Problem 298: Binary Tree Longest Consecutive Sequence

Problem Information

Difficulty: Medium
Acceptance Rate: 54.28%
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 sequence path**_.

A **consecutive sequence path** is a path where the values **increase by one** along the path.

Note that the path can start **at any node** in the tree, and you cannot go from a node to its parent in the path.

Example 1:

Input: root = [1,null,3,2,4,null,null,null,5] **Output:** 3 **Explanation:** Longest consecutive sequence path is 3-4-5, so return 3.

Example 2:

Input: root = [2,null,3,2,null,1] **Output:** 2 **Explanation:** Longest consecutive sequence path is 2-3, not 3-2-1, so return 2.

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:
```