437 Path Sum III

2018年4月11日 13:25

You are given a binary tree in which each node contains an integer value.
Find the number of paths that sum to a given value.
The path does not need to start or end at the root or a leaf, but it must go downwards (traveling only from parent nodes to child nodes).
The tree has no more than 1,000 nodes and the values are in the range -1,000,000 to 1,000,000.

Example:

root = [10,5,-3,3,2,null,11,3,-2,null,1], sum = 8

10
/ \
5 -3
**/\ **
3 2 11
**/\ **
3 -2 **1**

Return 3. The paths that sum to 8 are:

1. 5 -> 3
2. 5 -> 2 -> 1
3. -3 -> 11

来自 <https://leetcode.com/problems/path-sum-iii/description/>
给定一个二叉树，二叉树的每个节点含有一个整数。
找出路径和等于给定数的路径总数。
路径不需要从根节点开始，也不需要在叶节点结束，当路径方向必须是向下的（只从父节点到子节点）。
二叉树不超过1000个节点，节点的整数值的范围是[-1000000,1000000]。

Solution for Python3:

```python
# Definition for a binary tree node.
# class TreeNode:
#     def __init__(self, x):
#         self.val = x
#         self.left = None
#         self.right = None

class Solution1:
    def pathSum(self, root, sum):
        """
        :type root: TreeNode
        :type sum: int
        :rtype: int
        """
        if not root:
            return 0
        return self.sumUp(root, 0, sum) + self.pathSum(root.left, sum) + self.pathSum(root.right, sum)
    def sumUp(self, root, pre, sum):
        if not root:
            return 0
        current = pre + root.val
        return (current == sum) + self.sumUp(root.left, current, sum) + self.sumUp(root.right, current, sum)

class Solution2:
    def pathSum(self, root, sum):
        """
        :type root: TreeNode
        :type sum: int
        :rtype: int
        """
        if not root:
            return 0
        return self.sumUp(root, sum) + self.pathSum(root.left, sum) + self.pathSum(root.right, sum)
    def sumUp(self, root, sum):
        if not root:
            return 0
        return (root.val == sum) + self.sumUp(root.left, sum - root.val) + self.sumUp(root.right, sum - root.val)
```

Solution for C++:

```cpp
/**
 * Definition for a binary tree node.
 * struct TreeNode {
```

```
 *     int val;
 *     TreeNode *left;
 *     TreeNode *right;
 *     TreeNode(int x) : val(x), left(NULL), right(NULL) {}
 * };
 */
class Solution {
public:
    int pathSum(TreeNode* root, int sum) {
        if (!root)
            return 0;
        return sumUp(root, 0, sum) + pathSum(root->left, sum) + pathSum(root->right, sum);
    }
private:
    int sumUp(TreeNode* root, int pre, int sum) {
        if (!root)
            return 0;
        int current = pre + root->val;
        return (current == sum) + sumUp(root->left, current, sum) + sumUp(root->right, current, sum);
    }
};
```