

298. Binary Tree Longest Consecutive Sequence

Previous

Next

March 5, 2016

|

21.7K views



Average Rating: 4.71 (21 votes)

Given a binary tree, find 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 need to be from parent to child (cannot be the reverse).

Example 1:

Input:

```
      1
     / \
    3   4
   / \   \
  2  4   5
     \
      5
```

Output: 3

Explanation: Longest consecutive sequence path is 3-4-5, so return 3.

Example 2:

Input:

```
      2
     / \
    3   2
   /   \
  1     2
 /
1
```

Output: 2

Explanation: Longest consecutive sequence path is 2-3, not 3-2-1, so return 2.

Solution

Approach #1 (Top Down Depth-first Search) [Accepted]

Algorithm

A top down approach is similar to an in-order traversal. We use a variable `length` to store the current consecutive path length and pass it down the tree. As we traverse, we compare the current node with its parent node to determine if it is consecutive. If not, we reset the length.

```
private int maxLength = 0;
public int longestConsecutive(TreeNode root) {
    dfs(root, null, 0);
    return maxLength;
}

private void dfs(TreeNode p, TreeNode parent, int length) {
    if (p == null) return;
    length = (parent != null && p.val == parent.val + 1) ? length + 1 : 1;
    maxLength = Math.max(maxLength, length);
    dfs(p.left, p, length);
    dfs(p.right, p, length);
}
```

@lightmark presents a neat approach without storing the `maxLength` as a global variable.

```
public int longestConsecutive(TreeNode root) {
    return dfs(root, null, 0);
}

private int dfs(TreeNode p, TreeNode parent, int length) {
    if (p == null) return length;
    length = (parent != null && p.val == parent.val + 1) ? length + 1 : 1;
    return Math.max(length, Math.max(dfs(p.left, p, length),
                                     dfs(p.right, p, length)));
}
```

Complexity analysis

- Time complexity : $O(n)$. The time complexity is the same as an in-order traversal of a binary tree with n nodes.
- Space complexity : $O(n)$. The extra space comes from implicit stack space due to recursion. For a skewed binary tree, the recursion could go up to n levels deep.

Approach #2 (Bottom Up Depth-first Search) [Accepted]

Algorithm

The bottom-up approach is similar to a post-order traversal. We return the consecutive path length starting at current node to its parent. Then its parent can examine if its node value can be included in this consecutive path.

```
private int maxLength = 0;
public int longestConsecutive(TreeNode root) {
    dfs(root);
    return maxLength;
}

private int dfs(TreeNode p) {
    if (p == null) return 0;
    int L = dfs(p.left) + 1;
    int R = dfs(p.right) + 1;
    if (p.left != null && p.val + 1 != p.left.val) {
        L = 1;
    }
    if (p.right != null && p.val + 1 != p.right.val) {
        R = 1;
    }
    int length = Math.max(L, R);
    maxLength = Math.max(maxLength, length);
    return length;
}
```

Complexity analysis

- Time complexity : $O(n)$. The time complexity is the same as a post-order traversal in a binary tree, which is $O(n)$.
- Space complexity : $O(n)$. The extra space comes from implicit stack space due to recursion. For a skewed binary tree, the recursion could go up to n levels deep.


Rate this article: 

Previous


Next

Comments: 14


Sort By





Type comment here... (Markdown is supported)





 Preview


Post





better365  54  January 7, 2018 10:02 PM

Yup, the first one should be preorder traversal! But it is still a good solution!





11    Share  Reply






yunshengyao  53  October 15, 2018 2:26 AM

They difference is not top-down and bottom-up. To me these are the terms used in dynamic programming, where recursive solutions are usually called top-down and the iterative solutions are bottom-up.

In fact, the first solution is a tail-recursion, and the second solution is a non-tail recursion.

8    Share  Reply







vigjadel  30  August 19, 2016 11:04 PM


Hi,



Isn't the first approach called the preorder traversal instead of inorder traversal ?

Thanks,





Vignesh


5    Share  Reply





snapfinger  43  January 6, 2017 4:12 AM





Agree, the first approach should be called preorder traversal manner instead of inorder


3    Share  Reply





ZiyueYang  1  January 14, 2019 6:08 PM





Hi, seems the standard program gives 1 when input is a nullptr. It should be 0.

1    Share  Reply






cindy0092  52  February 18, 2019 5:54 AM

why is tree traversal time not a $O(N\log N)$ time?





0    Share  Reply

SHOW 2 REPLIES






LearningMind  174  December 25, 2018 8:04 PM

Why Space Complexity is "n" it should be "log n" where n is number of nodes?





0    Share  Reply


SHOW 1 REPLY





liush100  0  October 23, 2016 10:06 PM





@vigjadel, I think there is a typo. The first approach should be preorder traversal


0    Share  Reply





xurc238  0  October 16, 2016 9:57 PM





Beautiful Solution!

0    Share  Reply



lijingyabeyond  17  2 days ago

Am I the only one that can't understand the question?

0    Share  Reply

