课程尚未开始
请大家耐心等待

关注微信公共账号，获得最新面试题信息及解答

http://www.mikecrm.com/f.php?t=vaIwMR

Facebook: http://www.facebook.com/ninechapter

Weibo:
http://www.weibo.com/ninechapter

Renren:
http://page.renren.com/601712402

4. 两根指针

高级算法班IT求职面试培训 第4章
www.ninechapter.com

两根指针

1. 一个数组，从两边往中间移动(对撞型)
2. 一个数组，同时向前移动(前向型)
3. 两个数组（并行型）

1. 对撞型

Two sum 类 和 Partition 类

Two sum 类型题目

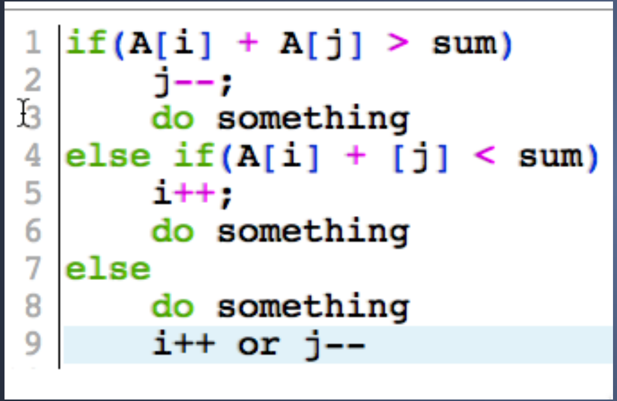
Two sum II

Given an array of integers, count the number of pairs that this pair’s two number sum is larger than target number.

(5,4,3,7,8) sum=9

这一类题目思路

Two sum



```
if(A[i] + A[j] > sum)
     j--;
     do something
else if(A[i] + [j] < sum)
     i++;
     do something
else
     do something
     i++ or j--
```

这一类通过对撞型指针优化算法，根本上其实要证明就是不用扫描多余状态

Triangle Count

http://www.lintcode.com/en/problem/triangle-count/

(3,4,6,7,8,9)

灌水 类型题目

Trapping Rain Water

http://www.lintcode.com/en/problem/trapping-rain-water/

(3, 0, 1, 4, 0, 1, 2)

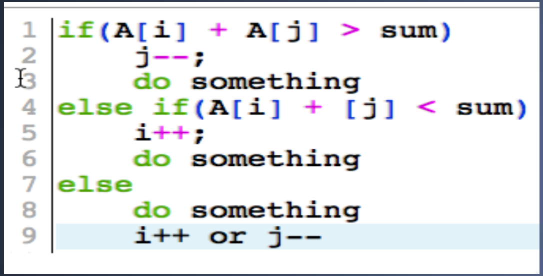
Container With Most Water

http://www.lintcode.com/en/problem/container-with-most-water/

[2,1,4,6,2,3]

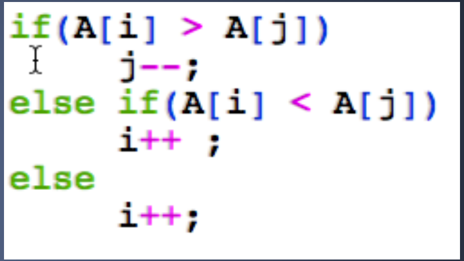
这一类题目思路

Two sum



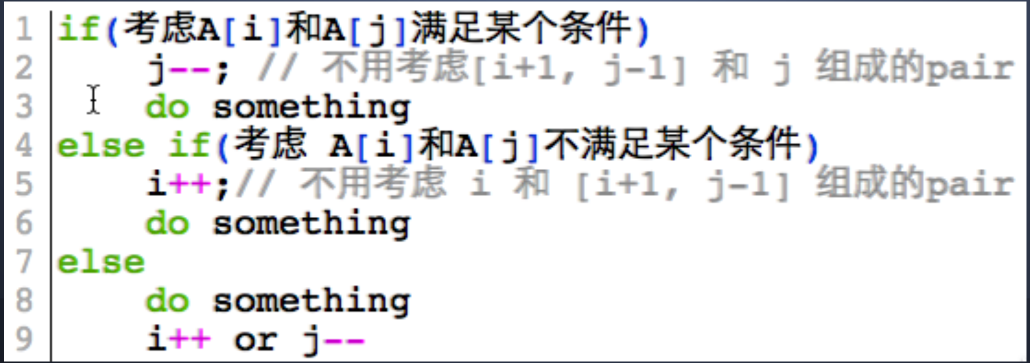
```
if(A[i] + A[j] > sum)
    j--;
    do something
else if(A[i] + [j] < sum)
    i++;
    do something
else
    do something
    i++ or j--
```

灌水



```
if(A[i] > A[j])
    j--;
else if(A[i] < A[j])
    i++ ;
else
    i++;
```

这一类通过对撞型指针优化算法，根本上其 实要证明就是不用扫描多余状态



```
if(考虑A[i]和A[j]满足某个条件)
    j--; // 不用考虑[i+1, j-1] 和 j 组成的pair
    do something
else if(考虑 A[i]和A[j]不满足某个条件)
    i++;// 不用考虑 i 和 [i+1, j-1] 组成的pair
    do something
else
    do something
    i++ or j--
```

对撞型指针题目

2 Sum 类（通过判断条件优化算法）
- 3 Sum Closest
- 4 Sum
- 3 Sum
- Two sum II
- Triangle Count
- Trapping Rain Water
- Container With Most Water

Partition 类
- Partition-array
- **Sort Colors**
- **Partition Array by Odd and Even**
- Sort Letters by Case
- **Valid Palindrome**

2. 前向型

窗口类 和 快慢类

窗口类

minimum-size-subarray-sum

http://www.lintcode.com/en/problem/minimum-size-subarray-sum/

窗口类指针移动模板

通过两层for循环改进算法

```
while(i < n-1){
   while(j < n-1){
     if(满足条件)
         j++;
         更新状态
     else(不满足条件)
         break;
   }
   i++;
   更新状态
}
```

Longest Substring Without Repeating Characters

http://www.lintcode.com/en/problem/longest-substring-without-repeating-characters/

1. 前向型**指针**
2. **Hash或者set记录上次访问**

Minimum Window Substring

http://lintcode.com/en/problem/minimum-window-substring/

[ABCZDEF, ACD]

Longest Substring with At Most Two Distinct Characters
Longest Substring with At Most K Distinct Characters

http://www.lintcode.com/en/problem/longest-substring-with-at-most-k-distinct-characters/

总结

两根指针

优化类型：

- 优化思想通过两层for循环而来
- 慢指针依然是依次遍历
- 快指针证明是否需要回退

前向型指针题目

- 窗口类
- Remove Nth Node From End of List
- minimum-size-subarray-sum
- Minimum Window Substring
- Longest Substring with At Most K Distinct Characters
- Longest Substring Without Repeating Characters

- 快慢类
- Find the Middle of Linked List
- Linked List Cycle I, II

两个数组两个指针

The Smallest Difference

http://www.lintcode.com/en/problem/the-smallest-difference/

O(N^2) -> O(nlog(n))

其他的题目

http://www.lintcode.com/en/problem/merge-two-sorted-lists/

Summary

两个指针

a. 对撞型 (2 sum 类　和 partition 类)

b. 前向型 (窗口类，快慢类)

c. 两个数组, 两个指针 (并行)

希望帮助大家把题目越做越少, 而不是越做越多
http://www.mikecrm.com/f.php?t=vaIwMR