Program -19

Aim:- . Write a Program to view a tree from left View

The **left view** of a binary tree is the set of nodes that are visible when the tree is viewed from the left side. In other words, it consists of the first node encountered at each level of the tree when traversing from top to bottom and left to right.

```
      1
     / \
    2   3
   / \   \
  4   5   6
 /
7
```

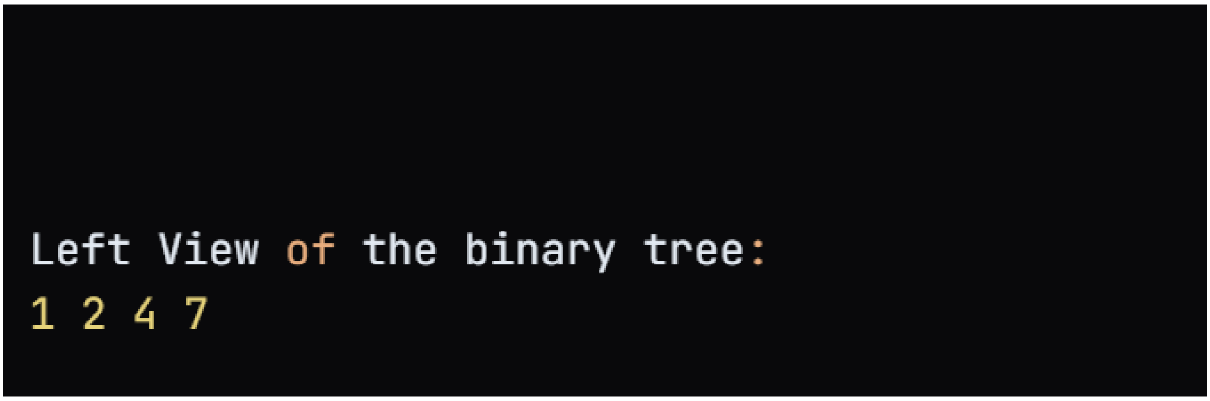
Left View Calculation:

- **Level 0**: The first node is **1**.
- **Level 1**: The first node is **2**.
- **Level 2**: The first node is **4**.
- **Level 3**: The first node is **7**.

Program:-

```
import java.util.LinkedList;

import java.util.Queue;
```

```
// Definition for a binary tree node
class TreeNode {
    int val;
    TreeNode left;
    TreeNode right;

    TreeNode(int x) {
        val = x;
        left = null;
        right = null;
    }
}

public class LeftViewOfBinaryTree {
    // Function to print the left view of the binary tree
    public void leftView(TreeNode root) {
        if (root == null) {
            return; // If the tree is empty, return
        }

        Queue<TreeNode> queue = new LinkedList<>();
        queue.add(root);

        while (!queue.isEmpty()) {
            int levelSize = queue.size(); // Number of nodes at the current level

            // Traverse all nodes of the current level
            for (int i = 0; i < levelSize; i++) {
                TreeNode currentNode = queue.poll();

                // Print the first node of this level
                if (i == 0) {
                    System.out.print(currentNode.val + " ");
                }

                // Add left child to the queue
                if (currentNode.left != null) {
                    queue.add(currentNode.left);
                }

                // Add right child to the queue
                if (currentNode.right != null) {
                    queue.add(currentNode.right);
                }
            }
        }
    }
    // Main method to test the left view function
    public static void main(String[] args) {
        // Create a sample binary tree
        TreeNode root = new TreeNode(1);
        root.left = new TreeNode(2);
        root.right = new TreeNode(3);
        root.left.left = new TreeNode(4);
        root.left.right = new TreeNode(5);
        root.right.right = new TreeNode(6);
        root.left.left.left = new TreeNode(7);

        LeftViewOfBinaryTree tree = new LeftViewOfBinaryTree();
        System.out.println("Left View of the binary tree:");
        tree.leftView(root); // Perform left view traversal
    }
}
```



```
Left View of the binary tree:
1 2 4 7
```