二分法
Binary Search

课程版本 3.4　　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://zhuanlan.zhihu.com/jiuzhang
官网: http://www.jiuzhang.com

新学员问题

九章算法

- 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
- 有事儿不能来上直播课怎么办？
  - 优惠价基础上再半价报名下一期的课程
  - 半价连成本都不够 T_T
    - GotoWebinar费用，LintCode服务费，Amazon 服务器费
    - 客服费，老师费，助教费
- 是否提供录像？
  - 九章的所有课程均为直播形式，不提供任何录像
  - 禁止录像与传播录像，否则将被追究法律责任与经济损失
    - 你如果录像了，是可以看到你的ID和基本信息的
    - 传播你录制的视频将影响到你的身份问题（在美留学生）和求职（企业看中你是否会泄密）

新学员问题

九章算法

- 学员QQ群是什么？怎么加？
  - 见 GotoWebinar 文字广播（付费之后您收到邮件里就有）
  - 加QQ群请 **附上报名邮箱** 供管理员验证
  - 不允许建QQ/微信私群
- LintCode 阶梯训练在课程结束后仍然可以使用么？
  - 一年之内可以
- 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

Outline

- 第一境界：会写程序
  - 通用的二分法模板 Binary Search Template
    - 解决二分程序的三大痛点
    - 权衡递归与非递归
- 第二境界：找到第一个/最后一个满足某个条件的位置/值
  - 二分位置 Binary Search on Index
  - 二分答案 Binary Search on Result
- 第三境界：保留有解的一半

- 点题时间 *new*

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.

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/

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

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

九章算法

- O(1) 极少
- O(logn) 几乎都是二分法
- O(√n) 几乎是分解质因数
- O(n) 高频
- O(nlogn) 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

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？
  - start <= end
  - start < end
  - start + 1 < end
- 指针变化到底是哪个？
  - start = mid
  - start = mid + 1
  - start = mid - 1

通用的二分法模板

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

start + 1 < end

start + (end - start) / 2

A[mid] ==, <, >

A[start] A[end] ? target

独孤九剑 —— 破剑式

比O(n)更优的时间复杂度
几乎只能是O(logn)的二分法

二分法——二分位置
Binary Search on Index

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

Search a 2D Matrix

http://www.lintcode.com/problem/search-a-2d-matrix/

http://www.jiuzhang.com/solutions/search-a-2d-matrix/

Last row that matrix[row][0] <= target

Search Insert Position

http://www.lintcode.com/problem/search-insert-position/

http://www.jiuzhang.com/solutions/search-insert-position/

First position >= target

(***Last*** position < target) + 1

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 <= Last Number

(WRONG: First position <= or < First Number)

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

二分法——二分答案
Binary Search on Result

往往没有给你一个数组让你二分

同样是找到满足某个条件的最大或者最小值

Sqrt(x)

http://www.lintcode.com/problem/sqrtx/

http://www.jiuzhang.com/solutions/sqrtx/

Last number that number^2 <= x

follow up: what if return a double, not an integer?

First Bad Version

http://www.lintcode.com/problem/find-bad-version/

http://www.jiuzhang.com/solutions/find-bad-version/

First version that is bad version

Wood Cut

http://www.lintcode.com/problem/wood-cut/

http://www.jiuzhang.com/solutions/wood-cut/

Last/Biggest length that can get >= k pieces

进一步理解二分法

保留有答案的那一半

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)

Related Questions

- Binary Search:
  - http://www.lintcode.com/problem/count-of-smaller-number/
  - http://www.lintcode.com/problem/search-for-a-range/
- Rotate Array
  - 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] → [5,4,1,2,3] → [5,4,3,2,1] → [1,2,3,4,5]

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
- 两类二分法
  - 二分位置 Binary search on index
  - 二分答案 Binary search on result
- 理解二分法的三个层次：
  1. 头尾指针，取中点，判断往哪儿走
  2. 寻找满足**某个条件**的**第一个**或是**最后一个**位置
  3. 保留剩下来一定有解的那一半

点题时间

与二分法相关的最新面试题

http://www.jiuzhang.com/qa/974/

九章算法班跟不上？

换到《Java入门与基础算法班》

http://www.jiuzhang.com/course/7/