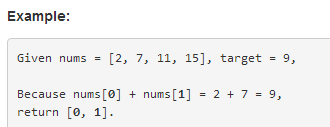
LeetCode刷题1-2

# Two Sum(Easy)

Given an array of integers, return indices of the two numbers such that they add up to a specific target. You may assume that each input would have exactly one solution, and you may not use the same element twice.



程序代码:

public int[] twoSum(int[] nums, int target) {

for (int i = 0; i < nums.length; i++) {

for (int j = i + 1; j < nums.length; j++) {

if (nums[j] == target - nums[i]) {

return new int[] { i, j };

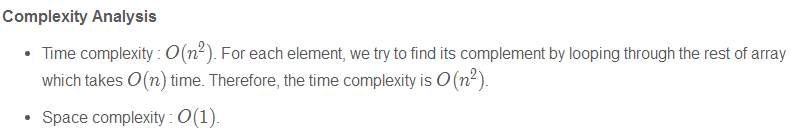
}

}

}

throw new IllegalArgumentException("No two sum solution");

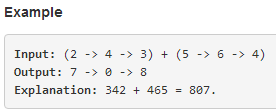
}



# Add Two Numbers(Medium)

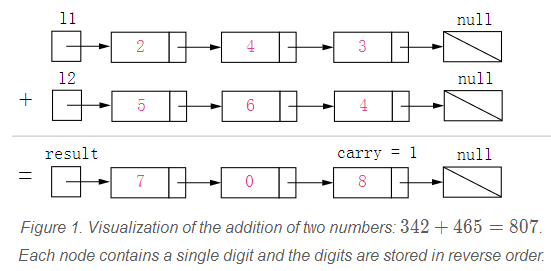
You are given **two non-empty linked lists** representing two non-negative integers. The digits are stored in reverse order and each of their nodes contain a single digit. Add the two numbers and return it as a linked list.

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



**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.



解答：

关键理解ListNode这个对象：

/\*\*

\* Definition for singly-linked list.

\* public class ListNode {

\* int val;

\* ListNode next;

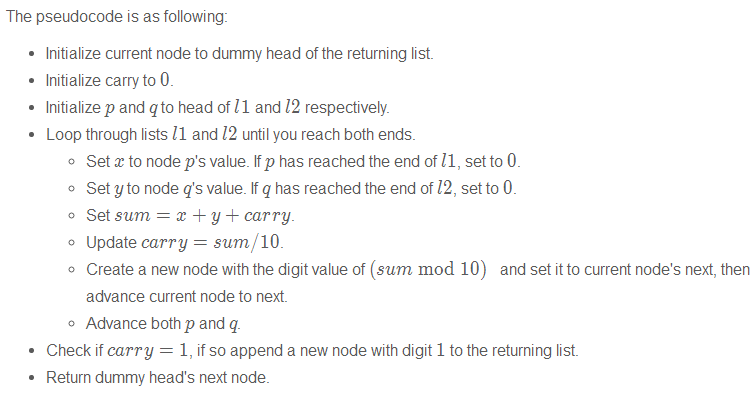
\* ListNode(int x) { val = x; }

\* }

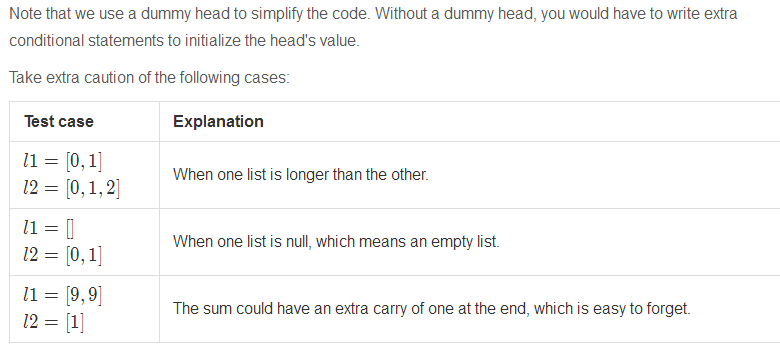
\*/

ListNode为一个List的节点，引用便量根据其指向的地址调出该List，如果在最开头就调出整个List，如果在中间某个节点，就会只调出后面一部分。

程序伪代码：



注意特殊情况：



程序代码：

public ListNode addTwoNumbers(ListNode l1, ListNode l2) {

ListNode dummyHead = new ListNode(0);

ListNode p = l1, q = l2, curr = dummyHead;//创建新的引用变量的作用：让dummyHead指向List的最开头，新的变量curr需要往后走。同理p、q也是需要往后走。

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) {

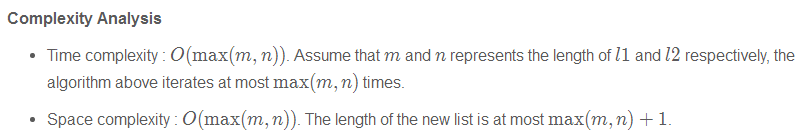
curr.next = new ListNode(carry);

}

return dummyHead.next; //由于dummyHead指向List的最开头，但是最开头的ListNode为0，因此需要往后挪一位。dummyHead指向List为0->7->0>8，dummyHead.next指向List为7->0>8。

}

程序的**复杂度分析**：



变式训练：**Follow up**

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

