129. Sum Root to Leaf Numbers

You are given the `root` of a binary tree containing digits from `0` to `9` only.

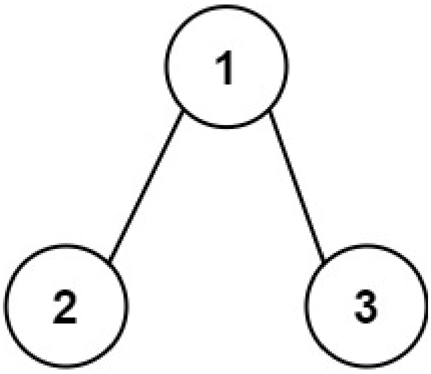
Each root-to-leaf path in the tree represents a number.

- For example, the root-to-leaf path `1 -> 2 -> 3` represents the number `123`.

Return *the total sum of all root-to-leaf numbers*.

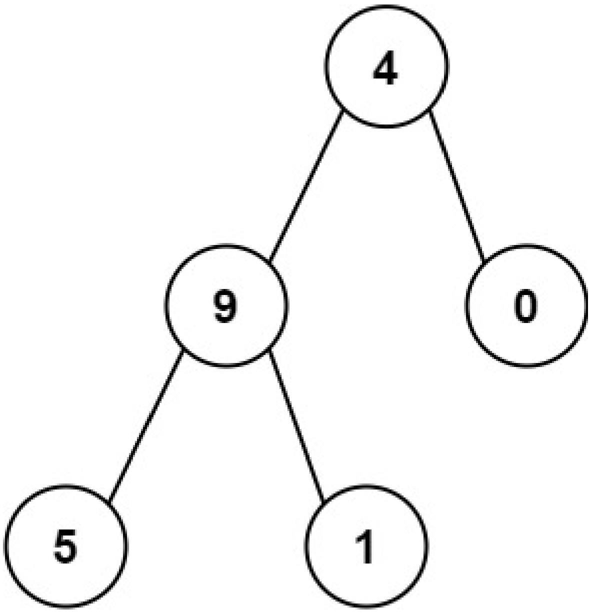
A **leaf** node is a node with no children.

Example 1:



Input: root = [1,2,3]
Output: 25
Explanation:
The root-to-leaf path `1->2` represents the number `12`. The root-to-leaf path `1->3` represents the number `13`. Therefore, sum = 12 + 13 = `25`.

Example 2:



Input: root = [4,9,0,5,1]
Output: 1026
Explanation:
The root-to-leaf path `4->9->5` represents the number 495. The root-to-leaf path `4->9->1` represents the number 491. The root-to-leaf path `4->0` represents the number 40. Therefore, sum = 495 + 491 + 40 = `1026`

```
def sumNumbers(self, root: TreeNode) -> int:
        temp = ""
        ans = [0]
        self.helper(root,temp,ans)
        return ans[0]


    def helper(self,root,temp,ans):
        if root is None:
            return
        temp = temp+str(root.val)
        if root.left is root.right:
            ans[0]=ans[0]+int(temp)
        self.helper(root.left,temp,ans)
        self.helper(root.right,temp,ans)
```