## 题目：

Given two binary trees and imagine that when you put one of them to cover the other, some nodes of the two trees are overlapped while the others are not.

You need to merge them into a new binary tree. The merge rule is that if two nodes overlap, then sum node values up as the new value of the merged node. Otherwise, the NOT null node will be used as the node of new tree.

给定两个二叉树，想象当你把其中一个树覆盖另一棵树时，这两棵树的一些节点是重叠的，而其他的不是。你需要把它们合并成一个新的二叉树。合并规则是，如果两个节点重叠，则将节点值作为合并节点的新值。否则，非空节点将被用作新树的节点。

example:

Input:

Tree 1 Tree 2

1 2

/ \ / \

3 2 1 3

/ \ \

5 4 7

Output:

Merged tree:

3

/ \

4 5

/ \ \

5 4 7

note:

struct TreeNode {

int val;

TreeNode \*left;

TreeNode \*right;

TreeNode(int x) : val(x), left(NULL), right(NULL) {}

};

## 思路：

1.假设两颗树的指针为 t1,t2。 则有4种情况：两棵都为空树，返回nullptr。 t1为空，返回t2。 t2为空，返回t1。 都不为空，两个val相加，还未结束，递归处理左右孩子,返回存放结果的树的指针。 可以这样理解：左右孩子又是相同的问题。

## 代码：