DSA Practice – 10

1.Left view of a BST

```
import java.util.*;
class TreeNode {
    int val;
    TreeNode left, right;
    public TreeNode(int val) {
        this.val = val;
        left = right = null;
    }
}
public class BSTLeftView {
    public static List<Integer> leftView(TreeNode root) {
        List<Integer> result = new ArrayList<>();
        if (root == null) {
            return result;
        }
        Queue<TreeNode> queue = new LinkedList<>();
        queue.add(root);
        while (!queue.isEmpty()) {
            int levelSize = queue.size();
            for (int i = 0; i < levelSize; i++) {
                TreeNode currentNode = queue.poll();
                if (i == 0) {
                    result.add(currentNode.val);
                }
                if (currentNode.left != null) {
                    queue.add(currentNode.left);
                }
```

```
            if (currentNode.right != null) {
                queue.add(currentNode.right);
            }
        }
    }
    return result;
  }
  public static void main(String[] args) {
    TreeNode root = new TreeNode(10);
    root.left = new TreeNode(5);
    root.right = new TreeNode(20);
    root.left.left = new TreeNode(3);
    root.left.right = new TreeNode(7);
    root.right.right = new TreeNode(25);
    List<Integer> leftViewResult = leftView(root);
    System.out.println("Left View of the BST: " + leftViewResult);
  }
}
```

Output

```
Microsoft Windows [Version 10.0.22631.4460]
(c) Microsoft Corporation. All rights reserved.

C:\Users\PRETHIKA>cd Documents

C:\Users\PRETHIKA\Documents>javac BSTLeftView.java

C:\Users\PRETHIKA\Documents>java BSTLeftView
Left View of the BST: [10, 5, 3]
```

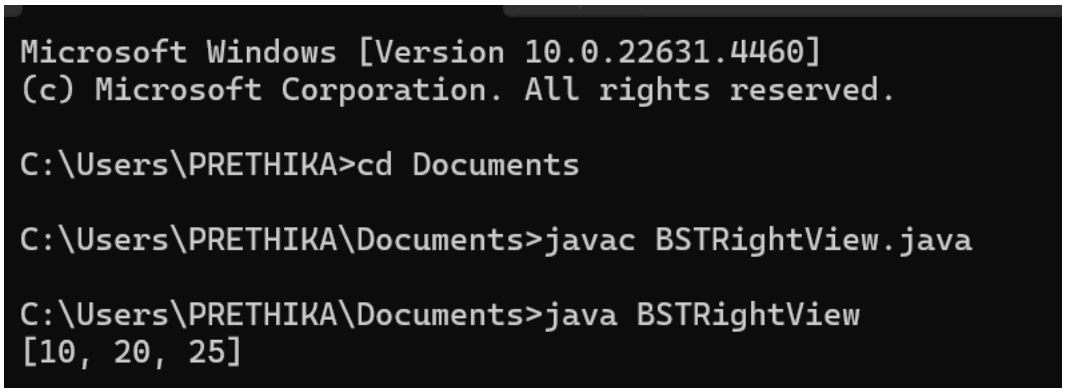
2.Right view of a BST

```
import java.util.*;
class TreeNode {
```

```
    int val;
    TreeNode left, right;
    TreeNode(int val) { this.val = val; }
}
public class BSTRightView {
    public static List<Integer> rightView(TreeNode root) {
        List<Integer> result = new ArrayList<>();
        if (root == null) return result;
        Queue<TreeNode> queue = new LinkedList<>();
        queue.add(root);
        while (!queue.isEmpty()) {
            int levelSize = queue.size();
            for (int i = 0; i < levelSize; i++) {
                TreeNode current = queue.poll();
                if (i == levelSize - 1) {
                    result.add(current.val);
                }
                if (current.left != null) queue.add(current.left);
                if (current.right != null) queue.add(current.right);
            }
        }
        return result;
    }
    public static void main(String[] args) {
        TreeNode root = new TreeNode(10);
        root.left = new TreeNode(5);
        root.right = new TreeNode(20);
        root.left.left = new TreeNode(3);
        root.left.right = new TreeNode(7);
        root.right.right = new TreeNode(25);
```

```
        System.out.println(rightView(root));
    }
}
```

Output

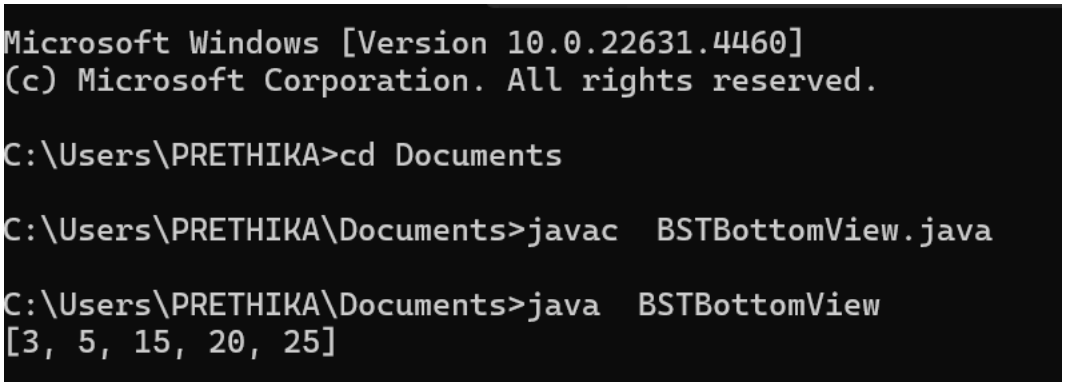


3.Bottom view of a BST

```
import java.util.*;
class TreeNode {
    int val;
    TreeNode left, right;
    TreeNode(int val) { this.val = val; }
}
public class BSTBottomView {
    static class Pair {
        TreeNode node;
        int hd;s
        Pair(TreeNode node, int hd) {
            this.node = node;
            this.hd = hd;
        }
    }
    public static List<Integer> bottomView(TreeNode root) {
        List<Integer> result = new ArrayList<>();
```

```
        if (root == null) return result;
        TreeMap<Integer, Integer> bottomViewMap = new TreeMap<>();
        Queue<Pair> queue = new LinkedList<>();
        queue.add(new Pair(root, 0));
        while (!queue.isEmpty()) {
            Pair current = queue.poll();
            TreeNode node = current.node;
            int hd = current.hd;
            bottomViewMap.put(hd, node.val);
            if (node.left != null) queue.add(new Pair(node.left, hd - 1));
            if (node.right != null) queue.add(new Pair(node.right, hd + 1));
        }
        result.addAll(bottomViewMap.values());
        return result;
    }
    public static void main(String[] args) {
        TreeNode root = new TreeNode(10);
        root.left = new TreeNode(5);
        root.right = new TreeNode(20);
        root.left.left = new TreeNode(3);
        root.left.right = new TreeNode(7);
        root.right.left = new TreeNode(15);
        root.right.right = new TreeNode(25);
        System.out.println(bottomView(root));
    }
}
```

Output


Microsoft Windows [Version 10.0.22631.4460]
(c) Microsoft Corporation. All rights reserved.
C:\Users\PRETHIKA>cd Documents
C:\Users\PRETHIKA\Documents>javac BSTBottomView.java
C:\Users\PRETHIKA\Documents>java BSTBottomView
[3, 5, 15, 20, 25]