Experiment : 2.2

Student Name: Nishant Kumar Mehta **UID:** 21BCS-*3402*
Branch: CSE **Section/Group:** 21BCS-IOT-602B
Semester: 5th **Date:** 29/09/23
Subject Name: Advanced Programming LAB **Subject Code:** 21CSP-314

AIM:

To implement the concept of Tree.

OBJECTIVE:

1). Given a pointer to the root of a binary tree, print the top view of the binary tree.The tree as seen from the top the nodes, is called the top view of the tree.

2.) You are given a pointer to the root of a binary search tree and values to be inserted into the tree. Insert the values into their appropriate position in the binary search tree and return the root of the updated binary tree. You just have to complete the function.

CODE:

Code 1:

```
import java.util.*;
import java.io.*;
class Node {
    Node left;
    Node right;
    int data;
    Node(int data) {
        this.data = data;
        left = null;
        right = null;
    }
}
class Solution {
    public static void topView(Node root) {
      int[] arr = new int[501]; //node data
      int[] lvs = new int[501]; //node level
      SetNode(root, arr, lvs, 250, 1);
      boolean ok = false;
      for(int i=0;i<501;i++)
```

```
  {
      int a = arr[i];
      if (a == 0) continue;
      if (ok) System.out.print(" ");
      System.out.print(a);
      ok = true;
  }
}
static void SetNode(Node root, int[] arr, int[] lvs, int pos, int level)
{
    if (root == null) return;
    int lv = lvs[pos];
    if (lv==0 || level < lv) {
        arr[pos] = root.data;
        lvs[pos] = level;
    }
    level++;
    SetNode(root.left, arr, lvs, pos - 1, level);
    SetNode(root.right, arr, lvs, pos + 1, level);
}
public static Node insert(Node root, int data) {
    if(root == null) {
        return new Node(data);
    } else {
        Node cur;
        if(data <= root.data) {
            cur = insert(root.left, data);
            root.left = cur;
        } else {
            cur = insert(root.right, data);
            root.right = cur;
        }
        return root;
    }
}
public static void main(String[] args) {
    Scanner scan = new Scanner(System.in);
    int t = scan.nextInt();
    Node root = null;
    while(t-- > 0) {
        int data = scan.nextInt();
        root = insert(root, data);
    }
    scan.close();
    topView(root);
    }
}
```

Code 2

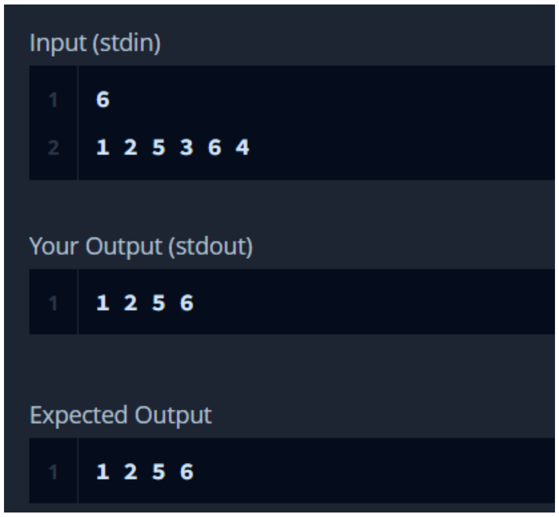
```
import java.util.*;
import java.io.*;
class Node {
    Node left;
    Node right;
    int data;
        Node(int data) {
        this.data = data;
        left = null;
        right = null;
    }
}
class Solution {
    public static void preOrder( Node root ) {
        if( root == null)
            return;
        System.out.print(root.data + " ");
        preOrder(root.left);
        preOrder(root.right);

    }
    public static Node insert(Node root,int data) {
Node head = root;

        if (root == null) {
            head = new Node(data); return head;
        }

        while (root != null) {
            if (data < root.data) {
                if (root.left == null) {
                    root.left = new Node(data); break;
                } else {
                    root = root.left;
                }
            } else {
                if (root.right == null) {
                    root.right = new Node(data); break;
                } else {
                    root = root.right;
                }
            }
        }
        return head;
    }
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        int t = scan.nextInt();
        Node root = null;
        while(t-- > 0) {
            int data = scan.nextInt();
            root = insert(root, data);
        }
        scan.close();
        preOrder(root);
    }
}
```

OUTPUT 1:



Input (stdin)

```
6
1 2 5 3 6 4
```

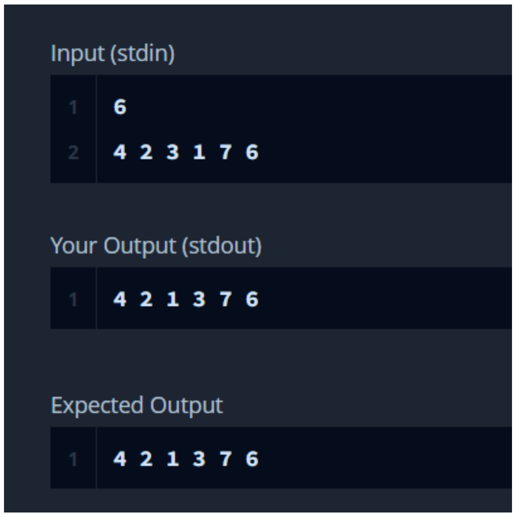
Your Output (stdout)

```
1 2 5 6
```

Expected Output

```
1 2 5 6
```

OUTPUT 2



Input (stdin)

```
6
4 2 3 1 7 6
```

Your Output (stdout)

```
4 2 1 3 7 6
```

Expected Output

```
4 2 1 3 7 6
```

LEARNING OUTCOMES:

1. *Understood the concept of Tree.*
2. *Understood the concept how to search in tree and perform different operations.*
3. *Learn about algorithm thinking*
4. *Learn about mathematical logic*