**Experiment 2.2**

**Competitive Coding Lab 5(Tree)**

**Student Name: Sahul Kr. Parida UID: 20BCS4919**

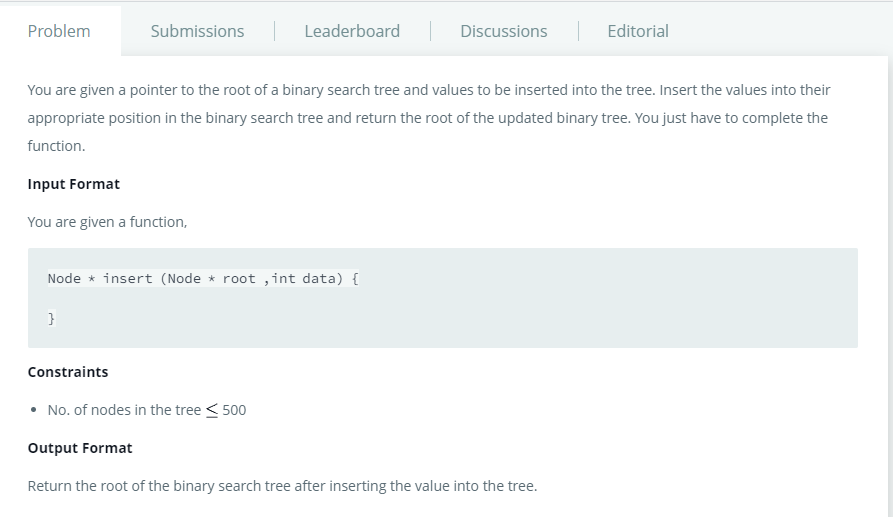
**Branch: CSE Section/Group: WM-904/B**

**Semester: 5th Date of Performance: 17/10/22**

**Subject Name: Competitive Coding(CC) Subject Code: 20CSP-314**

**PROBLEM STATEMENT 6.1: -**

<https://www.hackerrank.com/challenges/binary-search-tree-insertion/problem?isFullScreen=false>



**SOLUTION:**

Node \* insert(Node \* root, int data)

{

if(!root)

return new Node(data);

if(root->data < data)

root->right = insert(root->right, data);

else

root->left = insert(root->left, data);

return root;

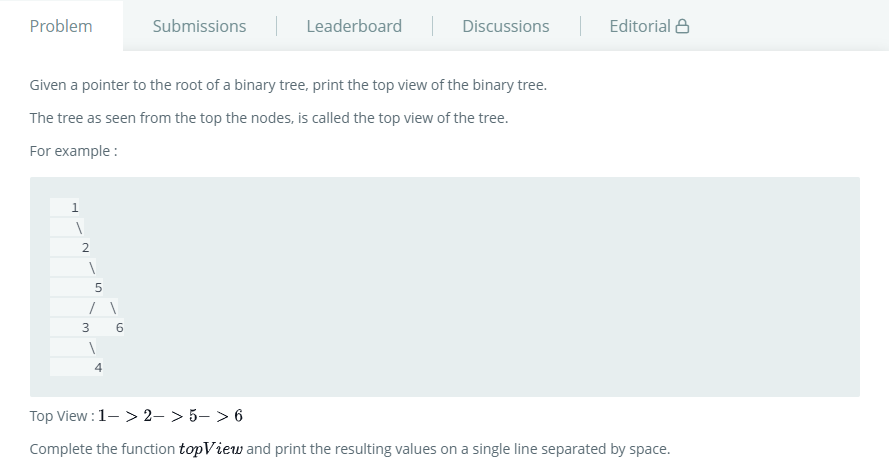
}

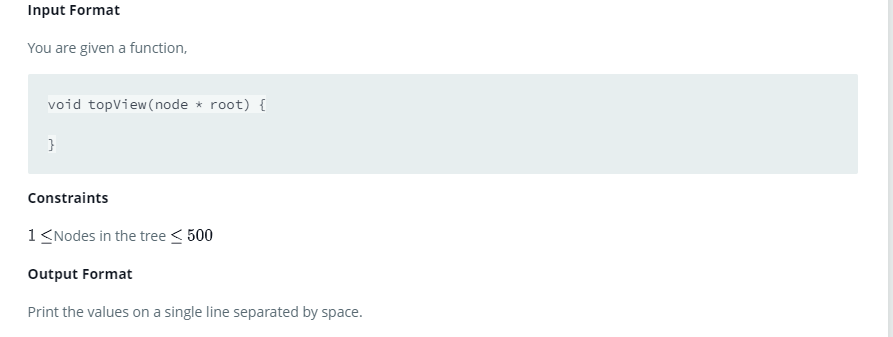
# TEST CASES:

# 

**PROBLEM STATEMENT 6.2: -**

<https://www.hackerrank.com/challenges/tree-top-view/problem?isFullScreen=false>





**SOLUTION:**

void topView(Node \* root)

    {

        queue<pair<int,Node\*>> q; q.push(make\_pair(0,root));

        map<int,Node\*> ans;

        for(auto i=q.front();!q.empty();q.pop(),i=q.front())

        {

            if(!i.second)

            continue;

            ans.insert(i);

            q.push(make\_pair(i.first+1,i.second->right));

            q.push(make\_pair(i.first-1,i.second->left));

        }

        for(auto i:ans) cout<<i.second->data<<" ";

    }

# TEST CASES:

# 