# 题目

给定一个二叉树，计算整个树的坡度。

一个树的节点的坡度定义即为，该节点左子树的结点之和和右子树结点之和的差的绝对值。空结点的的坡度是0。

整个树的坡度就是其所有节点的坡度之和。

**示例：**

输入：

1

/ \

2 3

输出：1

**解释：**

结点 2 的坡度: 0

结点 3 的坡度: 0

结点 1 的坡度: |2-3| = 1

树的坡度 : 0 + 0 + 1 = 1

**提示：**

任何子树的结点的和不会超过 32 位整数的范围。

坡度的值不会超过 32 位整数的范围。

# 分析

## 方法一：递归法

/\*\*

\* 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 helper(TreeNode\* root,int &tilt)

{

if(root==NULL) return 0;

int left=helper(root->left,tilt);

int right=helper(root->right,tilt);

tilt+=abs(left-right);

return left+right+root->val;

}

int findTilt(TreeNode\* root) {

int tilt=0;

helper(root,tilt);

return tilt;

}

};