

MCS – 253P ADVANCED PROGRAMMING AND PROBLEM SOLVING

LAB 4 Write Up (Add Two Numbers II)

Aswin Sampath
saswin@uci.edu

53844684

Date: 26/10/2023

Question:

DescriptionEditorialSolutions (4K)Submissions

445. Add Two Numbers II

Medium✔👍 5.7K🗨 279☆🔄

🔒 Companies

You are given two **non-empty** linked lists representing two non-negative integers. The most significant digit comes first 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.

DescriptionEditorialSolutions (4K)Submissions

Example 1:

```
graph LR; n1((7)) --> n2((2)); n2 --> n3((4)); n3 --> n4((3)); m1((5)) --> m2((6)); m2 --> m3((4)); r1((7)) --> r2((8)); r2 --> r3((0)); r3 --> r4((7))
```

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

Example 2:

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

Example 3:

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

Constraints:

- The number of nodes in each linked list is in the range [1, 100].
- 0 ≤ Node.val ≤ 9

Code

```
1  class Solution {
2  public:
3      ListNode* addTwoNumbers(ListNode* l1, ListNode* l2) {
4          stack<int> s1, s2;
5          ListNode* current = l1;
6
7          while (current) {
8              s1.push(current->val);
9              current = current->next;
10         }
11
12         current = l2;
13
14         while (current) {
15             s2.push(current->val);
16             current = current->next;
17         }
18
19         int carry = 0;
20         ListNode* result = nullptr;
21
22         while (!s1.empty() || !s2.empty() || carry) {
23             int sum = carry;
24
25             if (!s1.empty()) {
26                 sum += s1.top();
27                 s1.pop();
28             }
29
30             if (!s2.empty()) {
31                 sum += s2.top();
32                 s2.pop();
33             }
34
35             carry = sum / 10;
36             sum %= 10;
37
38             ListNode* newNode = new ListNode(sum);
39             newNode->next = result;
40             result = newNode;
41         }
42
43         return result;
44     }
45 };
```

Output:

Problem List

Dynamic LayoutPremium0

DescriptionEditorialSolutionsSubmissions

Accepted

Runtime

27 ms

Beats 44.35% of users with C++

Memory

73.99 MB

Beats 18.93%

More challenges

1634. Add Two Polynomials Represented as Linked List

Status

Language

Runtime

Memory

Accepted

29 minutes ago

C++

27 ms

74 MB

Aswin

Oct 26, 2023 06:57

Details

+ Solution

C++

Runtime 27 ms

Beats 44.35%

Memory 74 MB

Beats 18.93%

Click the distribution chart to view more details

Related Tags

Select related tags

0/5

class Solution {

public:

ListNode* addTwoNumbers(ListNode* l1, ListNode* l2) {