617. Merge Two Binary Trees

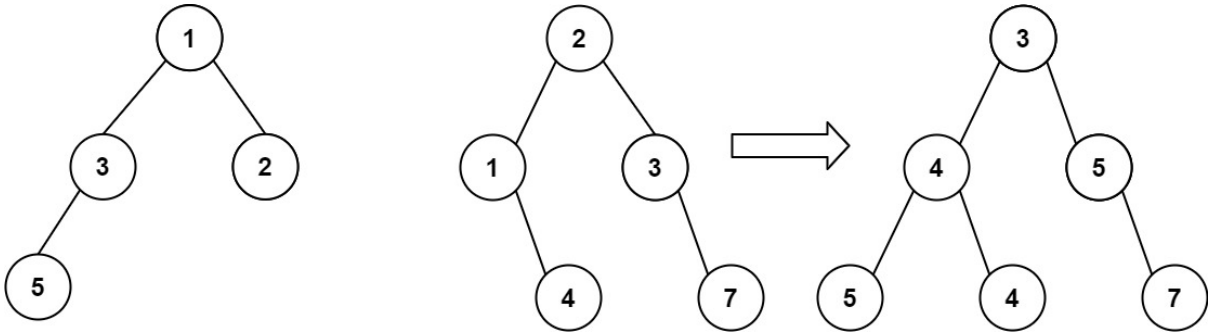
You are given two binary trees root1 and root2.

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 the two trees 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 the new tree.

Return *the merged tree*.

Note: The merging process must start from the root nodes of both trees.

Example 1:



Input: root1 = [1,3,2,5], root2 = [2,1,3,null,4,null,7]

Output: [3,4,5,5,4,null,7]

Example 2:

Input: root1 = [1], root2 = [1,2]

Output: [2,2]

Constraints:

- The number of nodes in both trees is in the range [0, 2000].
- -10^4 <= Node.val <= 10^4

```
# Definition for a binary tree node.
# class TreeNode(object):
#     def __init__(self, val=0, left=None, right=None):
#         self.val = val
#         self.left = left
#         self.right = right
class Solution(object):
    def mergeTrees(self, root1, root2):
        """
        :type root1: TreeNode
        :type root2: TreeNode
        :rtype: TreeNode
        """
        if not root1:
            return root2
        if not root2:
            return root1

        merged_root = TreeNode(root1.val + root2.val)
        merged_root.left = self.mergeTrees(root1.left,
root2.left)
        merged_root.right = self.mergeTrees(root1.right,
root2.right)

        return merged_root
```