21 Lecture - CS301

Important Subjective

1. **What is an AVL Tree and what is its significance in computer science?**

Answer: An AVL Tree is a self-balancing binary search tree in which the heights of the left and right subtrees of any node differ by at most one. Its significance in computer science lies in its ability to maintain efficient search, insertion and deletion operations with a guaranteed time complexity of O(log n).

2. **What is the difference between a balanced binary search tree and an unbalanced binary search tree?**

Answer: A balanced binary search tree maintains a balance between the height of the left and right subtrees of any node, ensuring a time complexity of O(log n) for its operations. An unbalanced binary search tree, on the other hand, can have a skewed structure that results in a time complexity of O(n) for its operations.

3. **How does the AVL Tree maintain balance during insertion and deletion operations?**

Answer: The AVL Tree maintains balance during insertion and deletion operations by performing rotations at the nodes where the balance factor (the difference between the heights of the left and right subtrees) is greater than one. The rotations are performed to move the subtrees and maintain the balance factor within the acceptable range.

4. **What is the height of a perfectly balanced AVL Tree with 10 nodes?**

Answer: The height of a perfectly balanced AVL Tree with 10 nodes would be 4.

5. **What is the difference between a single rotation and a double rotation in an AVL Tree?**

Answer: A single rotation is performed to balance an AVL Tree when the balance factor of a node is greater than one, and it involves rotating the node and its subtree once. A double rotation, on the other hand, involves performing two rotations to balance the tree, either in the same or opposite directions.

6. **Can a binary search tree be both height-balanced and weight-balanced?**

Answer: Yes, a binary search tree can be both height-balanced and weight-balanced. AVL Trees are an example of a height-balanced binary search tree, while Red-Black Trees are an example of a weight-balanced binary search tree.

7. **Why is it important to maintain balance in a binary search tree?**

Answer: It is important to maintain balance in a binary search tree to ensure that the search, insertion and deletion operations have a guaranteed time complexity of O(log n), making them

efficient for large datasets.

8. **What is the time complexity of searching for a node in an AVL Tree?**

Answer: The time complexity of searching for a node in an AVL Tree is O(log n).

9. **Can an AVL Tree have duplicate nodes?**

Answer: Yes, an AVL Tree can have duplicate nodes, but they would need to be stored in a specific way, such as storing a count for each duplicate node.

10. **What is the root node of an AVL Tree?**

Answer: The root node of an AVL Tree is the topmost node in the tree, from which all other nodes are descended.