Problem 742: Closest Leaf in a Binary Tree

Problem Information

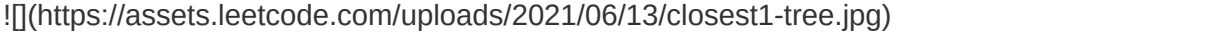
Difficulty: Medium
Acceptance Rate: 47.29%
Paid Only: Yes
Tags: Tree, Depth-First Search, Breadth-First Search, Binary Tree

Problem Description

Given the `root` of a binary tree where every node has **a unique value** and a target integer `k`, return _the value of the**nearest leaf node** to the target _`k` _in the tree_.

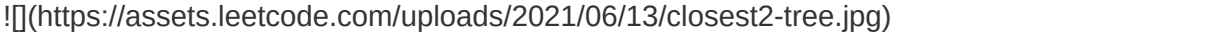
Nearest to a leaf means the least number of edges traveled on the binary tree to reach any leaf of the tree. Also, a node is called a leaf if it has no children.

Example 1:



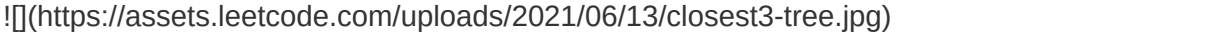
Input: root = [1,3,2], k = 1 **Output:** 2 **Explanation:** Either 2 or 3 is the nearest leaf node to the target of 1.

Example 2:



Input: root = [1], k = 1 **Output:** 1 **Explanation:** The nearest leaf node is the root node itself.

Example 3:



Input: root = [1,2,3,4,null,null,null,5,null,6], k = 2 **Output:** 3 **Explanation:** The leaf node with value 3 (and not the leaf node with value 6) is nearest to the node with value 2.

Constraints:

* The number of nodes in the tree is in the range `[1, 1000]`. * `1 <= Node.val <= 1000` * All the values of the tree are **unique**. * There exist some node in the tree where `Node.val == k`.

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 findClosestLeaf(TreeNode* root, int k) {

}
};
```

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 findClosestLeaf(TreeNode root, int k) {

}
}
```

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 findClosestLeaf(self, root: Optional[TreeNode], k: int) -> int:
```