역순으로 저장된 연결 리스트의 숫자를 더하라.

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

```
# Definition for singly-linked list.
# class ListNode:
#     def __init__(self, val=0, next=None):
#         self.val = val
#         self.next = next
```

1.브루트 포스

```
class Solution:
    def toList(self, head: ListNode) -> List[str]:
        result = []
        while head:
            result.append(str(head.val))
            head = head.next
        return result
    def addTwoNumbers(self, l1: ListNode, l2: ListNode) -> ListNode:
        result = None
        left = int(''.join(self.toList(l1)[::-1]))
        right = int(''.join(self.toList(l2)[::-1]))
        for c in str(left + right):
            result = ListNode(val = int(c), next = result)
        return result
```

2.전가산기 활용

```
class Solution:
    def addTwoNumbers(self, l1: ListNode, l2: ListNode) -> ListNode:
        root = head = ListNode(0)
        carry = 0
        while l1 or l2 or carry:
            sum = 0
            if l1:
                sum += l1.val
                l1 = l1.next
            if l2:
                sum += l2.val
                l2 = l2.next
            carry, val = divmod(sum + carry, 10)
            head.next = ListNode(val = val)
            head = head.next
        return root.next
```

- 가산기 알고리즘 활용하여 carry on을 처리하며 계산한다.