

## 2. Add Two Numbers

1032

372

Add to List

[Description \(/problems/add-two-numbers/description/\)](/problems/add-two-numbers/description/)

[Hints \(/problems/add-two-numbers/hints/\)](/problems/add-two-numbers/hints/)

[Submissions \(/problems/add-two-numbe](/problems/add-two-numbers/submissions/)

### Solution

#### Intuition

Keep track of the carry using a variable and simulate digits-by-digits sum starting from the head of list, which contains the least-significant digit.

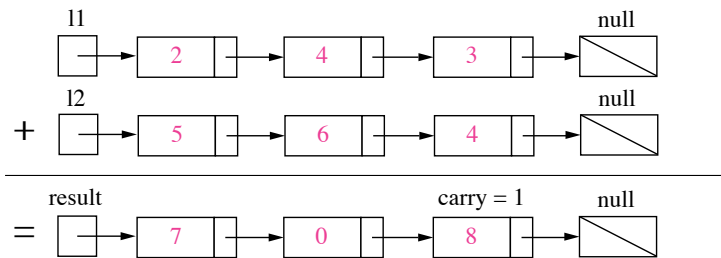


Figure 1. Visualization of the addition of two numbers:  $342 + 465 = 807$ .  
Each node contains a single digit and the digits are stored in reverse order.

#### Algorithm

Just like how you would sum two numbers on a piece of paper, we begin by summing the least-significant digits, which is the head of *l1* and *l2*. Since each digit is in the range of  $0 \dots 9$ , summing two digits may "overflow". For example  $5 + 7 = 12$ . In this case, we set the current digit to 2 and bring over the *carry* = 1 to the next iteration. *carry* must be either 0 or 1 because the largest possible sum of two digits (including the carry) is  $9 + 9 + 1 = 19$ .

The pseudocode is as following:

- Initialize current node to dummy head of the returning list.
- Initialize carry to 0.
- Initialize *p* and *q* to head of *l1* and *l2* respectively.
- Loop through lists *l1* and *l2* until you reach both ends.
  - Set *x* to node *p*'s value. If *p* has reached the end of *l1*, set to 0.
  - Set *y* to node *q*'s value. If *q* has reached the end of *l2*, set to 0.
  - Set *sum* = *x* + *y* + *carry*.
  - Update *carry* = *sum*/10.
  - Create a new node with the digit value of (*sum* mod 10) and set it to current node's next, then advance current node to next.
  - Advance both *p* and *q*.
- Check if *carry* = 1, if so append a new node with digit 1 to the returning list.
- Return dummy head's next node.

Note that we use a dummy head to simplify the code. Without a dummy head, you would have to write extra conditional statements to initialize the head's value.

Take extra caution of the following cases:

Test case	Explanation
$l1 = [0, 1]$ $l2 = [0, 1, 2]$	When one list is longer than the other.

Notes

$l1 = []$ $l2 = [0, 1]$	When one list is null, which means an empty list.
$l1 = [9, 9]$ $l2 = [1]$	The sum could have an extra carry of one at the end, which is easy to forget.

**Java**

```

public ListNode addTwoNumbers(ListNode l1, ListNode l2) {
    ListNode dummyHead = new ListNode(0);
    ListNode p = l1, q = l2, curr = dummyHead;
    int carry = 0;
    while (p != null || q != null) {
        int x = (p != null) ? p.val : 0;
        int y = (q != null) ? q.val : 0;
        int sum = carry + x + y;
        carry = sum / 10;
        curr.next = new ListNode(sum % 10);
        curr = curr.next;
        if (p != null) p = p.next;
        if (q != null) q = q.next;
    }
    if (carry > 0) {
        curr.next = new ListNode(carry);
    }
    return dummyHead.next;
}

```

**Complexity Analysis**

- Time complexity :  $O(\max(m, n))$ . Assume that  $m$  and  $n$  represents the length of  $l1$  and  $l2$  respectively, the algorithm above iterates at most  $\max(m, n)$  times.
- Space complexity :  $O(\max(m, n) + 1)$ . The length of the new list is at most  $\max(m, n) + 1$ .

**Follow up**

What if the digits in the linked list are stored in non-reversed order? For example:

$$(3 \rightarrow 4 \rightarrow 2) + (4 \rightarrow 6 \rightarrow 5) = 8 \rightarrow 0 \rightarrow 7$$



Join the conversation

Signed in as **Xiaotian\_Fu**.

Post a Reply

B

**BobLee** commented 14 hours ago

I can this problem can be solved by space complexity  $O(1)$

(<https://discuss.leetcode.com/user/boblee>)

V

**VINAYVKK** commented yesterday

@luckyever619 (<https://discuss.leetcode.com/uid/94427>) consider this example number 235, this number is stored in linked list nodes : 5->3->2->Null

(<https://discuss.leetcode.com/user/vinayvkk>)

L

**luckyever619** commented yesterday

Why it is not returning curr.next node?

(<https://discuss.leetcode.com/user/luckyever619>)