

12 Lecture - CS301

Important Subjective

1. What is a binary tree?

Answer: A binary tree is a tree data structure in which each node has at most two children, referred to as the left child and the right child.

2. What is the height of a binary tree?

Answer: The height of a binary tree is the length of the longest path from the root node to a leaf node.

3. What is the difference between a binary tree and a binary search tree?

Answer: A binary search tree is a binary tree with the property that the value of each node is greater than or equal to the values in its left subtree and less than or equal to the values in its right subtree.

4. What is the time complexity of inserting a node in a binary tree?

Answer: The time complexity of inserting a node in a binary tree is $O(h)$, where h is the height of the tree.

5. What is the time complexity of deleting a node in a binary tree?

Answer: The time complexity of deleting a node in a binary tree is $O(h)$, where h is the height of the tree.

6. What is the difference between pre-order traversal and post-order traversal?

Answer: Pre-order traversal visits the root node first, followed by the left subtree and then the right subtree, while post-order traversal visits the left subtree first, followed by the right subtree and then the root node.

7. How do you determine if a binary tree is balanced?

Answer: A binary tree is balanced if the height of its left subtree and the height of its right subtree differ by at most one.

8. What is the difference between complete binary tree and a full binary tree?

Answer: A complete binary tree is a binary tree in which every level except possibly the last is completely filled, while a full binary tree is a binary tree in which every node has either two children or zero children.

9. What is the time complexity of finding the maximum element in a binary tree?

Answer: The time complexity of finding the maximum element in a binary tree is $O(n)$, where n is the number of nodes in the tree.

10. What is the time complexity of finding the height of a binary tree using dynamic programming?

Answer: The time complexity of finding the height of a binary tree using dynamic programming is $O(n)$, where n is the number of nodes in the tree.