

# Problem 129: Sum Root to Leaf Numbers

## Problem Information

**Difficulty:** Medium

**Acceptance Rate:** 69.24%

**Paid Only:** No

**Tags:** Tree, Depth-First Search, Binary Tree

## Problem Description

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. Test cases are generated so that the answer will fit in a **32-bit** integer.

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->5` represents the number `405`. Therefore, `sum = 495 + 491 + 405 = 1391`.

**\*\*Constraints:\*\***

\* The number of nodes in the tree is in the range `[1, 1000]`. \* `0 <= Node.val <= 9` \* The depth of the tree will not exceed `10`.

## Code Snippets

**C++:**

```
/**
 * Definition for a binary tree node.
 * struct TreeNode {
 *     int val;
 *     TreeNode *left;
 *     TreeNode *right;
 *     TreeNode() : val(0), left(nullptr), right(nullptr) {}
 *     TreeNode(int x) : val(x), left(nullptr), right(nullptr) {}
 *     TreeNode(int x, TreeNode *left, TreeNode *right) : val(x), left(left),
 *     right(right) {}
 * };
 */
class Solution {
public:
    int sumNumbers(TreeNode* root) {

    }
};
```

**Java:**

```
/**
 * Definition for a binary tree node.
 * public class TreeNode {
 *     int val;
 *     TreeNode left;
 *     TreeNode right;
 *     TreeNode() {}
 *     TreeNode(int val) { this.val = val; }
 *     TreeNode(int val, TreeNode left, TreeNode right) {
 *         this.val = val;
 *         this.left = left;
 *     }
 * }
```

```

* this.right = right;
* }
* }
*/
class Solution {
public int sumNumbers(TreeNode root) {

}
}

```

### Python3:

```

# Definition for a binary tree node.
# class TreeNode:
# def __init__(self, val=0, left=None, right=None):
# self.val = val
# self.left = left
# self.right = right
class Solution:
def sumNumbers(self, root: Optional[TreeNode]) -> int:

```