# Data Structure (Viva Questions)

## Trees:

1.Explain the concept of a tree. Discuss its applications.

2. What are the two ways of representing binary trees in the memory? Which one do you prefer? and why?

3.List all possible non-similar binary trees having four nodes.

4.What is the maximum number of levels that a binary search tree with 100 nodes can have?

5. What is the maximum height of a tree with 32 nodes?

6. What is the maximum number of nodes that can be found in a binary tree at levels 3, 4, and 12?

7. Draw all possible non-similar binary trees having three nodes.

8. Explain the concept of binary search trees.

9. Explain the operations on binary search trees.

10. How does the height of a binary search tree affect its performance?

11. How many nodes will a complete binary tree with 27 nodes have in the last level? What will be the height of the tree?

12. Write a short note on threaded binary trees.

13. Why are threaded binary trees called efficient binary trees? Give the merits of using a threaded binary tree.

14. Discuss the advantages of an AVL tree.

15. How is an AVL tree better than a binary search tree?

16. How does a red-black tree perform better than a binary search tree?

17. List the merits and demerits of a splay tree.

18. Why is a large value of m needed in a B tree?

19. Compare B trees with B+ trees.

20. In what conditions will you prefer a B+ tree over a B tree?

21.Give a brief summary of M-way search trees.

## Whether True or False, Explain and Justify with Reasons (Trees)

1. Nodes that branch into child nodes are called parent nodes.

2. The size of a tree is equal to the total number of nodes.

3. A leaf node does not branch out further.

4. A node that has no successors is called the root node.

5. A binary tree of n nodes has exactly `n – 1` edges.

6. Every node has a parent.

7. The Huffman coding algorithm uses a variable length code table.

8. The internal path length of a binary tree is defined

as the sum of all path lengths summed over each path from the root to an external node.

9.In a binary search tree, all the nodes in the left sub-tree have a value less than that of the root node.

10. If we take two empty binary search trees and insert the same elements but in a different order, then the resultant trees will be the same.

11. When we insert a new node in a binary search tree, it will be added as an internal node.

12. Mirror image of a binary search tree is obtained by interchanging the left sub-tree with the right sub-tree at every node of the tree.

13. If the thread appears in the right field, then it will point to the in-order successor of the node.

14. If the node to be deleted is present in the left subtree of `A`, then `R` rotation is applied.

15. Height of an AVL tree is limited to `O(log n)`.

16. Critical node is the nearest ancestor node on the path from the root to the inserted node whose balance factor is –1, 0, or 1.

17. RL rotation is done when the new node is inserted in the right sub-tree of the right sub-tree of the critical node.

18. In a red-black tree, some leaf nodes can be red.

# Graphs:

1.Explain the relationship between a linked list structure and a digraph.

2. What is a graph? Explain its key terms.

3. How are graphs represented inside a computer's memory? Which method do you prefer and why?

4.Explain the graph traversal algorithms in detail with example.

5. Differentiate between depth-first search and breadth-first search traversal of a graph.

6. Explain the topological sorting of a graph G.

7. Define spanning tree.

8. When is a spanning tree called a minimum spanning tree? Take a weighted graph of your choice and find out its minimum spanning tree.

9. Explain Prim's algorithm.

10. Write a brief note on Kruskal's algorithm.

11. Write a short note on Dijkstra's algorithm.

12. Differentiate between Dijkstra's algorithm and minimum spanning tree algorithm.

## Whether True or False, Explain and Justify with Reasons:(Graphs)

1. Graph is a linear data structure.

2. In-degree of a node is the number of edges leaving that node.

3. The size of a graph is the total number of vertices in it.

4. A sink has a zero in-degree but a positive outdegree.

5. The space complexity of depth-first search is lower than that of breadth-first search.

6. A node is known as a sink if it has a positive outdegree but the in-degree = 0.

7. A directed graph that has no cycles is called a directed acyclic graph.

8. A graph G can have many different spanning trees.

9. Fringe vertices are not a part of T, but are adjacent to some tree vertex.

10. Kruskal's algorithm is an example of a greedy algorithm.