Tree : Top View

164 more points to get your next star!

Rank: 718746 | Points: 311/475

Problem Solving ★★★

Your Tree : Top View submission got 20.00 points. Share Post

You are now 164 points away from the 4th star for your problem solving badge.

Try the next challenge | Try a Random Challenge

Problem Submissions Leaderboard Editorial

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.

For example :

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

Complete the function ***topView*** and print the resulting values on a single line separated by space.

Input Format

You are given a function,

```
void topView(node * root) {

}
```

Constraints

$1 \leq$ Nodes in the tree ≤ 500

Output Format

Print the values on a single line separated by space.

Sample Input

Sample Output

```
1 2 5 6
```

Explanation

From the top, only nodes **1, 2, 5, 6** are visible.

Change Theme Language Python 3

```
class Node:…


"""
Node is defined as
self.left (the left child of the node)
self.right (the right child of the node)
self.info (the value of the node)
"""
from collections import deque

def topView(root):
    #Write your code here
    ans = []
    mpp = {}
    q = deque([(root, 0)])
    while q:
        node, line = q.popleft()
        if line not in mpp:
            mpp[line] = node.info
        if node.left:
            q.append((node.left, line-1))
        if node.right:
            q.append((node.right, line+1))
    for k,v in sorted(mpp.items()):
        ans.append(v)
    print(" ".join(map(str, ans)))
…
```

EMACS Line: 1 Col: 1

Upload Code as File Test against custom input Run Code Submit Code

You have earned 20.00 points!

You are now 164 points away from the 4th star for your problem solving badge.

40% 311/475

Congratulations

You solved this challenge. Would you like to challenge your friends?

Next Challenge

- Test case 0
- Test case 1
- Test case 2
- Test case 3
- Test case 4
- Test case 5

Compiler Message

Success

Input (stdin) Download

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

Expected Output Download

```
1 1 2 5 6
```