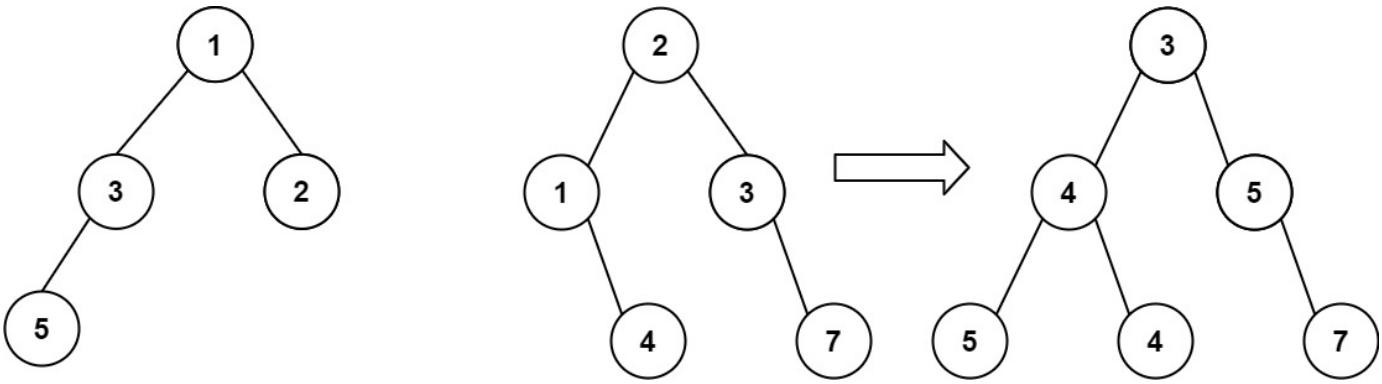
617. Merge Two Binary Trees

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.



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

```
def mergeTrees(self, root1: TreeNode, root2: TreeNode) -> TreeNode:
        if root1 is None or root2 is None:
            return root1 if root2 is None else root2
        root = self.helper(root1,root2)
        return root

    def helper(self,root1,root2):
        if root1 is None or root2 is None:
            return
        temp = root1.val+root2.val
        node = TreeNode(temp)
        node.left = self.helper(root1.left,root2.left)
        if root1.left is not None and root2.left is None:
            temp = root1.left
            root1.left = None
            node.left = temp
```

```
        elif root1.left is None and root2.left is not None:
            temp = root2.left
            root2.left = None
            node.left = temp

        node.right = self.helper(root1.right,root2.right)
        if root1.right is not None and root2.right is None:
            temp = root1.right
            root1.right = None
            node.right = temp
        elif root1.right is None and root2.right is not None:
            temp = root2.right
            root2.right = None
            node.right = temp

        return node
```