DSA VIVA QUESTIONS:-

UNIT – 1

1. **What is Data structure?**

Answer:- Data Structure can be defined as the group of data elements which provides an efficient way of storing and organising data in the computer.

2. **What is Data?**

Answer:- The term 'DATA' simply referes to a a value or a set of values.

3. **What is Group items and elementary items?**

Answer:- Data items that can be divided into sub items are called group items where as those who can not be divided in to sub items are called elementary items.

4. **What are the applications of data structure?**

Answer:-

5. **What is linear data structure?**

Answer:- A linear data structure is a structure in which the elements are stored sequentially, and the elements are connected to the previous and the next element.

6. **What are the examples of linear data structure?**

Answer:- Array, Queue, Stack, Linked List.

7. **What is non-linear data structure?**

Answer:- A non-linear data structure is also another type of data structure in which the data elements are not arranged in a contiguous manner.

8. **What are the examples of non-linear data structure?**

Answer:- Trees and Graphs.

9. **What is asymptomatic notation?**

Answer:- Asymptotic notations are mathematical tools to represent the time complexity of algorithms for asymptotic analysis.

10. **Name 3 popularly used asymptomatic notation.**

Answer:- Theeta Notation, BIG-O notation, Omega Notation.

11. **What do you mean by Theeta notation?**

Answer:- The theta notation bounds a functions from above and below, so it **defines exact asymptotic behavior.**

12. **What do you mean by BIG-O notation?**

Answer:- The Big O notation defines an upper bound of an algorithm.

13. **What do you mean by Omega Notation?**

Answer:- The Omega notation defines the lower bound of an algorithm.

14. **What is Complexity?**

Answer:- The complexity of an algorithm is a function describing the efficiency of the algorithm in terms of the amount of data the algorithm must process.

15. **What is time complexity?**

Answer:- Time complexity is a function describing the amount of time an algorithm takes in terms of the amount of input to the algorithm.

16. **What is worst-case time complexity?**

Answer:- In the worst case analysis, we calculate upper bound on running time of an algorithm

17. **What is average case time complexity?**

Answer:- In average case analysis, we take all possible inputs and calculate computing time for all of the inputs. Sum all the calculated values and divide the sum by total number of inputs.

18. **What is best case time complexity?**

Answer:- In the best case analysis, we calculate lower bound on running time of an algorithm.

19. **What is space complexity?**

Answer:- Space complexity is a function describing the amount of memory (space) an algorithm takes in terms of the amount of input to the algorithm.

20. **Name few operations available in Data Structure?**

Answer:- Traversing, Searching, sorting, insertion, deletion, etc

21. **What is traversing/searching/sorting/insertion/deletion ?**

Answer:-

- **i.)** **Traversing:-** Accessing each record exactly once so that certain items in the record may be processed.
- **ii.)** **Searching:**- Finding the location of a particular record with a given key value.
- **iii.)** **Inserting:**- Adding a new record to the structure.
- **iv.)** **Deleting:**- removing the record from the structure.
- **v.)** **Sorting:**- Managing the data or record in some logical order

22. **Name two Searching Method.**

Answer:- Linear search and binary search

23. **What is threaded list?**

Answer:- threaded list A list in which additional linkage structures, called threads, have been added to provide for traversals in special orders.

24. Name few sorting methods.

Answer:- Bubble sort, Selection sort, insertion sort, merge sort, quick sort.

25. What is selection Sort and what we exactly do in it?

Answer:- Selection sort is a simple sorting algorithm. This sorting algorithm is an in-place comparison-based algorithm in which the list is divided into two parts, the sorted part at the left end and the unsorted part at the right end. Initially, the sorted part is empty and the unsorted part is the entire list.

The smallest element is selected from the unsorted array and swapped with the leftmost element, and that element becomes a part of the sorted array. This process continues moving unsorted array boundary by one element to the right.

26. What is bubble sort and what we exactly do in it?

Answer:- Bubble sort is a simple sorting algorithm. This sorting algorithm is comparison-based algorithm in which each pair of adjacent elements is compared and the elements are swapped if they are not in order. This algorithm is not suitable for large data sets as its average and worst case complexity are of O(n2) where n is the number of items.

27. What is insertion sort and what we exactly do in it?

Answer:- This is an in-place comparison-based sorting algorithm. Here, a sub-list is maintained which is always sorted. For example, the lower part of an array is maintained to be sorted. An element which is to be 'insert'ed in this sorted sub-list, has to find its appropriate place and then it has to be inserted there. Hence the name, insertion sort.

28. What is merge sort and what we exactly do in it?

Answer:- Merge sort is a sorting technique based on divide and conquer technique. With worst-case time complexity being O(n log n), it is one of the most respected algorithms. Merge sort first divides the array into equal halves and then combines them in a sorted manner.

29. Name any stable sorting algorithm.

Answer:- Merge sort, Insertion sort, Bubble sort, count sort

30. Why is it called Stable?

Answer:- Because in this two objects with equal keys appear in the same order in sorted output as they appear in the input array to be sorted.

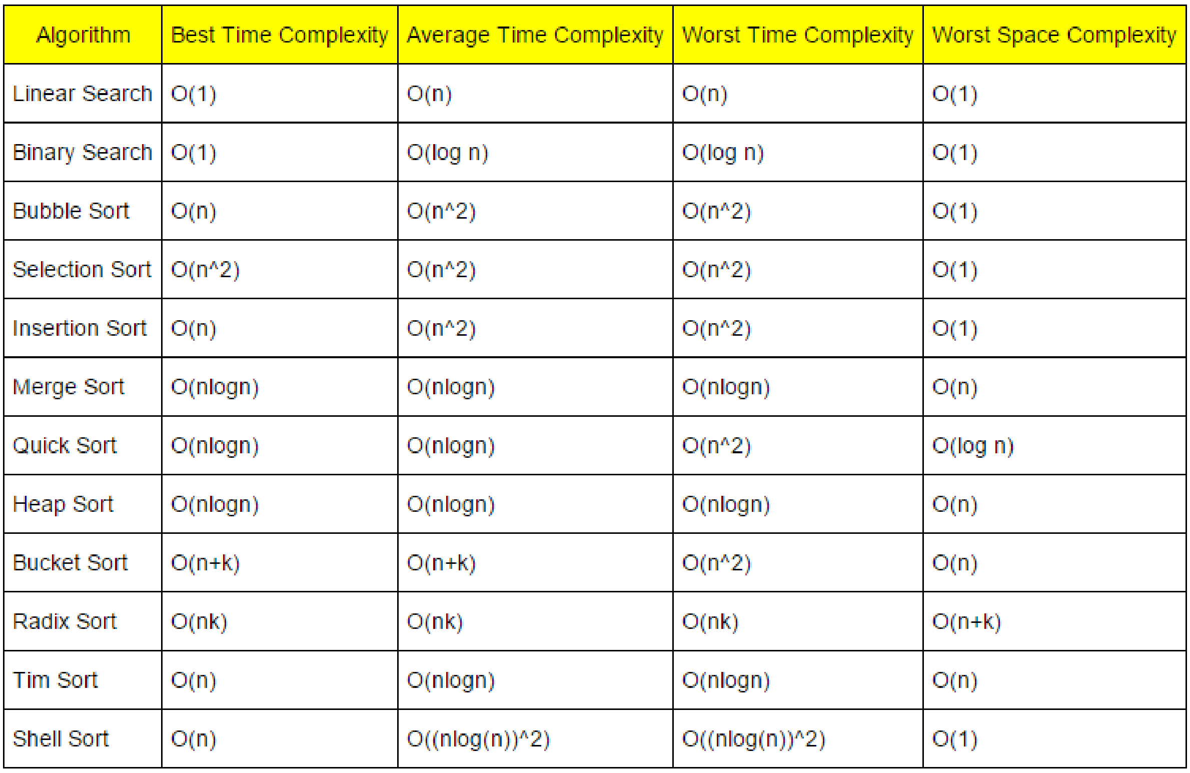
31. What is binary search and what exactly we do in it?

Answer:- Binary search looks for a particular item by comparing the middle most item of the collection. If a match occurs, then the index of item is returned. If the middle item is greater than the item, then the item is searched in the sub-array to the left of the middle item. Otherwise, the item is searched for in the sub-array to the right of the middle item. This process continues on the sub-array as well until the size of the subarray reduces to zero.

MOST IMPORTANT QUESTION TYPE of this unit- complexity of Various sorting and searching algo:-

QUESTION WILL BE LIKE:- what is the worst case/Best case/ average case of Bubble sort/Selection sort/ insertion sort/ merge sort/ quick sort/ Binary search/ linear search.

Answer can be find in this table:-



Algorithm	Best Time Complexity	Average Time Complexity	Worst Time Complexity	Worst Space Complexity
Linear Search	O(1)	O(n)	O(n)	O(1)
Binary Search	O(1)	O(log n)	O(log n)	O(1)
Bubble Sort	O(n)	O(n^2)	O(n^2)	O(1)
Selection Sort	O(n^2)	O(n^2)	O(n^2)	O(1)
Insertion Sort	O(n)	O(n^2)	O(n^2)	O(1)
Merge Sort	O(nlogn)	O(nlogn)	O(nlogn)	O(n)
Quick Sort	O(nlogn)	O(nlogn)	O(n^2)	O(log n)
Heap Sort	O(nlogn)	O(nlogn)	O(nlogn)	O(n)
Bucket Sort	O(n+k)	O(n+k)	O(n^2)	O(n)
Radix Sort	O(nk)	O(nk)	O(nk)	O(n+k)
Tim Sort	O(n)	O(nlogn)	O(nlogn)	O(n)
Shell Sort	O(n)	O((nlog(n))^2)	O((nlog(n))^2)	O(1)

NOW FOR UNIT 2-5 these are some basics operation time complexity which can be asked in any viva.

Worst Case time complexity of different data structures for different operations

Data structure	Access	Search	Insertion	Deletion
Array	O(1)	O(N)	O(N)	O(N)
Stack	O(N)	O(N)	O(1)	O(1)
Queue	O(N)	O(N)	O(1)	O(1)
Singly Linked list	O(N)	O(N)	O(1)	O(1)
Doubly Linked List	O(N)	O(N)	O(1)	O(1)
Hash Table	O(N)	O(N)	O(N)	O(N)
Binary Search Tree	O(N)	O(N)	O(N)	O(N)
AVL Tree	O(log N)	O(log N)	O(log N)	O(log N)
Binary Tree	O(N)	O(N)	O(N)	O(N)
Red Black Tree	O(log N)	O(log N)	O(log N)	O(log N)

UNIT – 2

1. **What are arrays?**

Answer:- An array is a collection of items of similar type stored at contiguous memory locations.

2. **What is the difference between array and structure?**

Answer:- Array is a collection of item of similar data type whereas the structure is a collection of items of different data types.

3. **What is the index of 1st element of array by default?**

Answer:- 0

4. **What the some operations we can do on array?**

Answer:- Traversing, Searching, sorting, etc.

5. **What is the advantage of array over other data structure like Linked list?**

Answer:- Random access is available in array.

6. **What is 2-D array?**

Answer:- 2D array can be defined as an array of arrays.

7. **What is the formula of finding address of 1-D array?**

Answer:- Address of A[I] = B + W * (I – LB) , B=Base address, W= Size of Data type I= subset of element whose address is to be found. LB= Lower Bound

8. **What is the formula of finding address of row major order matrix.**

Answer:- Address of A[I][J] = B + W * ((I – LR) * N + (J – LC))

9. **What is the formula of finding address of column major order matrix.**

Answer:- Address of A[I][J] = B + W * ((J – LC) * M + (I – LR))

10. **What is 3-d array?**

Answer:- A 3-Dimensional array is a collection of 2-Dimensional arrays.

11. **What is linked List?**

Answer:- A linked list is a linear data structure, in which the elements are not stored at contiguous memory locations and each node contains a data field and a pointer to the next node.

12. **What is the difference between linked list and array?**

Answer:- Arrays store elements in contiguous memory locations whereas Linked list store element in non-contiguous memory location.

13. **What is the advantage of Linked list over arrays?**

Answer:- some basic operations like insertion and deletion can be done quickly and easily.

14. What are the basic operations we can do on linked List?

Answer:- Insertion, Deletion, Traversal

15. What are the application of Linked List?

Answer:- It is used in undo functionality of Photoshop or MS Word. It is also used in cache of our browser which allows us to hit Back button. It is used to represent Polynomials.

16. What are the various types of Linked list?

Answer:- There are three types of linked list:-

i.) Singly linked lists
ii.) Doubly linked lists
iii.) Circular linked lists

17. What is doubly linked list?

Answer:- In a doubly linked list, each node contains two links the first link points to the previous node and the next link points to the next node in the sequence.

18. What is Circular linked list?

Answer:- In the circular linked list the last node of the list contains the address of the first node and forms a circular chain.

19. What is sparse matrix?

Answer:- Sparse matrix is a matrix which contains very few non-zero elements.

SOME POINTS :-

If the linked list is empty, then the value of the head points to NULL.

UNIT – 3

1. What is stack?

Answer:- Stack is a linear data structure which follows a particular order in which the operations are performed. The order may be LIFO(Last In First Out) or FILO(First In Last Out).

2. What is the full form of LIFO & FILO?

Answer:- LIFO - > Last In First Out)   FILO -> First In Last Out

3. Give the live example of Stack.

Answer:- You can Give multiple example of your choice. My choice is plate pile example.

We have a lot of plates which we can use, now we are putting one plate and then put every other plate on top of it. Now if we want to take 1 plate out we have to pick the last plate which we have put on the pile. So this is the example of LIFO or FILO. The plate which we put first will we out at last. And plate which we put it last will we out first.

4. **What are the basic operation we can do on Stack?**

Answer:- POP:- Delete an item   PUSH:- insert an item   ISEMPTY():- if(top == -1)

ISFULL() :- if(top==MAXSIZE)

5. **What are the application of Stack?**

Answer:-

i.) Stacks can be used for expression evaluation.
ii.) Stacks can be used to check parenthesis matching in an expression.
iii.) Stack can be used to solve Tower of Hanoi problem

6. **What is the various notations to write arithmetic expression?**

Answer:- Infix, Postfix and Prefix

7. **What is Infix Notation?**

Answer:- Notation in which operators are used in-between operands are infix Notation.

8. **What are prefix Notation?**

Answer:- In this notation, operator is prefixed to operands, i.e. operator is written ahead of operands.

9. **What are postfix Notation?**

Answer:- In this notation style, operator is postfixed to the operands i.e., operator is written after the operands.

10. **What is recursion?**

Answer:- The process in which a function calls itself directly or indirectly is called recursion and the corresponding function is called as recursive function.

11. **Recursion is an application of which data structure?**

Answer:- Stack

12. **What is Queue?**

Answer:- A Queue is a linear structure which follows a particular order in which the operations are performed. The order is First In First Out (FIFO)

13. **Give the live example of Queue.**

Answer:- Queue can be understand by the train ticket booking row. What exactly happening in this is that the first person who went to book the train ticket will get the ticket first and get out of the row first. The person who entered last will get ticket last.

NOTE:- In a queue data structure, the insertion operation is performed at a position which is known as **'rear'** and the deletion operation is performed at a position which is known **as 'front'.**

14. **What are the operations of Queue?**

Answer:- enqueue():- Insert an element

dequeue() :- delete an element

15. How is insertion and deletion performed in Queue?

Answer:- the insertion operation is performed using a function called "enQueue()"

deletion operation is performed using a function called "deQueue()".

16. What are the various types of Queue?

Answer:- There are four different types of queues:

i.) Simple Queue
ii.) Circular Queue
iii.) Priority Queue
iv.) Double Ended Queue

17. What is deque?

Answer:- Deque or Double Ended Queue is a generalized version of Queue data structure that allows insert and delete at both ends.

18. What is priority queue?

Answer:- Priority Queue is an abstract data type, which is similar to a queue, however, in the priority queue, every element has some priority.

19. What are the application of Priority Queue?

Answer:- it is used to implement prim’s algorithm, Dijkstra’s Shortest Path Algorithm

20. What is circular queue?

Answer:- Circular Queue is a linear data structure in which the operations are performed based on FIFO (First In First Out) principle and the last position is connected back to the first position to make a circle.

21. Can we implement Queue using array/Lnked list?

Answer:- YES

UNIT – 4

1. What is tree?

Answer:- Tree is a non-linear data structure which organizes data in hierarchical structure

2. What is ROOT, LEAF, AND OTHER TERMINOLOGIES OF TREES?

Answer-

Tree Data Structure

* Tree is a non-linear data structure where each node is connected to a number of nodes with the help of pointers or references.
Tree store data in heriechial system.

Basic Terminology :-

Root :- The root of a tree is the first node of the tree. Every tree must have root node. We cant have multiple root in a tree.

Edge :- The connecting link between any two nodes is called as Edge. In a tree with N nodes there will be N-1 edges.

Parent :- The node ~~with~~ which is predecessor of any node is called as Parent Node.

Child :- The node which is descendent of any node is called as Child Node. In a tree, all the nodes except root node are child.

Siblings :- The children node of same parent is called siblings. That is node with same parents.

Leaf :- The node which does not have any child is called leaf node.
(Also known as External nodes / Terminal nodes)

Internal nodes :- The node which has atleast one children is called ~~Exter~~ Internal node.

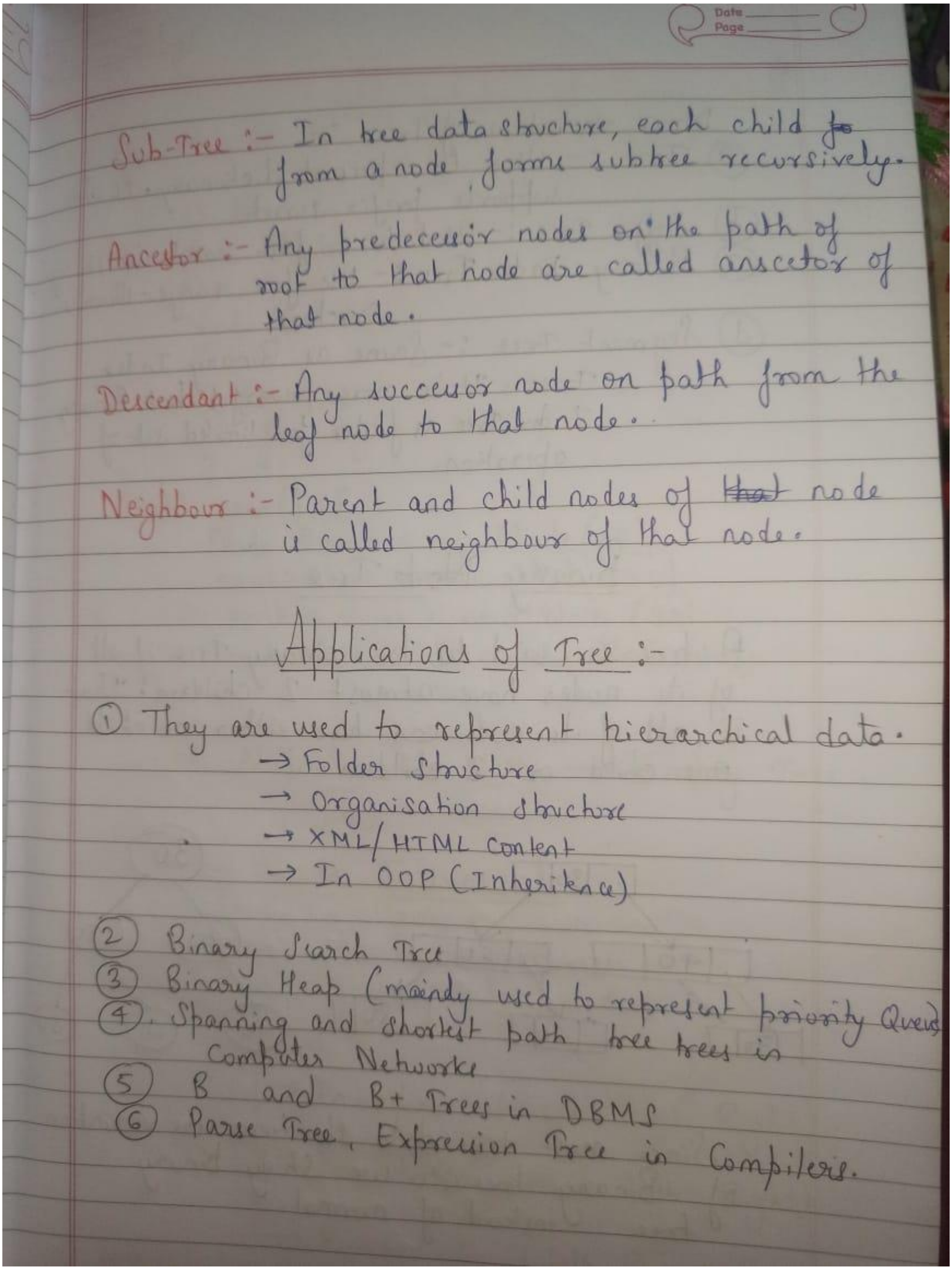
Degree :- The total no. of children of a node is called degree of that node. The highest degree of a node among all the nodes in a tree is called Degree of Tree.

Level :- In tree each step from top to bottom is called level of tree and level count starts from 0.

Height :- The total number of edges from leaf node to a particular node in the longest path is called Height of ~~tree~~ node. Height of root node is height of tree.

Depth :- The total number of edges from root node to particular node is called depth of that node. In a tree, the total number of edges from root node to a leaf node in longest path is called depth of tree.

Path :- The sequence of Node and Edge from one node to another is called path between that two nodes. Length of path is total number of nodes in that path.



Sub-Tree :- In tree data structure, each child ~~to~~ from a node forms subtree recursively.

Ancestor :- Any predecessor nodes on the path of root to that node are called anscetor of that node.

Descendant :- Any succesor node on path from the leaf node to that node.

Neighbour :- Parent and child nodes of ~~that~~ node is called neighbour of that node.

Applications of Tree :-

① They are used to represent hierarchical data.
→ Folder Structure
→ Organisation structure
→ XML/HTML content
→ In OOP (Inheritance)

② Binary Search Tree
③ Binary Heap (mainly used to represent priority Queue)
④. Spanning and shortest path tree trees in Computer Networks
⑤ B and B+ Trees in DBMS
⑥ Parse Tree, Expression Tree in Compilers.

3. What is the various ways to represent binary tree?

Answer:- 1. Array representation 2. Linked list representation

4. What are the various types of traversals.

Answer:- There are three types of binary tree traversals.

1. In - Order Traversal :- LEFT -> ROOT -> RIGHT

2. Pre - Order Traversal :- ROOT -> LEFT -> RIGHT

3. Post - Order Traversal :- LEFT -> RIGHT -> ROOT

4. Level-order tranversal is also ther in which we traverse level by level.

5. What is threaded binary tree?

Answer:- a threaded binary tree is a binary tree variant that facilitates traversal in a particular order.

6. **What is strictly Binary tree?**

Answer:- A binary tree in which every node has either two or zero number of children is called Strictly Binary Tree. It is also called as Full Binary Tree or Proper Binary Tree or 2-Tree.

7. **What is complete binary tree?**

Answer:- A binary tree in which every internal node has exactly two children and all leaf nodes are at same level is called Complete Binary Tree. It is also called as Perfect Binary Tree.

8. What is extended binary tree?

Answer:- The full binary tree obtained by adding dummy nodes to a binary tree is called as Extended Binary Tree.

9. What is binary search tree?

Answer:- Binary Search Tree is a binary tree in which every node contains only smaller values in its left subtree and only larger values in its right subtree.

10. Name 2 self balancing tree.

Answer:- AVL TREE & RED-BLACK TREE

11. What is balance factor?

Answer:- Balance factor of a node is the difference between the heights of left and right subtrees of that node.

12. What is rotation?

Answer:- Rotation is the process of moving the nodes to either left or right to make tree balanced.

13. What are the types of rotation?

Answer:- 4 types:-

1.) LL ROTATION :- In LL Rotation every node moves one position to left from the current position.
2.) RR ROTATION :- In RR Rotation every node moves one position to right from the current position.
3.) LR ROTATION:- In LR Rotation, first every node moves one position to left then one position to right from the current position.
4.) RL ROTATION:- In RL Rotation, first every node moves one position to right then one position to left from the current position.

14. **What is b-tree?**

Answer:- B-Tree is a self-balanced search tree with multiple keys in every node and more than two children for every node.

15. Name 2-3 property of B-TREE.

Answer:-

- All the leaf nodes must be at same level.
- All the key values within a node must be in Ascending Order.
- If the root node is a non leaf node, then it must have at least 2 children.

16. What is splay tree?

Answer:- Splay Tree is a self - adjusted Binary Search Tree in which every operation on an element rearrange the tree so that the element is placed at the root position of the tree.

17. What do you mean by splaying?

Answer:- Splaying an element is the process of bringing it to the root position by performing suitable rotation operations.

18. What are the various type of rotation in Splay tree?

Answer:- Rotations in Splay Tree:-

1. Zig Rotation:- The Zig Rotation in a splay tree is similar to the single right rotation in AVL Tree rotations. In zig rotation every node moves one position to the right from its current position.

2. Zag Rotation:- The Zag Rotation in a splay tree is similar to the single left rotation in AVL Tree rotations. In zag rotation every node moves one position to the left from its current position.

3. Zig - Zig Rotation:- The Zig-Zig Rotation in a splay tree is a double zig rotation. In zig-zig rotation every node moves two position to the right from its current position.

4. Zag - Zag Rotation:- The Zag-Zag Rotation in a splay tree is a double zag rotation. In zag-zag rotation every node moves two position to the left from its current position.

5. Zig - Zag Rotation:- The Zig-Zag Rotation in a splay tree is a sequence of zig rotation followed by zag rotation. In zig-zag rotation every node moves one position to the right followed by one position to the left from its current position.

6. Zag - Zig Rotation:- The Zag-Zig Rotation in a splay tree is a sequence of zag rotation followed by zig rotation. In zag-zig rotation every node moves one position to the left followed by one position to the right from its current position.

19. What is red-black tree?

Answer:- Red Black Tree is a Binary Search Tree in which every node is colored eigther RED or BLACK.

20. Name any 2-3 property of red-black tree.

Answer:-

- The ROOT node must colored BLACK.
- The children of Red colored node must colored BLACK.
- In all the paths of the tree there must be same number of BLACK colored nodes.
- Every leaf (e.i. NULL node) must colored BLACK.

UNIT- 5:-

1. What is a graph?

Answer:- A Graph is a non-linear data structure consisting of nodes and edges. The nodes are sometimes also referred to as vertices and the edges are lines or arcs that connect any two nodes in the graph.

Graph Terminology

Path:- A path can be defined as the sequence of nodes that are followed in order to reach some terminal node V from the initial node U.

Closed Path:- A path will be called as closed path if the initial node is same as terminal node. A path will be closed path if V_0=V_N.

Simple Path:- If all the nodes of the graph are distinct with an exception V_0=V_N, then such path P is called as closed simple path.

Cycle:- A cycle can be defined as the path which has no repeated edges or vertices except the first and last vertices.

Connected Graph:- A connected graph is the one in which some path exists between every two vertices (u, v) in V. There are no isolated nodes in connected graph.

Complete Graph:- A complete graph is the one in which every node is connected with all other nodes. A complete graph contain n(n-1)/2 edges where n is the number of nodes in the graph.

Weighted Graph:- In a weighted graph, each edge is assigned with some data such as length or weight. The weight of an edge e can be given as w(e) which must be a positive (+) value indicating the cost of traversing the edge.

Digraph:- A digraph is a directed graph in which each edge of the graph is associated with some direction and the traversing can be done only in the specified direction.

Loop:- An edge that is associated with the similar end points can be called as Loop.

Adjacent Nodes:- If two nodes u and v are connected via an edge e, then the nodes u and v are called as neighbours or adjacent nodes.

Degree of the Node:- A degree of a node is the number of edges that are connected with that node. A node with degree 0 is called as isolated node.

2. What is the basic operations in graph?

Answer:- Following are basic primary operations of a Graph which are following.

1.) Add Vertex − add a vertex to a graph.
2.) Add Edge − add an edge between two vertices of a graph.
3.) Display Vertex − display a vertex of a graph.

3. What is the various traversal method of graph?

Answer:- Following are the 2 traversal method of graph:-

1.) **DFS:-** Depth First Search algorithm(DFS) traverses a graph in a depth ward motion and uses **a stack to** remember to get the next vertex to start a search when a dead end occurs in any iteration.
2.) **BFS:**- Breadth First Search algorithm(BFS) traverses a graph in a breadth wards motion and **uses a queue to** remember to get the next vertex to start a search when a dead end occurs in any iteration.

4. What is spanning tree?

Answer:- A spanning tree is a subset of Graph G, which has all the vertices covered with minimum possible number of edges. A complete undirected graph can have maximum nn-2 number of spanning trees, where n is number of nodes.

5. **What are the application of spanning tree?**

Answer:- Spanning tree is basically used to find minimum paths to connect all nodes in a graph. Common application of spanning trees are − 1. Civil Network Planning 2. Computer Network Routing Protocol 3. Cluster Analysis

6. **What is minimum spanning tree?**

Answer:- In a weighted graph, a minimum spanning tree is a spanning tree that has minimum weight that all other spanning trees of the same graph.

7. **Name any 2 minimum spanning tree algorithm.**

Answer:- 1. Kruskal's Algorithm

2. Prims Algorithm

8. **Shortest Path Algorithm ----> Dijkstra**
9. **what is biconnected graph?**

Answer:- A biconnected graph is a connected graph on two or more vertices having no articulation vertices.

10. **What is Hashing?**

Answer:- Hashing is a technique or process of mapping keys, values into the hash table by using a hash function.

11. What is hash table?

Answer:- Hash Table is a data structure which stores data in an associative manner. In a hash table, data is stored in an array format, where each data value has its own unique index value.

12. **What is hash function?**

Answer:- A hash function is any function that can be used to map data of arbitrary size to fixed-size values.

13. **What are the various operation of hash table?**

Answer:- Following are basic primary operations of a hashtable which are following.

1.) Search − search an element in a hashtable.
2.) Insert − insert an element in a hashtable.
3.) delete − delete an element from a hashtable.

14. **What is collision?**

Answer:- Since a hash function gets us a small number for a key which is a big integer or string, there is possibility that two keys result in same value. The situation where a newly inserted key maps to an already occupied slot in hash table is called collision.

15. **What are the various method to handle collision?**

Answer:- There are mainly two methods to handle collision:

1) Separate Chaining 2) Open Addressing