404 Sum of Left Leaves

2018年4月10日 19:57

Question:

Find the sum of all left leaves in a given binary tree.

Example:

```
   3
  / \
 9  20
   /  \
  15   7
```

There are two left leaves in the binary tree, with values **9** and **15** respectively. Return **24**.

来自 <https://leetcode.com/problems/sum-of-left-leaves/description/>

计算给定二叉树的所有左叶子之和。

来自 <https://leetcode-cn.com/problems/sum-of-left-leaves/description/>

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 sumOfLeftLeaves(self, root, isleft = False):
        """
        :type root: TreeNode
        :rtype: int
        """
        if not root:
            return 0
        if not root.left and not root.right:
            return root.val if isleft else 0
        return self.sumOfLeftLeaves(root.left, True) + self.sumOfLeftLeaves(root.right, False);

class Solution2:
    def sumOfLeftLeaves(self, root, isleft = False):
        """
        :type root: TreeNode
        :rtype: int
        """
        if not root:
            return 0
        sum = 0
        from collections import deque
        deq = deque([root])
        while deq:
            node = deq.pop()
            if node.left:
                if not node.left.left and not node.left.right:
                    sum += node.left.val
                else:
                    deq.append(node.left)
            if node.right:
                if node.right.left or not node.right.right:
                    deq.append(node.right)
        return sum


class Solution3:
    def sumOfLeftLeaves(self, root, isleft = False):
        """
        :type root: TreeNode
        :rtype: int
        """
        if not root:
            return 0
        sum = 0
        if root.left and not root.left.left and not root.left.right:
            sum += root.left.val
        else:
            sum += self.sumOfLeftLeaves(root.left)
        sum += self.sumOfLeftLeaves(root.right)
        return sum
```

Solution for C++:

```
/**
 * Definition for a binary tree node.
 * struct TreeNode {
 *     int val;
 *     TreeNode *left;
 *     TreeNode *right;
 *     TreeNode(int x) : val(x), left(NULL), right(NULL) {}
 * };
 */
class Solution1 {
public:
    int sumOfLeftLeaves(TreeNode* root, bool isleft = false) {
        if (!root)
            return 0;
        if (!root->left && !root->right)
            return isleft ? root->val : 0;
        return sumOfLeftLeaves(root->left, true) + sumOfLeftLeaves(root->right, false);
    }
};

class Solution2 {
public:
    int sumOfLeftLeaves(TreeNode* root, bool isleft = false) {
        if (!root)
            return 0;
        int sum = 0;
        queue<TreeNode*> que;
        que.push(root);
        while (!que.empty()) {
            TreeNode* node = que.front();
            que.pop();
            if (node->left)
                if (node->left->left == NULL && node->left->right == NULL)
                    sum += node->left->val;
                else
                    que.push(node->left);
            if (node->right)
                if (node->right->left || node->right->right)
                    que.push(node->right);
        }
        return sum;
    }
};
```