Problem 445: Add Two Numbers II

Problem Information

Difficulty: Medium
Acceptance Rate: 62.16%
Paid Only: No
Tags: Linked List, Math, Stack

Problem Description

You are given two **non-empty** linked lists representing two non-negative integers. The most significant digit comes first and each of their nodes contains a single digit. Add the two numbers and return the sum as a linked list.

You may assume the two numbers do not contain any leading zero, except the number 0 itself.

Example 1:

![](https://assets.leetcode.com/uploads/2021/04/09/sumii-linked-list.jpg)

Input: l1 = [7,2,4,3], l2 = [5,6,4] **Output:** [7,8,0,7]

Example 2:

Input: l1 = [2,4,3], l2 = [5,6,4] **Output:** [8,0,7]

Example 3:

Input: l1 = [0], l2 = [0] **Output:** [0]

Constraints:

* The number of nodes in each linked list is in the range `[1, 100]`. * `0 <= Node.val <= 9` * It is guaranteed that the list represents a number that does not have leading zeros.

Follow up: Could you solve it without reversing the input lists?

Code Snippets

C++:

```
/**
* Definition for singly-linked list.
* struct ListNode {
* int val;
* ListNode *next;
* ListNode() : val(0), next(nullptr) {}
* ListNode(int x) : val(x), next(nullptr) {}
* ListNode(int x, ListNode *next) : val(x), next(next) {}
* };
*/
class Solution {
public:
ListNode* addTwoNumbers(ListNode* l1, ListNode* l2) {

}
};
```

Java:

```
/**
* Definition for singly-linked list.
* public class ListNode {
* int val;
* ListNode next;
* ListNode() {}
* ListNode(int val) { this.val = val; }
* ListNode(int val, ListNode next) { this.val = val; this.next = next; }
* }
*/
class Solution {
public ListNode addTwoNumbers(ListNode l1, ListNode l2) {

}
}
```

Python3:

```
# Definition for singly-linked list.
# class ListNode:
# def __init__(self, val=0, next=None):
# self.val = val
# self.next = next
class Solution:
def addTwoNumbers(self, l1: Optional[ListNode], l2: Optional[ListNode]) ->
Optional[ListNode]:
```