

Add Two Numbers

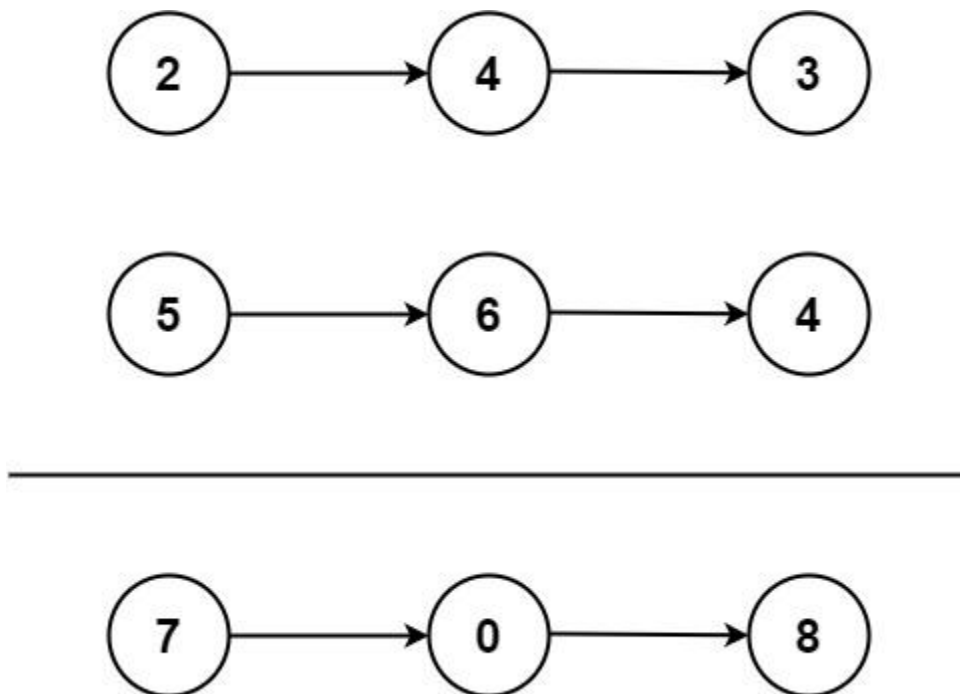
Question:

<https://leetcode.com/problems/add-two-numbers/>

You are given two non-empty linked lists representing two non-negative integers. The digits are stored in reverse order, 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:



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

Output: $[7, 0, 8]$

Explanation: $342 + 465 = 807$.

Example 2:

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

Output: [0]

Example 3:

Input: l1 = [9,9,9,9,9,9,9], l2 = [9,9,9,9]

Output: [8,9,9,9,9,0,0,0,1]

Approach 1:

So my first approach was to loop over both the linked lists and directly sum the digits.

Now there are many things wrong with this approach.

1. I was not considering unequal linked lists into account.
2. I was not handling the boundary cases.

Solution 1:

Code: <https://gist.github.com/vermaayush680/a51ff9e9990b77e2643aabbdd8cb48a4>

class Solution:

def addTwoNumbers(self,l1,l2):

 l3=l1

 temp=l3

 carry=0

while l1 and l2:

 l3.val=l1.val+l2.val+carry

 carry=l3.val//10

 l3.val=l3.val%10

 l1=l1.next

 l2=l2.next

```
l3=l3.next  
return temp
```

Approach 2:

So I first solved the problem of unequal linked lists and handled the boundary case separately.

Solution 2:

Code: <https://gist.github.com/vermaayush680/7c4f4fbdfc4b64eccd04fba12af7d10>

```
class Solution:  
    def addTwoNumbers(self,l1,l2):  
        l3=ListNode(0)  
        temp=l3  
        carry=0  
        while l1 or l2:  
            if l1:  
                carry+=l1.val  
                l1=l1.next  
            if l2:  
                carry+=l2.val  
                l2=l2.next  
            l3.next=ListNode(carry%10)  
            l3=l3.next  
            carry=carry//10  
        if carry:  
            l3.next=ListNode(carry)  
        return temp.next
```

Approach 3:

Finally, I merged the boundary case inside the loop instead of separate handling.

While there is carry, keep adding values to the linked list. This ensures that the last carry is also included in the linked list.

Solution 3:

Code: <https://gist.github.com/vermaayush680/2d054e6346fd4a5cc377915401d2e4eb>

```
class Solution:
    def addTwoNumbers(self, l1, l2):
        l3=ListNode(0)
        temp=l3
        carry=0
        while l1 or l2 or carry:
            if l1:
                carry+=l1.val
                l1=l1.next
            if l2:
                carry+=l2.val
                l2=l2.next
            l3.next=ListNode(carry%10)
            l3=l3.next
            carry=carry//10
        return temp.next
```