Description | Editorial | Solutions (4.7K) | Submissions

119. Pascal's Triangle II

Easy | 3.8K | 304

Companies

Given an integer `rowIndex`, return the `rowIndex`th (**0-indexed**) row of the **Pascal's triangle**.

In **Pascal's triangle**, each number is the sum of the two numbers directly above it as shown:

Example 1:

```
Input: rowIndex = 3
Output: [1,3,3,1]
```

Example 2:

```
Input: rowIndex = 0
Output: [1]
```

Example 3:

Java | Auto

```java
class Solution {
    public List<Integer> getRow(int rowIndex) {
        ArrayList<Integer> result = new ArrayList<Integer>();
        if (rowIndex < 0)
            return result;
        result.add(1);
        for (int i = 1; i <= rowIndex; i++)
        {
            for (int j = result.size() - 2; j >= 0; j--)
            {
                result.set(j + 1, result.get(j) + result.get(j + 1));
            }
            result.add(1);
        }
        return result;
    }
}
```

Testcase | Result

Accepted Runtime: 0 ms

Case 1 | Case 2 | Case 3

Input

rowIndex =
3

Output

[1,3,3,1]

Expected

[1,3,3,1]

Console | Run | Submit

Description | Editorial | Solutions (2.2K) | Submissions

1161. Maximum Level Sum of a Binary Tree

Hint

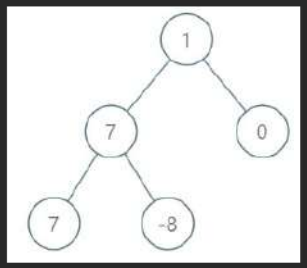
Medium | 2.7K | 83

Companies

Given the `root` of a binary tree, the level of its root is `1`, the level of its children is `2`, and so on.

Return the **smallest** level `x` such that the sum of all the values of nodes at level `x` is **maximal**.

Example 1:




```
Input: root = [1,7,0,7,-8,null,null]
Output: 2
Explanation:
Level 1 sum = 1.
Level 2 sum = 7 + 0 = 7.
Level 3 sum = 7 + -8 = -1.
So we return the level with the maximum sum which is level 2.
```

Example 2:

```
Input: root =
[989,null,10250,98693,-89388,null,null,null,-32127]
Output: 2
```

Java | Auto

```
14  * }
15  */
16 class Solution {
17     // tc -> n, sc-> n
18     public int maxLevelSum(TreeNode root) {
19         int minLevel = 1, maxVal = Integer.MIN_VALUE;
20
21         Queue<TreeNode> q = new LinkedList<>();
22         q.offer(root);
23
24         int level = 1;
25
26         while(!q.isEmpty()){
27
28             int size = q.size();
29
30             int sum = 0;
```

Testcase | Result

Accepted Runtime: 0 ms

Case 1 | Case 2

Input

root =
[1,7,0,7,-8,null,null]

Output

2

Expected

2

Console | Run | Submit