PRACTICE COMPETE JOBS LEADERBOARD Search swatantragoswam1

All Contests > OLT Code TEST-6 > Tree : Top View

Tree : Top View

Problem Submissions Leaderboard

You are given a pointer to the root of a binary tree. Print the top view of the binary tree.
Top view means when you look the tree from the top the nodes, what you will see will be called the top view of the tree. See the example below.
You only have to complete the function.
For example :

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

Top View : 1 -> 2 -> 5 -> 6

Input Format

You are given a function,

```
void topView(node * root) {

}
```

Constraints

1 $\leq$Nodes in the tree $\leq$ 500

Output Format

Print the values on a single line separated by space.

Sample Input

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

Sample Output

1 2 5 6

Explanation

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

From the top only nodes 1,2,5,6 will be visible.

Contest ends in an hour

Submissions: 244

Max Score: 10

Rate This Challenge:

☆☆☆☆☆

More

Current Buffer (saved locally, editable)

C++

```
#include<↔

using namespace std;

class Node {
    public:
        int data;
        Node *left;
        Node *right;
        Node(int d) {
            data = d;
            left = NULL;
            right = NULL;
        }
};

class Solution {
    public:
        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;
           }
        }
```

```
/*
class Node {
    public:
        int data;
        Node *left;
        Node *right;
        Node(int d) {
            data = d;
```

head

```
            left = NULL;
            right = NULL;
        }
};

*/
#include<bits/stdc++.h>
    void topView(Node * root) {
    map<int,int> m;
    queue<pair<struct Node*,int> > que;
    que.push(make_pair(root,0));

    while(!que.empty())
    {
        pair<struct Node*,int> cur=que.front();
        que.pop();
        if(m.find(cur.second)==m.end())
            m[cur.second]=cur.first->data;

        if(cur.first->left)
        que.push(make_pair(cur.first->left,cur.second-1));
        if(cur.first->right)
        que.push(make_pair(cur.first->right,cur.second+1));

    }
    auto p=m.begin();
    while(p!=m.end())
    {
        cout<<p->second<<" ";
        p++;
    }




```

```

  }
}; //End of Solution

int main() {

    Solution myTree;
    Node* root = NULL;

    int t;
    int data;

    std::cin >> t;

    while(t-- > 0) {
        std::cin >> data;
        root = myTree.insert(root, data);
    }

    myTree.topView(root);
    return 0;
}

```

Line: 47 Col: 1

Upload Code as File

□ Test against custom input

Run Code

Submit Code

Contest Calendar | Interview Prep | Blog | Scoring | Environment | FAQ | About Us | Support | Careers | Terms Of Service | Privacy Policy | Request a Feature