# 两根指针

九章算法强化班 第4章

**扫描二维码关注微信/微博**
**获取最新面试题及权威解答**

微信: ninechapter
微博: http://www.weibo.com/ninechapter
知乎: http://zhuanlan.zhihu.com/jiuzhang
官网: http://www.jiuzhang.com

# Overview



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. 对撞型或者相会型

Two sum 类 和 Partition 类

# Two sum II

http://www.lintcode.com/zh-cn/problem/two-sum-ii/
http://www.jiuzhang.com/solutions/two-sum-ii/

## ☆ 两数之和 II

描述　笔记　数据　评测

给一组整数，问能找出多少对整数，他们的和大于一个给定的目标值

**i 注意事项**

使用 O(1) 的额外空间和 O(nlogn) 的时间。

你在真实的面试中是否遇到过这个题？ Yes

**样例**

对于 numbers = `[2, 7, 11, 15]`, target = `24` 的情况，返回 `1`。因为只有11 + 15可以大于24。

# Triangle Count

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

http://www.jiuzhang.com/solutions/triangle-count/

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

# 382. Triangle Count ★

Description | Notes | >_ Testcase | Judge

Given an array of integers, how many three numbers can be found in the array, so that we can build an triangle whose three edges length is the three numbers that we find?

Have you met this question in a real interview? Yes

**Example**

Given array S = `[3,4,6,7]`, return `3`. They are:

```
[3,4,6]
[3,6,7]
[4,6,7]
```

Given array S = `[4,4,4,4]`, return `4`. They are:

```
[4(1),4(2),4(3)]
[4(1),4(2),4(4)]
[4(1),4(3),4(4)]
[4(2),4(3),4(4)]
```

**Tags**

Two Pointers | LintCode Copyright

# Two Sum类题目思路



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

# 灌水 类型题目

# Trapping Rain Water

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

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

# Container With Most Water

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

http://www.jiuzhang.com/solutions/container-with-most-water/

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

# 383. Container With Most Water ☆

Description | Notes | >_ Testcase | Judge

Given n non-negative integers $a_1$, $a_2$, ..., $a_n$, where each represents a point at coordinate `(i, ai)`. *n* vertical lines are drawn such that the two endpoints of line *i* is at `(i, ai)` and `(i, 0)`. Find two lines, which together with *x*-axis forms a container, such that the container contains the most water.

**Notice**

You may not slant the container.

Have you met this question in a real interview? Yes

**Example**

Given `[1,3,2]`, the max area of the container is `2`.

**Tags**

Array | Two Pointers

# Two Sum类题目总结思路



- Two sum　　灌水

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

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

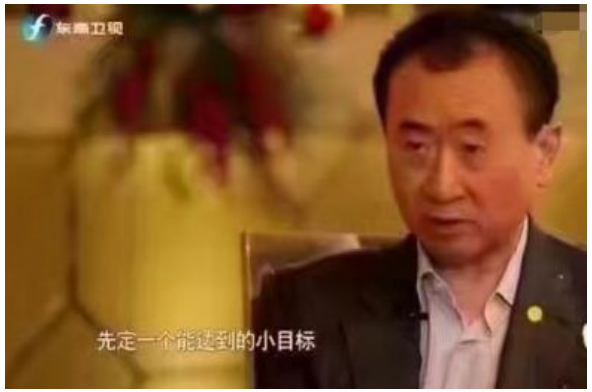
- 这一类通过**对撞型**指针优化算法，根本上其实要证明就是不用扫描多余状态

```
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--
```

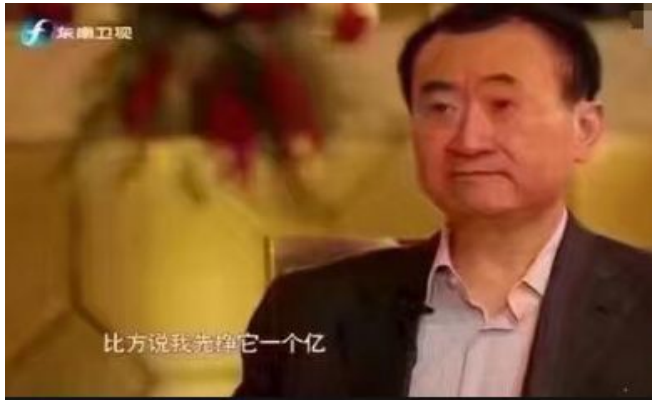
# Partition 类

# Quick select

http://www.lintcode.com/en/problem/kth-largest-element/
http://www.jiuzhang.com/solutions/kth-largest-element/




找世界第3富?



# 5. Kth Largest Element

Description | Notes | Testcase | Judge

Find K-th largest element in an array.

**Notice**

You can swap elements in the array

Have you met this question in a real interview? Yes

**Example**

In array `[9,3,2,4,8]`, the 3rd largest element is `4`.

In array `[1,2,3,4,5]`, the 1st largest element is `5`, 2nd largest element is `4`, 3rd largest element is `3` and etc.

**Challenge**

O(n) time, O(1) extra memory.

**Tags**

Sort | Quick Sort

- PriorityQueue
- 时间复杂度O(nlogk)
- 更适合Topk

- QuickSelect
- 时间复杂度O(n)
- 更适合第k大

# Partition 问题模板



- Partition 模板

- 问题？
- [5,5,5,3,5,5,5]

```
public int partition(int[] nums, int l, int r) {
    // 初始化左右指针和pivot
    int left = l, right = r;
    int pivot = nums[left];

    // 进行partition
    while (left < right) {
        while (left < right && nums[right] >= pivot) {
            right--;
        }
        nums[left] = nums[right];
        while (left < right && nums[left] <= pivot) {
            left++;
        }
        nums[right] = nums[left];
    }

    // 返还pivot点到数组里面
    nums[left] = pivot;
    return left;
}
```

# Nuts & Bolts Problem

http://www.lintcode.com/en/problem/nuts-bolts-problem/

http://www.jiuzhang.com/solutions/nuts-bolts-problem/

# 399. Nuts & Bolts Problem ☆

Description | Notes | Testcase | Judge

Given a set of *n* nuts of different sizes and *n* bolts of different sizes. There is a one-one mapping between nuts and bolts. Comparison of a nut to another nut or a bolt to another bolt is not allowed. It means nut can only be compared with bolt and bolt can only be compared with nut to see which one is bigger/smaller.

We will give you a compare function to compare nut with bolt.

Have you met this question in a real interview? Yes

**Example**

Given nuts = `['ab','bc','dd','gg']`, bolts = `['AB','GG', 'DD', 'BC']`.

Your code should find the matching bolts and nuts.

one of the possible return:

nuts = `['ab','bc','dd','gg']`, bolts = `['AB','BC','DD','GG']`.

we will tell you the match compare function. If we give you another compare function.

the possible return is the following:

nuts = `['ab','bc','dd','gg']`, bolts = `['BC','AA','DD','GG']`.

So you must use the compare function that we give to do the sorting.

`The order of the nuts or bolts does not matter. You just need to find the matching bolt for each nut.`

**Tags**

Sort | Quick Sort

# 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
- k 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**
- **quick sort/ quick select/ nuts bolts problem/wiggle sort II**

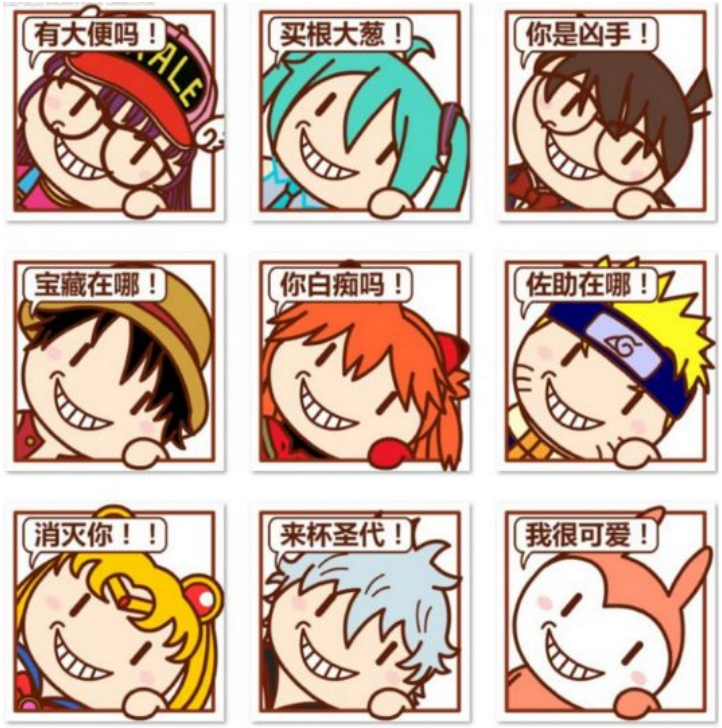
# Break

休息5分钟

# 2. 前向型或者追击型

窗口类 和 快慢类

# 窗口类



# Minimum Size Subarray Sum

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

http://www.jiuzhang.com/solutions/minimum-size-subarray-sum/

# 406. Minimum Size Subarray Sum ☆

Description | Notes | >_ Testcase | Judge

Given an array of n positive integers and a positive integer s, find the minimal length of a subarray of which the sum ≥ s. If there isn't one, return -1 instead.

Have you met this question in a real interview? Yes

**Example**

Given the array `[2,3,1,2,4,3]` and s = `7`, the subarray `[4,3]` has the minimal length under the problem constraint.

**Challenge**

If you have figured out the O(n) solution, try coding another solution of which the time complexity is O(n log n).

**Tags**

Array | Two Pointers | Facebook

# 窗口类指针移动模板



```
通过两层for循环改进算法      ---------------不同于sliding window
for (i =0; i < n; i++)
    while(j < n){
        if(满足条件)
            j++;
            更新j状态
        else(不满足条件)
            break;
    }
    更新i状态
}
```

与sliding windows 区别 ？

# Longest Substring Without Repeating Characters

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

1. 前向型**指针**
2. **Hash或者set记录上次访问**

# 384. Longest Substring Without Repeating Characters ☆

Description | Notes | >_ Testcase | Judge

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

Have you met this question in a real interview? Yes

**Example**

For example, the longest substring without repeating letters for `"abcabcbb"` is `"abc"`, which the length is `3`.

For `"bbbbb"` the longest substring is `"b"`, with the length of `1`.

**Challenge**

O(n) time

**Tags**

Two Pointers | Hash Table | String

# Minimum Window Substring

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

[ABCZDEF, ACD]

# 32. Minimum Window Substring ☆

Description | Notes | >_ Testcase | Judge

Given a string source and a string target, find the minimum window in source which will contain all the characters in target.

**Notice**

If there is no such window in source that covers all characters in target, return the emtpy string `""`.

If there are multiple such windows, you are guaranteed that there will always be only one unique minimum window in source.

Have you met this question in a real interview? Yes

**Clarification**

Should the characters in minimum window has the same order in target?

- Not necessary.

**Example**

For source = `"ADOBECODEBANC"`, target = `"ABC"`, the minimum window is `"BANC"`

**Challenge**

Can you do it in time complexity O(n) ?

**Tags**

Hash Table | LinkedIn | Facebook

# Longest Substring with At Most K(two) Distinct Characters

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

http://www.jiuzhang.com/solutions/longest-substring-with-at-most-k-distinct-characters/

# ☆ Longest Substring with At Most K Distinct Characters

Description | Notes | >_ Testcase | Judge

Given a string *s*, find the length of the longest substring T that contains at most k distinct characters.

Have you met this question in a real interview? Yes

**Example**

For example, Given s = `"eceba"`, `k = 3`,

T is `"eceb"` which its length is `4`.

# 总结



- 优化类型:
  - 优化思想通过两层for循环而来
  - 外层指针依然是依次遍历
  - 内层指针证明是否需要回退

```
通过两层for循环改进算法
for (i =0; i < n; i++)
    while(j < n){
        if(满足条件)
            j++;
            更新j状态
        else(不满足条件)
            break;
    }
    更新i状态
}
```

# 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

# 两个数组各找一个元素，使得和等于target

1. 找一种
2. 找全部种类

# The Smallest Difference

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

http://www.jiuzhang.com/solutions/the-smallest-difference/

# 387. The Smallest Difference ☆

Description | Notes | >_ Testcase | Judge

Given two array of integers(the first array is array `A`, the second array is array `B`), now we are going to find a element in array A which is A[i], and another element in array B which is B[j], so that the difference between A[i] and B[j] (|A[i] - B[j]|) is as small as possible, return their smallest difference.

Have you met this question in a real interview? Yes

**Example**

For example, given array A = `[3,6,7,4]`, B = `[2,8,9,3]`, return `0`

**Challenge**

O(*n* log *n*) time

# 其他的题目

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

# 165. Merge Two Sorted Lists ☆

Description | Notes | Testcase | Judge

Merge two sorted (ascending) linked lists and return it as a new sorted list. The new sorted list should be made by splicing together the nodes of the two lists and sorted in ascending order.

Have you met this question in a real interview? Yes

**Example**

Given `1->3->8->11->15->null`, `2->null`, return `1->2->3->8->11->15->null`.

**Tags**

Linked List | LinkedIn

# 今日经典三题



- Triangle Count
  - Two Sum的变种，灵活运用
- Nuts & Bolts Problem
  - 怎么样想不到快速排序还能这样考。
- Minimum Size Subarray Sum
  - 前向指针题目的经典入门题

# Summary



- 两个指针
  - 对撞型 (2 sum 类  和 partition 类)
  - 前向型 (窗口类，快慢类）
  - 两个数组，两个指针 (并行)

- 模板
  - 2 Sum类模板
  - Partition 类模板
  - 窗口类模板

# Thank You