        Then
                Report an error
```

**7. Give the Features of balancing symbols.**

1. It is clearly linear.
2. Makes only one pass through the input.
3. It is online and quite fast.
4. It must be decided what to do when an error is reported.

**8. Convert the given infix to postfix.**

(j*k)+(x+y)

Ans:jk* xy++

9. **Convert into postfix and evaluate the following expression.** (a+b*c)/d, a=2 b=4 c=6 d=2

Ans: Post fix: abc*+d/ Evaluation: 2 4 6 * +2/ =13

**10. Write the features of representing calls in a stack.**

1. When a function is called the register values and return address are saved.

2. After a function has been executed the register values are resumed on returning to the calling statement.

3. The stack is used for resuming the register values and for returning to the calling statement.

4. The stack overflow leads to fatal error causing loss of program and data.

5. Recursive call at the last line of the program is called tail recursion and also leads to error.

**11. Define queue with examples.**

A Queue is an ordered collection of items from which items may be deleted at one end called the front of the queue and into which tems may be inserted at the other end called rear of the queue. Queue is called as First –in-First-Out(FIFO).

e.g: Ticket counter, Phone calls waiting in a queue for the operator to receive.

**12. List the operations of queue.**

Two operations

1. Enqueue-inserts an element at the end of the list called the rear.
2. Dequeue-delets and returns the element at the start of the list called as the front.

**13. List the Applications of queue?**

- Graph Algorithm

- Priority Algorithm
- Job scheduling
- Categorizing Data

**13. What is a circular queue?**

The queue, which wraps around upon reaching the end of the array is called as circular queue.

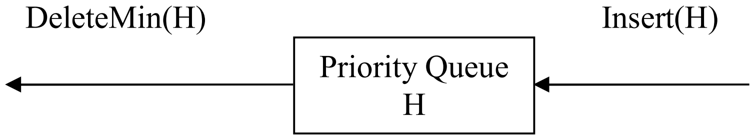
**14. Define priority queue with diagram and give the operations.**

Priority queue is a data structure that allows at least the following two operations.

1. Insert-inserts an element at the end of the list called the rear.

2. DeleteMin-Finds, returns and removes the minimum element in the priority Queue.

3.Ascending and Descending priority queue are the two types of Priority queue.



**15. Give the applications of priority queues.**

There are three applications of priority queues

1. External sorting.
2. Greedy algorithm implementation.
3. Discrete even simulation.
4. Operating systems.

**16. Write postfix from of the expression –A+B-C+D?**

A-B+C-D+

**17. How do you test for an empty queue?**

To test for an empty queue, we have to check whether READ=HEAD where REAR is a pointer pointing to the last node in a queue and HEAD is a pointer that pointer to the dummy header. In the case of array implementation of queue, the condition to be checked for an empty queue is READ<FRONT.

**18. What are the postfix and prefix forms of the expression?**

A+B*(C-D)/(P-R)

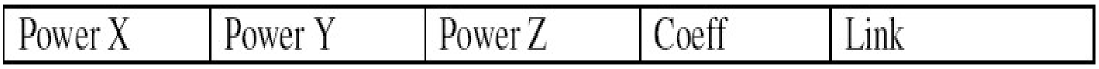
Postfix form: ABCD-*PR-/+

Prefix form: +A/*B-CD-PR

**19. Explain the usage of stack in recursive algorithm implementation?**

In recursive algorithms, stack data structures is used to store the return address when a recursive call is encountered and also to store the values of all the parameters essential to the current state of the procedure.

**20. Draw term of a polynomial in the variables x, y and z and explain?**



| Power X | Power Y | Power Z | Coeff | Link |
|---|---|---|---|---|

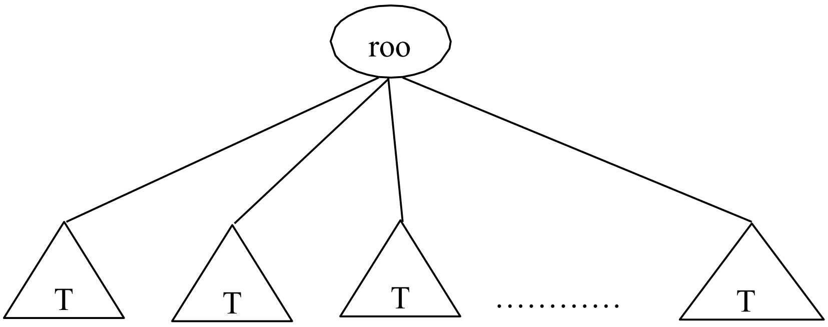
The first three fields represent the power of the variables X, Y and Z respectively. The fourth and fifth fields represent the coefficient of the term in the polynomial and address of the next term in the polynomial respectively.

## UNIT III: TREES

### Part-A

**1. Define Tree .Give an example.**

A tree is a collection of nodes .The collection can be empty .Otherwise a tree consists of a distinguished node r called the root and 0 or more non empty sub-trees $T_1$,$T_2$,$T_3$ ............................ $T_k$ each of whose roots are connected by a directed edge from r.

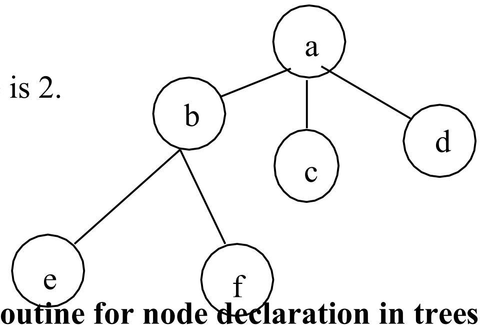


Eg: directory structure hierarchy

**2. Define depth of a node in a tree. Give example**.

For any node $n_i$ the depth of $n_i$ is the length of the unique path from the root to $n_i$ eg:

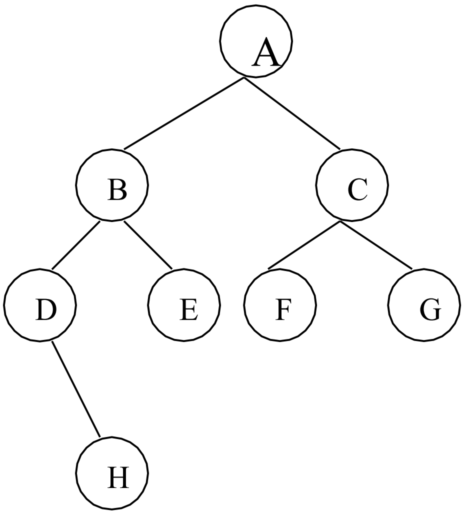
The depth of e is 2.



**3. Write the routine for node declaration in trees.**

```
typedef struct TreeNode *PtrToNode;
struct TreeNode
{
        ElementType Element;
        PtrToNode FirstChild;
        PtrToNode NextSibling;
};
```

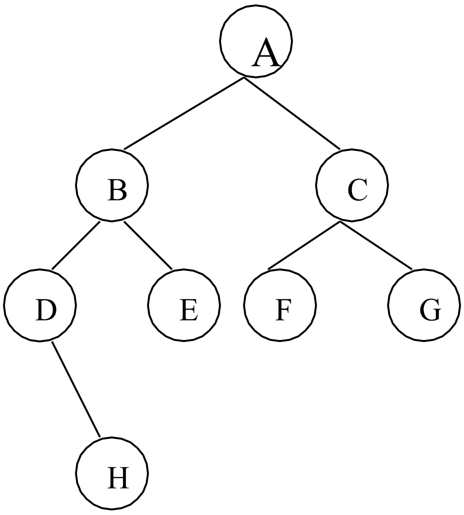
**4. Define length of the path in a tree with an example.**



Length of a path is the number of edges on the path. The length of the path from A-H is 3.

**5. Define a path in a tree. Give example.**

A path from a node $n_1$ to $n_k$ is defined as the sequence of nodes $n_1$, $n_2$…..$n_k$ such that $n_i$ is the parent of $n_{i+1}$ for $1 \le i < k$.



Eg:

The path from A-H is A-B-D-H

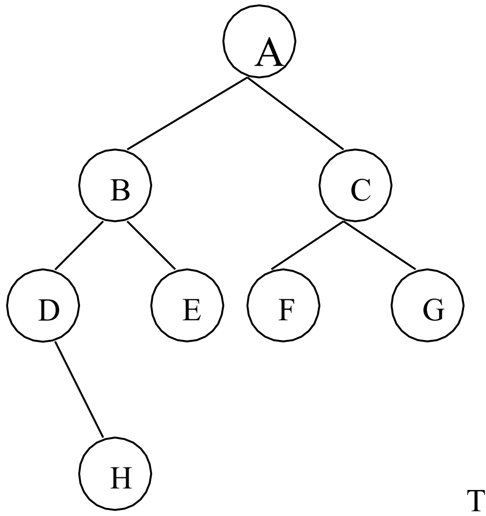
**6. List the applications of trees.**

- Binary search trees
- Expression trees
- Threaded binary trees

**7. Define height of the node in a tree. Give example.**

The height of node $n_i$ the length of the longest path from $n_i$ to a leaf

Eg:



The height of node B is 2.

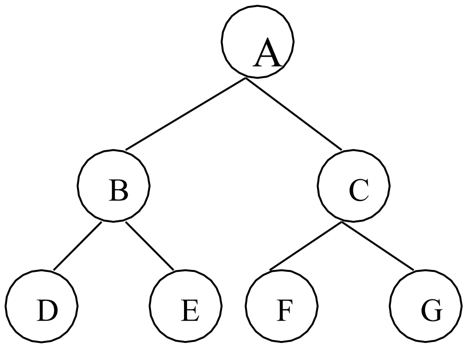
**8. List the tree traversal applications**.

1. Listing a directory in an hierarchal file system (preorder)
2. Calculating the size of a directory (post order)

**9. Define binary tree ADT with an example.**

A binary tree is a tree in which no node can have more than two children.

For Eg:,



**10. Define binary search tree?**

Binary Search tree is a binary tree in which each internal node *x* stores an element such that the element stored in the left sub tree of *x* are less than or equal to *x* and elements stored in the right sub tree of *x* are greater than or equal to *x*. This is called binary-search-tree

**11. List the Types of binary search trees**

i) Performance comparisons

ii) Optimal binary search trees

**12. List the uses of binary tree.**

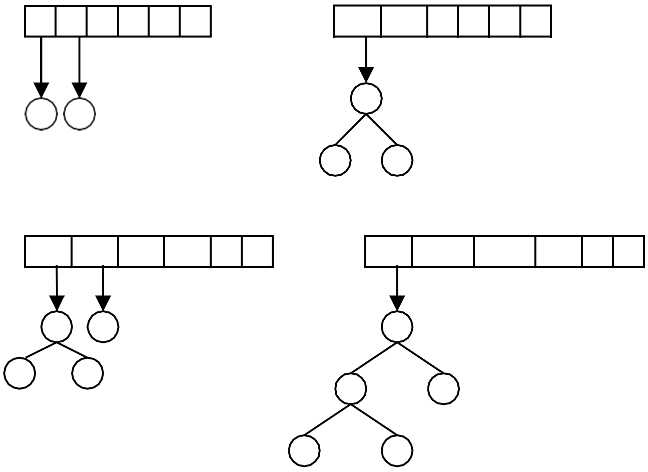
1. Searching. 2. Compiler design.

**13. List the Operations of binary search tree?**

- Make Empty

- Find
- Insert
- Delete
- Search
- Display

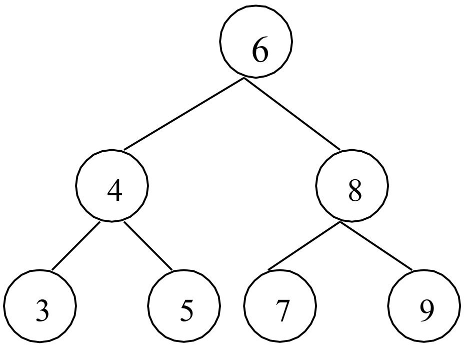
**14. Draw the expression tree for the given postfix expression using stack.**



AB*C+

**15. Define binary search tree ADT with an example.**

A binary search tree is a tree in which for every node X, the values of all the keys in its left sub tree are smaller than the key value in X and the values of all the keys in its right sub tree are larger than the key value in X.



**16. Define internal path length.**

It is the sum of the depths of all nodes in a tree.

**17. How deletion is performed in a binary search tree.**

Once the node to be deleted is found there are three possibilities

1. If the node is a leaf, it can be deleted immediately.

2. If the node has one child the node can be deleted after its parent adjusts a pointer to bypass the node.

3. If the node has two children the general strategy is to replace the data of this node with the smallest data of the right sub tree and recursively delete the node which is empty.

**18. What is the average depth of all nodes in an equally likely tree?**

The average depth of all nodes in an equally likely tree is O (log N).

**19. List the disadvantages of Binary search tree.**

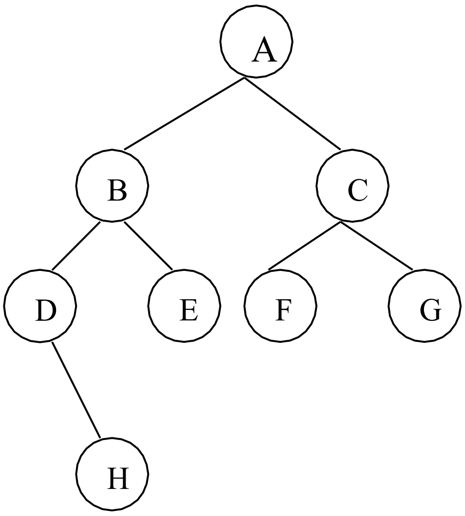
1. Deletions in a binary search tree lead to trees which are not equally likely.
2. Absence of balanced search tree.
3. The average depth is not O (log N) in trees which are not equally likely.

**20. Define tree traversal, List out the types of Tree traversal?**

Traveling through all the nodes of the tree in such a way that each node is visited exactly once. There are three types of tree traversal

1. Preorder traversal
2. In order traversal
3. Post order traversal

**21. Perform preorder traversal for the given tree.**



Preorder: **ABDHECFG**

**22. Define the following.**

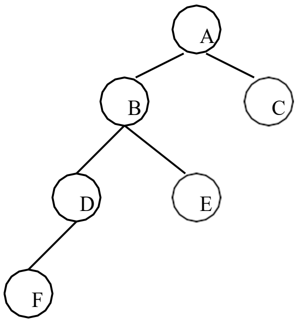
**i)** Leaf - Nodes at the bottommost level of the tree are called **leaf nodes**

ii) Sibling - The nodes with common parent are called Sibling

**23. Define expression trees?**

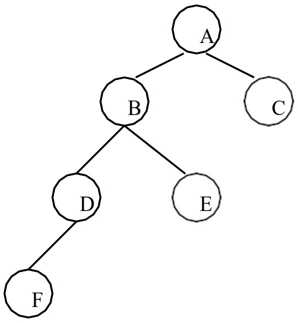
The leaves of an expression tree are operands such as constants or variable names and the other nodes contain operators.

**24. Perform in order traversal for the given tree.**



Inorder:FDBEAC

**25. Perform postorder traversal for the given tree.**



Postorder:FDEBCA

**26. Define expression trees?**

The leaves of an expression tree are operands such as constants or variable names and the other nodes contain operators.

**27. Define strictly binary tree?**

If every nonleaf node in a binary tree has nonempty left and right subtrees, the tree is termed as a strictly binary tree.

**28. Define complete binary tree?**

A complete binary tree of depth d is the strictly binary tree all of whose are at level d.

**29. What is an almost complete binary tree?**

A binary tree of depth d is an almost complete binary tree if :Each leaf in the tree is either at level d or at level d-1.For any node nd in the tree with a right descendant at level d,all the left descendants of nd that are leaves are at level d.

**30. What is mean by Full Binary Tree?**

A binary tree is said to be full, is all its leaves are at the same level and every interior node has two children.

**31. Define right – in-threaded tree?**

Right –in –threaded binary tree is defined as one in which threads replace NULL pointers in nodes with empty right sub trees.

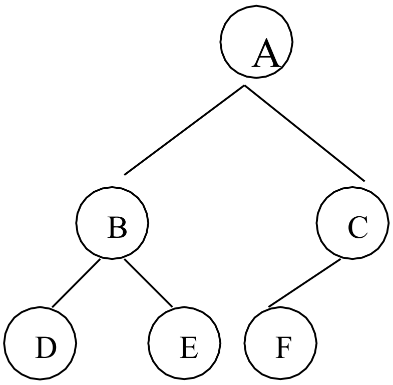
**32. Define left – in –threaded tree?**

A left-in-threaded binary tree may be defined as one in which each NULL pointers is altered to contain a thread to that node's in order predecessor.

**33. Define AVL Tree. Give Example.**

An AVL Tree is a binary search tree with a balance condition, which is easy to maintain and ensure that the depth of the tree is O(logN). Balance condition require that the left and the right sub trees have the same height.

Example:



**34. Define Balance factor.**

The balance factor of a node in binary tree is defined to be $|h_L\text{-}h_R|$<=1, where $h_L$ and $h_R$ are heights of left and right sub trees of T. For any node in AVL tree the balance factor should be 1,0 or -1.

**35. What are the various transformation performed in AVL tree?**

1. Single rotation:

-Single L rotation

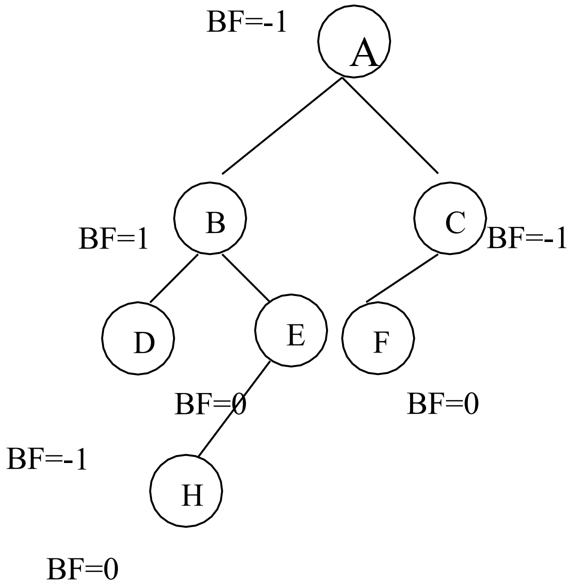
-Single R rotation

2. Double rotation

-LR rotation

-RL rotation

**36. Give the balance factor of each node for the following tree**.



**37. When AVL tree property is violated and how to solve it?**

After insertion of any node in an AVL tree if the balance factor of any node becomes other than -1, 0, or 1 then it is said that AVL property is violated. So the node on the path from the inserted node to the root needs to be readjusted. Check the balance factor for each node in the path from inserted node to the root node and adjust the affected sub tree such that the entire sub tree should satisfy the AVL property.

**38. Mention the four cases to rebalance the AVL tree.**

- An insertion of new node into Left sub tree of Left child (LL).
- An insertion of new node into Right sub tree of Left child (LR).
- An insertion of new node into Left sub tree of Right child (RL).
- An insertion of new node into Right sub tree of Right child (RR).

**39. List the Operations on Binary heap.**

- Adding to the heap
- Deleting the root from the heap

**40. What is priority queue?**

A Queue in which we are able to insert items or remove items from any position based on some priority is after referred to as a priority Queue.A priority queue is a data structure that allows at least the following two operations: insert which does the obvious thing; and Deletemin, which finds, returns, and removes the minimum element in the priority queue. The Insert operation is the equivalent of Enqueue.

**41. Define min heap?**

A heap in which the parent has a smaller key than the child's is called a min heap.

**42. Define max heap?**

A heap in which the parent has a larger key than the child's is called a max heap.

**43. Define binary heaps.**

A binary heap is a heap data structure created using a binary tree. It can be seen as a binary tree with two additional constraints:

- The *shape property*: the tree is an *almost complete binary tree*; that is, all levels of the tree, except possibly the last one (deepest) are fully filled, and, if the last level of the tree is not complete, the nodes of that level are filled from left to right.
- The *heap property*: each node is greater than or equal to each of its children according to some comparison predicate which is fixed for the entire data structure.

**44. List the Applications of Binary heap**

- Heap sort
- Selection Algorithm and Graph Algorithm

**45. Application of priority queues?**

- Schedule the process in operating system.
- It is used for external sorting.
- To implement of greedy algorithm.

**46. What are the main properties of a binary heap?**

- Structure property
- Heap order property

**47. List the Operations on B-Trees.**

- Search
- Create
- Insert

**48. List the B-Trees Applications.**

- Databases
- Concurrent Access to B-Trees

**49. Define B-Tree.**

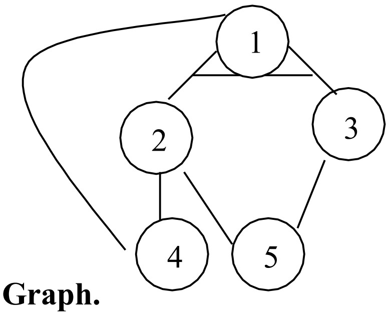
A search tree that is not a binary tree is called B-Tree. That satisfies the following structural properties

- Root is either a leaf or has between 2 and M children
- All non leaf nodes except the root have between [M/2] and M children.
- All leafs are at the same depth.

## UNIT IV: GRAPHS

**1. Define Graph with illustration.**

A Graph is defined as G = (V, E) where V is the set of vertices (nodes) and E is the set of edges (arcs) connecting the vertices. An edge is represented as a pair of vertices (u,v).



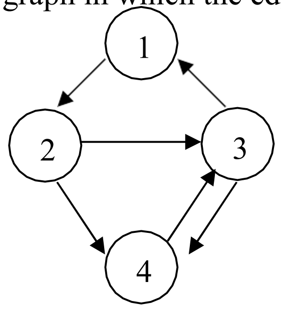
V = {1, 2, 3, 4, 5}

E = {(1, 2),(1, 3),(1,4),(2,3)

(2,4),(2,5),(3,5)}

**2. Define Directed Graph.**

A Directed graph is a graph in which the edges are directed. It is also called Digraph.



V={1,2,3,4}

E={(1,2),(2,3),(2,4)

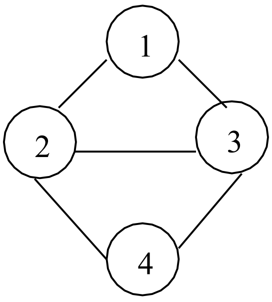
(3,1),(3,4),(4,3)}

**3. Mention the ways of representing a graph?**

- Adjacency Matrix representation
- Adjacency List representation

**4. What do you mean by Undirected Graph?**

Edges in the graph do not have directions marked. Such graphs are referred to as undirected graphs.



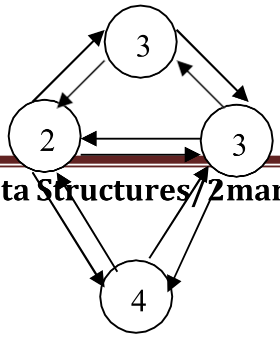
V={1,2,3,4}

E={(1,2),(2,1),(2,3),(3,2),(2,4),

(4,2),(3,4),(4,3),(3,1),(1,3)}

**5. Define Symmetric Digraph.**

Every edge has an edge in the reverse direction i.e for every edge (U,V) there is an edge (V,U). Such type of graph is called Symmetric Digraph.



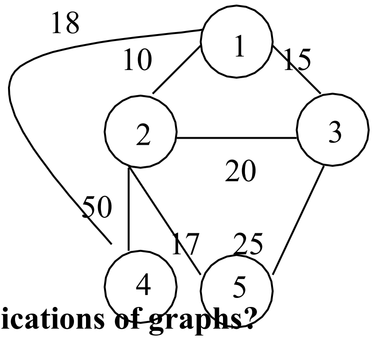
V={1,2,3,4}

E={(1,2),(2,1),(2,3),(3,2)

(2,4),(4,2),(3,4),(4,3),(3,1),(1,3)}

**6. What do you mean by weighted graph?**

Weighted graphs are such graphs where the edges are associated with weights. These weights are used to mark the importance of edges in representing a problem. Ex. Road map represented as graph where the weight is the distance between two places.



V={1,2,3,4,5}

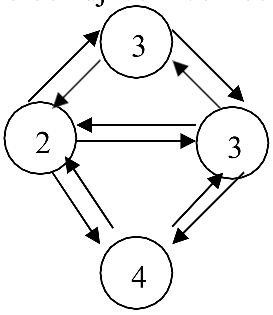
E={(1,2),(1,3),(1,4),(2,3)
(2,4),(2,5),(3,5)}

**7. What are the applications of graphs?**

- Transport systems
- Computer Networks
- Electric circuits

**8. Define adjacent vertices.**

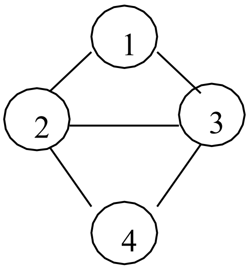
Two vertices are said to be adjacent vertices if there is an edge between them.



Vertices 1&2, 2&3, 2&4, 3&4 are adjacent vertices.

**9. Define path.**

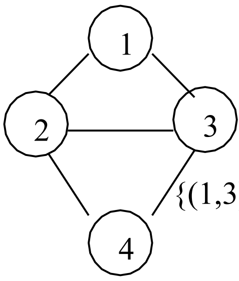
A Path between vertices (u,v) is a sequence of edges which connects vertices u & v.



The path between the vertices 1 and 4 is 1-2-4 or 1-3-4

**10. Define Length of the Path.**

Length of the path is the number of edges in a path which is equal to N-1. N represents the number of vertices.



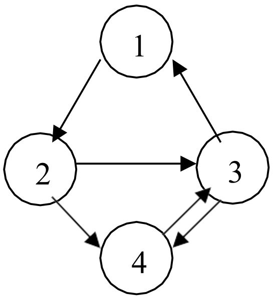
The Length of the path 1 -4 {(1, 2), (2, 3), (3,4) } is 3 {(1,3),(3,4)} is 2 and also {(1,2),(2,4)} is 2.

**11. What is a forest?**

A forest may be defined as an acyclic graph in which every node has one or no of predecessors. A tree may be defined as a forest in which only a single node called root has no predecessors.

**12. Define cycle.**

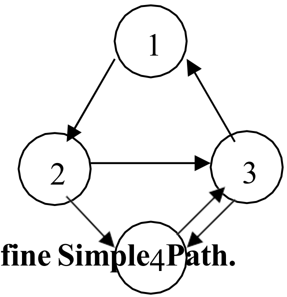
A Cycle in a graph is a path that starts and ends at the same vertex. i.e. path in which the first and last vertices are same.



Ex: {(3, 1), (1, 2), (2, 4), (4, 3)} is a Cycle.

**13. Define simple cycle.**

Simple cycle means the vertex should not be repeated and the first and last vertex should be the same.

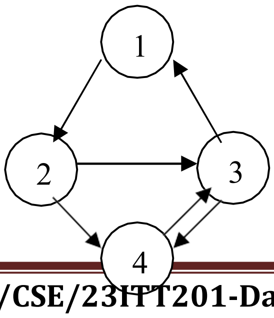


Example: 1-2-4-3-1 is a simple cycle

1-2-3-4-3-1 is not a simple Cycle

**14. Define Simple Path.**

All vertices are distinct, except that the first and last may or may not be the same.



Example: 1-2-4-3-1 is a simple path

1-2-3-4-3-1 is not a simple path

1-2-4-3 is a simple path

**15. Explain unweighted path length**

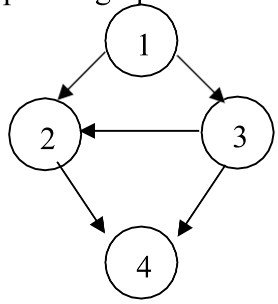
Unweighted path length is the number of edges on the path namely, N-1(where N is the number of vertices).

**16. What is a tree edge?**

Traversal from one vertex to the next vertex in a graph is called as a tree edge.

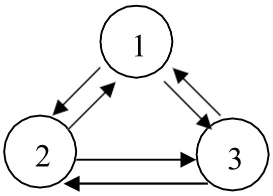
**17. Define acyclic graph.**

An acyclic graph is a graph with no cycle.



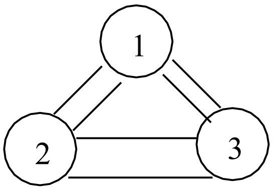
**18. What do you mean by a strongly connected graph?**

A Directed graph is strongly connected if there is a path from every vertex to every other vertex.



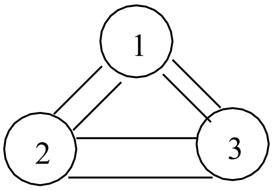
**19. What do you mean by weakly connected graph?**

If the directed graph is not strongly connected, but the underlying graph (without direction to the arcs) is connected, then the graph is said to be weakly connected.



**20. Define complete graph.**

A graph in which there is an edge between every pair of vertices.



**21. Explain weighted path length.**

N-1

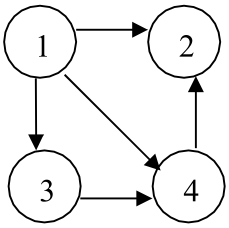
The cost of the path $V_1, V_2 \ldots V_N$ is $\Sigma$ $C_{i, i+1}$.. This is referred to as the weighted path length.

i=1

**22. What do you mean by Adjacency Matrix representation?**

The adjacency matrix M of a graph G = (V, E) is a matrix of order $V_i$ x$V_j$ and the elements of the unweighted graph M are defined as, M [i] [j] = 1, if $(V_i, V_j) \in E$ = 0 Otherwise, For a weighted graph the elements of M are defined as M [i] [j] = $W_{ij}$, if $(V_i, V_j) \in E$ and $W_{ij}$ is the weight of edge $(V_i, V_j)$ = 0 Otherwise

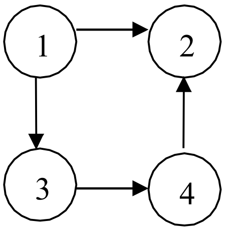
**Example:**

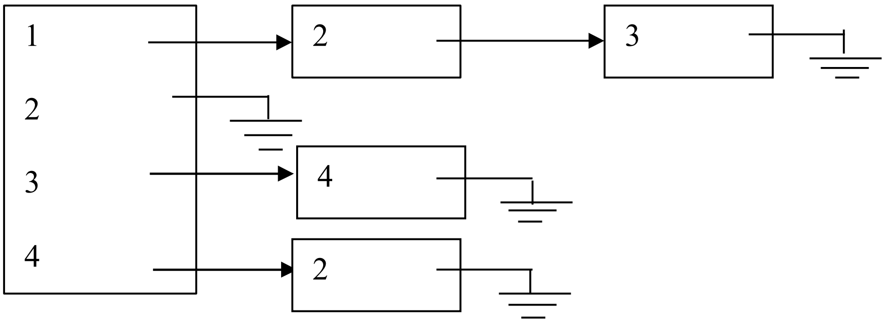


$$\begin{array}{c|cccc|} & 1 & 2 & 3 & 4 \\ 1 & 0 & 1 & 1 & 1 \\ 2 & 0 & 0 & 0 & 1 \\ 3 & 0 & 0 & 0 & 1 \\ 4 & 0 & 1 & 0 & 0 \end{array}$$

**23. What do you mean by adjacency List representation?**

It is an array of linked list, for each vertex a linked list of all adjacent vertices is maintained. It consists of a table with the no. of entries for each vertex for each entry a linked List is initiated for the vertices adjacent to the corresponding table entry.





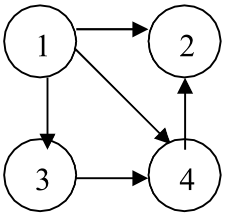
**24. What are the conditions for a graph to become a tree?**

A graph is a tree if it has two properties. i. If it is a connected graph. ii. There should not be any cycles in the graph.

**25. What do you mean by Indegree and Outdegree of a graph?**

- Indegree of a vertex in a graph is the number of incoming edges.
- Outdegree of a vertex is the number of edges that leaves the vertex.

Example:



Indegree for the vertex 1 is 0, vertex 2 is 2, vertex 3 is 1, and vertex 4 is 2

Outdegree for the vertex 1 is 3, vertex 2 is 0, vertex 3 is 1, and vertex 4 is 1.

**26. Define Topological sort.**

Topological sort is defined as an ordering of vertices in a directed acyclic graph. Such that if there is a path from $V_i$ to $V_j$, then $V_j$ appears after $V_i$ in the ordering.

**27. Explain the principle of topological sort.**

- Find the vertex with no incoming edge.
- Print the vertex and remove it along with its edges from the graph.
- Apply the same strategy to the rest of the graph.
- Finally all recorded vertices give topological sorted list.

**28. What is the disadvantage of topological sort?**

Ordering of vertices is not possible if the graph is a cyclic graph. because if there are two vertices v and w on the cycle, v proceeds w and w proceeds v, so ordering not unique.

**29. What is the running time for topological sort?**

The running time of the algorithm for topological sort is O( $V^2$).For the algorithm using Queue, the running time is O (|E| +|V| ) if adjacency list are used.

**30. What is a back edge?**

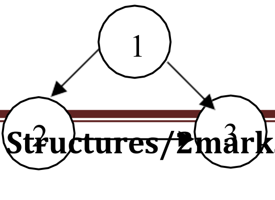
The possibility of reaching an already marked vertex is indicated by a dashed line, in a graph is called as back edge.

**31. State the shortest path problem or single source shortest path problem.**

Given as input a weighted graph or an unweighted graph G= (V, E) and a distinguished vertex s , the shortest path problem is to find the shortest weighted or unweighted path from s to every other vertex.

**32. What do you mean by Negative edge?**

Negative edge means a graph having at least one edge with a negative weight.



**33. Give examples for problems solved by shortest path algorithm.**

- Cheapest way of sending electronic news from one computer to another.
- To compute the best route

**34. What is the running time for the weighted and unweighted shortest path?**

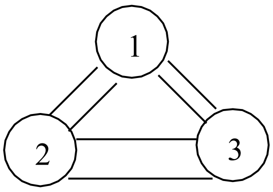
- The running time for the weighted shortest path is O ( |E| + | V| )
- The running time for the Unweighted shortest path is O (|E| log |V| )

**35. What is the advantage of unweighted shortest path algorithm?**

There is no calculation with weights. Only the number of edges on the shortest path is found.

**36. What is a connected graph?**

An undirected graph is connected if there is a path from every vertex to every other vertex. A directed graph with this property is called as strongly connected graph. If a directed graph is not strongly connected but the underline graph without direction is connected it is called as a weakly connected graph.



**37. What are the two traversal strategies used in traversing a graph?**

a. Breadth first search/traversals (BFS/BFT)

b. Depth first search /traversals (DFS/DFT)

**38. What is a minimum spanning tree?**

A minimum spanning tree of an undirected graph G is a tree formed from graph edges that connects all the vertices of G at the lowest total cost.

**39. What is an undirected acyclic graph?**

When every edge in an acyclic graph is undirected, it is called an undirected acyclic graph. It is also called as undirected forest.

**40. What is a minimum spanning tree?**

A minimum spanning tree of an undirected graph G is a tree formed from graph edges. That connects all the vertices of G at lowest total cost.

**41. What is a single source shortest path problem?**

Given as an input, a weighted graph, G=<V,E> and a distinguished vertex 'S' as the source vertex. Single source shortest path problem finds the shortest weighted path from s to every other vertex in G.

**42. Explain about Unweighted shortest path**

Single source shortest path finds the shortest path from the source to each and every vertex present in a unweighted graph. Here no cost is associated with the edges connecting the vertices. Always unit cost is associated with each edge.

**43. Explain about Weighted shortest path**

Single source shortest path finds the shortest path from the source to each and Every vertex present in a weighted graph. In a weighted graph some cost is always associated with the edges connecting the vertices.

**44. What are the methods to solve minimum spanning tree?**

a) Prim's algorithm b) Kruskal's algorithm

**45. Explain briefly about Prim's algorithm**

Prim's algorithm creates the spanning tree in various stages. At each stage, a node is picked as the root and an edge is added and thus the associated vertex along with it.

**46. Define a depth first spanning tree.**

The tree that is formulated by depth first search on a graph is called as depth first spanning tree. The depth first spanning tree consists of tree edges and back edges.

**47. What is activity node in graph?**

The activity-node graph is a vertex-weighted graph. An activity involves both synchronization and branching constructs, similar to but more powerful than a traditional flow chart, which only supports sequential and branching constructs.

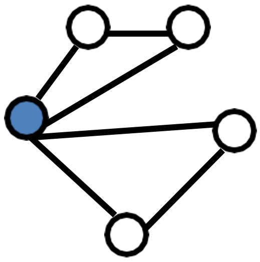
**48. Define Euler tours/circuits?**

An Euler cycle, Euler circuit or Euler tour in an undirected graph is a cycle that uses each edge exactly once. If such a cycle exists, the graph is called unicursal. While such graphs are Euler graphs, not every Euler graph possesses an Euler cycle.

**49. Define articulation points.**

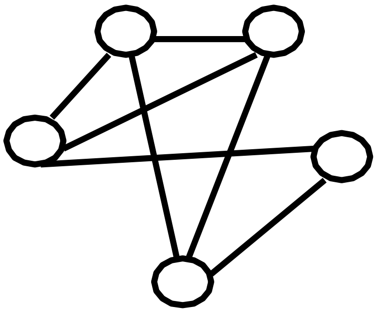
If a graph is not biconnected, the vertices whose removal would disconnect the graph are known as articulation points. We can compute articulation points using depth-first search and a special numbering of the vertices in the order of their visiting. For example

**C is an articulation point**

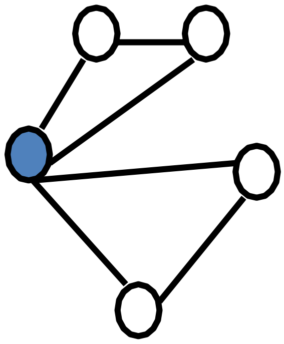


**50. What is biconnected graph?**

A graph is biconnected, if there are no vertices whose removal will disconnect the graph. For example.



Biconnected



Not biconnected

**51. Distinguish BFS and DFS.**

| BFS | DFS |
|---|---|
| BFS removes them from the beginning, which results in maintaining the list as a queue | DFS removes them from the end, maintaining the list as a stack. |
| For BFS in directed graphs each edge of the graph either:<br>• connects two vertices at the same level<br>• Goes down exactly one level.<br>• Goes up any number of levels. | For DFS, each edge either connects an ancestor to :<br>• a descendant<br>• a descendant to an ancestor<br>• One node to a node in a previously visited sub tree. |

**52. List the Applications of Graphs.**

- Computer Networks
- Transport
- Electronic Circuits

## UNIT V: SEARCHING, SORTING AND HASHING TECHNIQUES

**1. General idea of hashing and what is the use of hashing function?**

A hash table similar to an array of some fixes size-containing keys. The keys specified here might be either integer or strings, the size of the table is taken as table size or the keys are mapped on to some number on the hash table from a range of 0 to table size.

**2. What is the need for hashing?**

Hashing is used to perform insertions, deletions and find in constant average time.

**3. List out the different types of hashing functions?**

The different types of hashing functions are,

a. The division method

b. The mind square method

c. The folding method

d. Multiplicative hashing

e. Digit analysis

**4. What are the problems in hashing?**

When two keys compute in to the same location or address in the hash table through any of the hashing function then it is termed collision.

**5. Define hashing.**

It is the implementation of hash tables. Hashing is a technique used for performing insertions, deletions and finds in constant average time.

**6. Define a Key in hashing.**

A key is a string with an associated value. Example: Salary information.

**7. Define table size of a hash table.**

Table size is the size of the table and it is part of the hash data structure. The table runs from 0 to Tablesize-1.

**8. Define Hash table. Give an example.**

The hash table data structure is an array of fixed size containing the keys.

| | |
|---|---|
| 0 | 50 |
| 1 | |
| 2 | |
| 3 | |
| 4 | |
| 5 | |
| 6 | |
| 7 | |
| 8 | |
| 9 | 49 |

**9. List the types of hash function.**

1. Division method
2. Mid square
3. Multiplicative hash function
4. Digit folding
5. Digit analysis

**10. Define a Hash function with an example.**

Hash function maps each key into some number in the range 0 to Tableaize-1and places the key in the appropriate cell.

Example: Key mod Table size is a basic hash function

List the properties of a hash function.

1. Hash function should be simple to compute.
2. Should ensure that any two distinct keys get different cells.
3. It should distribute the keys evenly among the cells.

**11. What is collision in hashing?**

If an element is inserted and if it hashes to the same value as an already inserted element collision takes place.

**12. What are the methods of resolving collision in hashing?**

1. Separate chaining hashing.
2. Open addressing hashing.

**13.Define separate chaining hashing**

Separate chaining means to keep a list of all elements that hash to the same value.

**14. What are the operations of separate chaining?**

1. Find

2. Insert

**15. List the applications of hashing.**

1. DBMS.
2. Computer networks
3. Storage of secret data
4. Cryptography.
5. Securing the database
6. Database applications
7. Storing data in a database

**16. What are the advantages of separate chaining?**

1. Avoids collision by maintaining linked lists.
2. The link lists will be short if the table is large and the hash function is good.

**17. What are the disadvantages of separate chaining?**

1. Requires pointers.
2. Slows down the algorithm
3. Time is required to allocate new cells.
4. Requires the implementation of a second data structure.

**18. What is open addressing?**

In open addressing if a collision occurs alternative cells are tried until an empty cell is found. It is an alternative to resolving collisions with linked list.(i.e.) $h_0(X)$,$h_1(X)$,……Are tried in succession.

Where $h_i(X)$=(Hash(X)+F(i))mod Tablesize with F(0)=0 where F is the collision resolution strategy.

**19. What are the types of open addressing hashing?**

- Linear probing.
- Quadratic probing
- Double hashing

**20. Define primary clustering.**

It is the forming of blocks of occupied cells. It means that any key that hashes into the cluster will require several attempts to resolve the collision and then it will add to the cluster.

**21. What is heap order property?**

The smallest element should be at the root .Any node should be smaller than all of its descendants.

**22. What is Linear probing?**

It amounts to trying cells sequentially with wraparound in search of an empty cell. Function: It is linear ( i.e.)F(i)=i

**23. List out advantages and disadvantages of Linear probing.**

Advantages:

- Does not use pointers
- No second data structure
- Time is not required for allocating new cells.

Disadvantages:

- Time for inserting is quite large.
- Primary clustering.
- Linear probing is a bad idea if the table is more than half full.
- Bigger table is needed than separate chaining.

**19. What do you mean by separate chaining?**

Separate chaining is a collision resolution technique to keep the list of all elements that hash to the same value. This is called separate chaining because each hash table element is a separate chain (linked list). Each linked list contains all the elements whose keys hash to the same index.

**20. Write the advantage and Disadvantages of separate chaining.**

Adv: • More number of elements can be inserted as it uses linked lists.

Dis Adv. • The elements are evenly distributed. Some elements may have more elements and some may not have anything.

• It requires pointers. This leads to slow the algorithm down a bit because of the time required to allocate new cells, and also essentially requires the implementation of a second data structure.

**21. What do you mean by open addressing?**

Open addressing is a collision resolving strategy in which, if collision occurs alternative cells are tried until an empty cell is found. The cells h0(x), h1(x), h2(x),…. are tried in succession, where hi(x)=(Hash(x)+F(i))mod Tablesize with F(0)=0. The function F is the collision resolution strategy.

**22. What do you mean by Probing?**

Probing is the process of getting next available hash table array cell.

**23. What do you mean by linear probing?**

Linear probing is an open addressing collision resolution strategy in which F is a linear function of i, F(i)=i. This amounts to trying sequentially in search of an empty cell. If the table is big enough, a free cell can always be found, but the time to do so can get quite large.

**24. What do you mean by primary clustering?**

In linear probing collision resolution strategy, even if the table is relatively empty, blocks of occupied cells start forming. This effect is known as primary clustering means that any key hashes into the cluster will require several attempts to resolve the collision and then it will add to the cluster.

**25. What do you mean by quadratic probing?**

Quadratic probing is an open addressing collision resolution strategy in which F(i)=i2. There is no guarantee of finding an empty cell once the table gets half full if the table size is not prime. This is because at most half of the table can be used as alternative locations to resolve collisions.

**26. What do you mean by secondary clustering?**

Although quadratic probing eliminates primary clustering, elements that hash to the same position will probe the same alternative cells. This is known as secondary clustering.

**27. List the limitations of linear probing.**

- Time taken for finding the next available cell is large.
- In linear probing, we come across a problem known as clustering.

**28. Mention one advantage and disadvantage of using quadratic probing.**

Advantage: The problem of primary clustering is eliminated.

Disadvantage: There is no guarantee of finding an unoccupied cell once the table
is nearly half full.

**29.What is meant by sorting?**

Ordering the data in an increasing or decreasing fashion according to some relationship among the data item is called sorting.

**30.What are the two main classifications of sorting based on the source of data?**

a. Internal sorting b. External sorting

**31. What is meant by external sorting?**

External sorting is a process of sorting in which large blocks of data stored in storage devices are moved to the main memory and then sorted.

**32.What is meant by internal sorting?**

Internal sorting is a process of sorting the data in the main memory.

**33. What are the various factors to be considered in deciding a sorting algorithm?**

a. Programming time

b. Execution time of the program

c. Memory needed for program environment

**33. What is the main idea in Bubble sort?**

The basic idea underlying the bubble sort is to pass through the file sequentially several times.Each pass consists of comparing each element in the file with its successor (x[i] and x[i+1] and interchanging the two elements if they are not in proper order.

**34. What is the main idea behind insertion sort?**

The main idea of insertion sort is to insert in the ith pass the ith element in A (1) A (2)...A (i) in its rightful place.

**35. What is the main idea behind selection sort?**

The main idea behind the selection sort is to find the smallest element among in A (I) A(J+1)...A (n) and then interchange it with a (J). This process is then repeated for each value of J.

**36. What is the basic idea of shell sort?**

Instead of sorting the entire array at once, it is first divide the array into smaller segments, which are then separately sorted using the insertion sort.

**37. What is the other name for shell sort?**

Diminishing increment sort.

**38. What is the purpose of quick sort?**

The purpose of the quick sort is to move a data item in the correct direction, just enough for to reach its final place in the array.

**39. What i the advantage of quick sort?**

Quick sort reduces unnecessary swaps and moves an item to a greater distance, in one move.

**40. What is the average efficiency of heap sort?**

The average efficiency of heap sort is 0 (n(log2 n)) where, n is the number of elements sorted.

**41. Define segment?**

When large blocks of data are to be sorted, only a portion of the block or file is loaded in the main memory of the computer since, it cannot hold the entire block.

This small portion of file is called a segment.

**42. Name some of the external sorting methods?**

a. Polyphase merging b. Oscillation sorting c. Merge sorting

**16. When is a sorting method said to be stable?**

A sorting method is said to be stable, it two data items of matching values are guaranteed to be not rearranged with respect to each other as the algorithm progresses.

**17. Name some simple algorithms used in external sorting?**

a. Multiway merge b. Polyphase merge c. Replacement selection

**18. When can we use insertion sort?**

Insertion sort is useful only for small files or very nearly sorted files.

**19. How many passes are required fork-way merging?**

The number of passes required using k-way merging is [log k (n/m)] because the N H S get k times as large in each pass.

**20. Define max heap?**

A heap in which the parent has a larger key than the child's is called a max heap.

**21. Define min heap?**

A heap in which the parent has a smaller key than the child is called a min heap.

**22. What is the idea behind Address calculation sort ?**

This sorting is also called Hashing.In this method a function f is applied to each key. The result of the function determines into which of several subfiles the record is to be placed. The function should have the property that if x<=y ,f(x)<=f(y).Such a function is called order preserving.

**23. What is binary search?**

Binary search is a remarkably efficient algorithm for searching in a sorted array. It works by comparing a search key K with the arrays middle element A[m]. if they match the algorithm stops; otherwise the same operation is repeated recursively for the first half of the array if K < A[m] and the second half if K > A[m].

```
                              K
A[0]………A[m-1] A[m] A[m+1]………A[n-1]
          search here if K<A[m]            search here if K>A[m]
```

**24. What is a binary tree extension and what is its use?**

The binary tree extension can be drawn by replacing the empty subtrees by special nodes in a binary tree. The extra nodes shown as little squares are called external & the original nodes shown as little circles called internal. The extension of a empty binary tree is a single external node. The binary tree extension helps in analysis of tree algorithms.

**25. What are the classic traversals of a binary tree?**

The classic traversals are as follows

i. Preorder traversal: the root is visited before left & right subtrees

ii. Inorder traversal: the root is visited after visiting left subtree and before visiting right subtree

iii. Postorder traversal: the root is visited after visiting the left and right subtrees

**26. Mention an algorithm to find out the height of a binary tree.**

ALGORITHM

```
    Height(T) //Compares recursively the height of a binary tree
    //Input: A binary tree T //Output: The height of T
        if T = Φ
            return –1
        else
                return max{Height(TL),
         Height(TR)}+1
```

**27. What are binary search trees and what is it mainly used for?**

Binary search trees is one of the principal data structures for implementing dictionaries. It is a binary tree whose nodes contain elements of a set of orderable items, one element per node, so that all elements in the left subtree are smaller than the element in the subtree’s root and all elements in the right subtree are greater than it.

**28. Define Divide and Conquer algorithm?**

Divide and Conquer algorithm is based on dividing the problem to be solved into several, smaller sub instances, solving them independently and then combining the sub instances solutions so as to yield a solution for the original instance.

**29. Mention some application of Divide and Conquer algorithm?**

a. Quick Sort b. Merge Sort c. Binary search

**30. What are the two stages for heap sort?**

Stage 1 : Construction of heap Stage 2 : Root deletion N-1 times

**31. What is divide and conquer strategy?**

In divide and conquer strategy the given problem is divided into smaller Problems and solved recursively. The conquering phase consists of patching together the answers . Divide – and – conquer is a very powerful use of recursion that we will see many times.