

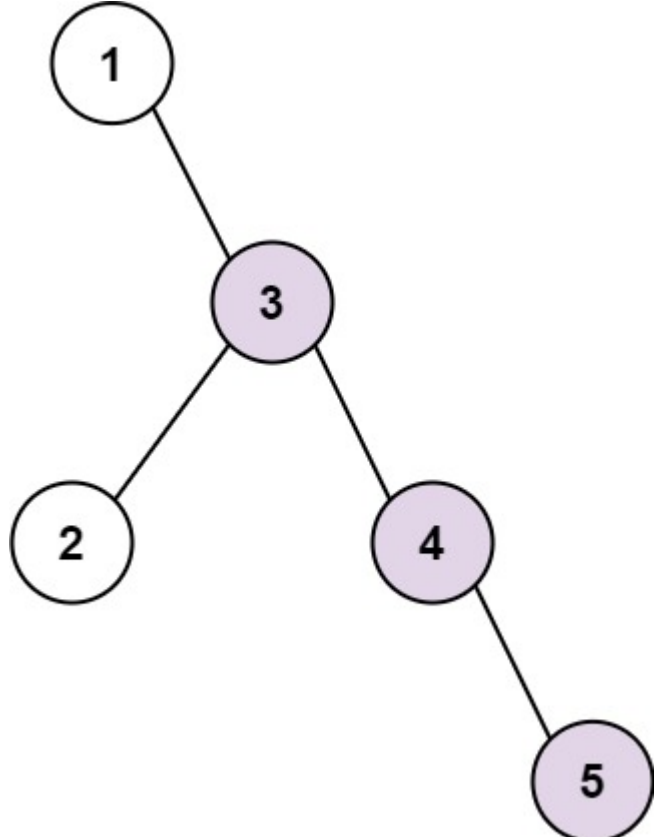
298. Binary Tree Longest Consecutive Sequence

Medium👍 791🗨 195❤️ Add to List🔗 Share

Given the `root` of a binary tree, return the length of the longest consecutive sequence path.

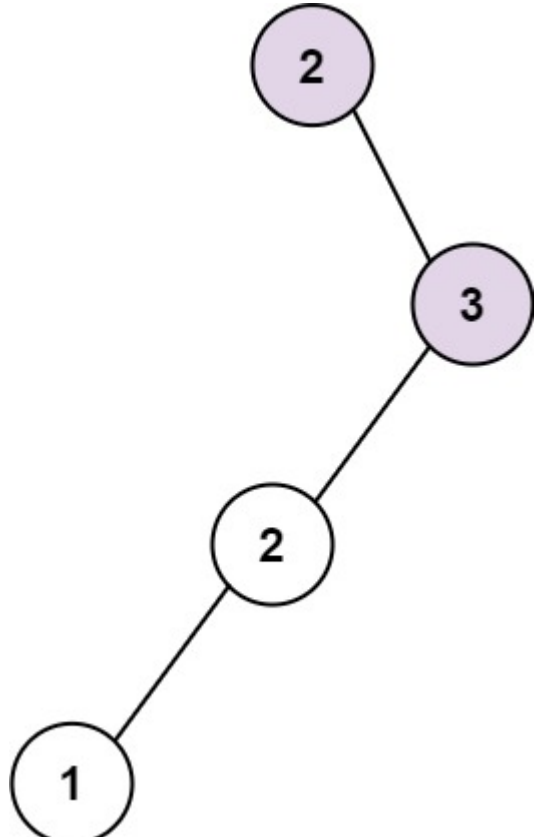
The path refers to any sequence of nodes from some starting node to any node in the tree along the parent-child connections. The longest consecutive path needs to be from parent to child (cannot be the reverse).

Example 1:



**Input:** root = [1,null,3,2,4,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  $[1, 3 \times 10^4]$ .
- $-3 \times 10^4 \leq \text{Node.val} \leq 3 \times 10^4$

Accepted108,988Submissions218,568

Seen this question in a real interview before?

YesNo

Companies🔍

0 ~ 6 months6 months ~ 1 year1 year ~ 2 years

Facebook6ByteDance3Google2Pinterest2tiktok2

Related Topics

TreeDepth-First SearchBinary Tree

Similar Questions

Longest Consecutive SequenceMedium

Binary Tree Longest Consecutive Sequence IIMedium

```
1  /**
2   * Definition for a binary tree node.
3   * public class TreeNode {
4   *     int val;
5   *     TreeNode left;
6   *     TreeNode right;
7   *     TreeNode() {}
8   *     TreeNode(int val) { this.val = val; }
9   *     TreeNode(int val, TreeNode left, TreeNode right) {
10  *         this.val = val;
11  *         this.left = left;
12  *         this.right = right;
13  *     }
14  * }
15  */
16
17  class Solution {
18  private int maxLength = 0;
19  public int longestConsecutive(TreeNode root) {
20      dfs(root);
21      return maxLength;
22  }
23
24  private int dfs(TreeNode p) {
25      if (p == null) return 0;
26      int L = dfs(p.left) + 1;
27      int R = dfs(p.right) + 1;
28      if (p.left != null && p.val + 1 != p.left.val) {
29          L = 1;
30      }
31      if (p.right != null && p.val + 1 != p.right.val) {
32          R = 1;
33      }
34      int length = Math.max(L, R);
35      maxLength = Math.max(maxLength, length);
36      return length;
37  }
```