each iteration to create bigger sorted lists, which are combined recursively until there is a single sorted list.

29. How is a variable stored in memory when using data structures?

A variable is stored according to the required amount of memory. The steps to store a variable in memory are as follows:

- It starts by assigning the amount of memory that is needed.
- It is stored as per the data structure.
- Dynamic allocation ensures enhanced efficiency, and one can access units based on their real-time requirements.

30. What is dynamic memory management?

In the dynamic memory management technique, storage units are allocated according to the requirements. Dynamic memory allocation allows storing each data structure either separately or together to form entities known as composites. You can work on these entities whenever required.

31. What is a postfix expression?

Post-fix expression consists of operators and operands, and every operator precedes the operands. With the concept of operator precedence, it eliminates the need for using parentheses or sub-expressions.

32. What is a binary tree?

A binary tree is used to organise data, making data retrieval and manipulation efficient. This data structure has two nodes, left and right, called leaves and nodes. Leaves represent data, while nodes represent relationships between leaves.

Every node has two children, also known as siblings, with each child having one parent. The parent node is closest to the root tree. If a node is deleted from the tree, it gets deleted from its child and parent. According to programming language, a binary tree is an extension to the linked list.

33. What are the applications of binary trees?

Common applications of a binary tree are:

- In data compression methods in the Huffman coding tree
- Display hierarchical data
- Search for a specific element
- Put priority queues into action
- Implement file systems where every individual node represents a directory or file.
- Implement decision trees, a machine learning algorithm for classification and regression analysis.

- In editing apps of spreadsheets and Microsoft Excel
- Encoding and decoding operations
- Implement efficient sorting algorithms

34. What is the AVL tree data structure?

An AVL tree, named after its inventors Adelson, Velskii, and Landi, is the highest-balancing binary search tree. It compares the heights of the left and right subtrees of nodes and ensures that the difference is less than or equal to one. The difference is called the Balance Factor. It is used for a large set of data with continuous pruning through insertion and deletion.

35. List the operations, rotations, and applications of an AVL tree.

An AVL tree is used to perform the following operations:

a) Insertion

Although insertion in an AVL tree is the same as in a binary search tree, it may cause a violation of the property. Hence, the tree must be balanced, for which rotations can be used.

b) Deletion

It is also the same as in the binary search tree. However, deletion can lead to disruption in the tree’s balance, so rotations are used to balance it.

To balance an AVL tree, it performs the following rotations:

c) Left Rotation

When inserting a node into the right subtree of the right subtree cause the tree to be unbalanced, users need to perform a single left rotation.

d) Right Rotation

If the tree is unbalanced because of insertion in the left subtree of the left subtree, it requires a single right rotation.

e) Left-Right Rotation

It is the extended version of the single rotation and is also known as double rotation. When node insertion is into the right subtree of the left subtree, a left rotation is followed by a right rotation to balance the tree.

f) Right-Left Rotation

When node insertion is done into the left subtree of the right subtree, a right rotation is followed by a left rotation.

Here are a few real-life applications of an AVL tree:

- In-memory sets and dictionaries
- Database applications with fewer insertions and deletions but frequent data lookups.
- Applications that need enhanced searching.

36. What is the difference between the B tree and the B+ tree?

a) B Tree

It refers to a self-balancing m-way tree, where ‘m’ signifies the tree’s order. It is an extension of the binary search tree where nodes can have more than two children and one key. The data is sorted in a B tree with lower values placed on the left subtree while higher values are placed on the right subtree.

b) B+ Tree

It refers to an advanced self-balanced tree where every path from the root to the leaf is of the same length. This shows that all the paths occurred at the same level. So, when all leaf nodes appear on the second level, no other lead node can appear on the third level.

37. What are the applications for B-trees?

Before discussing the real-life applications of B-tress, let’s have a look at some of its properties:

The following are the key properties of a B-tree data structure:

- All the leaves are at the same height.
- Minimum degree ‘t’ refers to a B-tree, and its value is determined by the disk block size.
- Each node must have at least t-1 keys. The root is the only exception to this, which must contain at least one key.
- In B-tree, node insertion happens at Leaf Node.
- The number of children of a node= number of keys in the node plus one.
- Unlike the binary search tree, which grows and shrinks from downward, B-tree grows and shrinks from the root.
- Nodes, including root, can’t have more than 2*t-1 keys.
- Keys are sorted in ascending order. A child between k1 and k2 consists of all keys ranging from k1 to k2.

The applications of B-tree are:

- In CAD systems, to sort and search geometric data.
- In the case of a large database, to access data stored on a disc
- Most servers use B-tree
- To achieve multilevel indexing
- To search data in less time

- In areas such as computer networks, natural language processing (NLP), and cryptography.

It is among the top interview questions in data structures.

38. What is a graph data structure?

It is a non-linear data structure with nodes or vertices connected by a set of ordered edges. These edges are lines or arcs connecting two nodes in a graph so that data can be accessed and stored based on requirements.

39. What are the applications of a graph data structure?

Real-time applications of a graph data structure are:

- Social network graphs to track users’ data.
- Google Maps for building transportation systems
- Transport grids with stations represented as vertices and routes as edges
- Dijkstra algorithm to determine the smallest path between two or more nodes
- Utility graph or water or power with vertices representing connection points and edges of the wires.
- Operating System, in the Resource Allocation Graph where each process and resource are represented as vertices
- World Wide Web, where web pages are considered as vertices
- Computer science to represent the flow of computation

Advanced Data Structure Interview Questions

40. What is the difference between the Breadth First Search (BFS) and Depth First Search (DFS)?

Breadth First Search (BFS)	**Depth First Search (DFS)**
Uses Queue data structure to find the shortest path.	Uses the Stack data structure to find the sl
Need to walk through all the nodes on the same level to go to the next level in BFS.	Begins at the root node and then proceeds the node with no unvisited nearby nodes.
It is slower.	Faster compared to BFS.
Performs better if the target is closer to the source.	Performs better if the target is farther fron
Needs more memory.	Needs less memory.
No backtracking is possible.	As it is a recursive algorithm, it employs t
Based on the FIFO principle (First In First Out).	Based on the LIFO principle (Last In Firs