

# Problem 1530: Number of Good Leaf Nodes Pairs

## Problem Information

**Difficulty:** Medium

**Acceptance Rate:** 71.78%

**Paid Only:** No

**Tags:** Tree, Depth-First Search, Binary Tree

## Problem Description

You are 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.

**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 between them is 3. This is the only good pair.

**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 not good because the length of their shortest path between them is 4.

**Example 3:**

**\*\*Input:\*\*** root = [7,1,4,6,null,5,3,null,null,null,null,2], distance = 3 **\*\*Output:\*\*** 1

**\*\*Explanation:\*\*** The only good pair is [2,5].

**\*\*Constraints:\*\***

\* The number of nodes in the `tree` is in the range `[1, 210].` \* `1 <= Node.val <= 100` \* `1 <= distance <= 10`

## Code Snippets

**C++:**

```
/**  
 * Definition for a binary tree node.  
 * struct TreeNode {  
 *     int val;  
 *     TreeNode *left;  
 *     TreeNode *right;  
 *     TreeNode() : val(0), left(nullptr), right(nullptr) {}  
 *     TreeNode(int x) : val(x), left(nullptr), right(nullptr) {}  
 *     TreeNode(int x, TreeNode *left, TreeNode *right) : val(x), left(left),  
 *     right(right) {}  
 * };  
 */  
class Solution {  
public:  
    int countPairs(TreeNode* root, int distance) {  
  
    }  
};
```

**Java:**

```
/**  
 * Definition for a binary tree node.  
 * public class TreeNode {  
 *     int val;  
 *     TreeNode left;  
 *     TreeNode right;  
 *     TreeNode() {}  
 *     TreeNode(int val) { this.val = val; }  
 }
```

```
* TreeNode(int val, TreeNode left, TreeNode right) {
*     this.val = val;
*     this.left = left;
*     this.right = right;
* }
*
class Solution {
    public int countPairs(TreeNode root, int distance) {
        ...
    }
}
```

### Python3:

```
# Definition for a binary tree node.
# class TreeNode:
#     def __init__(self, val=0, left=None, right=None):
#         self.val = val
#         self.left = left
#         self.right = right
class Solution:
    def countPairs(self, root: Optional[TreeNode], distance: int) -> int:
```