

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:

```
1 # Definition for a binary tree node.
2 # class TreeNode:
3 #     def __init__(self, x):
4 #         self.val = x
5 #         self.left = None
6 #         self.right = None
7
8 class Solution1:
9     def sumOfLeftLeaves(self, root, isleft = False):
10         """
11         :type root: TreeNode
12         :rtype: int
13         """
14         if not root:
15             return 0
16         if not root.left and not root.right:
17             return root.val if isleft else 0
18         return self.sumOfLeftLeaves(root.left, True) + self.sumOfLeftLeaves(root.right, False);
19
20 class Solution2:
21     def sumOfLeftLeaves(self, root, isleft = False):
22         """
23         :type root: TreeNode
24         :rtype: int
25         """
26         if not root:
27             return 0
28         sum = 0
29         from collections import deque
30         deq = deque([root])
31         while deq:
32             node = deq.pop()
33             if node.left:
34                 if not node.left.left and not node.left.right:
35                     sum += node.left.val
36                 else:
37                     deq.append(node.left)
38             if node.right:
39                 if node.right.left or not node.right.right:
40                     deq.append(node.right)
41         return sum
42
43
44 class Solution3:
45     def sumOfLeftLeaves(self, root, isleft = False):
46         """
```

```

47         :type root: TreeNode
48         :rtype: int
49         """
50         if not root:
51             return 0
52         sum = 0
53         if root.left and not root.left.left and not root.left.right:
54             sum += root.left.val
55         else:
56             sum += self.sumOfLeftLeaves(root.left)
57         sum += self.sumOfLeftLeaves(root.right)
58         return sum

```

Solution for C++:

```

1  /**
2   * Definition for a binary tree node.
3   * struct TreeNode {
4   *     int val;
5   *     TreeNode *left;
6   *     TreeNode *right;
7   *     TreeNode(int x) : val(x), left(NULL), right(NULL) {}
8   * };
9   */
10 class Solution1 {
11 public:
12     int sumOfLeftLeaves(TreeNode* root, bool isleft = false) {
13         if (!root)
14             return 0;
15         if (!root->left && !root->right)
16             return isleft ? root->val : 0;
17         return sumOfLeftLeaves(root->left, true) + sumOfLeftLeaves(root->right, false);
18     }
19 };
20
21 class Solution2 {
22 public:
23     int sumOfLeftLeaves(TreeNode* root, bool isleft = false) {
24         if (!root)
25             return 0;
26         int sum = 0;
27         queue<TreeNode*> que;
28         que.push(root);
29         while (!que.empty()) {
30             TreeNode* node = que.front();
31             que.pop();
32             if (node->left)
33                 if (node->left->left == NULL && node->left->right == NULL)
34                     sum += node->left->val;
35                 else
36                     que.push(node->left);
37             if (node->right)
38                 if (node->right->left || node->right->right)
39                     que.push(node->right);
40         }
41         return sum;
42     }
43 };

```