

二分法

Binary Search

课程版本 4.2 主讲 令狐冲



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://zhuankan.zhihu.com/jiuzhang>

官网: <http://www.jiuzhang.com>

版权声明

九章的所有课程均受法律保护，不允许录像与传播录像
一经发现，将被追究法律责任和赔偿经济损失

- **课程错过不补课, 也不提供任何视频**
 - 你才会把在两个小时内集中精力, 全神贯注
 - 你才会把学习放在第一位, 而不是先 LoL 一把, 先逛个街, 先和朋友吃个饭
 - 你才会获得最佳的课程体验
 - 良苦用心希望同学们理解
- **不允许建私群(包括QQ群, 微信群)**
 - 在QQ群中拉人私下组群的将被踢群并不再提供QQ答疑服务
- LintCode 阶梯训练如何使用: <http://www.jiuzhang.com/faq/31/>
- 课程各类服务的有效期为一年
 - LintCode阶梯训练访问权限
 - QQ群答疑
 - QA答疑
 - 课件
 - 知识点小视频

- 新学员必读常见问题解答
 - <http://www.jiuzhang.com/qa/3/>
- 第一节课错过了怎么办？
 - 报名下一期的《九章算法班》第一节课免费试听即可
- 学员QQ群是什么？怎么加？
 - 请登录官网在右上角“我的课程”中查看QQ群号
- 九章的账户绑定到LintCode之后可以解除绑定么？
 - 不可以
 - 因此不要把你的九章账户给别人使用
 - 一些老学员的 LintCode 账号绑定了其他人的九章账户是因为你以前把账号共享给了其他人
 - 你可以申请新的 LintCode 账户和你现在的账户进行绑定
 - LintCode 相关问题请参见：<http://www.jiuzhang.com/qa/683/>
- 更多问题的答案请见：<http://www.jiuzhang.com/faq/>

每节课的具体上课时间(请注意日期和时区, 特别是周几)



章节	内容	北京时间	美东时间	美西时间
1	Introducing Algorithm Interview & Coding Style 【免费试听】	2017/09/10 09:30:00	2017/09/09 21:30:00	2017/09/09 18:30:00
2	二分搜索 Binary Search	2017/09/16 09:30:00	2017/09/15 21:30:00	2017/09/15 18:30:00
3	二叉树问题与分治算法 Binary Tree & Divide Conquer	2017/09/17 09:30:00	2017/09/16 21:30:00	2017/09/16 18:30:00
4	宽度优先搜索 Breadth First Search	2017/09/23 09:30:00	2017/09/22 21:30:00	2017/09/22 18:30:00
5	深度优先搜索 Depth First Search	2017/09/24 09:30:00	2017/09/23 21:30:00	2017/09/23 18:30:00
6	链表与数组 Linked List & Array	2017/09/30 09:30:00	2017/09/29 21:30:00	2017/09/29 18:30:00
7	双指针算法 Two Pointers	2017/10/01 09:30:00	2017/09/30 21:30:00	2017/09/30 18:30:00
8	数据结构 Data Structure	2017/10/07 09:30:00	2017/10/06 21:30:00	2017/10/06 18:30:00
9	动态规划 Dynamic Programming	2017/10/08 09:30:00	2017/10/07 21:30:00	2017/10/07 18:30:00

- 第一境界 二分法模板
 - 时间复杂度小练习
 - 递归与非递归的权衡
 - 二分的三大痛点
 - 通用的二分法模板
- 第二境界: 二分位置 之 圈圈叉叉 Binary Search on Index - OOXX
 - 找到满足某个条件的第一个位置或者最后一个位置
- 第三境界: 二分位置 之 保留一半 Binary Search on Index - Half half
 - 保留有解的一半, 或者去掉无解的一半

Binary Search

Given an sorted integer array - nums, and an integer - target.

Find the **any/first/last** position of target in nums

Return **-1** if target does not exist.

$$T(n) = T(n/2) + O(1) = O(\log n)$$

通过 $O(1)$ 的时间, 把规模为 n 的问题变为 $n/2$

思考: 通过 $O(n)$ 的时间, 把规模为 n 的问题变为 $n/2$?

Time Complexity in Coding Interview

- $O(1)$ 极少
- $O(\log n)$ 几乎都是二分法
- $O(\sqrt{n})$ 几乎是分解质因数
- $O(n)$ 高频
- $O(n \log n)$ 一般都可能要排序
- $O(n^2)$ 数组, 枚举, 动态规划
- $O(n^3)$ 数组, 枚举, 动态规划
- $O(2^n)$ 与组合有关的搜索
- $O(n!)$ 与排列有关的搜索

独孤九剑 —— 破剑式

比 $O(n)$ 更优的时间复杂度
几乎只能是 $O(\log n)$ 的二分法

经验之谈: 根据时间复杂度倒推算法是面试中的常用策略

Recursion or While Loop?

R: Recursion

W: While loop

B: Both work

Recursion or Non-Recursion

- 面试中是否使用 Recursion 的几个判断条件
 1. 面试官是否要求了不使用 Recursion (如果你不确定, 就向面试官询问)
 2. 不用 Recursion 是否会造成实现变得很复杂
 3. Recursion 的深度是否会很深
 4. 题目的考点是 Recursion vs Non-Recursion 还是就是考你是否会 Recursion ?
- 记住: 不要自己下判断, 要跟面试官讨论 !

二分法常见痛点

- 又死循环了！ what are you 弄撒捏！
- 循环结束条件到底是哪个？
 - $\text{start} \leq \text{end}$
 - $\text{start} < \text{end}$
 - $\text{start} + 1 < \text{end}$
- 指针变化到底是哪个？
 - $\text{start} = \text{mid}$
 - $\text{start} = \text{mid} + 1$
 - $\text{start} = \text{mid} - 1$

第一境界 二分法模板

<http://www.jiuzhang.com/solutions/binary-search/>

$start + 1 < end$

$start + (end - start) / 2$

$A[mid] ==, <, >$

$A[start] A[end] ? target$

令狐大师兄手把手教你写代码

<http://www.lintcode.com/problem/classical-binary-search/>

<http://www.lintcode.com/problem/first-position-of-target/>

<http://www.lintcode.com/problem/last-position-of-target/>

第二境界

二分位置 之 OOXX

一般会给你一个数组

让你找数组中第一个/最后一个满足某个条件的位置

OOOOOOO...O**O****X**X...XXXXXX

First Bad Version

<http://www.lintcode.com/problem/first-bad-version/>

<http://www.jiuzhang.com/solutions/first-bad-version/>

First version that is bad version

Search In a Big Sorted Array

<http://www.lintcode.com/problem/search-in-a-big-sorted-array/>

<http://www.jiuzhang.com/solutions/search-in-a-big-sorted-array/>

Take a break

5 分钟后回来

Find Minimum in Rotated Sorted Array

<http://www.lintcode.com/problem/find-minimum-in-rotated-sorted-array/>

<http://www.jiuzhang.com/solutions/find-minimum-in-rotated-sorted-array/>

First position \leq Last Number

(WRONG: First position \leq or $<$ First Number)

- Search a 2D Matrix
 - <http://www.lintcode.com/en/problem/search-a-2d-matrix/>
 - <http://www.lintcode.com/en/problem/search-a-2d-matrix-ii/>
 - 不是二分法, 但是是常考题
- Search for a Range
 - <http://www.lintcode.com/en/problem/search-for-a-range/>
 - <http://www.lintcode.com/en/problem/total-occurrence-of-target/>
- Maximum Number in Mountain Sequence
 - <http://www.lintcode.com/en/problem/maximum-number-in-mountain-sequence/>
- Smallest Rectangle Enclosing black Pixels
 - <http://www.lintcode.com/problem/smallest-rectangle-enclosing-black-pixels/>
-
- 以上题目的答案请在 <http://www.jiuzhang.com/solutions> 中搜索

第三境界

二分位置 之 Half half

并无法找到一个条件，形成 OOX 的模型
但可以根据判断，保留下有解的那一半或者去掉无解的一半

Maximum Number in Mountain Sequence

<http://www.lintcode.com/problem/maximum-number-in-mountain-sequence/>

<http://www.jiuzhang.com/solutions/maximum-number-in-mountain-sequence/>

在先增后减的序列中找最大值

Find Peak Element

<http://www.lintcode.com/problem/find-peak-element/>

<http://www.jiuzhang.com/solutions/find-peak-element/>

follow up: Find Peak Element II (by 算法强化班)

Search in Rotated Sorted Array

<http://www.lintcode.com/problem/search-in-rotated-sorted-array/>

<http://www.jiuzhang.com/solutions/search-in-rotated-sorted-array/>

会了这道题，才敢说自己会二分法

总结 —— 我们今天学到了什么

- 使用递归与非递归的权衡方法
- 使用T函数的时间复杂度计算方式
- 二分法模板的四点要素
 - $start + 1 < end$
 - $start + (end - start) / 2$
 - $A[mid] ==, <, >$
 - $A[start] A[end] ? target$
- 三个境界
 - 二分法模板
 - OOX
 - Half half

- Search in a 2D Matrix II
 - 小视频: <http://www.jiuzhang.com/video/search-a-2d-matrix-ii/>
 - 不是二分法, 但是是常考题
- Binary Search:
 - <http://www.lintcode.com/en/tag/binary-search/>
- Rotate Array
 - 小视频: <http://www.jiuzhang.com/video/3-step-reverse/>
 - <http://www.lintcode.com/problem/recover-rotated-sorted-array/>
 - <http://www.lintcode.com/problem/rotate-string/>
 - 三步翻转法:
 - $[4, 5, 1, 2, 3] \rightarrow [5, 4, 1, 2, 3] \rightarrow [5, 4, 3, 2, 1] \rightarrow [1, 2, 3, 4, 5]$
- 点题时间:
 - <http://www.jiuzhang.com/qa/974/>

想学习更难的分法？

《九章算法强化班》

<http://www.jiuzhang.com/course/5/>

第四境界(至高境界):二分答案

例题: <http://www.lintcode.com/problem/copy-books/>

二分法相关题目的解题报告

参考程序+详细的思路描述

<http://www.jiuzhang.com/article/?tags=binary-search>