

Description

Editorial

Solutions (9.7K)

Submissions

118. Pascal's Triangle

Easy

10.5K

339



Companies

Given an integer `numRows`, return the first `numRows` of **Pascal's triangle**.

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



Example 1:

Input: `numRows = 5`

Output: `[[1], [1, 1], [1, 2, 1], [1, 3, 3, 1], [1, 4, 6, 4, 1]]`

Example 2:

Input: `numRows = 1`

Output: `[[1]]`

Constraints:

Java

Auto

```
1 class Solution {
2
3     public static int nCr(int n, int r) {
4         long res = 1;
5         // calculating nCr:
6         for (int i = 0; i < r; i++) {
7             res = res * (n - i);
8             res = res / (i + 1);
9         }
10        return (int) res;
11    }
12
13    public List < List < Integer >> generate(int n) {
14
15        List < List < Integer >> ans = new ArrayList < > ();
16
17        // Store the entire Pascal's triangle:
18        for (int row = 1; row <= n; row++) {
19            List < Integer > templst = new ArrayList < > (); // temporary list
20            for (int col = 1; col <= row; col++) {
```

Testcase

Result

Accepted

Runtime: 0 ms

• Case 1

• Case 2

Input

`numRows =``5`

Output

`[[1], [1, 1], [1, 2, 1], [1, 3, 3, 1], [1, 4, 6, 4, 1]]`

Expected

Console



Run

Submit

863. All Nodes Distance K in Binary Tree

Medium

9.6K

184

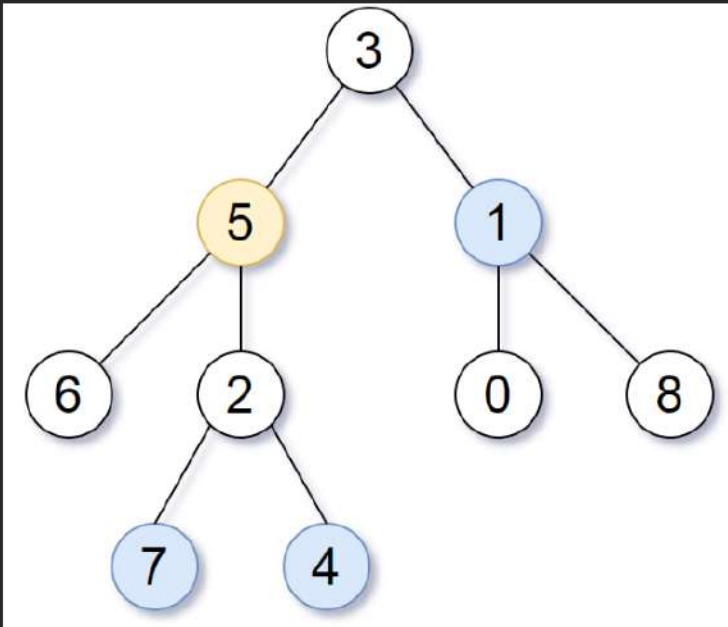


Companies

Given the `root` of a binary tree, the value of a target node `target`, and an integer `k`, return an array of the values of all nodes that have a distance `k` from the target node.

You can return the answer in **any order**.

Example 1:



Input: root = [3,5,1,6,2,0,8,null,null,7,4], target = 5, k = 2

Output: [7,4,1]

```

10 class Solution {
11     public List<Integer> distanceK(TreeNode root, TreeNode target, int k) {
12         List<Integer> ans = new ArrayList<>();
13         Map<Integer, TreeNode> parent = new HashMap<>();
14         Queue<TreeNode> queue = new LinkedList<>();
15         queue.offer(root);
16
17         while (!queue.isEmpty()) {
18             int size = queue.size();
19             for (int i = 0; i < size; i++) {
20                 TreeNode top = queue.poll();
21
22                 if (top.left != null) {
23                     parent.put(top.left.val, top);
24                     queue.offer(top.left);
25                 }
26
27                 if (top.right != null) {
28                     parent.put(top.right.val, top);
29                     queue.offer(top.right);
30                 }
31             }
32         }
33     }
34 }
  
```

Testcase Result

Accepted Runtime: 14 ms

Case 1 Case 2

Input

[3,5,1,6,2,0,8,null,null,7,4]

5

2

Output

Console



Run

Submit