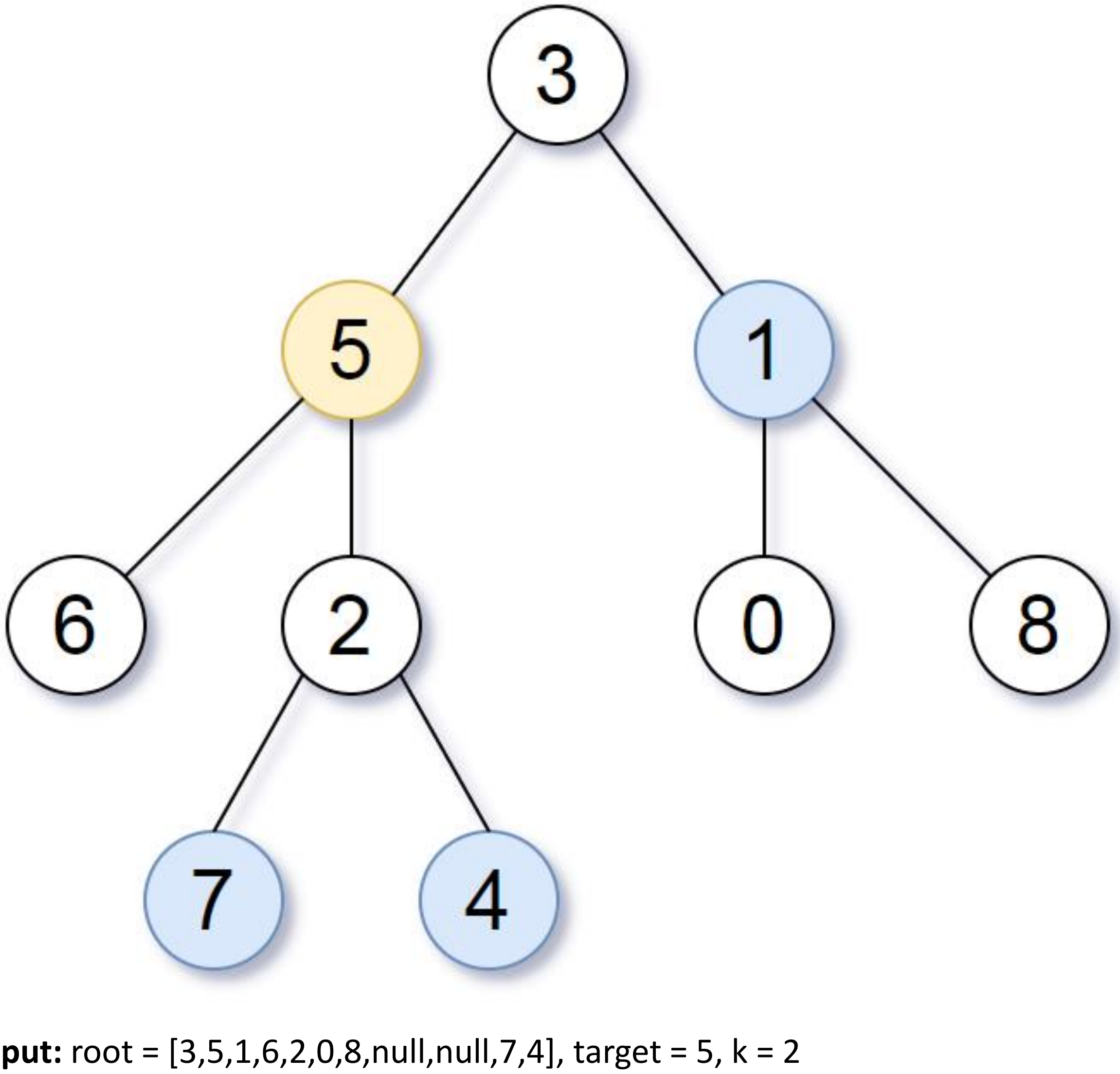
863. All Nodes Distance K in Binary Tree

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: []

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

class Solution(object):
    def distanceK(self, root, target, k):
        def helper(node, parent):
            if not node:
                return

            node.parent = parent
            helper(node.left,node)
            helper(node.right,node)

        helper(root,None)
        ans = []
        seen = set()

        def trav(node,dist):
            if not node or node in seen or dist>k:
                return
            seen.add(node)
            if dist == k:
                ans.append(node.val)
                return
            if dist+1 <= k:
                trav(node.parent,dist+1)
                trav(node.left,dist+1)
                trav(node.right,dist+1)

        trav(target,0)
        return ans
```