Problem 549: Binary Tree Longest Consecutive Sequence II

Problem Information

Difficulty: Medium
Acceptance Rate: 49.60%
Paid Only: Yes
Tags: Tree, Depth-First Search, Binary Tree

Problem Description

Given the `root` of a binary tree, return _the length of the longest consecutive path in the tree_.

A consecutive path is a path where the values of the consecutive nodes in the path differ by one. This path can be either increasing or decreasing.

* For example, `[1,2,3,4]` and `[4,3,2,1]` are both considered valid, but the path `[1,2,4,3]` is not valid.

On the other hand, the path can be in the child-Parent-child order, where not necessarily be parent-child order.

Example 1:

Input: root = [1,2,3] **Output:** 2 **Explanation:** The longest consecutive path is [1, 2] or [2, 1].

Example 2:

Input: root = [2,1,3] **Output:** 3 **Explanation:** The longest consecutive path is [1, 2, 3] or [3, 2, 1].

Constraints:

* The number of nodes in the tree is in the range `[1, 3 * 104]`. * `-3 * 104 <= Node.val <= 3 * 104`

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 longestConsecutive(TreeNode* root) {

}
};
```

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 longestConsecutive(TreeNode root) {

}
}
```

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