5474. Number of Good Leaf Nodes Pairs

My Submissions (/contest/weekly-contest-199/problems/number-of-good-leaf-nodes-pairs/submissions/)

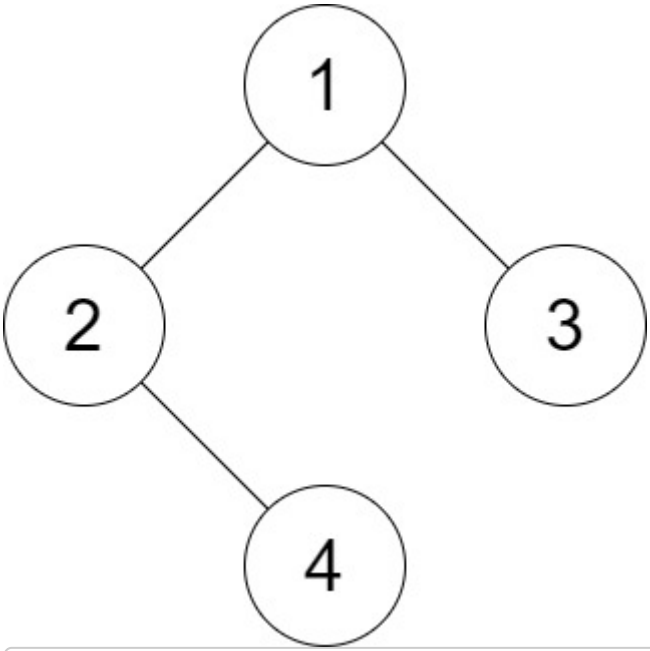
Back to Contest (/contest/weekly-contest-199/)

Given the `root` of a binary tree and an integer `distance`. A pair of two different **leaf** nodes of a binary tree is said to be good if the length of **the shortest path** between them is less than or equal to `distance`.

Return *the number of good leaf node pairs* in the tree.

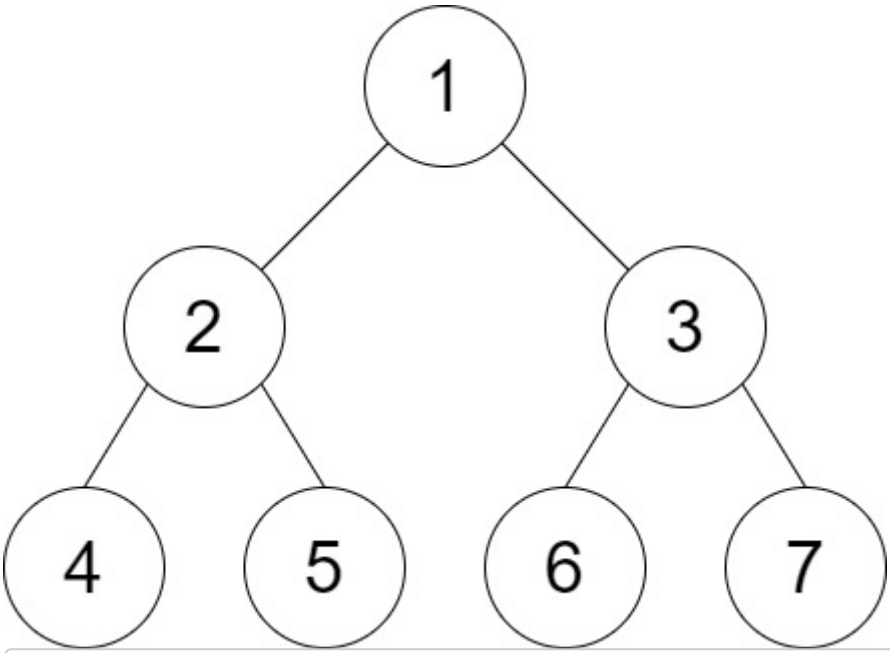
User Accepted:	13
User Tried:	15
Total Accepted:	13
Total Submissions:	15
Difficulty:	Medium

Example 1:




```
Input: root = [1,2,3,null,4], distance = 3
Output: 1
Explanation: The leaf nodes of the tree are 3 and 4 and the length of the shortest path be
```

Example 2:




```
Input: root = [1,2,3,4,5,6,7], distance = 3
Output: 2
Explanation: The good pairs are [4,5] and [6,7] with shortest path = 2. The pair [4,6] is
```

Example 3:

```
Input: root = [7,1,4,6,null,5,3,null,null,null,null,null,2], distance = 3
Output: 1
Explanation: The only good pair is [2,5].
```

Example 4:

```
Input: root = [100], distance = 1
Output: 0
```

Example 5:

```
Input: root = [1,1,1], distance = 2
Output: 1
```

Constraints:

- The number of nodes in the `tree` is in the range `[1, 2^10]`.
- Each node's value is between `[1, 100]`.
- `1 <= distance <= 10`

JavaScript

```
/**
 * Definition for a binary tree node.
```

```
 * function TreeNode(val, left, right) {
 *     this.val = (val===undefined ? 0 : val)
 *     this.left = (left===undefined ? null : left)
 *     this.right = (right===undefined ? null : right)
 * }
 */
/**
 * @param {TreeNode} root
 * @param {number} distance
 * @return {number}
 */
var countPairs = function(root, distance) {

};
```

Custom Testcase | Use Example Testcases

Run | Submit