

# 0002. 两数相加

👤 [ITCharge](#) ⌚ 大约 1 分钟

---

- 标签：递归、链表、数学
- 难度：中等

## 题目链接

---

- [0002. 两数相加 - 力扣](#)

## 题目大意

---

**描述：**给定两个非空的链表  $l_1$  和  $l_2$ 。分别用来表示两个非负整数，每位数字都是按照逆序的方式存储的，每个节点存储一位数字。

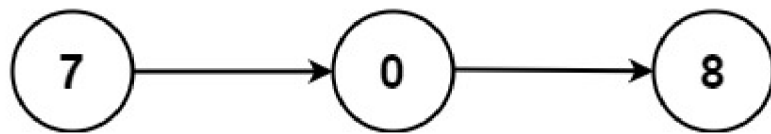
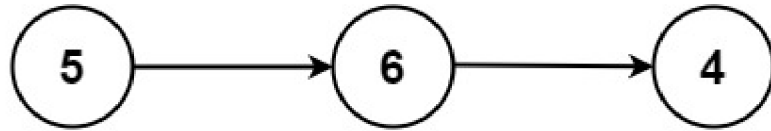
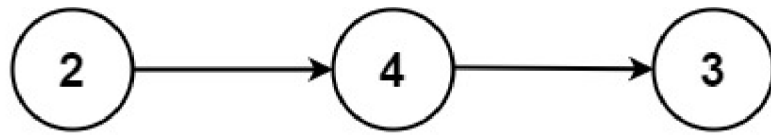
**要求：**计算两个非负整数的和，并逆序返回表示和的链表。

**说明：**

- 每个链表中的节点数在范围  $[1, 100]$  内。
- $0 \leq Node.val \leq 9$ 。
- 题目数据保证列表表示的数字不含前导零。

**示例：**

- 示例 1：



输入:  $l1 = [2, 4, 3]$ ,  $l2 = [5, 6, 4]$

输出:  $[7, 0, 8]$

解释:  $342 + 465 = 807$ .

py

- 示例 2:

输入:  $l1 = [0]$ ,  $l2 = [0]$

输出:  $[0]$

py

## 解题思路

### 思路 1: 模拟

模拟大数加法，按位相加，将结果添加到新链表上。需要注意进位和对 10 取余。

## 思路 1：代码

py

```
class Solution:
    def addTwoNumbers(self, l1: ListNode, l2: ListNode) -> ListNode:
        head = curr = ListNode(0)
        carry = 0
        while l1 or l2 or carry:
            if l1:
                num1 = l1.val
                l1 = l1.next
            else:
                num1 = 0
            if l2:
                num2 = l2.val
                l2 = l2.next
            else:
                num2 = 0

            sum = num1 + num2 + carry
            carry = sum // 10

            curr.next = ListNode(sum % 10)
            curr = curr.next

        return head.next
```

## 思路 1：复杂度分析

- **时间复杂度：** $O(\max(m, n))$ 。其中， $m$  和  $n$  分别是链表  $l1$  和  $l2$  的长度。
- **空间复杂度：** $O(1)$ 。