

双指针算法

Two Pointers Algorithm

课程版本 v5.0 主讲 令狐冲



扫描二维码关注微信小程序/公众号
获取第一手求职资料

请在随课教程中预习如下内容：

什么是同向双指针？

什么是相向双指针？

双指针的鼻祖题 —— 两数之和 Two Sum

链表上的快慢指针算法

快速排序 & 归并排序

- 同向双指针
- 相向双指针
 - 几乎所有 Two Sum 变种
 - Partition
 - Quick Select
 - 分成两个部分
 - 分成三个部分
 - 一些你没听过的(但是面试会考的)排序算法

同向双指针

两根指针一前一后，直到前面的指针走过头

时间复杂度 $O(n)$

随课教程预习题

<http://www.lintcode.com/problem/window-sum/>

<http://www.lintcode.com/problem/remove-duplicate-numbers-in-array/>

<http://www.lintcode.com/problem/intersection-of-two-linked-lists/>

<http://www.lintcode.com/problem/linked-list-cycle/>

<http://www.lintcode.com/problem/linked-list-cycle-ii/>

<http://www.lintcode.com/problem/intersection-of-two-linked-lists/>

Move Zeroes

<http://www.lintcode.com/problem/move-zeroes/>

<http://www.jiuzhang.com/solution/move-zeroes>

将数组中非 0 的元素移动到数组的后半部分

Follow ups:

0 多 vs 0 少 算法有什么区别？

不需要维持相对顺序 vs 需要维持相对顺序 算法有什么区别？

如果需要保证最少修改次数如何做？

相向双指针

两根指针一头一尾，向中间靠拢直到相遇

时间复杂度 $O(n)$

相向双指针的问题主要可以分为下面三类：

1. Reverse 类
2. Two Sum 类
3. Partition 类

Reverse 类

前面学过的三步翻转法就是这一类

Valid Palindrome

验证一个字符串是否为回文串，忽略大小写和非英文字母字符

<http://www.lintcode.com/problem/valid-palindrome/>

<http://www.jiuzhang.com/solution/valid-palindrome/>

Follow up: 可以删掉一个字符

<http://www.lintcode.com/problem/valid-palindrome-ii/>

其实是一道证明题

证明见随课教程

<http://www.jiuzhang.com/tutorial/algorithm/390>

Two Sum 类

先修内容中我们已经讲解了双指针的经典题 Two Sum
接下来我们来看这类问题可能的变化

只能使用 HashMap:

<http://www.lintcode.com/problem/two-sum-data-structure-design/>

<http://www.jiuzhang.com/solutions/two-sum-data-structure-design/>

使用 Two Pointers 会更快:

<http://www.lintcode.com/problem/two-sum-input-array-is-sorted/>

<http://www.jiuzhang.com/solutions/two-sum-input-array-is-sorted/>

Two Sum - Unique pairs

<http://www.lintcode.com/en/problem/two-sum-unique-pairs/>

<http://www.jiuzhang.com/solutions/two-sum-unique-pairs/>

问: 是否可以先去重?

3Sum

<http://www.lintcode.com/problem/3sum/>

<http://www.jiuzhang.com/solutions/3sum/>

统计所有的和为 0 的三元组 (Triples)

Triangle Count

<http://www.lintcode.com/problem/triangle-count/>

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

独孤九剑 —— 破掌式

对于求 2 个变量如何组合的问题
可以循环其中一个变量, 然后研究另外一个变量如何变化

统计所有和 $\leq \text{target}$ 的配对数

<http://www.lintcode.com/problem/two-sum-less-than-or-equal-to-target/>

<http://www.jiuzhang.com/solutions/two-sum-less-than-or-equal-to-target/>

统计所有和 $\geq \text{target}$ 的配对数

<http://www.lintcode.com/en/problem/two-sum-greater-than-target/>

<http://www.jiuzhang.com/solutions/two-sum-greater-than-target/>

休息5分钟

take a break

Two Sum Closest

<http://www.lintcode.com/problem/two-sum-closest-to-target/>

<http://www.jiuzhang.com/solutions/two-sum-closest/>

Follow Up: 3Sum Closest

<http://www.lintcode.com/problem/3sum-closest/>

<http://www.jiuzhang.com/solutions/3sum-closest/>

- **4Sum**
- <http://www.lintcode.com/problem/4sum/>
- <http://www.jiuzhang.com/solutions/4sum/>
-
- **Two Sum - difference equals to target (同向双指针)**
- <http://www.lintcode.com/problem/two-sum-difference-equals-to-target/>
- <http://www.jiuzhang.com/solutions/two-sum-difference-equals-to-target/>

Partition Array

<http://www.lintcode.com/problem/partition-array/>

<http://www.jiuzhang.com/solutions/partition-array/>

Quick Select

随课教程: <http://www.jiuzhang.com/tutorial/algorithm/321>

<http://www.lintcode.com/problem/kth-smallest-numbers-in-unsorted-array/>

<http://www.lintcode.com/problem/kth-largest-element/>

- **Partition Array by Odd and Even**

- <http://www.lintcode.com/problem/partition-array-by-odd-and-even/>
- <http://www.jiuzhang.com/solutions/partition-array-by-odd-and-even/>

-

- **Interleaving Positive and Negative Numbers**

- <http://www.lintcode.com/problem/interleaving-positive-and-negative-numbers/>
- <http://www.jiuzhang.com/solutions/interleaving-positive-and-negative-integers/>

-

- **Sort Letters by Case**

- <http://www.lintcode.com/problem/sort-letters-by-case/>
- <http://www.jiuzhang.com/solutions/sort-letters-by-case/>

Sort Colors

<http://www.lintcode.com/problem/sort-colors/>

<http://www.jiuzhang.com/solutions/sort-colors/>

分成两个部分 vs 分成三个部分

随课教程 <http://www.jiuzhang.com/tutorial/algorithm/354>

排序 Rainbow Sort

<http://www.lintcode.com/en/problem/sort-colors-ii/>

<http://www.jiuzhang.com/solutions/sort-colors-ii/>

问：猜一猜最优的时间复杂度？

更多 Two Pointers 的题和知识点

请报名《九章算法强化班》

烙饼排序 Pancake Sort(有可能会考哦)

https://en.wikipedia.org/wiki/Pancake_sorting

<http://www.geeksforgeeks.org/pancake-sorting/>

睡眠排序 Sleep Sort

https://rosettacode.org/wiki/Sorting_algorithms/Sleep_sort

面条排序 Spaghetti Sort

https://en.wikipedia.org/wiki/Spaghetti_sort

猴子排序 Bogo Sort

<https://en.wikipedia.org/wiki/Bogosort>

随课教程: <http://www.jiuzhang.com/tutorial/algorithm/335>

三指针算法

烙饼排序

<http://www.lintcode.com/problem/pancake-sorting/>