FA22-BCS-183

Muhammad Ahmad

Section C

DSA assignment#3

Question 1:

1. Parent of a given node:

- Start at the root.

- Traverse the tree while comparing the given node value with the current node’s values.

- Keep track of the parent node until the given node is found.

2. Depth/Level of a given node:

- Perform a depth-first or breadth-first traversal, keeping track of the level as you traverse.

- When the given node is found, return its level.

3. Height of the tree:

- Recursively calculate the height of the left and right subtrees.

- Return the maximum height plus 1.

4. Check if a tree is a BST:

- Perform an in-order traversal.

- Check if the values are in ascending order.

5. Check if two nodes are at the same level:

- Use a modified level order traversal.

- Record the level of the first and second nodes.

- If levels are the same, return true.

6. Total number of nodes:

- Recursively count nodes in the left and right subtrees.

- Return the sum plus 1.

7. InOrderSuccessor of a given node:

- If the node has a right subtree, return the leftmost node of the right subtree.

- If the node doesn’t have a right subtree, go up the tree until finding the first ancestor where the node is in the left subtree.

Question 2:

Deletion Algorithm (Iterative) for BST:

1. Start at the root and traverse the tree to find the node to delete.

2. Identify the type of node (leaf, single child, or both children).

3. For leaf node deletion, remove the node directly.

4. For a node with a single child, replace the node with its child.

5. For a node with both children, find the in-order successor (or predecessor), replace the node's value with the successor's value, and then recursively delete the successor.

Question 3:

Priority Queue using Binary Heap (linked

list implementation):

1. Implement a binary heap using a linked list.

2. For insertion, add the new element at the end and then heapify up.

3. For deletion (removal of the highest priority), swap the root with the last element, remove the last element, and then heapify down.

4. Test the class by inserting elements with different priorities, removing elements, and checking the order of removal.