# 题目

给定一个二叉树，它的每个结点都存放一个 0-9 的数字，每条从根到叶子节点的路径都代表一个数字。

例如，从根到叶子节点路径 1->2->3 代表数字 123。

计算从根到叶子节点生成的所有数字之和。

说明: 叶子节点是指没有子节点的节点。

**示例 1:**

输入: [1,2,3]

1

/ \

2 3

输出: 25

解释:

从根到叶子节点路径 1->2 代表数字 12.

从根到叶子节点路径 1->3 代表数字 13.

因此，数字总和 = 12 + 13 = 25.

**示例 2:**

输入: [4,9,0,5,1]

4

/ \

9 0

 / \

5 1

输出: 1026

解释:

从根到叶子节点路径 4->9->5 代表数字 495.

从根到叶子节点路径 4->9->1 代表数字 491.

从根到叶子节点路径 4->0 代表数字 40.

因此，数字总和 = 495 + 491 + 40 = 1026.

# 分析

/\*\*

\* 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 sumNumbers(TreeNode\* root) {

return helper(root,0);

}

int helper(TreeNode\* root, int i){

if (root == nullptr) return 0;

int temp = i \* 10 + root->val;

if (root->left == nullptr && root->right == nullptr)

return temp;

return helper(root->left, temp) + helper(root->right, temp);

}

};