Problem 863: All Nodes Distance K in Binary Tree

Problem Information

Difficulty: Medium
Acceptance Rate: 67.14%
Paid Only: No
Tags: Hash Table, Tree, Depth-First Search, Breadth-First Search, Binary Tree

Problem Description

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] Explanation: The nodes that are a distance 2 from the target node (with value 5) have values 7, 4, and 1.

Example 2:

Input: root = [1], target = 1, k = 3 **Output:** []

Constraints:

* The number of nodes in the tree is in the range `[1, 500]`. * `0 <= Node.val <= 500` * All the values `Node.val` are **unique**. * `target` is the value of one of the nodes in the tree. * `0 <= k <= 1000`

Code Snippets

C++:

```
/**
* Definition for a binary tree node.
* struct TreeNode {
* int val;
* TreeNode *left;
* TreeNode *right;
* TreeNode(int x) : val(x), left(NULL), right(NULL) {}
* };
*/
class Solution {
public:
vector<int> distanceK(TreeNode* root, TreeNode* target, int k) {

}
};
```

Java:

```
/**
* Definition for a binary tree node.
* public class TreeNode {
* int val;
* TreeNode left;
* TreeNode right;
* TreeNode(int x) { val = x; }
* }
*/
class Solution {
public List<Integer> distanceK(TreeNode root, TreeNode target, int k) {

}
}
```

Python3:

```
# Definition for a binary tree node.
# class TreeNode:
# def __init__(self, x):
# self.val = x
# self.left = None
# self.right = None
```

```
class Solution:
def distanceK(self, root: TreeNode, target: TreeNode, k: int) -> List[int]:
```