

## 个人背景 & 面试准备

个人背景: 985 本科, 毕业好几年了, 离职之前在一家外企做 Java 后端开发。

面试准备: 因为选了微软作为目标公司, 因此完全是根据微软的面试特点来准备的。微软面试内容大概是项目 + 算法 + 系统设计 + 英语。因此主要把时间放在了准备算法和系统设计上。算法方面从今年 2 月份开始坚持在 leetcode 刷题, 刷了大概大半年, 刷题量 700+。系统设计看完了 Grokking system design 和数据密集型应用系统设计。

## 面试流程

微软今年 7 月份以后有一批 hc 出来, 投递了简历之后一两天就收到笔试通知的邮件。

笔试提交通过第二天就会有 HR 来协调面试时间, 面试一共五轮, 都是用 teams 视频面试。前两轮是组里的开发或者资深开发面试, 第三轮和四轮是不同组的 manager 面试。第五轮是 AA 面, 就是最后的大老板。每一轮都有技术问题, 大概方向是 30 分钟项目细节或者系统设计题目 + 30 分钟 1 道算法题目。

## 线上笔试

线上笔试是在 codility 这个平台做的, 60 分钟两道题目, 题目都不难, 关键是不要想复杂了, 尽量往简单的方向想。

## 一面

一面是 30 分钟项目问题和 30 分钟 1 道算法题目。

## 项目:

项目的准备主要分为项目核心功能的梳理和对项目架构和技术选型的思考。建议面试前几周多花一点准备自己项目这两部分的问题。

梳理完核心功能后在介绍起自己项目的时候会比较精准有侧重点, 让面试官很快了解到你做的项目是什么。

项目技术的思考也是被提问的热点, 知道要怎么做还不够, 重要的是知道为什么要这么做, 还有没有更好的?

比如我们当时的项目要从单体架构转型微服务架构, 所以一个经常被问到的问题是为什么要转微服务架构。

如果能提前总结此类问题, 那么面试被问到的时候回答起来逻辑就比较清晰。如果当场想的话, 很可能出现遗漏、回答不全、逻辑不清晰的情况。

## 算法:

一道简化版本的拓扑排序问题。理解了拓扑排序的思想, 代码写起来并不难。

用 teams 共享屏幕然后用自己的 IDE 写代码, 写完代码要保证测试样例能通过, 并且能准确说出时间和空间复杂度。

尽可能的考虑到更全的测试样例。

## 二面

二面是一道系统设计题目，时间充足的话建议看完 Grokking system design 和数据密集型应用系统设计这两份资料。

时间不够的话优先看 Grokking system design。重点是掌握系统设计题目的回答套路，如何一步一步地展开自己的系统设计思路以及和面试官提哪些问题来确定设计细节。一定要多和面试官沟通，多提问。看完 Grokking system design 想要实际操练一下可以在 [pramp.com](https://www.pramp.com) 上发起模拟面试，这样在真实的面试环节就不会特别紧张和卡壳。可能因为系统设计题目回答的比较让面试官满意，算法题目并没有为难，也没有要求写，只是说了下思路。

题目: 151. 翻转字符串里的单词

Follow up: 如果字符串长度很长内存装不下应该怎么办？

三面

三面考察了项目的深度, 有没有研究过同类型的项目别人是怎么做的？有没有横向对比？算法是面试官结合自身业务出的一道题目，涉及到的知识点是基本数据结构的运用（链表，栈，队列等）。

四面

四面是 30 分钟英文项目介绍和技术问答和 30 分钟一道算法题目

用英文介绍项目然后针对简历上的技术用英文做了提问和回答。

Manager 面试比较看重数据和结果，如果简历上提到优化了某些功能，那么性能提高了多少？内存降低了多少？系统响应时间降低了多？如果能拿出来具体的数据，那么给面试官的印象会更好。

题目: 54. 螺旋矩阵

AA 面

AA 面考察的东西比较多，技术和非技术方面的都有涉及。

技术方面给一个需求，最后转化成算法的实现方案，尽可能地给出优化方案。

类似的题目是: 307. 区域和检索 - 数组可修改

非技术问题涉及到为什么想加入微软，团队合作等问题, 面试前可以提前想一想怎么回答。

总结

如何拿到心仪公司的 offer？我的感觉是要了解清楚心仪公司的面试特点并做针对性的准备。

相信努力都是有回报的，拿到 offer 只是起点，进入到心仪的公司后面还有很长的路要走，保持学习，共勉。

