

Leetcode2: 两数相加:

2. 两数相加

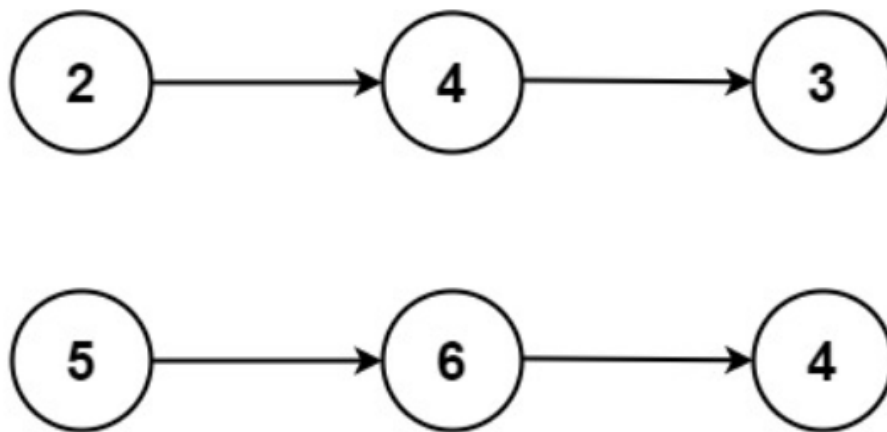
难度 中等 9388 ☆ 代码 文 铃 问

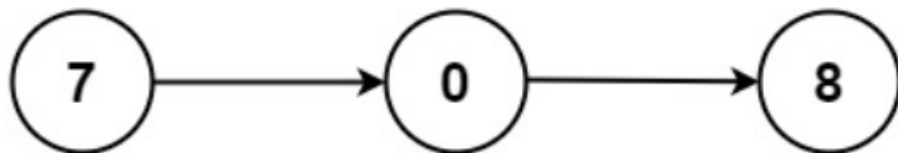
给你两个 **非空** 的链表，表示两个非负的整数。它们每位数字都是按照 **逆序** 的方式存储的，并且每个节点只能存储 **一位** 数字。

请你将两个数相加，并以相同形式返回一个表示和的链表。

你可以假设除了数字 0 之外，这两个数都不会以 0 开头。

示例 1:





输入: $l1 = [2,4,3]$, $l2 = [5,6,4]$

输出: $[7,0,8]$

解释: $342 + 465 = 807$.

示例 2:

输入: $l1 = [0]$, $l2 = [0]$

输出: $[0]$

示例 3:

输入: $l1 = [9,9,9,9,9,9,9]$, $l2 = [9,9,9,9]$

输出: $[8,9,9,9,0,0,0,1]$

算法思路: 模拟 (大数加法)

1: 同时从头开始枚举两个链表, 将 $l1$ 和 $l2$ 指针指向的元素存到 t 中, 再将 $t \% 10$ 的元素存到 dummy 链表中, 再 $t / 10$ 去掉存进去的元素, $l1$ 和 $l2$ 同时往后移动一格

2: 当遍历完所有元素时, 如果 $t \neq 0$, 再把 t 存入到 dummy 链表中

时间复杂度: $O(n + m)$

n 和 m 分别代表两个链表的长度

```
1  /**
2   * Definition for singly-linked list.
3   * public class ListNode {
4   *     int val;
5   *     ListNode next;
6   *     ListNode() {}
7   *     ListNode(int val) { this.val = val; }
8   *     ListNode(int val, ListNode next) { this.val = val; this.next = next; }
9   * }
10  */
11  class Solution {
12      public ListNode addTwoNumbers(ListNode l1, ListNode l2) {
13          ListNode dummy = new ListNode(-1);
14          ListNode cur = dummy;
```

```
15     int t = 0;
16     while(l1 != null || l2 != null){
17         if(l1 != null){
18             t += l1.val;
19             l1 = l1.next;
20         }
21         if(l2 != null){
22             t += l2.val;
23             l2 = l2.next;
24         }
25         cur.next = new ListNode(t % 10);
26         t /= 10;
27         cur = cur.next;
28     }
29     if(t != 0){
30         cur.next = new ListNode(t);
31     }
32     return dummy.next;
33 }
34 }
```