FB面试官揭秘面试速成技巧——
如何做到 Bug Free 和刷100题=刷300题

主讲人 令狐冲

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

我们是谁？

九章算法于 2013 年由 3 位来自 Facebook 和 Google 工程师成立于美国硅谷。致力于帮助更多中国人找到好工作，用技术助力中国科技行业腾飞！

九章的合作讲师均来自 Facebook, Google, Apple, Amazon, Microsoft, Alibaba, Bytedance 等世界 500强 IT 企业。

九章已经服务超过 30000+ 学员拿到自己心仪的 Offer。

九章开设了数十门 IT 求职课程，包含面试算法，人工智能，大数据，Python，Java ，Web 前端，面向对象，系统设计等方向。

九章和其他培训机构的的最大区别

1. 课程均为面试求职导向，只讲面试需要的知识，不讲浪费时间的

2. 严把课程质量和师资质量，在工程师级别和授课质量上都有很高的要求
   a. 工程师级别至少是脸书E5/谷歌L5/阿里P7/腾讯T10
   b. 算法老师要求 NOI/ACM 金牌且刷1000+题
   c. 课程试讲评分 4.5+ （满分5分）

讲师&助教介绍

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

今天聊些什么？

面试官眼中的求职者和评价体系

如何跟面试官正确的沟通

刷题刷到什么程度去面试才够

如何才能修炼 Bug Free 的能力

面试算法的考察范围到底是什么

去FMAG这些大厂是不是一定要刷难题

SDE/SWE/MLE/DS/FTE等岗位的算法面试难度有什么不同

算法以外，还有哪些类型的面试？

OOD 要会些什么

什么岗位会问 System Design

Behavior Question 考什么

如何做到刷100题=别人刷300题

面试官眼中的求职者

HR 主要通过你的衣着，谈吐和外貌来看你

而技术面试官则是通过你的代码来看你

写代码不超过 3 个月

```
for(int i=0;i<n;i++){
for(int j=0;j<n;j++){
for(int k=0;k<n;k++){
for(int l=0;l<n;l++){
                    ...
}
}
}
}
```

```
for i in range(n):
    for j in range(n):
        for k in range(n):
            for l in range(n):
                # ...
```

代码从没被人 Review 过

```
if (grid[i][j] == 1) {
    ...
} else if (grid[i][j] == 2) {
    ...
}
```

```
if grid[i][j] == 1:
    # do something
elif grid[i][j] == 2:
    # do something
```

写代码经验太少

```java
int i = 0, j = 0, k = 0;
int[] arr3 = new int[arr1.length + arr2.length];

while (i < arr1.length || j < arr2.length) {
    if (arr1[i] < arr2[j]) {
        arr3[k++] = arr1[i];
    } else {
        arr3[k++] = arr2[j];
    }
}

return arr3;
```

```python
def merge(list1, list2):
    i, j = 0, 0
    list3 = []
    while i < len(list1) or j < len(list2):
        if list1[i] < list2[j]:
            list3.append(list1[i])
            i += 1
        else:
            list3.append(list2[j])
            j += 1
    return list3
```

缺乏项目经验

耦合度高

重复代码

全局变量

代码耦合度太高，逻辑交杂在一起

这是今后我们会学到的克隆图问题

https://www.lintcode.com/problem/clone-graph/

使用的是 BFS 宽度优先搜索算法右边的代码

一边做宽度优先搜索找到所有的点

一边又复制所有的点

一边又复制所有的边

并且在复制边的时候又复制点

```python
def cloneGraph(self, node):
    if not node:
        return None
    queue = [node]
    start = 0
    mapping = {}
    while start < len(queue):
        curt_node = queue[start]
        start += 1
        if curt_node in mapping:
            new_node = mapping[curt_node]
        else:
            new_node = UndirectedGraphNode(curt_node.label)
            mapping[node] = new_node
        for neighbor in curt_node.neighbors:
            if neighbor in mapping:
                new_neighbor = mapping[neighbor]
            else:
                new_neighbor = UndirectedGraphNode(neighbor.label)
                mapping[neighbor] = new_neighbor
                queue.append(neighbor)
            new_node.neighbors.append(new_neighbor)
    return mapping[node]
```

代码耦合度（Coupling）高容易导致：

- 难维护
- 难读懂
- 易出错

解决办法：解耦合 Decouple

拆散一对是一对

劝分不劝合

更好的实现方法

将整个算法分解为三个步骤：

1. 找到所有点
2. 复制所有点
3. 复制所有边

```python
def cloneGraph(self, node):
    if not node:
        return None

    # step 1: find nodes
    nodes = self.find_nodes_by_bfs(node)
    # step 2: copy nodes
    mapping = self.copy_nodes(nodes)
    # step 3: copy edges
    self.copy_edges(nodes, mapping)

    return mapping[node]
```

```python
def find_nodes_by_bfs(self, node):
    queue = collections.deque([node])
    visited = set([node])
    while queue:
        curt_node = queue.popleft()
        for neighbor in curt_node.neighbors:
            if neighbor in visited:
                continue
            visited.add(neighbor)
            queue.append(neighbor)
    return list(visited)

def copy_nodes(self, nodes):
    mapping = {}
    for node in nodes:
        mapping[node] = UndirectedGraphNode(node.label)
    return mapping

def copy_edges(self, nodes, mapping):
    for node in nodes:
        new_node = mapping[node]
        for neighbor in node.neighbors:
            new_neighbor = mapping[neighbor]
            new_node.neighbors.append(new_neighbor)
```

卖个关子

重复代码和全局变量的问题

我们将在后面的课程中陆续学习到

你认为的 Code Quality：代码要加注释

使用含义清晰的变量名命名+简单易读的处理逻辑

>>

用注释去解释让人看不懂的代码

你认为的 Code Quality：代码越短越好

通过适当的子函数化的代码包装，多加空行

虽然代码更长了，但是能够让你的代码：

易读，易维护，不易错

好的代码质量真的那么重要么？

代码就像一件艺术品，越是高级的程序员，越有代码洁癖

越是高级别的面试官，越在意你的代码质量

好的代码质量真的那么重要么？

拥有好的代码质量，会让面试官在心里为你默默加分

如果质量很差，面试官会在心里为你默默扣分

最终是否导致 Hire / No Hire，就是一个量变引起质变的问题

好的代码质量真的那么重要么？

拥有好的代码质量，还能够让你的代码少出 BUG

你以为只需要细心就可以不出 BUG

但是通过**子函数化、避免全局变量**等手段可以让你出 BUG 的概率大大降低

面试评价体系

Coding(Algorithm) Interview 的评价体系主要有如下一些方面

Logicality / Code Quality / Communication

面试评价体系

Logicality 逻辑思维能力

1、是否能很快的想到一个 Work Solution
2、是否能够在面试官点出问题后优化自己的 Solution

Code Quality 代码质量

1、代码到底写完没有
2、代码风格好不好
   a、可读性
   b、变量名、函数名命名
   c、空格与空行的正确使用
3、异常检测
4、Bug Free

Communication 沟通能力

把面试官当作 Co-worker 而不是考官

让面试官愿意和你一起工作

几个面试沟通法则

•做一个题之前，先沟通清楚，得到面试官肯定，再开始写代码，写完以后再解释

- 不要闷头写
- 也不要一边写一边解释太多（容易写不完）

•别和面试官吵架

- 面试官带着答案来面试你的
- 不同意见在大部分情况下，都是你自己想错了

几个面试沟通法则

- 可以要提示，经过提示做出来的题，也是可以拿到 Hire 的
  - 但是先自己努力想一下，别太容易放弃，容易让人觉得不会主动思考问题

- 会就会，不会就不会，不要遮遮掩掩，坦诚很重要
  - 容易让人觉得和你沟通“不顺畅”
  - 做过的题就说做过，不要故意说没做过
  - 因为他既然已经怀疑你做过了，即使你说没有，他也无法打消这个顾虑，还不如让他换题

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

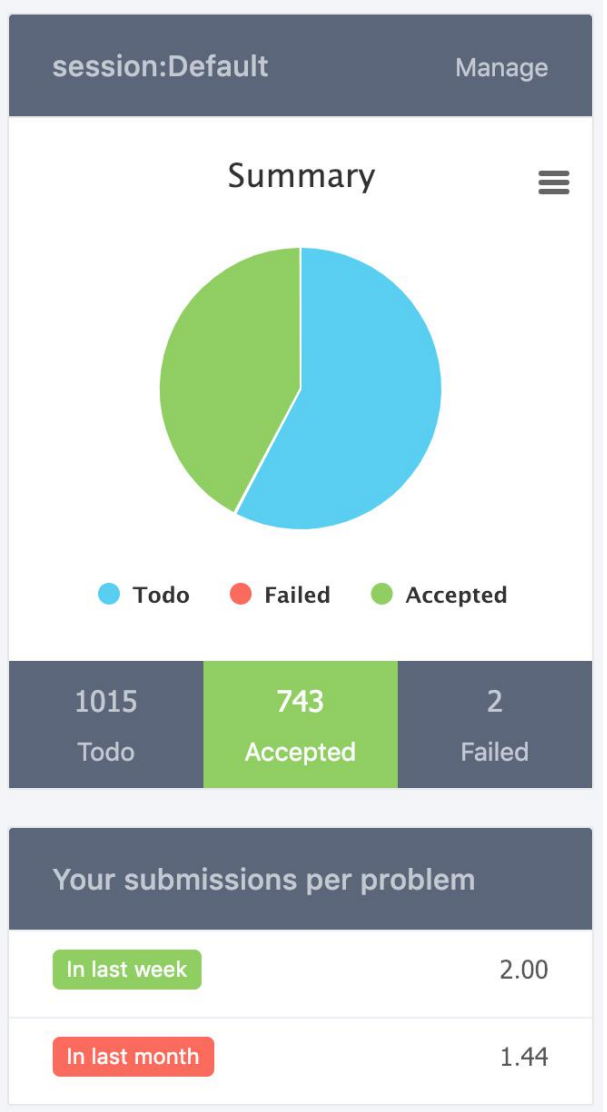
很多人只关心第三个维度，但这个维度是最弱的维度

如何评估算法能力？LintCode CAT 来帮你！

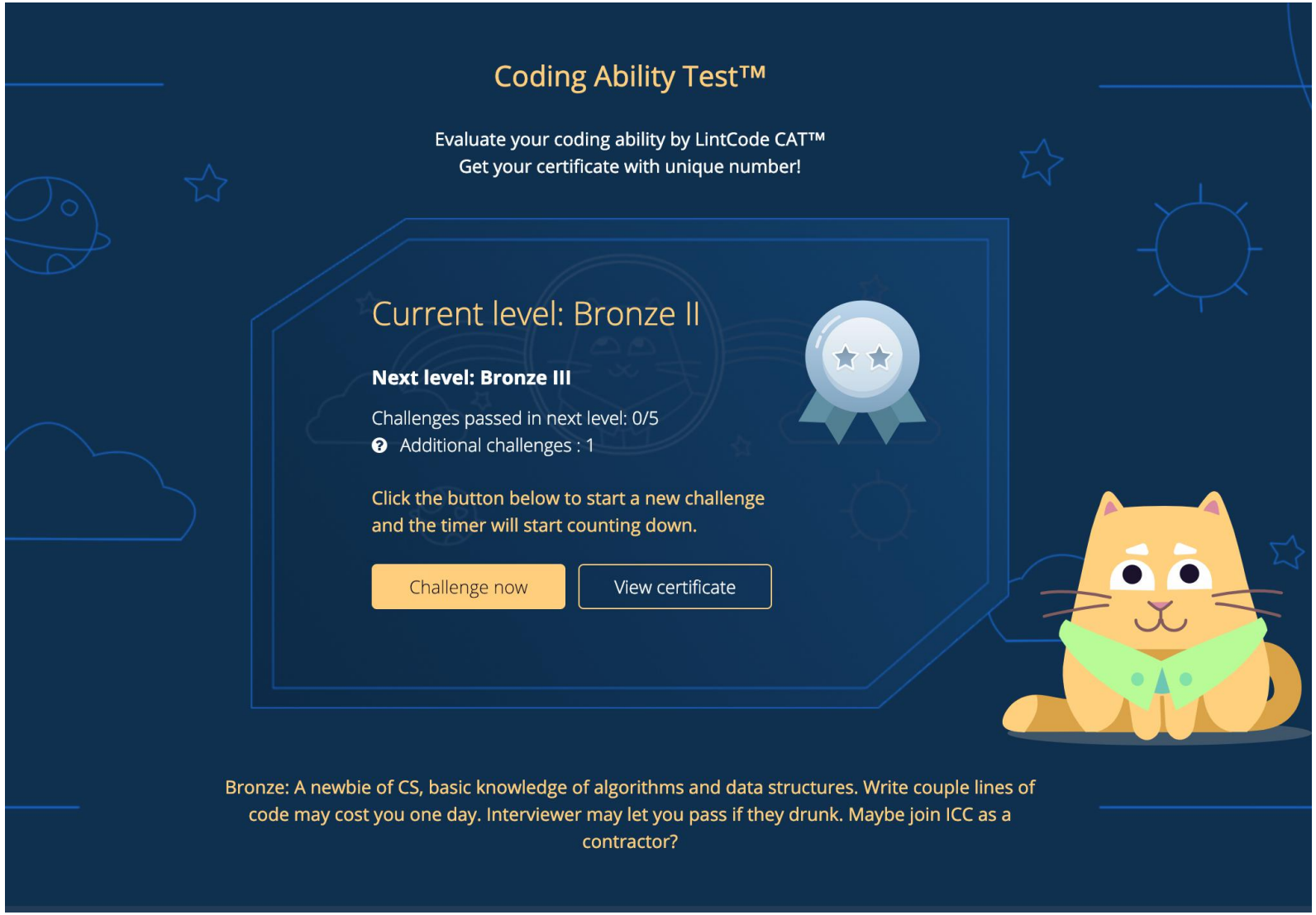
https://www.lintcode.com/cat/

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

面试算法和算法有什么区别

你还在看算法导论么？

面试算法 vs 算法

如果你还在看算法导论？赶紧扔掉

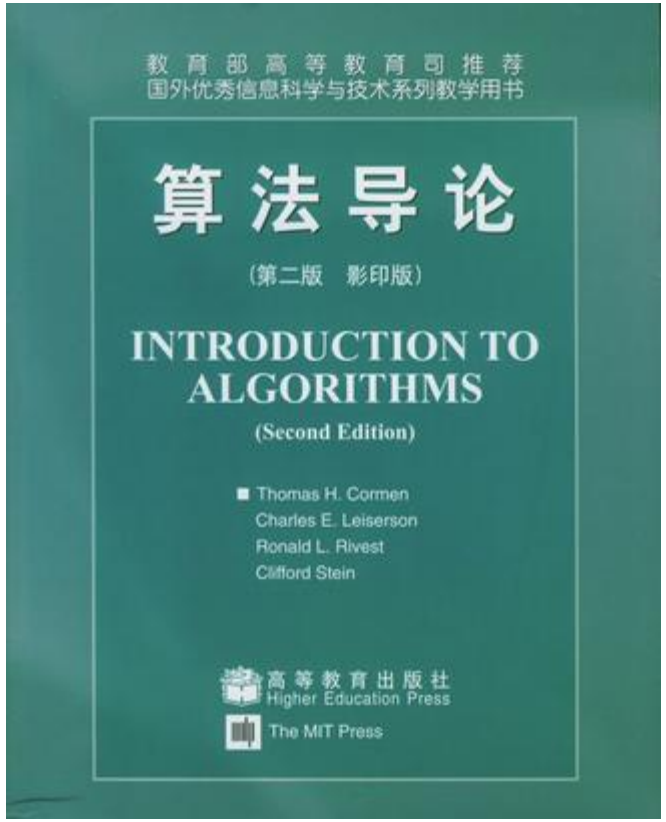
——我宁可你看的是 《Cracking The Coding Interview》

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
www.jiuzhang.com

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

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

一个判断某算法考不考的技巧

带名字的都不考

Morris / Dijkstra / Floyd / Manacher / Tarjan / Dinic / KMP

九章成立以来 2013-2020 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)

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
排序算法	中	高	中	2~5	高	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
双指针	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 / 并查集	中	低	低	2~5	高	九章算法强化班

去 FMAG 是否一定要刷难题？

Facebook, Microsoft, Amazon / Apple, Google

这些公司的面试题难度到底如何？

去 FMAG 是否一定要刷难题

九章算法
www.jiuzhang.com

Google

(活少钱多)

★ 必须要刷难题!

★ 算法考察范围很广，特别喜欢DP和红黑树 Red-black Tree，线段树 Segment Tree

Microsoft

(活多钱多)

facebook

(活多钱少)

amazon

(活少钱少)

★ 刷中等题就够了

★ 算法考察范围见 Cheat Sheet

其他中小厂

★ 刷中等题就够了

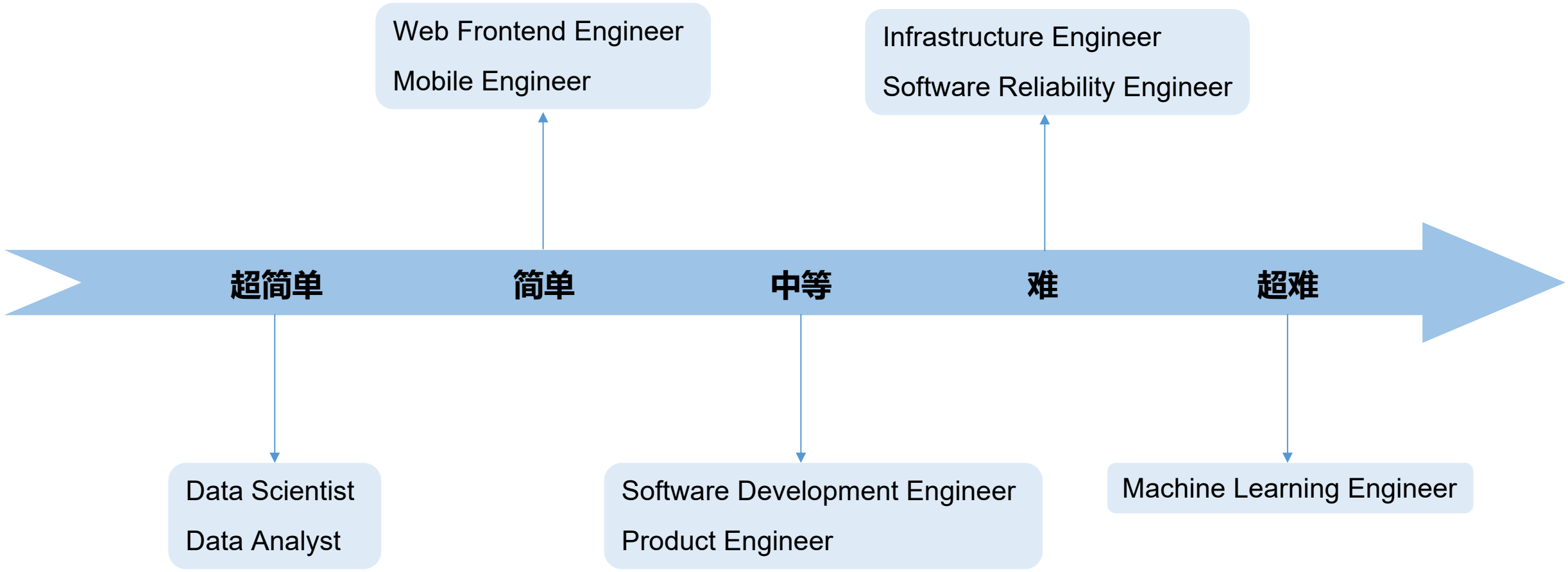
★ 算法面试考察范围很窄 Binary Tree, LinkedList, String, Array 这些基本数据结构相关的题掌握即可

不同的岗位算法面试难度不同么

是的，不同

越后端越难，越前端越简单

不同岗位的算法面试难度



技术岗除了面算法题
还有其他哪些类型的面试？

系统设计 System Design / Architechture Design

面向对象设计 Objected Oriented Design

行为面试 Behavior Question

简历面试 Experience Interview

System Design 考么？

系统设计 System Design 是一种非常常见的面试形式

通常出现在后端相关的岗位中

前端工程师和数据工程师一般不太考察

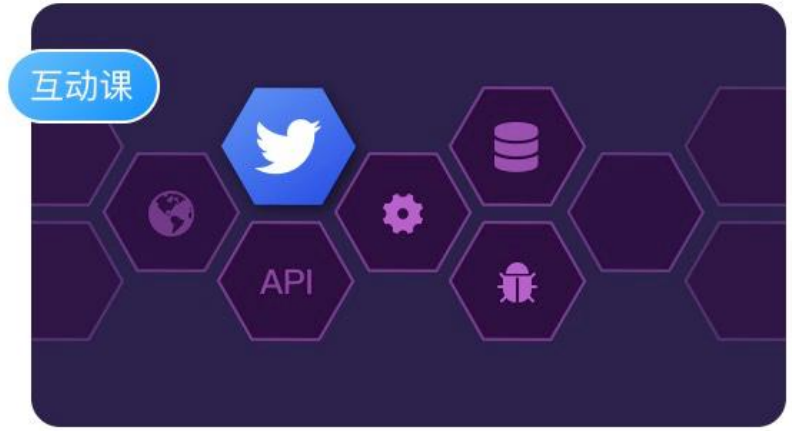
应届生也可能会考一些简单的系统设计

全网最专业的系统设计课程

包含10个章节，20课时的课程

涵盖如下高频系统设计知识点：

- Web知识
- 缓存 Memcached / Redis
- 关系型数据库 MySQL / PostgreSQL
- 非关系型数据库 Cassandra
- 分布式数据库系统 Big Table
- 分布式计算系统 Map Reduce
- 分布式文件系统 GFS
- 爬虫 Crawler
- API设计 / Restful
- 消息队列 Message Queue
- 推拉模式 Push & Pull
- Web Socket
- LBS



系统设计 System Design

系统设计面试是常见的一种面试类型，
特别是针对后端工程师，全栈工程师...

什么岗位考 OOD？

互联网公司一般考 System Design 多，软件公司一般考 OOD

互联网公司 = Facebook / Google / Amazon 之流

软件公司 = Microsoft / Apple / Oracle 之流

应届生 New Grad 通常会考 OOD 多于 System Design

面试导向的 OOD 课程

5 个章节 10 个课时

掌握 OOD 高频面试题和知识点

- 设计模式
- SOLID 原则
- 电梯设计
- 停车场设计

面向对象设计 OOD

应届生及亚马逊面试必考，IT求职必备基础

Behavior Question 考什么？

BQ 面试是现在各大公司技术岗位几乎必须面的一轮面试

通常是其他非技术岗位的人面试你

比如 HR, Product/Project Manager, Designer

你为什么来我们公司?

- **错误:** 因为三番天气好，因为离家近，因为活少钱多
- **一般:** 因为认可公司的文化，喜欢公司的使命，这里有很多优秀的员工
- **优秀:** 我在 Facebook 上认识了我的初恋女友，我被 Facebook 连接世界的使命深深的感触到了，我也发现了很多 FB 的 BUG，我想来改掉

你为什么离开现在的公司?

- **错误:** 因为老板是傻逼，因为同事傻逼
- **一般:** 因为没有成长空间了，要寻求更好的发展
- **优秀:** 我非常喜欢上一家公司，之前的老板也非常希望我能够留下来，这是一个很难的决定，我希望寻求改变，之前的公司我已经基本熟悉各类业务，能够带团队了，现在希望能够加入到贵司获得更大的成长，因为贵司的上升空间更大，能够提供一个更大的平台让我实现我的价值

- **如何去平衡帮助新人和完成自己的工作?**

- **错误**：优先完成自己的工作的同时，利用空余时间去帮助新人
- **一般**：我很乐于去帮助新人，我会根据事情的优先级去安排和平衡
- **优秀**：帮助新人也是我的工作之一，如果影响到了我自己的工作，我就自己加班完成

你还有什么想问我的?

- **错误**：工作压力大吗？加班吗？年假几天？有401k吗（五险一金）？
- **一般**：你们公司的技术栈是啥？我加入的团队多少人？
- **优秀**：我如何能够参与到更重要的项目中去？我提前可以为这份工作做哪些准备？

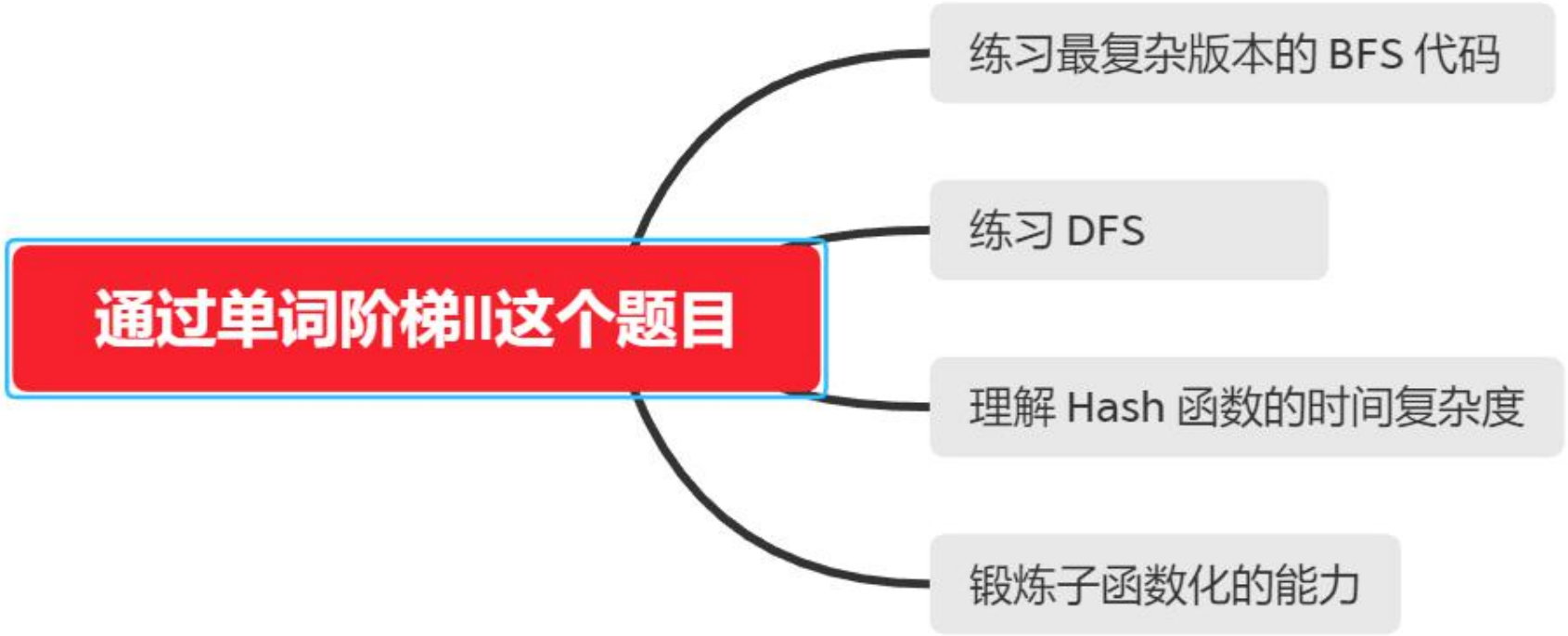
如何做到刷100题=别人刷300题

九章算法精选 100 道有代表性的面试题目

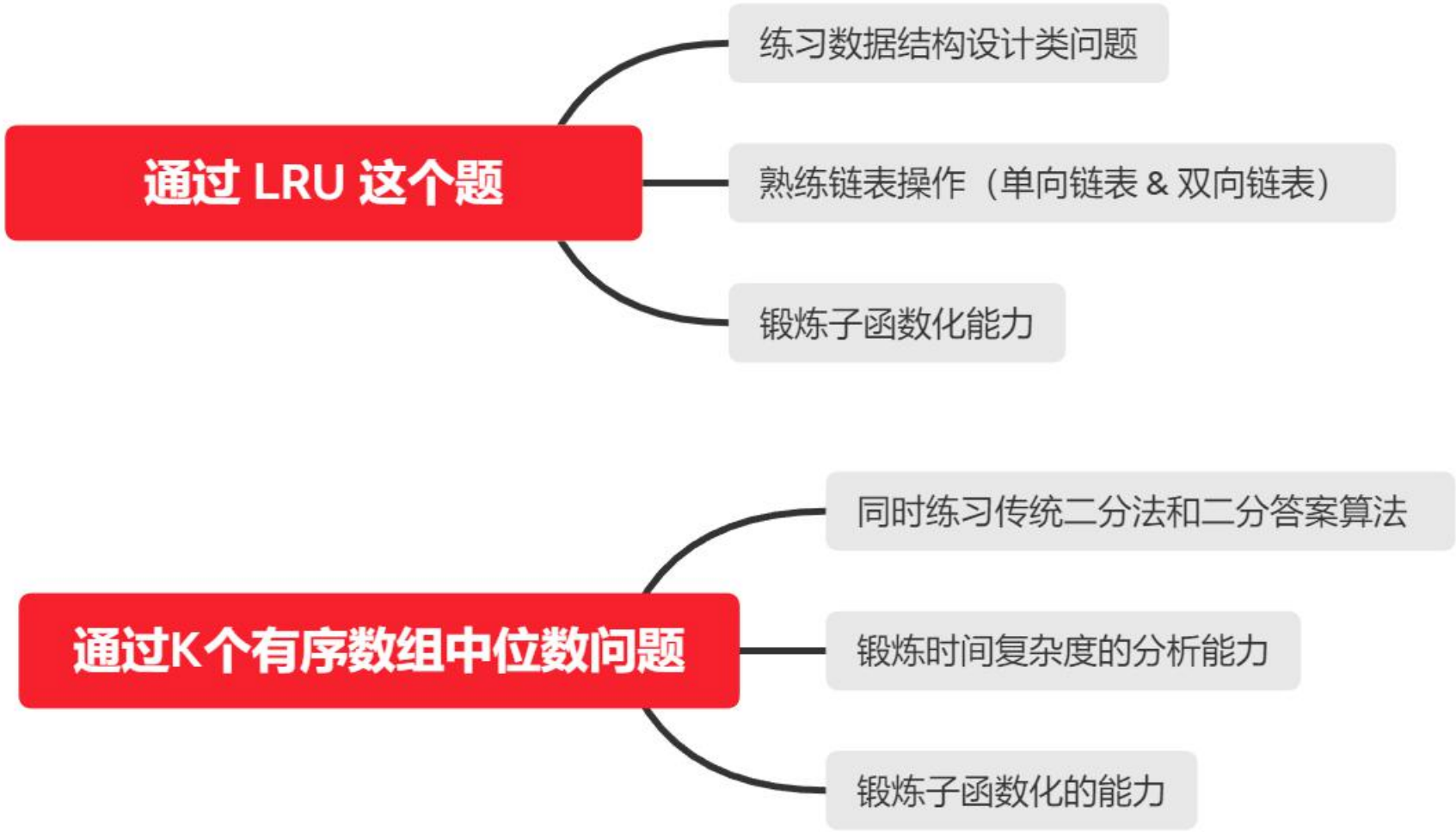
每个题做完都相当于啃下 3-5 道类似题

请扫码加花花拉你进群

举个例子



举个例子



九章算法班 2020 版有啥更新？

9节直播课 → 9节直播课(18课时) + 34节互动课(至少34课时)

什么是互动课？文字、图片、视频等多媒体形式的教学模式

一边学习一边练习

视频会向你提问检测你的学习成果

并个性化推送你不会的知识点给你

第二节课和第三节课是免费试听的互动课

有录播么？

从 2020 版开始，直播和互动课均提供7天回放

为什么不永久回放？

督学+防盗版

实在是需要更长的回放？可以通过购买增值服务实现

《九章算法班 2020 版》的价值

节约时间

自己需要三个月才能学下来的，我一个月带你准备好

课程覆盖 **90%** 的面试高频知识点

有班主任督学，有助教答疑

其他算法视频 vs 九章算法班

九章算法
www.jiuzhang.com

	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。	直播课+互动课，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，学员微信群里一起学习，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，都可以在微信群，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

后序课程安排

第二节课和第三节课是免费试听的互动课

第四、五、六、七节付费后即可开始学习

直播课程按直播时间上课，互动课程每周开放8节课

每周4小时直播课，8~12小时互动课

https://www.jiuzhang.com/course/71

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

Q & A