如何一个月攻破算法面试
Defeat Algorithm Interview

课程版本 v6.0　　主讲 令狐冲

课件获取：关注左侧微信公众号，回复“算法6”

加课程顾问九妹
了解更多课程信
息和优惠方式

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

优惠活动

TBD

讲师：令狐冲

算法竞赛国家队，多年算法教学经验

曾在2家北美顶尖IT企业就职并担任面试官

国内TOP 1学校毕业

国外顶级Offer 10+ 个 国内 Offer 20+ 个

刷题数超过 **3000** 题

助教团队：

均获得过算法竞赛金奖

刷题数均超过 **1000** 题

随课教程（预习和复习的去处）

九章算法

基础知识先修

重点知识直播

补充知识拓展

注意事项：

- 先修内容通常需要 1~2 小时，务必在课前完成否则难以跟上课程进度
- 补充知识有空的时候看，内容也很重要不能不看
- 有效期为第一节课开始后三个月内

Python 3
人工智能时代的语言

在一场 45 分钟的面试中

Python 相对于 Java 节约 **10** 分钟的 Coding 时间

课程习题参考答案包含主流的 Python, Java, C++

其他语言的同学也可以跟上节奏

算法是通用的，不受限于编程语言的

这个班不是“基础”算法班

这个班需要一定的算法基础，至少 LintCode Easy 的题要会做

最好先上《九章算法基础班（Java）》（限时免费哦！）

https://www.jiuzhang.com/course/23/

《九章算法强化班》是这个班的后序衔接课程，建议也要上

九章基础算法班（Java）

九章算法

第1章 第一章 零基础找 CS 工作如何准备? 试听章节

习题：已做1/全部17　　已做正确率：0%

第2章 第二章 Java基础一 —— 变量，程序控制流 VIDEO 已过期

习题：已做7/全部28　　已做正确率：43%

第3章 第三章 Java基础二 —— 函数，String， 面向对象 VIDEO 已过期

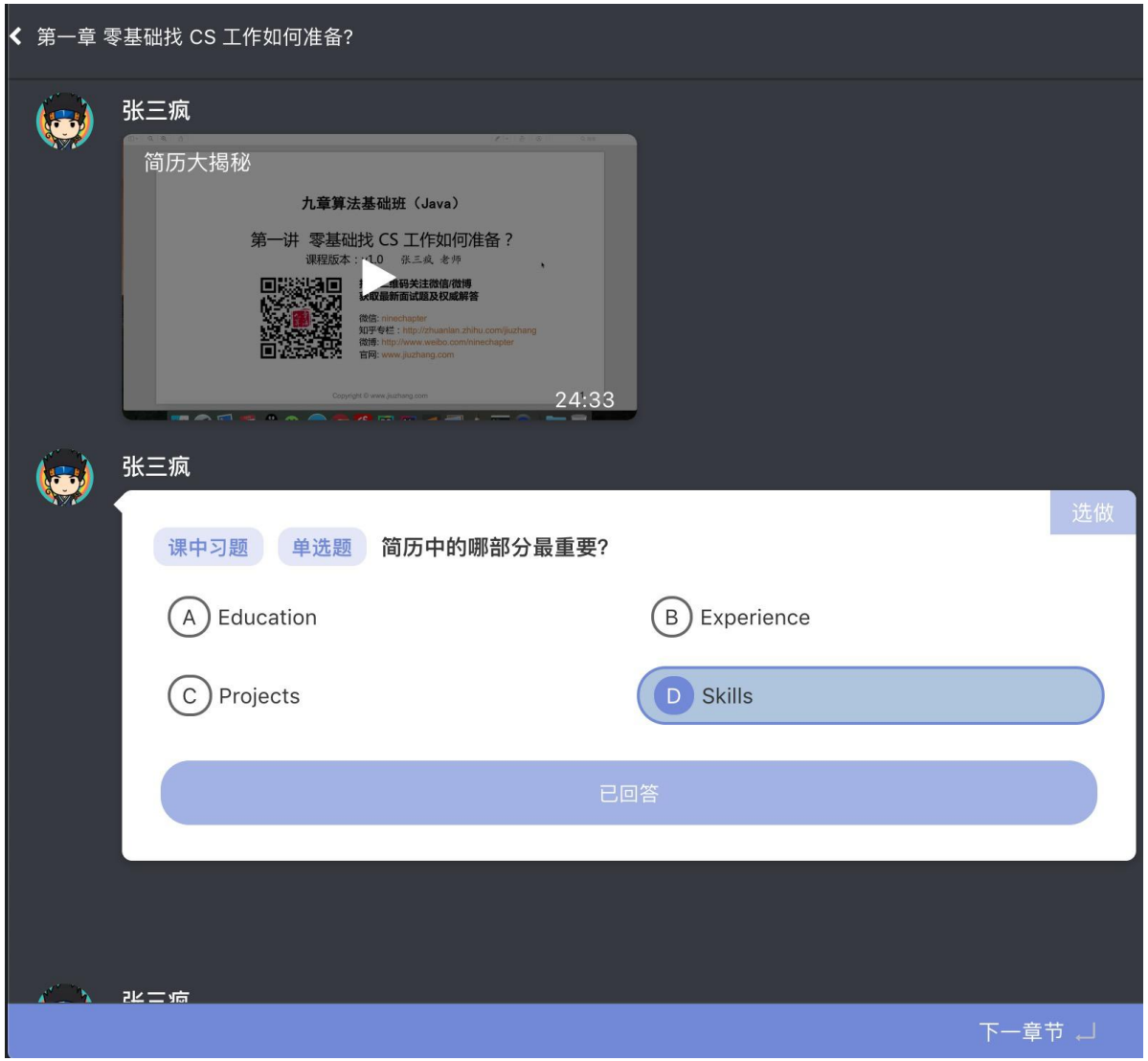
第4章 第四章 Reference、数据结构 VIDEO 已过期

第5章 第五章 链表及其操作，算法的时间复杂度 VIDEO 已过期

第6章 第六章 栈，队列，哈希表 VIDEO 已过期

第7章 第七章 递归及二叉树的深度遍历 VIDEO 已过期

第8章 第八章 二叉树的宽度优先遍历及分治法 VIDEO 已过期



G家常见面试题

开场热身：最长回文子串
Longest Palindromic Substring

G

http://www.lintcode.com/problem/longest-palindromic-substring/

我现在是你的面试官，你可以问我任何问题

打开 www.collabedit.com 开始敲代码

写完的同学可以通过 Zoom 的 QA 功能分享你的代码链接给我

我会挑选一些同学的代码进行点评

写完再解释还是一边写一边解释？

该如何与面试官进行“有效”沟通？

常见错误：我知道有个算法叫 Manacher's Algorithm

该算法可以在 O(n) 的时间内求得最长回文子串，n=字符串长度

这是该问题的最优算法，却绝对不是面试官想让实现的算法

为什么？

因为他自己也不会

```python
class Solution:
    """
    @param s: input string
    @return: the longest palindromic substring
    """
    def longestPalindrome(self, s):
        start, longest = 0, 0
        for i in range(len(s)):
            for j in range(i, len(s)):
                if j - i + 1 > longest and self.is_palindrome(s, i, j):
                    longest = j - i + 1
                    start = i

        return s[start:start + longest]

    def is_palindrome(self, s, i, j):
        while i < j and s[i] == s[j]:
            i += 1
            j -= 1
        return i >= j
```

```python
class Solution:
    """
    @param s: input string
    @return: the longest palindromic substring
    """
    def longestPalindrome(self, s):
        if not s:
            return ""

        for length in range(len(s), 0, -1):
            for i in range(len(s) - length + 1):
                l, r = i, i + length - 1
                while l < r and s[l] == s[r]:
                    l += 1
                    r -= 1
                if l >= r:
                    return s[i:i + length]
        return ""
```

大家来找茬 III

```python
class Solution:
    """
    @param s: input string
    @return: the longest palindromic substring
    """
    def longestPalindrome(self, s):
        if not s:
            return ""

        start, longest = 0, 0
        for middle in range(len(s)):
            # odd
            left, right = middle, middle
            while left >= 0 and right < len(s) and s[left] == s[right]:
                left -= 1
                right += 1
            if longest < right - left - 1:
                longest = right - left - 1
                start = left + 1

            # even
            left, right = middle, middle + 1
            while left >= 0 and right < len(s) and s[left] == s[right]:
                left -= 1
                right += 1
            if longest < right - left - 1:
                longest = right - left - 1
                start = left + 1

        return s[start:start + longest]
```

基于中心点枚举法 Enumeration 的最佳实践


```python
class Solution:
    """
    @param s: input string
    @return: the longest palindromic substring
    """
    def longestPalindrome(self, s):
        if not s:
            return ""

        self.start, self.longest = 0, 0
        for middle in range(len(s)):
            self.find_longest_palindrome_from(s, middle, middle)
            self.find_longest_palindrome_from(s, middle, middle + 1)

        return s[self.start:self.start + self.longest]


    def find_longest_palindrome_from(self, s, left, right):
        while left >= 0 and right < len(s) and s[left] == s[right]:
            left -= 1
            right += 1
        if self.longest < right - left - 1:
            self.longest = right - left - 1
            self.start = left + 1
```

Follow up: 不能枚举中心点

基于第一种方法，如何进行优化？

基于动态规划 Dynamic Programming 的最佳实践


```python
class Solution:
    """
    @param s: input string
    @return: the longest palindromic substring
    """
    def longestPalindrome(self, s):
        if not s:
            return ""

        n = len(s)
        is_palindrome = [[False] * n for _ in range(n)]
        for i in range(n):
            is_palindrome[i][i] = True
        for i in range(1, n):
            is_palindrome[i][i - 1] = True

        start, longest = 0, 1
        for length in range(2, n + 1):
            for i in range(n - length + 1):
                j = i + length - 1
                is_palindrome[i][j] = is_palindrome[i + 1][j - 1] and s[i] == s[j]
                if is_palindrome[i][j] and length > longest:
                    longest = length
                    start = i

        return s[start:start + longest]
```

从最长回文子串中我们可以得到的面试考点

- 面试不一定会要求你用最优复杂度的算法来解决问题
  - 因此单纯只刷LC之类的OJ，容易让你产生一定要用最优解来解决这样的误区
- 代码真的不是写出来就可以过
  - 代码质量（Coding Quality）很重要
  - 好的代码质量包括：
    - Bug Free
    - 好的代码风格（Coding Style），包括变量名命名规范有意义，合理的使用空格，善用空行
    - 容易让人读懂的逻辑。要把复杂的事情用简单的方式，别把简单的事情写复杂了。
    - 没有冗余代码
    - 有边界检测和异常处理

独孤九剑 —— 总决式

想要做到 Bug Free 最重要的是优化你的 Coding Style
技巧：子函数 + 好的命名风格

Longest Palindromic Substring 的全部算法及时间复杂度

Manacher's Algorithm - O(n) // **学有余力**可以阅读全文并背诵

后缀数组 Suffix Array - O(n) // 完全不用学

动态规划 Dynamic Programming - O(n^2) // 必须掌握

枚举法 Enumeration - O(n^2) // 必须掌握

参考代码：

http://www.jiuzhang.com/solution/longest-palindromic-substring/

今后所有课上讲的题目的参考答案都可以在这里查询到

从 Longest Palindromic Substring 看面试的评分体系

- **Strong Hire**
- 使用 O(n) 或者 O(nlogn) 的算法实现出来 (Manacher's Algorithm or Suffix Array)，并且代码质量合格，无 Bug 或者 有很小的bug但是能自己发现并解决，无需太多提示
- **Hire**
- 能够分别使用枚举法和动态规划实现时间复杂度 O(n^2) 的算法。并且代码质量优秀，无Bug，无重复代码，无需面试官给提示
- **Weak Hire**
- 只使用了其中一种 O(n^2) 的算法实现出来，代码质量还不错，可以有一些小 Bug，面试官可以给一些小提示
- **No Hire**
- 只能想出一种 O(n^2) 的算法，但是 Bug 太多，或者需要很多提示
- **Strong No Hire**
- 连一种 O(n^2) 的算法都想不到

面试评分和 Offer 的关系

有 >= 1 个 Strong No Hire => No offer

有 >= 2 个 No hire => No offer

有 1 个 No Hire + 1 个 Weak Hire => No Offer

有 1 个 No Hire，其他都是 Hire => Offer or 加面（取决于公司招人多不多，门槛高不高）

有 1 个 Weak Hire => Offer or 加面

特殊情况：

一个 Strong Hire + 一个 Strong No Hire => 开个会一起讨论一下，通常结果是加面或者No Offer。

Implement strStr

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

在一个字符串中查询另外一个字符串第一次出现的位置

常见错误：
我知道一个算法叫做KMP

A同学: 论坛上有人说考到了KMP呢！你骗人！

真问我比O(n^2)更好的算法怎么办？

这个概率只有1%

可以学习一个比KMP算法更简单的算法: Rabin-Karp

http://www.jiuzhang.com/video/rabin-karp

休息 5 分钟

福利发放

上课须知

九章算法

- 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
  - 良苦用心希望同学们理解
- 不允许建私微信群
  - 在学员群中拉人私下组群的将被踢群并不再提供学员答疑服务
- LintCode 需要单独先注册一个账户，不要使用九章的账号密码去登陆
- LintCode 阶梯训练必须先完成上一节课的作业，才能做下一节课的作业
- 付费学员微信群（助教提供答疑服务）
  - 官网右上角我的课程
  - https://www.jiuzhang.com/accounts/profile/
- 新学员必读常见问题解答
  - http://www.jiuzhang.com/qa/3/

各类服务有效期

- 以下服务有效期为**一年（按第一节课上课时间起算）**
  - 课件
  - LintCode阶梯训练访问权限（不含 LintCode VIP）
- 以下服务有效期为**3个月（按第一节课上课时间起算）**
  - 随课教程
  - 学员群答疑

v6.0 版本有何更新？

新增两节动态规划课程

换血一些新题和高频题

调整课程讲解顺序和内容组织架构

刷题刷到什么程度去面试才够？

你永远没有觉得自己准备好的那一天！

LintCode 可以帮你解决烦恼！

LintCode 可以帮你！

三个维度：

1. 算法能力
2. Bug Free 能力
3. 题量

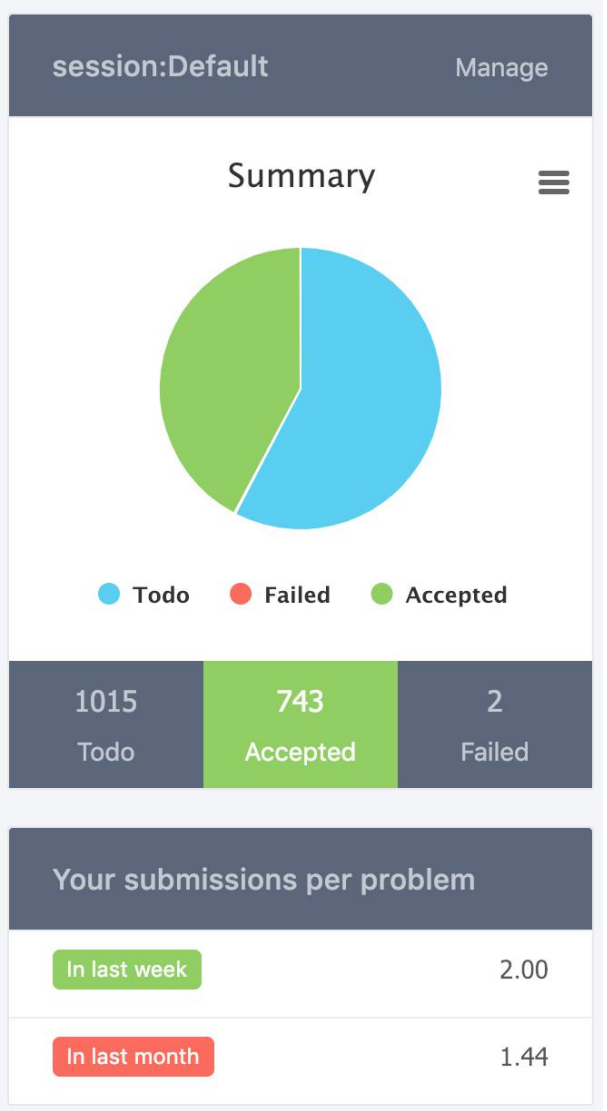
很多人只关心第三个维度，但这个维度是最弱的维度

如何评估算法能力？LintCode CAT 来帮你！

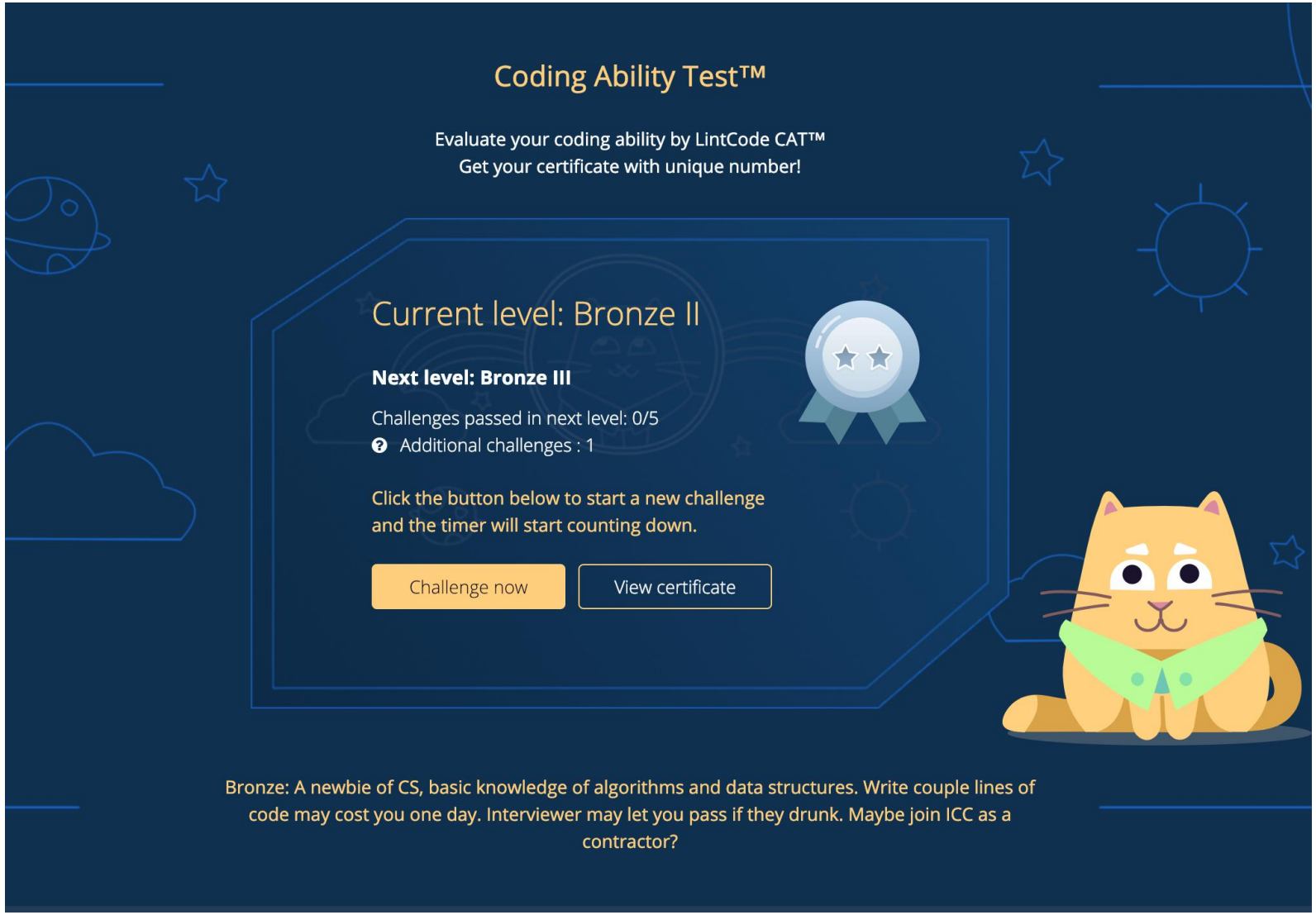
https://www.lintcode.com/cat/

如何评估 Bug Free 的能力？每道题的平均提交次数

https://www.lintcode.com/problem/



LintCode CAT - 按照算法难易程度升级打怪



LintCode CAT - 想拿到 Offer，至少要刷到黄金

LintCode CAT - 拥有属于你的 Coding 能力证书

如何一个月攻破算法面试？

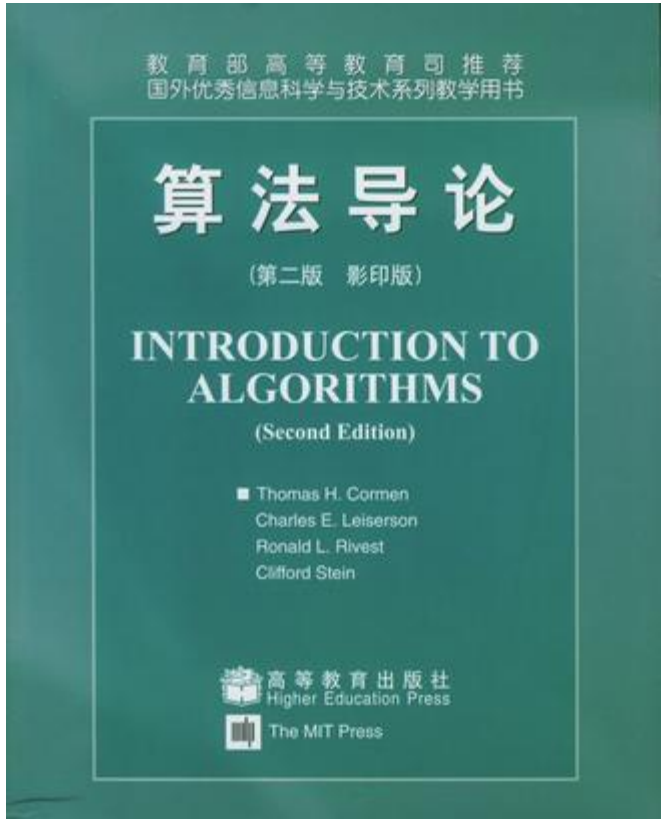
上面我们讲了面试的软技巧，学会了如何与面试官沟通，如何通过短时间的训练，让你的代码看上去不像一个初学者

下面我们来讲，你到底怎样才不至于在面试的时候什么都写不出来

面试算法 vs 算法

如果你还在看算法导论？赶紧扔掉

——我宁可你看的是 《Cracking The Coding Interview》



也请不要去看普林斯顿的算法公开课

——很多内容面试依然不考，或考得很少

为什么？

——面试算法 != 算法

算法面试最“虚”的部分

不知道的算法那么多

你根本不知道可能考到什么样的问题

如果让你给算法面试“划考点”

请列举你觉得会考的知识点（算法与数据结构）

到目前为止，下面哪些算法和数据结构，不在面试考察范围内？

九章算法

最短路算法
Dijkstra / Floyd / SPFA

拓扑排序算法
Topological Sorting

Morris 算法
O(1)额外空间前序遍历

贪心法
Greedy

Manacher 算法
求最长回文子串

KMP算法
strstr / indexOf

最小生成树算法
Minimum Spanning Tree

二分法
Binary Search

分治法
Divide & Conquer

网络流算法
Network Flow

希尔排序
Shell Sort

动态规划
Dynamic Programming

线段树
Segment Tree

平衡排序二叉树
如 Red-black Tree

字典树
Trie

并查集
Union Find

跳跃表
Skip List

哈希表
Hash Table

堆
Heap

KD树
KD-Tree

B树/B+树
B-Tree / B+ Tree

二叉查找树
Binary Search Tree

越红考得越多，灰色不考或者出现概率低于千分之一

九章算法

最短路算法
Dijkstra / Floyd / SPFA

拓扑排序算法
Topological Sorting

Morris 算法
O(1)额外空间前序遍历

贪心法
Greedy

Manacher 算法
求最长回文子串

KMP算法
strstr / indexOf

最小生成树算法
Minimum Spanning Tree

二分法
Binary Search

分治法
Divide & Conquer

网络流算法
Network Flow

希尔排序
Shell Sort

动态规划
Dynamic Programming

线段树
Segment Tree

平衡排序二叉树
如 Red-black Tree

字典树
Trie

并查集
Union Find

跳跃表
Skip List

哈希表
Hash Table

堆
Heap

KD树
KD-Tree

B树/B+树
B-Tree / B+ Tree

二叉查找树
Binary Search Tree

2013-2019 的面试难度变化

名词中英文对照：

动态规划 - Dynamic Programming
链表 - Linked List
递归 - Recursion
二叉树 - Binary Tree
二分法 - Binary Search
深度优先搜索 -Depth First Search (DFS)

面试常见知识点的考察频率，学习难度

九章算法

算法/数据结构	大公司考察频率	其他公司考察频率	难度	建议刷题数	性价比	包含在哪些九章课程中
字符串，模拟法	高	高	低	20~50	中	九章基础算法班
二分法	高	高	中	10~20	高	九章算法班，九章算法强化班
二叉树，链表	高	高	低	30~50	高	九章算法班，九章基础算法班
递归，DFS	高	高	高	20~40	中	九章算法班，九章算法强化班
BFS，拓扑排序	高	高	中	5~10	超高	九章算法班
堆（优先队列）	低	低	中	5~10	中	九章算法班，九章算法强化班
哈希表	高	高	中	10~30	高	九章算法班
两根指针	高	高	中	10~20	高	九章算法班，九章算法强化班
动态规划	中	低	高	40~60	低	九章算法班（入门） 九章算法强化班（部分） 动态规划专题班（全部）
字典树	中	低	低	2~5	高	九章算法强化班
并查集	低	无	低	2~5	高	九章算法强化班

对这门课的错误期望

老师把300题一题一题给我讲一遍，带着我都写一遍

—— 臣妾做不到，9节课18个课时，每节课只能”精讲2个题+粗讲7-8个题“。如有这个需求 Youtube 上一大堆的视频等着你

把 Facebook, Google, Amazon, Microsoft 等公司的所有明天就会面的题目给我讲一遍，让我面试能够押到题

—— 押题你可以自己来，网上随时能搜到很多面经，LC高级账号买一个也能解决你的需求

我上完课一定得拿到 Offer，否则这个课就是辣鸡

—— 我能带你学会一个知识，但无法保证你拿到 Offer。面试是 5 分运气，5 分实力。带着尽人事知天命的心态去准备面试，才能更从容的应对。

老师会把那些我平时练习的时候不会做的很难的题讲给我听

—— 你只有 1% 的几率碰到这样的题，性价比不高的东西我不会放在课上来讲。课上讲述的是那些最高频，性价比最高的**知识点**。这些高频知识点的题目不一定很难。

老师会把所有要考的知识点都讲一遍

—— 因为课时限制，这门课只涵盖 70% 的最高频的知识点。

—— 《九章算法班》+《九章算法强化班》会覆盖 99% 的面试中的算法与数据结构知识点

—— 有 1% 的知识点虽然有的公司考过，但是很低频。

《九章算法班》能带给你什么？

节约时间

你自己需要三个月才能准备下来的东西

我用一个月带你准备好

其他算法视频 vs 九章算法直播课

九章算法

	其他算法视频	九章算法直播课
形式	录制，滋长惰性，学习的时候无人可以问问题。虽然可以反复观看，但是不懂的东西直接问人是更节约时间的。反复观看只是浪费时间，不懂的还是不懂。	直播，定时定量学习，没有再来一次的机会会更加珍惜和集中精力。课程配备直播助教实时答疑
氛围	一个人在战斗，很难坚持下去	你不是一个人在战斗，学员QQ群里一起学习，学习积极性更高
课后	看不懂最多只能反复看视频，课后遇到新问题无人可以帮忙解决	课上没有掌握的知识，平时学习遇到的问题，都可以在QQ群，问答板块问老师，问助教
内容	陈题，没有面试官角度的分析，没有面试技巧编程技巧的讲解，没有题目在面试中评价标准的分析，通常是对单个题如何解决的讲解，通常只讲一个解法，知识没有连贯性，跳跃极大	永远是最新内容，从面试官角度分析，讲算法的同时讲解面试技巧和编程技巧，通常是由知识点带动题目讲解，学会如何解决一类问题而不是一个问题。知识点连贯性强，学习流程更加科学化
师资	作者一般只刷过 200-300 题	3000+ 的刷题经验，算法竞赛国家集训队员，ACM竞赛金牌
题库	不配套题库，或需要对题库进行额外付费	课上所涉及的题目，包括相关练习题约200道题，无需对题库重复付费，一年之内可以随便刷
评测	无，你根本不知道自己学得好不好	平时作业 + 期末考试，检验自己学习的水平，更有底气去面试 优秀学员还可以获得 FLAG 等企业的内推机会
价格	免费的东西是最贵的，因为你浪费了时间，不付钱你也不会珍惜	便宜的价格，不便宜的质量

我们卖的不是视频，而是服务

即便你搞来了九章往期的盗版视频（或者你正在观看盗版视频）

你也远远达不到九章直播课的学习效果

大纲和上课时间

http://www.jiuzhang.com/course/1/

注意时区，特别是周几上课，**错过不补**

这门课适合谁？

只要你有求职需求，只要你面试会面算法

无论是的主语言是 Java 还是 C++, Python

只要你先节约求职准备时间，提高求职准备效率

你可以获得哪些学员权限？

- LintCode专属阶梯训练题
  - 226 道精选题
  - 50+ 私有练习题
- 九章问答提问有专人回答
  - 助教老师100%回答
- 九章问答课程与内推板块浏览权限
  - 最新最热面试题面经实时分享
  - 让九章老学员帮你内推各大公司
- 九章课程学员群
  - 与同学们实时交流学习问题
  - 随时 @老师 @助教 答疑解惑
  - 认识更多志同道合的朋友，一起打鸡血
  - 学员线下活动（自行组织）

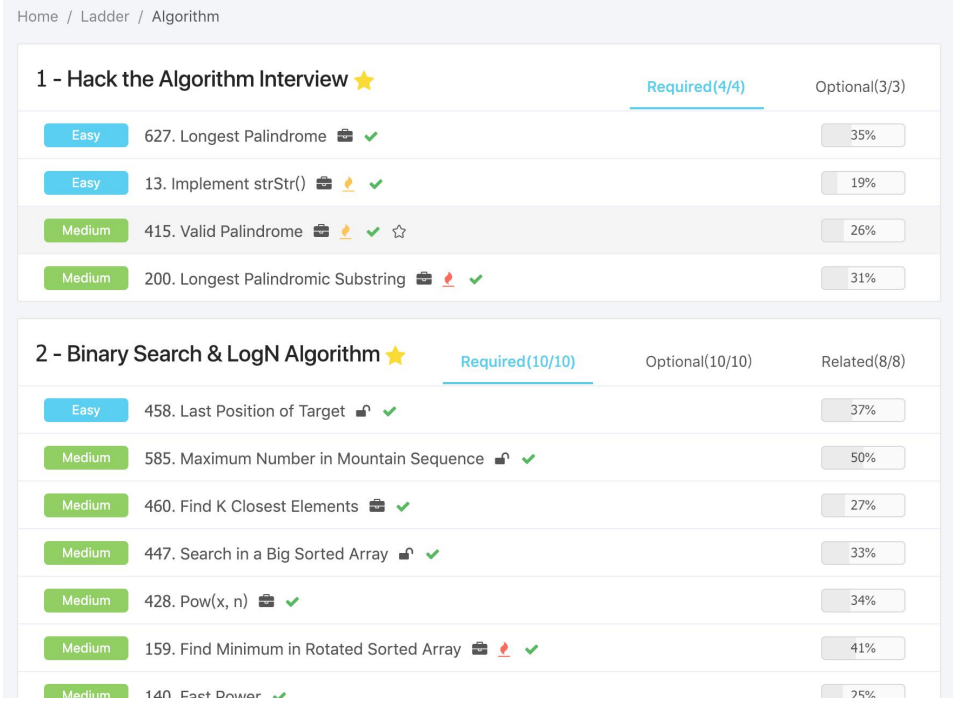
课程配套的 LintCode 阶梯训练

访问地址：

http://www.lintcode.com/ladder/1/

玩法：

解决上一节课的阶梯中的必做题（Required）才能看到下一节课的题目列表



Required - 通常是课上讲过的题

Optional - 其他近期高频的该类算法问题，但课上没讲

Related - 其他与该类算法或者某些题目相关的值得一练的题

你需要先注册一个 LintCode 账号，九章账号无法直接登录

有效期 1 年（自开课之日起算）

其他课程

完整课程列表请见：http://www.jiuzhang.com/course/

6 门 算法课程

5 门 1对1 私教服务课程

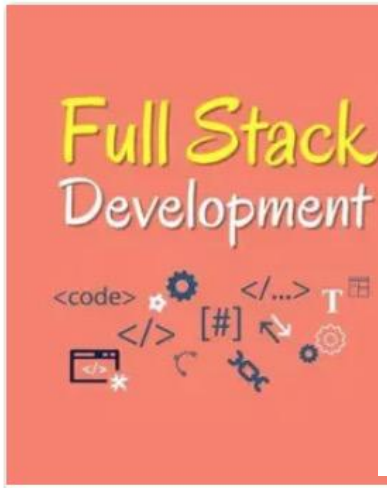
4 门 小班/1对1 VIP 集训营课程

9 门 直播课程

2 门 录播互动课程

2 门 大数据课程

4 门 项目课程



全栈开发基础项目课

Big Data 项目

硅谷工程师教你从零开始
Data!

系统设计班

系统设计面试是常
型，特别是针对后
工程师是必须的面试环节

面向对象设计专题班

面向求职，理论与实践并重，全网第一门可以刷题的Object-oriented design(OOD)课程！

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

课后拓展阅读

必读：

Google Coding Style: https://google.github.io/styleguide/javaguide.html

Rabin Karp: http://www.jiuzhang.com/video/rabin-karp

函数式编程：https://www.zhihu.com/question/28292740

选读：

Manacher’s Algorithm:

https://segmentfault.com/a/1190000003914228

https://www.geeksforgeeks.org/manachers-algorithm-linear-time-longest-palindromic-substring-part-1/

Q & A

常见问题 http://www.jiuzhang.com/qa/3/

加课程顾问九妹咨询更多课程信息