Previous Next

563. Binary Tree Tilt

April 22, 2017 | 36.9K views

Given a binary tree, return the tilt of the **whole tree**.

The tilt of a **tree node** is defined as the **absolute difference** between the sum of all left subtree node values and the sum of all right subtree node values. Null node has tilt 0.

The tilt of the **whole tree** is defined as the sum of all nodes' tilt.

Example:

```
Input:
         1
       /   \
      2     3
Output: 1
Explanation:
Tilt of node 2 : 0
Tilt of node 3 : 0
Tilt of node 1 : |2-3| = 1
Tilt of binary tree : 0 + 0 + 1 = 1
```

Note:

1. The sum of node values in any subtree won't exceed the range of 32-bit integer.
2. All the tilt values won't exceed the range of 32-bit integer.

Solution

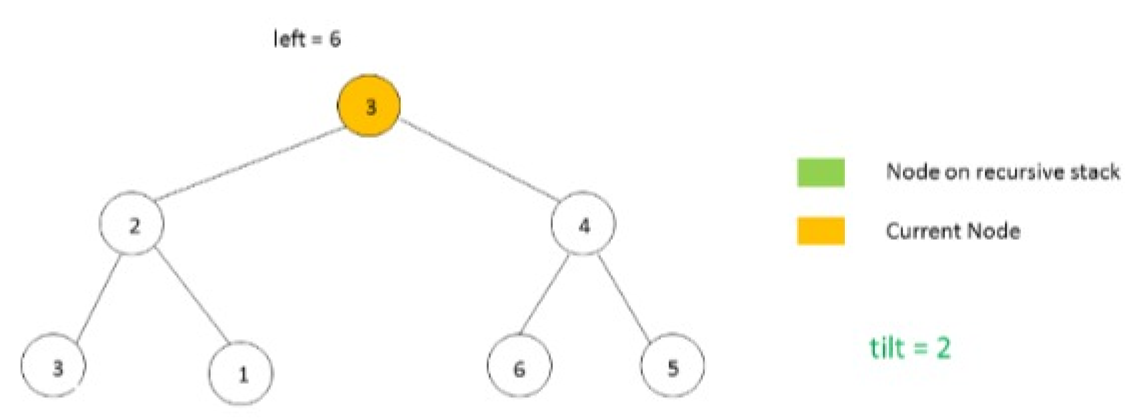
Approach 1: Using Recursion

Algorithm

From the problem statement, it is clear that we need to find the tilt value at every node of the given tree and add up all the tilt values to obtain the final result. To find the tilt value at any node, we need to subtract the sum of all the nodes in its left subtree and the sum of all the nodes in its right subtree.

Thus, to find the solution, we make use of a recursive function `traverse` which when called from any node, returns the sum of the nodes below the current node including itself. With the help of such sum values for the right and left subchild of any node, we can directly obtain the tilt value corresponding to that node.

The below animation depicts how the value passing and tilt calculation:



9 / 14

Java

```java
/**
 * Definition for a binary tree node.
 * public class TreeNode {
 *     int val;
 *     TreeNode left;
 *     TreeNode right;
 *     TreeNode(int x) { val = x; }
 * }
 */
public class Solution {
    int tilt = 0;
    public int findTilt(TreeNode root) {
        traverse(root);
        return tilt;
    }
    public int traverse(TreeNode root)
    {
        if (root == null )
            return 0;
        int left = traverse(root.left);
        int right = traverse(root.right);
        tilt += Math.abs(left-right);
        return left + right + root.val;
    }
}
```

Complexity Analysis

- Time complexity : $O(n)$. where n is the number of nodes. Each node is visited once.
- Space complexity : $O(n)$. In worst case when the tree is skewed depth of tree will be n. In average case depth will be $\log n$.

Rate this article:

Previous Next

Comments: 32 Sort By

Type comment here... (Markdown is supported)

Preview Post

theyselim ★ 87 November 4, 2017 10:28 PM

Something is not correct either in the question or the answer. The question states that the tilt is: "The tilt of a tree node is defined as the absolute difference between the sum of all left subtree node values and the sum of all right subtree node values". With that definition, tilt of root will be `|sum(left) - sum(right)|`. However this answer states that the root's tilt is `root.val + |sum(left) - sum(right)|`. Please advise if I am missing something here.

Read More

86 Share Reply

SHOW 2 REPLIES

McMasterAL ★ 44 June 29, 2018 8:17 PM

Obviously, the solution and the question are not consistent. For the example above:[3, 2, 4, 3, 1, 6, 5] leftSum = 6, rightSum = 15, result should be: abs(6-15) = 9. Not abs(6-15) + 3 = 12.
Look at the definition: Definition: "The tilt of a tree node is defined as the absolute difference between the sum of all left subtree node values and the sum of all right subtree node values. "!

29 Share Reply

SHOW 1 REPLY

arpitp ★ 22 September 8, 2018 6:12 PM Report

Some better example explained to avoid confusion ::

Input :
4
/

Read More

24 Share Reply

SHOW 2 REPLIES

joeandblack ★ 11 February 8, 2018 11:09 AM

Everybody knows the recursive - the trick is in the return statement: (left+right+val) -- it's not the tilt but return value. Actually it covers every situations of a given note - it works. But nobody explains or even mention it in their solution.

11 Share Reply

SHOW 1 REPLY

_game_on_ ★ 31 April 9, 2018 9:21 PM

Doesn't work for [1,2].

10 Share Reply

zoran.lovric ★ 43 March 13, 2018 7:38 AM

Ugly solution with the use of helper field. Doesn't support multiple calls, not thread safe etc. It seems number of lines is the only criteria of success on leetcode instead of readability and maintainability

9 Share Reply

dilrajio ★ 3 June 3, 2020 2:24 AM

THIS MAKES NO SENSE AND THE EXAMPLES ARE SHIT

2 Share Reply

charlieanna ★ 127 November 20, 2017 5:21 AM

Did the exact thing in python. Hurrah

1 Share Reply

jjtnx2 ★ 2 June 28, 2019 8:05 AM

Why can't we do this solution by preorder traversal? from top-down recursion I mean

0 Share Reply

jerome ★ 0 December 1, 2017 1:56 PM

JS without global variable:

var findTilt = function(root) {
return treeTilt(root, { val: 0 });
};

Read More

0 Share Reply

‹ 1 2 3 4 ›