**563. Binary Tree Tilt**

Given a binary tree, return the tilt of the **whole tree**.

The tilt of a **tree node** is defined as the **absolute difference** between the sum of all left subtree node values and the sum of all right subtree node values. Null node has tilt 0.

The tilt of the **whole tree** is defined as the sum of all nodes' tilt.

**Example:**

**Input:**

1

/ \

2 3

**Output:** 1

**Explanation:**

Tilt of node 2 : 0

Tilt of node 3 : 0

Tilt of node 1 : |2-3| = 1

Tilt of binary tree : 0 + 0 + 1 = 1

**Note:**

1. The sum of node values in any subtree won't exceed the range of 32-bit integer.
2. All the tilt values won't exceed the range of 32-bit integer.

给定一个二叉树，返回整棵树的倾斜程度。

一个节点的倾斜程度定义：左子树的所有节点和，与右子树所有节点和的绝对值差。空节点的倾斜程度定义为0。

整棵树的倾斜程度定义：所有节点的倾斜程度之和。

思路：递归

递归这东西有点烦人。但我小小的总结了一下，在我看来递归有两大部分：终点和自身调用部分。终点是递归停止的地方，没有函数就会一直循环下去，一般用if语句判断是否达到某一停止条件，我写递归一般先写这部分，比较容易写，拿这题举例子，递归的条件就是到达叶子结点（root = None），这时看看题目中给的例子，发现tilt应该是0.所以很自然的就return 0。

然后是自身调用部分：在这里我们需要明确何时用这部分（when），这部分计算什么（How），这部分返回什么（what）。when：当然是在没有到达停止条件的时候，这题就是非叶子结点的时候，用个else就行。How：当然是计算tilt，abs（root.left-root.right）,what：就是返回tilt

然后这题有些需要注意地方，一个是tilt是累积值sum，所以需要用变量来记录累加值，每次迭代都+=。还有一个就是leetcode是用class为答题模版的，所以在写答案的时候回用到python 定义类时，内部方法的互相调用<https://blog.csdn.net/yangnianjinxin/article/details/78171655>

这个博客举了例子

**总结：调用函数时，如果是内部方法，一定要用self，但是里面的参数不用加self**

**写内部函数时，参数里要有self**

