



中山大學
SUN YAT-SEN UNIVERSITY

Leetcode

题解

姓名：刘斯宇
学号：17341110

Contents

1	Two Sum	2
2	Add Two Numbers	3
3	Longest Substring Without Repeating Characters	4
4	Add Two Numbers	6

1 Two Sum

题目

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.

例子

Given nums = [2, 7, 11, 15], target = 9,

Because nums[0] + nums[1] = 2 + 7 = 9, return [0, 1].

思路

配对的题目，我们当然可以用两层额 for 循环来解决这个问题，但是这样的话，就没什么意义了，所以我们要降低时间复杂度，也就是要降低内层查找的时间复杂度，如何降低？用空间来换时间，所以很显然需要有个 map，这样就很容易的得到思路了。代码中比较巧妙的一点是如果出现了 vector 有相同的值的话，但是我们用后面的 induce 去覆盖前面的，这样我们遍历的时候就不会忘掉这种相同的值，但是 induce 不同的情况。

```
1 vector<int> twoSum(vector<int>& nums, int target)
2 {
3     unordered_map<int, int> value2index;
4     for(int i=0; i<nums.size(); ++i)
5         value2index[nums[i]] = i;
6     for(int i=0; i<nums.size(); ++i)
7     {
8         int temp = target - nums[i];
9         if (value2index.find(temp) != value2index.end() && i != value2index[
            temp])
10             return {i, value2index[temp]};
11     }
12     return {};
```

13 }

2 Add Two Numbers

题目

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.

例子

Input: (2 -> 4 -> 3) + (5 -> 6 -> 4)

Output: 7 -> 0 -> 8

Explanation: 342 + 465 = 807.

思路

链表的题目比较简单，一直向前遍历，直到找到结束的标志--两个链表都遍历完了以及没有进位了。

```
1  ListNode* addTwoNumbers(ListNode* l1 , ListNode* l2)
2  {
3      ListNode* temp1=l1 ;
4      ListNode* temp2=l2 ;
5      ListNode* start=new ListNode(0) ;
6      ListNode* mm= start ;
7      int jinwei=0;
8      while(temp1||temp2||jinwei)
9      {
10         int sum=0;
11         if (temp1)
12         {
```

```

13         sum+=temp1->val;
14         temp1=temp1->next;
15     }
16     if (temp2)
17     {
18         sum+=temp2->val;
19         temp2=temp2->next;
20     }
21     if (jinwei)
22         sum+=1;
23     if (sum>=10)
24         jinwei=1;
25     else
26         jinwei=0;
27     sum=sum%10;
28     ListNode*temp=new ListNode(sum);
29     start->next=temp;
30     start=temp;
31 }
32 return mm->next;
33 }

```

3 Longest Substring Without Repeating Characters

题目

Given a string, find the length of the longest substring without repeating characters.

例子

Input: "abcabcbb"

Output: 3

Explanation: The answer is "abc", with the length of 3.

Input: "bbbbbb"

Output: 1

Explanation: The answer is "b", with the length of 1.

Input: "pwwkew"

Output: 3

Explanation: The answer is "wke", with the length of 3. Note that the answer must be a substring, "pwke" is a subsequence and not a substring.

思路

因为要找子序列，并且子序列不能有重复元素，所以一定是滑动窗口法，一个元素记录最左边
+unordered_map 就可以了

```
1  int lengthOfLongestSubstring(string s)
2  {
3      if(s.empty())
4          return 0;
5      int left=0;
6      int res=0;
7      unordered_map<char, int>temp;
8      for(int i=0;i<s.size();++i)
9      {
10         if(!temp[s[i]] || left>temp[s[i]])
11             {
12                 res=max(res, i-left+1);
13             }
14         else
```

```

15     {
16         left=temp[s[i]];
17     }
18     temp[s[i]]=i+1;
19 }
20 return res;
21 }

```

4 Add Two Numbers

题目

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.

例子

Input: (2 -> 4 -> 3) + (5 -> 6 -> 4)

Output: 7 -> 0 -> 8

Explanation: 342 + 465 = 807.

思路

链表的题目比较简单，一直向前遍历，直到找到结束的标志--两个链表都遍历完了以及没有进位了。

```

1  ListNode* addTwoNumbers(ListNode* l1 , ListNode* l2)
2  {
3      ListNode*temp1=l1 ;
4      ListNode*temp2=l2 ;
5      ListNode*start=new ListNode(0) ;
6      ListNode*mm=start ;

```

```

7      int jinwei=0;
8      while(temp1||temp2||jinwei)
9      {
10         int sum=0;
11         if(temp1)
12         {
13             sum+=temp1->val;
14             temp1=temp1->next;
15         }
16         if(temp2)
17         {
18             sum+=temp2->val;
19             temp2=temp2->next;
20         }
21         if(jinwei)
22             sum+=1;
23         if(sum>=10)
24             jinwei=1;
25         else
26             jinwei=0;
27         sum=sum%10;
28         ListNode*temp=new ListNode(sum);
29         start->next=temp;
30         start=temp;
31     }
32     return mm->next;
33 }

```