298. Binary Tree Longest Consecutive Sequence (/problems/binary-tree-longest-consecutive-sequence/)

March 5, 2016 | 5.4K views

298. Binary Tree Longest Consecutive Sequence

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).

For example,

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

Longest consecutive sequence path is `3–4–5`, so return `3`.

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

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);
}
```

Articles >

@lightmark presents a neat approach (https://leetcode.com/discuss/66486/short-and-simple-c-solution) 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.

Articles ›

Rate this article: **(/ratings/107/14/?return=/articles/binary-tree-longest-consecutive-sequence/) (/ratings/107/14/?r**

Previous (/articles/3sum-smaller/) Next (/articles/walls-and-gates/)

Join the conversation

Signed in as **tan7**.

Post a Reply

snapfinger commented last year

Agree, the first approach should be called preorder traversal manner instead of inorder (https://discuss.leetcode.com/user/snapfinger)

L

liush100 commented last year

@vigjadel (https://discuss.leetcode.com/uid/41719), I think there is a typo. The first approach should be preorder traversal (https://discuss.leetcode.com/user/liush100)

X

xurc238 commented last year

Beautiful Solution! (https://discuss.leetcode.com/user/xurc238)

V

vigjadel commented last year

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

Thanks,
Vignesh
(https://discuss.leetcode.com/user/vigjadel)

View original thread (https://discuss.leetcode.com/topic/31)