

****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:

```