445. Add Two Numbers II

Medium 651 87 Favorite Share

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

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

Follow up:
What if you cannot modify the input lists? In other words, reversing the lists is not allowed.

Example:

```
Input: (7 -> 2 -> 4 -> 3) + (5 -> 6 -> 4)
Output: 7 -> 8 -> 0 -> 7
```

Accepted 83,023 | Submissions 167,845

Seen this question in a real interview before? Yes No

Contributor

```java
5
6       public static class ListNode{
7           int val;
8           ListNode next;
9           public ListNode(int val) {
10              this.val = val;
11          }
12      }
13      /*
14       * 这里的链表是顺序的数字，不和第2道一样。
15       * 思想是先计算两者的长度，保持l1.length > l2.l
16       */
17      public ListNode addTwoNumbers(ListNode l1,
18          ListNode pre = new ListNode(0);
19
20          int length1 = 0, length2 = 0;
21          ListNode next = l1, next2 = l2;
22
23          while (next != null) {
24              length1 ++;
25              next = next.next;
26          }
```

```
27
28          while (next2 != null) {
29              length2 ++;
30              next2 = next2.next;
31          }
32
33          if(length1 < length2)
34              return addTwoNumbers(l2, l1);
35
36          next = l1; next2 = l2;
37          int [] temp = new int [length1];
38          //因为l1.length > l2.length, 所以一开始te
39          for(int i = 0; i < length1; i ++) {
40              if(i < length1 - length2) {
41                  temp[i] = next.val;
42                  next = next.next;
43              }else {
44                  temp[i] = next.val + next2.val
45                  next = next.next;
46                  next2 = next2.next;
47              }
48          }
49          //然后逆序求, 插入节点是直接插在头节点的后面,
50          int carry = 0;
51          pre.next = null;
52          for(int j = length1 - 1; j >= 0; j --)
53              ListNode tmp = new ListNode((temp[
54              carry = (temp[j] + carry) / 10;
55              tmp.next = pre.next;
56              pre.next = tmp;
57          }
58          if(carry != 0) {
59              ListNode tmp = new ListNode(carry)
60              tmp.next = pre.next;
```

```
60                tmp.next = pre.next;
61                pre.next = tmp;
62            }
63            return pre.next;
64        }
```