

Google/Facebook 大公司高频题风格解析

课程不允许录像，否则将追究法律责任，赔偿损失



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](#)

知乎专栏: <http://zhuanlan.zhihu.com/jiuzhang>

微博: <http://www.weibo.com/ninechapter>

官网: www.jiuzhang.com

- 业余爱好：音乐剧，浙大灵韵音乐剧社创始人，常春藤盟校春晚导演

- ZOOM 的用法
- 老师/助教回答提问方式
- 课程Ladder: <http://www.lintcode.com/zh-cn/ladder/14/>
- 九章QA: <http://www.jiuzhang.com/qa/>

- 高频题是啥？
- **FLAG**等大公司面试技巧及面试风格解析——四道题目举例
- 高频题课会怎么上？

- 高频题是啥？
 - 不是难题，不是偏题，是面试出现频率很高的题
- 面试出现频率很高？
 1. 总结出公司出题风格
 2. 现在刷，面试时可能遇见原题

FLAG等大公司面试技巧及面试风格解析

——四道题目举例

Facebook: Decode Ways

<http://www.lintcode.com/zh-cn/problem/decode-ways/>

<http://www.jiuzhang.com/solutions/decode-ways/>

面试技巧 套原题

- 这一题类似哪道题呢？
 - 上楼梯问题（动态规划基础题）
 - 上楼梯问题： n 节楼梯从第0层开始，每次只能1层或2层，问上到第 n 层有多少种方法？

思路一：

- 枚举所有的走法？
 - 时间复杂度太高 $\approx O(2^{n/2}) \approx O(2^n)$

思路二：

- 考虑最后一步怎么走？（假设 $n=10$ ）
 - 最后一步一定是走1层 or 2层
 - 跳到第10层楼有且仅有这两类方法，并且两类方法不重合
 - 所以跳到第10层楼的方法数一定等于跳到第8层方法数+跳到第9层方法数



FLAG 高频题有什么特点——四道题目举例

思路:

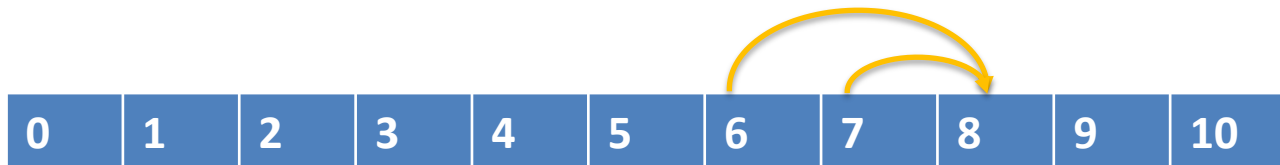
- 数学表示: $f[n]$ 表示从第0层走到第n层一共有多少种方法
 - $f[10] = f[8] + f[9]$

要计算 $f[10]$, 我们得先知道 $f[8]$, $f[9]$ 等于多少



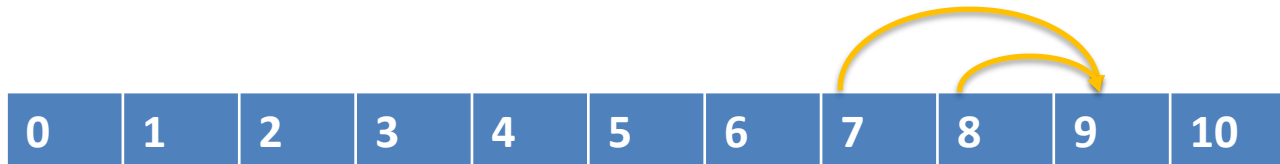
思路：

- 数学表示：f[n]表示从第0层走到第n层一共有多少种方法
 - $f[10] = f[8] + f[9]$
 - $f[8] = f[6] + f[7]$



思路：

- 数学表示：f[n]表示从第0层走到第n层一共有多少种方法
 - $f[10]=f[8]+f[9]$
 - $f[8] = f[6]+f[7]$
 - $f[9] = f[7]+f[8]$



FLAG 高频题有什么特点——四道题目举例

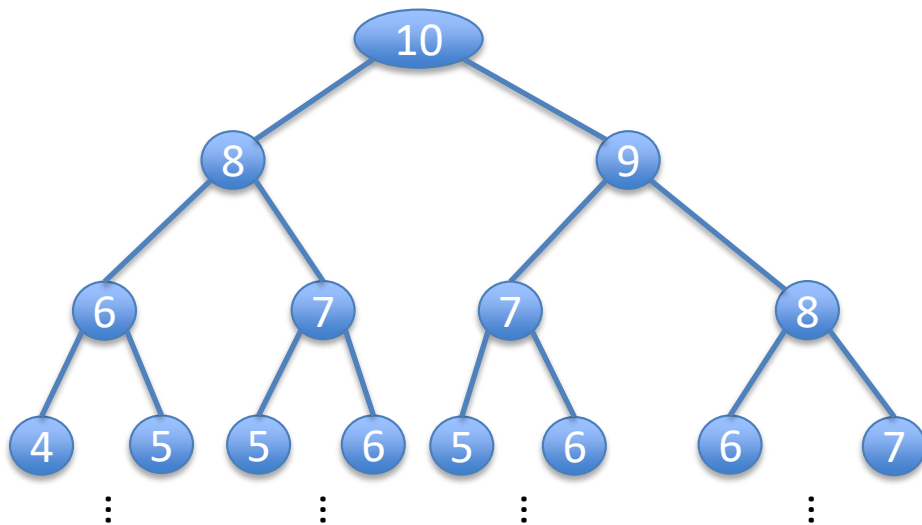
思路:

- 数学表示: $f[n]$ 表示从第0层走到第n层一共有多少种方法
 - $f[n] = f[n-1] + f[n-2]$



思路：

- 数学表示： $f[n]$ 表示从第0层走到第n层一共有多少种方法
 - $f[n] = f[n-1] + f[n-2]$



FLAG 高频题有什么特点——四道题目举例

思路:

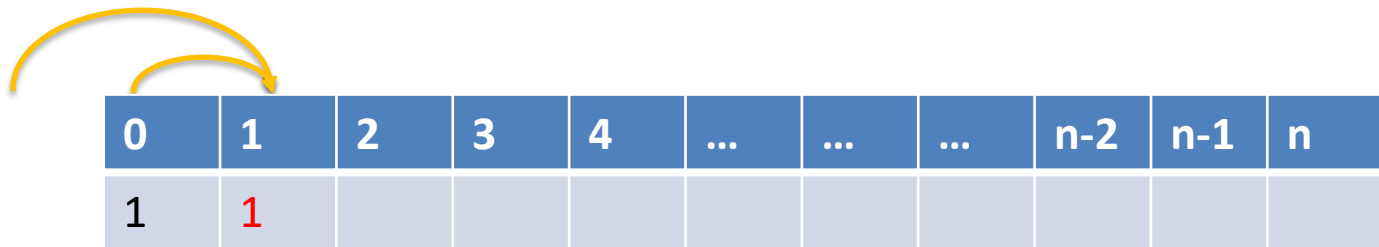
- 如何快速的计算 $f[10]$?
 - $f[n] = f[n-1] + f[n-2]$ $f[n]$ 只和比 n 小的状态有关
 - 按照 $f[1]$ $f[2]$ $f[3]$ $f[10]$ 的顺序依次计算



FLAG 高频题有什么特点——四道题目举例

思路:

- 如何快速的计算 $f[10]$?
 - $f[n] = f[n-1] + f[n-2]$ $f[n]$ 只和比 n 小的状态有关
 - 按照 $f[1]$ $f[2]$ $f[3]$ $f[10]$ 的顺序依次计算



FLAG 高频题有什么特点——四道题目举例

思路:

- 如何快速的计算 $f[10]$?
 - $f[n] = f[n-1] + f[n-2]$ $f[n]$ 只和比 n 小的状态有关
 - 按照 $f[1]$ $f[2]$ $f[3]$ $f[10]$ 的顺序依次计算

0	1	2	3	4	n-2	n-1	n
1	1	2								

FLAG 高频题有什么特点——四道题目举例

思路:

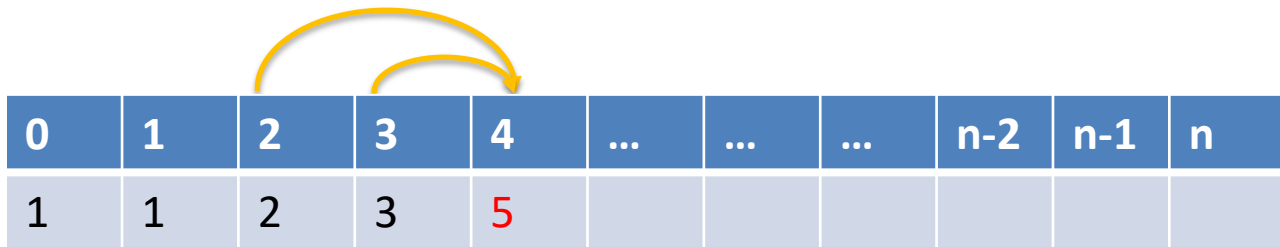
- 如何快速的计算 $f[10]$?
 - $f[n] = f[n-1] + f[n-2]$ $f[n]$ 只和比 n 小的状态有关
 - 按照 $f[1]$ $f[2]$ $f[3]$ $f[10]$ 的顺序依次计算

0	1	2	3	4	n-2	n-1	n
1	1	2	3							

FLAG 高频题有什么特点——四道题目举例

思路:

- 如何快速的计算 $f[10]$?
 - $f[n] = f[n-1] + f[n-2]$ $f[n]$ 只和比 n 小的状态有关
 - 按照 $f[1]$ $f[2]$ $f[3]$ $f[10]$ 的顺序依次计算



0	1	2	3	4	n-2	n-1	n
1	1	2	3	5						

总体时间复杂度 $O(n)$

Decode way 解法:

- 考虑最后一位怎么分解?
 - 最后1个数字当作一个字符 or 最后2个数字当作一个字符
 - 类似上楼梯跳1步 or 跳2步

s="7452.....519"

idx	0	1	2	3	4	n-2	n-1	n
s		7	4	5	2	5	1	9

9 → 'i'

OR

19 → 's'

Decode way 解法:

- $f[n]$ 代表 $s[1..n]$ 有多少种分解办法 (s 从 1 开始计数)

$s = "7452.....519"$

idx	0	1	2	3	4	n-2	n-1	n
s		7	4	5	2	5	1	9


9 → 'i'

OR

19 → 's'

Decode way 解法:

- $f[n] = f[n-1]$ (条件 $s[n] \neq 0$) + $f[n-2]$ (条件是 $s[n-1]$ 与 $s[n]$ 组成的数字在 10-26 之间)



idx	0	1	2	3	4	n-2	n-1	n
s		7	4	5	2	5	1	9

9 → 'i'

OR

19 → 's'

FLAG 高频题有什么特点——四道题目举例

- $f[n] = f[n-1](\text{条件 } s[n] \neq 0) + f[n-2](\text{条件是 } s[n-1] \text{ 与 } s[n] \text{ 组成的数字在 } 10-26 \text{ 之间})$
- 数据模拟:
 - $s = \text{"7452310519"}$

idx	0	1	2	3	4	5	6	7	8	9	10
s		7	4	5	2	3	1	0	5	1	9
f	1	1	1	1	1	2	2	2	2	2	4

FLAG 高频题有什么特点——四道题目举例



◆ Facebook特点:

- OJ (Online Judge)上的原题 DP (Dynamic Programming 动态规划)
- E.g.: [Wildcard Matching](#) [Regular Expression Matching](#)
- <http://www.lintcode.com/en/tag/facebook/>

LinkedIn: Isomorphic Strings

<http://www.lintcode.com/zh-cn/problem/isomorphic-strings/>

<http://www.jiuzhang.com/solution/isomorphic-strings/>

FLAG 高频题有什么特点——四道题目举例

题意：

- 验证相同位置上的两个字符是否是一一对应的

算法：

- 从左往右映射一遍，依次记录下映射关系，发现矛盾就false（比如a映射到f后 a又映射到k）
- 从右往左再做一遍

数据模拟：

- abad fekg 不是
- abad fefg 是

- 看到这里时，大家会发现这是一道**简单题**，算法都清楚了我们下一题吧 —— 不！我还要继续讲！
- 讲什么？——面试遇到简单题时该怎么玩？
 - 简单题时**TODO**: 怎么样简单的实现代码
 - 花式秀代码时间

注意：我们这里的“花式”代码是为了让程序更简洁、更易读、更容易 **bug free**，不是单纯的为了花式而花式

- 怎么表示“a 映射到f”？
 - Hash `map['a'] = 'f'`
- map用什么数据类型？（2种实现Hash的方式）
 - Java: `HashMap` C++: `unordered_map` Python: `{}` `dictionary`
 - 数组 `int[] map = new int[256];` （当标号是数字或者类数字时）

数组实现优势：代码短，**key**可以方便的顺序访问

劣势：只能在被**Hash**的对象是数字或者字符时使用，且受内存大小限制

◆ LinkedIn 特点:

- 基础算法、数据结构的快速实现
- 难度适中
- <http://www.lintcode.com/en/tag/linkedin/>

- 通过前面两道题我们学到了什么？
 - 难题：套原题
 - 简单题：代码写漂亮
- 所以如果想上高频班，小要求：
 - 课后把代码亲手写一遍

Amazon: Rectangle Overlap

<http://www.lintcode.com/zh-cn/problem/rectangle-overlap/>

<http://www.jiuzhang.com/solutions/rectangle-overlap/>

思路:

- 最基础的想法分类讨论
- 假设一个固定，另一个从左往右移，那么在一个维度上分别是：
- 不重叠 重叠 重叠 重叠 不重叠
- 两个维度上都重叠矩形才重叠

FLAG 高频题有什么特点——四道题目举例



- 正着想麻烦就反着来
- 考虑下不重叠的时候是什么情况？
- 要么上下、要么左右

面试套路
就是原题

FLAG 高频题有什么特点——四道题目举例



◆ Amazon特点:

- 题库比较固定 OA（Online Assessment）老9题等
- <http://www.lintcode.com/en/tag/amazon/>

FLAG 高频题有什么特点——四道题目举例



OA9题概要

Rectangle Overlap	Solution	基础题
K Closest Points	Solution	堆
Window Sum	Solution	Sliding window类
Longest Palindrome	Solution	模拟
Copy List with Random Pointer	Solution	链表操作
Course Schedule II	Solution	拓扑排序
Minimum Spanning Tree	Solution	基础最小生成树
High Five	Solution	Hash+堆
Maximum Subtree	Solution	树上DFS

FLAG 高频题有什么特点——四道题目举例

★ 0 - Amazon OA High Frequent 9 Problems		☰ 必做 (1/1)	☕ 选做 (8/8)	🔗 Related (0/0)
容易	627. 最长回文串 ☆		☑	25 %
容易	628. 最大子树 ☆		☑	43 %
容易	604. 滑动窗口内数的和 ☆		☑	33 %
中等	616. 安排课程 ☆		☑	22 %
中等	613. 优秀成绩 ☆		☑	30 %
中等	612. K个最近的点 ☆		☑	20 %
中等	105. 复制带随机指针的链表 ☆		☑	33 %
困难	629. 最小生成树 ☆		☑	26 %

Google: Valid Word Abbreviation

<http://www.lintcode.com/zh-cn/problem/valid-word-abbreviation/>

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

字符串收缩操作：

apple



a 3 e

apple



a 2 le

apple



4 e

思路:

- 直接模拟
- `s = "internationalization"`
- `t = "i12iz4n"`
- 一个串上一个指针
- 麻烦点: 逻辑小细节处理(怎样读数字、指针具体指向的位置)
- 怎么办: 写一遍才有体会 `coding time`

面试小技巧

遇到麻烦题不要怕，现场debug也加分

◆ Google特点:

- 网上原题少 原题变种 实现麻烦 代码功底
- <http://www.lintcode.com/en/tag/google/>

- 通过前面四道题我们学到了什么？
 - 难题：套原题
 - 简单题：代码写漂亮
 - 常考原题：刷题库
 - 面试debug：不要怕

- 四家公司面试特点：
 - Facebook: OJ (Online Judge)上的原题、DP (Dynamic Programming)
 - LinkedIn: 基础算法、数据结构的快速实现、难度适中
 - Amazon: 题库比较固定、OA老9题等
 - Google: 网上原题少、原题变种、实现麻烦、代码功底

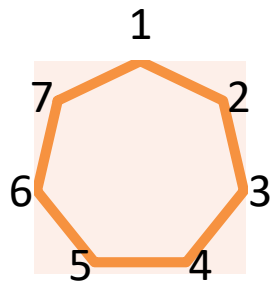
- 面试考察能力7维度：
 1. 理解问题
 2. 代码基础功力
 3. 基础数据结构/算法
 4. 逻辑思维/算法优化能力
 5. 细节处理（**corner case**）
 6. 算法分析（时间/空间复杂度）
 7. **debug**能力

- Decode Ways
 - 4. 逻辑思维/算法优化能力
 - 5. 细节处理 (corner case)
- Strings Homomorphism
 - 1. 理解问题
 - 3. 基础数据结构/算法

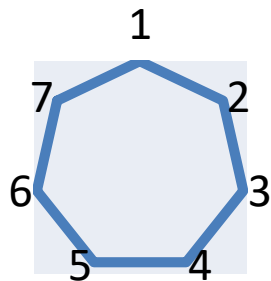
- Rectangle Overlap
 1. 理解问题
 4. 逻辑思维/算法优化能力
- Check Word Abbreviation
 2. 代码基础功力
 5. 细节处理（corner case）
 7. debug能力

FLAG 高频题有什么特点——四道题目举例

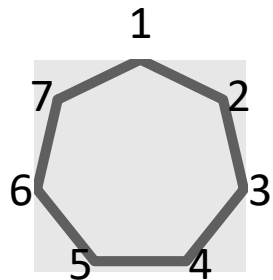
LinkedIn



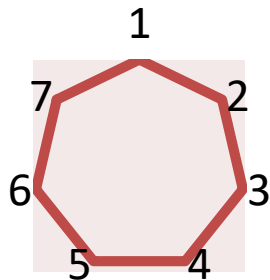
Facebook



Amazon



Google



高频题课会怎么上？

高频题课大纲

<http://www.jiuzhang.com/course/9/>

- 为什么是这个大纲？
 - 无论哪家，面试4类题： 1.模拟 2.搜索 3.DP 4.图论
- 所以我们课程是只针对FLAG四家公司吗？
 - 不是，因为考的内容都是一样的

课程时间：

- 美国时间每周五六
 - 美西 16:00-18:00
 - 美东 19:00-21:00
- 北京时间每周六日
 - 早上07:00 – 09:00
- 请在九章官网“我的课程”中查看具体每节课的时间安排

- 课程特色：

- 九章老师亲自总结各公司的高频题list
- FLAG个公司面试风格解剖
- 快速突破经典面试题
- 短期快速补充各种算法小知识点

- 课程定位：

- 短期冲刺班
- 一定的算法基础后，了解但不太熟练 → 熟练（+ 一点运气）

常见问题 Q & A:

1. 什么人适合上这门课？

有一定的语言基础（C++/Java/Python），
但没有算法基础，或算法能力薄弱
想听**FLAG**各个公司的面试技巧，面试偏好，已有高频题总结
希望短期高效有重点的准备大公司面试题

2. 课程目标是什么？

详细的分析**FLAG**各个公司面试风格，快速突破经典题目

常见问题 Q & A:

3. 有什么福利？

FLAG 常考题目，常考题目类型分布：

<http://www.lintcode.com/en/tag/linkedin/>

<http://www.lintcode.com/en/tag/facebook/>

<http://www.lintcode.com/en/tag/amazon/>

<http://www.lintcode.com/en/tag/google/>

常见问题 Q & A:

4. 上这门课有先修课程吗？

需要语言基础，对算法有基础了解即可（建议有30+以上的题量）

5. 高频题班 和九章算法、强化班之间是什么关系？

定位不同、互为补充，《九章算法班》《九章算法强化班》是从知识点的梳理和讲解出发，《高频题班》是从热点题目出发，是两个不同的课程体系，两门课程题目重叠度非常低，仅有2题左右相同

常见问题 Q & A:

6. 自己刷题不上课可以吗？

上课更高效、节省时间，和大家一起有学习的感觉和动力，更有独门的代码经验、小技巧的分享

7. 高频题班的高频题是如何统计的？

高频题由最新面经+OJ中的题目经过分类统计、结合老师经验归纳整理筛选而来，最近面试中最常见、最热门的题目都会在高频题班中出现

8. 这门课大概会有多少高频题？

总共近100道题目

常见问题 Q & A:

9. 课程题目难度如何？

题目难度适中，不会特别难，与今天的四道题类似

10. 大家对这门课程的期望是什么？



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](#)

知乎专栏: <http://zhuankan.zhihu.com/jiuzhang>

微博: <http://www.weibo.com/ninechapter>

官网: www.jiuzhang.com