

计算给定二叉树的所有左叶子之和。

示例:

```

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

在这个二叉树中，有两个左叶子，分别是 9 和 15，所以返回 24

分析，左子树之和，只需要判断，哪个节点是左子树就ok，那么遍历的时候，是左子树传递true，右子树传递false做标记，然后通过左右子树是否为NULL判断其是否为叶子节点，为叶子，就将自己的值传递上去，每一个节点接收自己左右子树的值，返回给自己上一层树。

代码

```
/**
 * 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 dfs(TreeNode* root, bool flag) {
        int num = 0;
        if (root == NULL) {
            return 0;
        }
        if (flag) {
            if (root->left == NULL && root->right == NULL) {
                return root->val;
            }
        }
        if (root->left != NULL) {
            num += dfs(root->left, true);
        }
        if (root->right != NULL) {
            num += dfs(root->right, false);
        }
        return num;
    }
    int sumOfLeftLeaves(TreeNode* root) {
```

```
        return dfs(root, false);  
    }  
};
```