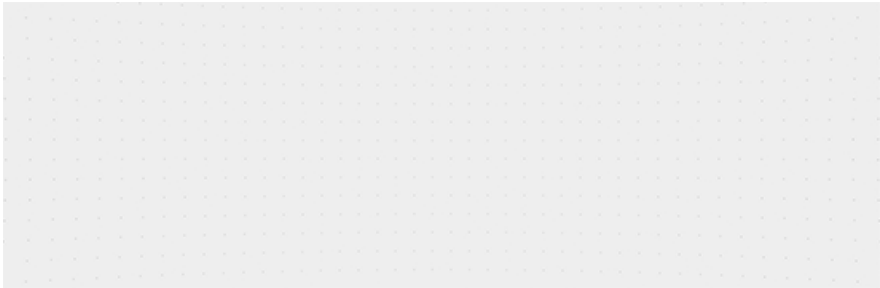


【面经系列】万字面试分享贴！从十二个角度解析面试全流程！附互联网公司Offer难度系数分析

CV开发者都爱看的 极市平台 2021-08-09 22:00:00 手机阅读

↑ 点击蓝字 关注极市平台



作者 | ELon Z
来源 | AI机器学习与知识图谱
编辑 | 极市平台
本文仅用于学术参考，如有侵权请联系小编做删文处理。

极市导读

本文作者在秋招期间面试了10+家互联网公司，总结了不同公司的考察点以及面试准备的全过程中的经验。帮助大家对各公司有个大致的认知，事半功倍。 >>加入极市CV技术交流群，走在计算机视觉的最前沿

目录

- 1、前言
- 2、时间线
- 3、考察维度
- 基础知识
- 算法能力
- 项目经历

一、前言

最近博客的阅读人数逐渐增加。回想起去年这个时候，我正在为校招做准备，很多文章也都是当时整理的。估计现在的读者很多都在为实习或校招做准备，所以趁现在有时间，分享一下自己的经验教训，希望能对大家有所帮助。

我的背景是 211本、985计算机硕，两段实习经历、一个项目/竞赛经历。我的校招面试从 2020 年 7 月初开始，到 9 月初结束，期间共面了 10+ 家互联网公司 (BAT/TMD/Y/K/P/MS 等)，全部都是后端开发岗位，技术栈不匹配挂了 2 家 (DD、JD)。面试一圈下来，感觉不同公司的考察点都大同小异，评价标准也比较相似。如果能提前有一个大致认知，复习时兴许能事半功倍，少走弯路。

二、时间线

壹伴图

极市平台
extreme

月发文数目: **
月平均阅读: **

文章工具

- 已发
- 采集图文 合成多
- 采集样式 查看

校招一般分为提前批、秋招、秋招补招、春招等阶段。

提前批一般在 7 月开始，但是 2020 年受到疫情影响，各家大厂早在 6 月就开始了提前批，比如字节提前批是 6.16~7.16、阿里是 6月中旬开始。今年预估还是 6 月开始。

我参加了提前批，所以是在 6 月下旬开始报名 + 笔试，7 月下旬集中面试（每天一两场面试），8月接近尾声（主要是三面 or HR 面），9 月上旬结束。也有同学不参加提前批，是在 8 月上旬开始，9 月集中面试，9 月下旬结束。

在这里，我的经验是：要尽可能早地准备、参与面试，有机会尽量参加提前批。

参加提前批意味着多一次面试机会，多投几家就能积累更多的经验，无论通过与否，对于之后的面试都有帮助。提前批也不是大佬专场，入场率和通过率都比想的要高。而且提前批大部分都免笔试，所以没什么试错成本。

提前批还可以集中面试很多家。校招面试就好比高三模考，天天面试就会习惯面试，好处一是不再紧张，二是能够维持面试的手感，三是在面试中暴露自己更多的问题，及时查漏补缺。

以往的说法是金九银十，但我感觉金八银九或许更准确些。去年身边的大部分同学在 9 月就已经拿到了心仪的 offer。当其他人手上至少一个保底 offer 而你还没有开始面试时，心里还是会比较焦虑的。所以还是要早做准备。

三、考察维度

各个公司对实习生的考察点大致可以分为以下几个维度：基础知识、算法、项目经历、实习经历、工程能力。我认为前两者更为重要。

1、基础知识

对于实习生/校招生来说，基础知识是最重要的考察点。

基础知识就是操作系统、计算机网络、数据库、计算机组成原理、编译原理等课程的内容（重要性递减）。作为实习生/校招生，大部分人都没有出彩的项目或实习经历，但都在学校系统学习了基础课程。因此，在很多面试官看来，基础知识最能反映校招候选人的水平。一方面，对基础知识的掌握程度体现了候选人的学习态度和接受能力；另一方面，实际工作也需要候选人有扎实的计算机基础。

2、算法能力

算法能力同样很重要。特别是对于后端同学来说，每轮面试至少 2 道算法已经成为标配，某些外企（如微软）甚至只考算法题。

面试官看重算法能力的原因：一方面，只要肯付出时间和精力去刷题，就可以达到一个不错的水平，所以算法能力至少应当达到平均线；另一方面，算法题需要深入积累、举一反三，才能解决面试中出现的新的题目，这反映了候选人的学习和思维能力。

算法题刷到什么程度就可以？其实很多面试官出的题目都是 LeetCode mid 难度，所以算法能力达到 LeetCode mid 以上就可以。刷完 LeetCode 前 300 道的 mid 题和一些高频/经典 hard 题，就可以应对大多数的面试了。

不过，相比于算法能力，我认为基础知识更为重要一些。算法是一项长期工作，见效慢，需要持续投入；而基础知识可以先广度覆盖各类知识点，再深入研究每个专题，在学习过程中可以随时获得反馈。算法的评判标准较为客观绝对，要么做出来要么做不出来；而基础知识的考察相对来说要宽松一些，不管会还是不会都可以说上一两句，面试过程中有更多的机会展示自

己。当然，这也取决于面试官的喜好，不同团队/方向的考察侧重点也不同，所以基础知识和算法还是要两手抓。

3、项目经历

在我看来，项目经历是一个锦上添花的内容，主要是给面试官一个提问的切入点。

项目/框架等工程能力，主要是针对工作3~5年的社招人员进行考核，因为他们理应有解决实际问题的经验。但是作为校招生，我们很难有大型项目的经历，面试官也不知道我们写过的项目的质量如何，或者不知道我们说的是真的还是在吹牛，所以会更关注我们的基础（如操作系统、数据库）与学习能力（如算法）。因此，不需要在项目经历上花费很长时间准备——性价比不高，也不需要将项目经历包装得高大上——面试官不信。

但是简历上还是要有一段项目经历，以给面试官一个不同的提问角度。

面试官对项目经历的提问，一般会从“介绍一下你做了什么”开始，然后会问一些业务细节，比如“为什么选这个方案？”/“有没有遇到xx问题？如何解决的？”如果我们的项目经历比较常规，大部分问题就会围绕里面用到的技术展开，而这些又最终会落到基础知识上。

面试官会注意候选人在项目过程中用了哪些技术、哪些框架，然后深入地问原理。面试官的关注点并不是“我在项目经历中，使用了哪些技术”，而是“我在项目经历中，使用了这些技术，并且我对其原理有深入的了解”。在项目中使⽤工程上很流行的技术和框架，这是最基本的水平；如果对这些框架的原理有深入的了解、对比过不同框架的优缺点、甚至自己尝试实现过一个 demo 框架，那在面试官那里一定是一个极大的加分项。

我们应当保证对项目中用到的技术/框架有较为深入的了解，而不只是停留在调用 API 上。比如项目里用了 gin 框架，面试官就会问为什么 gin 的速度快？有没有从源码的层面上理解过它的特点？这个时候，如果我们知道是前缀树，面试官就会点点头；如果我们再顺手写个前缀树的代码，那就会让面试官惊喜了。

一个常见的误区是：项目经历一定要高大上。比如，要不要做一个秒杀系统然后写在简历上？我认为没有太大必要，因为校招生很少有在大型分布式环境下编写 redis、消息队列等代码的条件，那这种项目经历不就是在 github 上照猫画虎吗？

项目不是只能做 Web 网站或者后台管理系统，还可以有其他选择。比如：

学习《CSAPP》课程，并做完若干 Labs，如内存缓存、内存分配回收、Web 服务器等 实现一个简单的操作系统，可以基于已经写好的系统，实现其中的部分模块，如进程调度、中断处理等 实现一个编译器，或者正则表达式（子集）的解析器 实现一个基于 TCP 的并发服务器框架，包括路由、Goroutine 池、消息队列等（如 zinx） 实现一个简单的 RPC 框架，考虑选择什么传输协议和编码协议、以及这样选择的原因 build-your-own-x 提供了许多实践项目，如 Docker、数据库、命令行工具、Shell、Git、搜索引擎等 这些项目不仅可以解决实际问题，还可以在过程中深入钻研底层原理，或者锻炼自己的算法能力。比起堆积木、搭脚手架，我认为这些项目更值得尝试。

4、实习经历

实习经历可以看作某种形式的项目经历，能够丰富简历、作为面试考点，大厂的实习经历也是对我们能力的背书。

面试官对实习经历的关注点：

在实习中做了什么工作？遇到哪些问题？如何解决？通过实习，得到了哪些成长？比如 Git 操作、内部工具、团队协作、开发流程/规范等。实习过程有哪些产出？有没有主动的贡献？比如

整理新人手册、发现并解决了某个问题、提升了业务指标等。对团队的现状和下一步规划，有没有自己的思考？在秋招（7、8月）之前，最好能够有至少一个实习经历。如果复习时间紧张，可以先投一些竞争较小的岗位，比如大厂的非核心部门、新业务/缺人的部门，或者二线互联网公司/独角兽企业等。

去哪儿找实习信息？首先是各个公司的官网，其次是牛客网（求职-实习广场）和实习僧，最后还有来自同学的内推。不要担心内推会麻烦别人，大部分正式员工都有招新同学的 KPI，我们只需要做一份简历，然后交给他们就好。

一个公司可以投递多个不同部门的岗位，但同一时刻只能进入一个岗位的面试流程。不同部门对候选人的要求不同，所以面试没有通过不一定代表水平不够，可以尝试投递同一个公司的其他部门。

因为培养实习生需要成本，所以部门希望实习生能够实习较长时间，一般都要 6 个月以上，至少 3 个月，但如果面试过程中说只能实习 3 个月很有可能被刷掉。

暑期实习有大概率转正。转正可以有一个保底 offer，面试其他公司可以拿来 argue，但是转正的评级一般不会太高 [如何提高?]。即使没有转正名额，也可以通过校招通道来争取更高级别的 offer。

5、工程能力

工程能力主要考察：对开发环境 / 工具的熟悉程度，如 Linux 概念、Linux 命令、Git 操作等；服务运维经验，如问题排查、性能优化等。面试题汇总中总结了部分常见题目。这里需要自己去搜集和动手操作。

四、开始复习

现在开始复习（3~5月），时间较为充裕。我们应当扎实掌握每一个知识点，而不是以覆盖更多面试题为目标。重质不重量，重深度不重广度，逐步构建个人知识体系。

现在网上有许多的面经汇总，包含一系列的题目和简短的回答。光靠这些面经的答案是远远不够的。面试官很喜欢在一道题上持续深入的追问，这很容易把浅尝辄止的候选人筛出去。我们应该将这些面经汇总看作问题集，每一个问题的答案，需要自己去深入挖掘。

许多面试题很常见，网上有详细的文章，但只会背八股文也远远不够。理解不牢，换个问法可能就不会了，背八股文也容易被面试官发现，或者被一连串的追问卡住。所以深入思考、主动归纳总结更为重要。

这里我的经验是做笔记。一份笔记，在写的过程中需要整理、思考和归纳，可以加深对知识点的理解，在复习过程中也是很好的参考资料。一系列笔记就是自己的知识库。

1、如何做笔记？

推荐使用 Markdown 做笔记。Markdown 的语法简单，书写过程中无需设置样式，不会打断思路。Markdown 也有较好的扩展性，可以和 HTML 语法结合使用。个人认为，Markdown 是最适合程序员的笔记格式。我的博客就是用 Markdown 写的。

Mac 系统下，推荐使用 MWeb 管理笔记。可选的替代方案有思源笔记、Typora、VS Code 等。MWeb 界面截图：

做笔记的过程中，应当遵循以下几个基本原则：

笔记不是简单地复制粘贴，而是要主动梳理，深入思考。笔记不是速记，即使不公开给他人，也要行文流畅规范。遣词造句的过程，实际是在脑海中梳理知识的过程。多问几个为什么，多发掘可能的考点。没有现成答案的问题，需要自己去探索和归纳。笔记的分类：

基础课程：操作系统、计算机网络、数据库等 算法：解题技巧、专题、题解等 语言：C/C++、Golang、Java 等 工具：Git、Linux、Docker 等 其他：分布式、设计模式、最佳实践等 Tips：

可以给笔记打标签，方便查找 可以在笔记的开头或结尾罗列参考文献 MWeb 可以通过 CMD + O 快捷键或 Alfred Workflow 搜索文档

2、基础知识复习指南

最重要的是操作系统、计算机网络，其次是数据库等。我的复习方法：

初步建立知识框架：选择一本经典本科教材，系统过一遍。本科教材知识点全，容易理解，但缺乏深度。深入研究各个专题：深入研究面经中列出的每个题目，特别是高频题目，如进程线程、HTTP 等。面经可以在很多渠道找到，比如牛客网、Github 上的面试题汇总。这里一定要自己整理笔记，我采用三个步骤：先搜集大量参考文章；阅读各个文章，同时梳理笔记大纲，粘贴一些可参考的关键段落；完善笔记内容，整理为一篇系统、连贯的文章。穿插阅读经典书籍：网上搜到的笔记大多是二手知识，直接阅读权威书籍，会有更原汁原味的体验。书籍有完整的上下文，随时可以向前或向后翻阅。时间有限的情况下，可以不从头到尾阅读，而是只阅读某个专题。可以去豆瓣或知乎查找高分/推荐图书，如《深入理解计算机系统》《现代操作系统》《Operating System: Three Easy Pieces》等。分清优先级：以基础知识、高频考点为主。像是分布式、中间件等进阶内容，应该是在有余力的情况下再深入了解。这些内容看起来高大上，但没有对应业务场景的情况下，只能是背八股文，一问就露馅。Tips：

不懂的名词多查查，多问几个为什么，发散思维，扩展相关知识点 网上的内容不一定正确，需要自己辨别或验证 网上的资料比较琐碎，自己整理一下，印象更为深刻 使用 Google 搜索；屏蔽内容质量低、抄来抄去的网站（比如 C**N）

可以尝试搜索英文关键字，内容质量较高

3、算法复习指南

算法能力达到 LeetCode mid 以上即可应对大部分面试。

我的个人情况：《剑指offer》50+ 道、LeetCode 242 道 (73 easy + 137 mid + 32 hard，范围主要是前 300 题 + 热题 HOT 100 + 腾讯精选练习 50)，题目数量不算多。

我的刷题方法：

持续学习：前期刷一些 easy 题热身，之后每天 1~2 道 mid 或 hard。定期复习：使用一个表格记录自己的错题、错误原因、笔记链接、上次复习时间，每周复习 3~5 道。按照标签刷：按照链表、数组、二叉树、动态规划等标签刷题，同一个标签的代码和方法类似，效率高。分清优先级：优先刷《剑指offer》、LeetCode 经典题、高频题、模板题；冷门题就没必要浪费时间了。可以参照网上的 LeetCode 笔记的顺序来刷题，见附录 - LeetCode 笔记。吃透每道题：刷题的目标不只是通过，而是要给出最优解。对于一题多解的情况，还需要参考他人的题解，将各种解法都实现一遍，并对比不同解法的优缺点。此外，很多不同的题目有着相似的解题思路，可以尝试用这道题的解法去做其他题，争取举一反三，比如 42. 接雨水和 84. 柱状图中的最大矩形。最后，总结自己的笔记。这样做下来，虽然刷题数量不多，但每道题都很会掌握得很扎实。Tips：学习数据结构时，配合可视化网站有奇效。推荐：

Data Structure Visualizations visualgo Algorithm Visualizer 关于“定期复习”，我使用一个表格来管理所有错题：

精选	题号	题目	Tag	难度	做错次数	复习记录
		▶ <input type="checkbox"/> 【格式说明】				
		▶ <input type="checkbox"/> 【数据结构 / 基础算法】				
		▶ <input type="checkbox"/> 【Linked List】				
		▶ <input type="checkbox"/> 【Array】				
		▶ <input type="checkbox"/> 【Tree】				
		▼ <input type="checkbox"/> 【动态规划】	Dynamic Programming			
<input checked="" type="checkbox"/>	121	▶ <input type="checkbox"/> 买卖股票系列				
<input checked="" type="checkbox"/>		▼ <input type="checkbox"/> 打家劫舍系列		Medium		
		198, 213, 打家劫舍				
	198	* <input type="checkbox"/> 打家劫舍 I		Medium	0	2020.9.6
	213	* <input type="checkbox"/> 打家劫舍 II: 环形		Medium	0	2020.9.6
	337	* <input type="checkbox"/> 打家劫舍 III: 二叉树		Medium		
<input checked="" type="checkbox"/>	62	* <input type="checkbox"/> 不同路径		Medium	1	2020.8.8
<input checked="" type="checkbox"/>	120	* <input checked="" type="checkbox"/> 三角形最小路径和		Medium	0	
<input checked="" type="checkbox"/>	152	* <input type="checkbox"/> 乘积最大子序列		Medium	2	2020.8.1
<input checked="" type="checkbox"/>	264	* <input type="checkbox"/> 丑数-ii		Medium	1	
<input checked="" type="checkbox"/>	300	* <input type="checkbox"/> 最长上升子序列		Medium	0	
<input checked="" type="checkbox"/>	91	* <input type="checkbox"/> 解码方法	String	Medium	0	
		▼ <input type="checkbox"/> 【背包问题】				
<input checked="" type="checkbox"/>	POJ3824	* <input type="checkbox"/> 01 背包问题		Easy	1	2020.8.9
		▼ <input type="checkbox"/> 完全背包问题				
<input checked="" type="checkbox"/>	322	* <input type="checkbox"/> 零钱兑换: 完全背包 + 恰好装满 + 最小值		Medium		2020.8.10
<input checked="" type="checkbox"/>	POJ1384	* <input type="checkbox"/> 完全背包 + 恰好装满 + 最小值		Medium	1	2020.8.9
<input checked="" type="checkbox"/>	416	* <input type="checkbox"/> 分割等和子集		Medium	0	

按照标签分组；相似题目可以汇总到一起，比如上图的“打家劫舍”。通过颜色区分题目的重要程度：橙色表示重点题目，优先复习；黑色表示常规错题，需要复习；灰色表示不重要的题目，有些细节需要注意，可以不复习；删除线表示已掌握的题目，不需要复习。每一行备注中的超链接是 MWeb 中的笔记链接，点击可跳转到笔记；备注里也可以记录需要注意的细节和易错点。软件是 OmniOutliner，使用 Excel 或 飞书云文档-多维表格可以实现同样的效果。首次复习时都需要重新写一遍代码，之后视情况选择写代码或者只回忆思路。及时更新各个题目的状态为“重点”“不重要”或“已掌握”，防止错题越攒越多。面试前快速浏览表格，回忆做题思路，优先看橙色高亮的重点题。

五、准备简历

简历从上到下大致为：个人信息、教育背景、实习经历、项目经历、荣誉奖励、专业技能/英语水平/个人评价等。我的简历模板：



Elon Z

求职意向: 后端开发工程师

路漫漫其修远兮, 吾将上下而求索

130-0000-0000

elonzzz@163.com

https://imageslr.github.io

北京市海淀区中关村

教育背景

2018.09 – 2021.06	北京XX大学	计算机科学与技术 (前x%)	硕士
2014.09 – 2018.06	北京XX大学	软件工程 (前x%)	本科

实习经历

某某公司 某某部门 – 后端开发实习生 2020.01 - 2021.01

– 主要负责 XXX, XXX 是 ... (介绍业务背景)

– 负责 XXX 的开发, 做了什么工作, 用了什么技术, 取得 XXX 结果, 最好由数据指标说明

– 负责 XXX 的开发, 做了什么工作, 用了什么技术, 取得 XXX 结果, 最好由数据指标说明

– 负责 XXX 的开发, 做了什么工作, 用了什么技术, 取得 XXX 结果, 最好由数据指标说明

某某公司 某某部门 – 后端开发实习生 2020.01 - 2021.01

- 主要负责 XXX, XXX 是 ... (介绍业务背景)
- 负责 XXX 的开发, 做了什么工作, 用了什么技术, 取得 XXX 结果, 最好由数据指标说明
- 负责 XXX 的开发, 做了什么工作, 用了什么技术, 取得 XXX 结果, 最好由数据指标说明
- 负责 XXX 的开发, 做了什么工作, 用了什么技术, 取得 XXX 结果, 最好由数据指标说明

项目经历

- “在线借书平台”小程序 - 前后端开发 2020.01 - 2021.01
- 作品主页: <https://github.com/imageslr/weapp-library> (550 Stars)
 - XXX 竞赛题目: 使用小程序技术, 为读者提供一个查找、借阅、续借、归还图书的线上平台
 - 利用 Sketch 绘制原型图, Swagger 编写 RESTful 文档; Vue / Golang 前后端分离开发, 前端实现组件化
 - 获得 XXX 竞赛全国一等奖, 获奖率 0.3%

荣誉奖励

- 校 X 等奖学金 (X 次)、国家奖学金 (X 次)
- XXX 竞赛全国一等奖
- XXX 竞赛全国一等奖
- XXX 竞赛全国一等奖
- XXX 竞赛全国一等奖
- XXX 竞赛全国一等奖

专业技能

- 编程能力: 熟悉 Golang、JavaScript; 了解 C++ / 深度学习 / NLP; 编程习惯良好, 具有代码洁癖
- 基础能力: 熟悉计算机原理; 熟悉 Linux、Git 的基本使用; 数据结构与算法功底良好
- 英语水平: 托福 94 分, 通过四六级, 能够无障碍阅读、交流与书写

上图使用 Sketch 绘制。制作一份这样的简历很简单: 首先确定一个主题色, 如 #353535, #3F59A8, #4F6F8A, #0089B2, #870509 等; 其次保证行高、行间距、页边距都是 4 的倍数, 如上图页边距为 24px、行间距为 4px / 8px / 12px; 矢量图标可以从 iconfont 查找。

简历可以使用在线平台制作, 如乔布简历、超级简历 CV 等, 也可以使用 Word / PS / Sketch 制作。拒绝花花绿绿、华而不实的风格, 关注内容而不是样式。采用 STAR 法则描述自己的项目和实习经历, 知乎的这个问题可以参考, geekcompany/ResumeSample 提供了一系列程序员的简历模板, resumejob/awesome-resume 则提供了许多程序员的简历例句。

Tips:

最好放在一页上, 如果一页放不下, 可以缩小字体/间距, 或改变页面大小; 部分在线平台也提供“自动填满一页”的布局功能。简历导出为 PDF 格式, 命名为“姓名_学校_岗位_手机号”。教育经历写预计毕业时间, 而不是“至今”。

六、开始面试

1、面试前

准备一份个人介绍 个人介绍相当于简历的文字形式。大部分面试官在开始面试时才会查看我们的简历, 所以个人介绍其实是引导面试官读简历的过程, 最好能和简历从上到下的内容对应。

如果面试外企, 还需要准备英文简历和英文个人介绍。

个人介绍大致可以分为三个部分:

背景介绍 (学校、专业、荣誉奖励等) 实习经历、项目经历 (遵循 STAR 法则) 业余生活 (博客、Github 开源经历、个人提升等) 个人介绍模板:

(背景介绍) 面试官您好, 我叫 XXX。来自 XX 大学 XX 学院, 目前研 X 在读, 明年 6 月毕业, 研究方向是 XXX。本科我就读于 XX 大学, XX 专业。在校期间, 我多次获得国家级、校级奖学金。此外, 我也参加过许多软件开发竞赛, 包括 XX 大赛、XX 大赛等, 拿到了 XX 奖项。

(实习经历) 去年,我在 XX 公司的 XX 团队实习 X 个月。这个部门主要是做 XXX (介绍团队业务背景),我在其中负责 XXX 工作,包括 XXX、XXX 等。此外,我还输出了 XX 篇文章,整理为部门新人手册。

(业余生活) 最后,我业余时间有写博客的习惯,会定期归纳总结自己的笔记。我也有开源经历,在 github 上有几个开源项目。

以上就是我的个人介绍,谢谢您。

Tips:

每个部分之间要有连接词。注意语速、节奏和停顿,避免变成高中生背课文;可以自己提前录音排练。时长控制在三分钟以内。一些经常被问到的问题,可以补充到自我介绍里。比如面试官总会问我:“你为什么从前端转到后端?”我就补充到了自我介绍的两个工作经历之间。准备一些常见问题的回答 简历里的项目经历、实习经历最有可能被面试官作为第一个问题,比如“从现在看来,你觉得这个工作还可以怎么改进?”“你觉得这些工作中最难的一点是什么?你是怎么解决的?”针对这些问题,应该提前准备。

附录 - 综合面试问题中总结了我在技术面和 HR 面遇到的综合面试问题,大家可以结合自己的实际经历,提前准备草稿。

准备反问面试官的话 面试结束后,面试官会说:“你还有什么要问我的吗?”这时借机可以了解一下团队。选工作是一个双向的过程,了解团队可以帮助做出选择。

反问技术面试官的问题:

组内主要的技术栈 / 语言? 部门具体业务 / 面试官的工作内容? 部门业务的挑战 / 难点? 我加入部门后可能负责的工作内容 (判断是否感兴趣) 部门的人数 / 人员构成 (判断是否核心) 部门业务的后续规划 (判断公司投入力度 / 未来机会) 针对新员工有哪些培训 (万能问题) 面试官的工作体验,比如工作是否枯燥乏味、对个人成长的帮助、工作压力是否很大、加班强度等。后续是什么流程? 还有几轮面试? (推断自己这一面是否通过) 大概多久能够出结果? (避免一直等结果心慌) 您对我刚才面试表现的评价? 您对我有哪些建议? / 哪些方面的知识、技能还可以再提升? / 您觉得我还有哪些方面的不足? 部门所在工区 / 地理位置 (考虑通勤成本) 上下班时间 / 是否需要打卡? 反问 HR 的问题:

针对新员工有哪些培训? 对我有哪些评价或者建议? 大概多久能够出结果 (避免一直等结果心慌) 之前的面评 (一般 HR 不会透露) 公司福利 / 三餐 / 房补 注意: HR 面试时,不要直接问户口、薪资等问题,等收到 Offer 后、在沟通薪资时再问。

实习生还需要关注这些问题:

部门有没有转正机会? 实习多长时间之后可以申请转正? 对实习出勤的要求是一周至少几天,至少实习多久? 部门的转正名额 / 转正概率? 如果我想转正,您期望我达到什么水平 / 有多少产出? 过去是签外包团队还是正式实习生? 不同身份的转正机会如何? 其他资源:

yifeikong/reverse-interview-zh - 反向面试 当面试官问“你有什么要问我的吗?”时,应该说什么? 安装录屏软件 如果是远程面试,推荐打开录屏 / 录音功能,录制面试过程。

为什么要录屏 / 录音? 面试过程中,我发现不同公司 / 不同轮次的面试题目有很大概率重复,这可能是因为面试官会根据候选人的经历、水平,针对性地挑选问题,我们的回答也可能影响面试官的下一个问题。因此,面试后的复盘非常重要。优先复习自己面试中遇到的题,其次再看网上面经的题,保证同一个题目只会答错一次。此外,也可以通过录屏 / 录音以第三人称的视角观察自己的面试表现,从而发现不足并及时改进。

Mac 系统默认只能录制麦克风的输入,需要配置 SoundFlower 来录制系统声音 (即面试官的声音)。Mac 系统的录屏软件可以使用 Kap。

我使用 2015 年 MBP + 关闭所有无关软件 + 打开录屏功能 + 面试一小时，系统没有出现明显卡顿。最好自己提前测试一下。

2、面试前一小时

快速浏览以下内容，遇到不会的不用深究，主要是有个思路：基础知识题目列表 综合面试问题 算法错题集，优先重点题目 牛客网面经 调试设备，包括网络、摄像头、耳机、麦克风、录屏功能等；视频画面里不要有杂物。面试前五分钟，排练一遍自我介绍。

3、面试中

如果遇到不会的题目怎么办？

即使不会做，也要主动和面试官交流。先说你如何考虑这道题目，你现在的解题思路是什么，然后向面试官要引导（比较晦涩）或提示（比较直接），最后给出一个自己水平内的最优解。

4、面试后

复盘。根据录屏或者回忆，整理面试中遇到的题目、自己没有发挥好的地方。距离面试结束的时间越短，回忆越清晰，可以尽快整理。

查漏补缺。有些题目可能只是我们自认为答对了，所以对于每道题目，都需要搜集资料、发掘所有的考点、并做笔记。

七、不同公司的面试流程对比

按照字母顺序排列；仅代表个人经历，不保证今年依然适用。技术面重点考察基础知识 / 算法；经理面以综合面试问题为主；交叉面是其他部门的人来面试。

1、阿里 面试难度：☆☆☆☆☆

面试流程：统一在线笔试。2 轮技术面 + 1 轮经理面 + 1 轮交叉面 + 1 轮 HR 面，每轮都是电话面试，时长 1 小时左右。一面面试官是未来的 leader，之后的面试官应该级别更高。一两面考察基础知识，三四面主要围绕项目和实习经历展开。

反馈周期：通过后会很快约下一轮面试，可以向你的内推人 / 师兄咨询面试结果；每轮面试间隔 2 天~1 周；Offer 统一在 9 月发出。

评价：由于是电话面试 + 已经笔试过，所以面试时不再考察算法。这导致面试题的覆盖面广、题量大、问得细，难度总体较高。HR 面体验很好，给我提了中肯的建议。

2、百度 面试难度：☆☆☆☆

面试流程：3 轮技术面 + 1 轮交叉面（可能没有）+ 1 轮 HR 面。技术面时长 1 小时，HR 面很短。是否有交叉面取决于部门和面试评级。技术面试包括算法题，中等偏难。使用百度的“如流”面试。

反馈周期：一面完可能紧接着就是二面。1~2 天约下一轮面试。HR 会加微信。

评价：不同部门的面试官水平差异极大。如果一个月还没给你一个明确的答复，大概率是被泡池子了，可以考虑终止流程，重新投递另一个部门。

3、快手 面试难度：☆☆☆☆

面试流程：2 轮技术面 + 1 轮经理面 + 1 轮 HR 面。技术面每轮都是牛客网面试，时长 1 小时。HR 面时长 30 分钟。

反馈周期：1 周左右约下一轮面试。

评价：基础架构部门问了很多 C/C++ 的问题，很少问上层的网络协议等知识，也没怎么考算法。面试官比较 nice，答错了会一步一步引导。

4、美团 面试难度：☆☆☆☆

面试流程：2~3 轮技术面，1 轮 HR 面。技术面时长 1 小时，HR 面时长 30 分钟。最后一轮技术面是 leader。使用牛客网面试。

反馈周期：3 天约下一轮面试。

评价：面试轮数和面试评级没有关联，2 轮也可以拿 SP。

5、拼多多 面试难度：☆☆☆☆

面试流程：2~3 轮技术面 + 1 轮 HR 面。技术面时长 1 小时左右，HR 面时长 30 分钟。技术面会考察两道算法题，都是 LeetCode 高频题，难度适中。使用自研平台面试，类似于牛客网，有代码编辑器和视频窗口。

反馈周期：一周左右约下一轮面试。

评价：确实很拼，约了 21:30 面试，面试官有事，等到 22:30 才开始，面完已经 23:30 了。从视频窗口能看出工位确实不大。

6、腾讯 面试难度：☆☆☆☆

面试流程：2 轮技术面 + 1 轮经理面 + 1 轮 HR 面。技术面时长 1 小时左右，HR 面时长 30 分钟。三面是 leader。面试形式有电话、牛客网、腾讯会议、QQ 视频。

反馈周期：2~3 天出面试结果 + 约下一轮面试。如果 5 天还没有约下一轮面试，建议联系 HR 查看进度。join.qq.com 上的进度超过 7 个工作日（一般 10 个自然日）没有更新时，流程会自动终止，简历重新放回池子里。面试通过后会先收到“云证”邮件。

评价：腾讯在面试实习生和校招生时，比较重视基础课程（操作系统、计算机网络），不会涉及太多高深技术，面试题目和面经重合度较高。每轮都有两三道算法，难度适中，不需要特殊技巧。

WXG：2 轮技术面 + 2 轮面委会 + 1 轮 HR 面。难度比其他部门高，会考察系统设计题（如高并发定时器）或复杂算法题（如判断点是否在封闭图形内）。

7、小米 面试难度：☆☆☆☆

面试流程：2 轮技术面，时长 1 小时。没有 HR 面。使用牛客网面试。

反馈周期：3 天约下一轮面试。

评价：面试官很耐心，会主动指出没答出的题目是哪个知识点、下来以后可以查一下。基本没有能够联系到 HR / 面试官的方法。感觉招人比较随意，二面完一个月没消息，一直认为自己挂了，结果 9 月底突然打电话，直接给出了薪资方案。

8、猿辅导 面试难度：☆☆☆☆

面试流程：先笔试，通过后进入面试。2 轮技术面 + 1 轮经理面，每轮时长 45 分钟，其中 15 分钟过简历 + 考察基础知识，25 分钟做两道中等难度的算法题，剩下 5 分钟反问。三面是 leader，没有 HR 面。使用牛客网面试。

反馈周期：一周内约下一轮面试。如果进入下一个阶段，内推人会先收到通知，可以问内推人。

评价：整体体验不错，HR 对校招生很用心，会加微信、拉微信群、送校招礼物。猿辅导笔试和面试的算法难度是 LeetCode 前 300（我也只刷了这些题）。基础知识 / 算法 / 实习经历至少有一项突出，会比较稳。

9、字节跳动 面试难度：☆☆☆☆

面试流程：2 轮技术面 + 1 轮经理面 + 1 轮 HR 面。技术面时长 1 小时，包括 2 道中等难度偏上的算法题。HR 面时间较短。日常实习可能没有经理面，校招可能会有加面。

反馈周期：一二面一般会连着，三面隔一两天。

评价：对算法和基础知识都很看重，问得比较细。无论通过与否，反馈都很快。会有 HR 加微信，一般是实习生。

八、收获 offer

如何选择 offer?

TODO 如何拒绝 offer? 拒 Offer! - 内推熊

如何和 HR 谈条件? TODO

九、LeetCode 题解

1、labuladong/fucking-algorithm：强烈推荐，非常赞的算法题解。

2、afatcoder/LeetcodeTop：各个大厂近期的高频面试题汇总。

3、MisterBooo/LeetCodeAnimation：可视化地呈现 LeetCode 题解。

4、yuanguangxin/LeetCode

5、azl397985856/leetcode

6、youngyangyang04/leetcode-master

7、Jack-Cherish/LeetCode

8、Github - Search · LeetCode

十、社区

- 1、牛客网：面经分享、笔试交流、Offer 选择
- 2、脉脉：部门评价、职场交流、Offer 选择 OfferShow
- 3、小程序：薪资分享

十一、综合面试问题

如果需要某个题目的答案模板，请评论指出。

- 1、实习期间的工作内容，介绍一下？
- 2、简单介绍一下你这个项目的内容？
- 3、你觉得它的亮点 / 难点有哪些？
- 4、实习期间遇到最大的挑战是什么？如何解决？
- 5、实习期间给你带来最大成长的工作是什么？
- 6、从现在看来，你觉得这个工作还可以怎么改进？
- 7、你觉得这些工作中最难的一点是什么？你是怎么解决的？
- 8、你觉得实习期间相比于实习前，对你的最大提升是什么？
- 9、有哪些印象深刻的事 fix？
- 10、实习期间有哪些工作以外的对团队的贡献？
- 11、之前三轮面试感觉如何？
- 12、有在面别的公司吗？有收到 offer 吗？正在哪里实习？转正通过没有？
可以说正在面试流程中，但尽量不要说已经进 HR 面、或者已经收到 offer 了
- 13、你如何选择不同 offer？ / 已经想好去哪个公司了吗？ / 有倾向吗？、
- 14、你是不是投过别的公司？如果都要你怎么办？
- 15、自己平时分析过源码吗？
- 16、你是怎么学习新技术的？
- 17、最近在学哪些新技术 / 在看哪些书？
- 18、你的长期职业规划是什么？1~3 年的规划是什么？
- 19、你能实习多长时间？你实习什么时候可以开始？
- 20、你们老师平时项目多吗？管的严吗？
- 21、你是什么专业？为什么不做算法而做开发？你如何看待很多本科生毕业后就去找算法岗？
- 22、为什么想要读研？
- 23、大量工作、有限时间怎么办？

- 24、你是如何团队合作的？
- 25、为什么想来我们公司？
- 26、为什么要应聘这个职位？
- 27、你哪里符合这个岗位的需求？
- 28、对这个职位，你还欠缺什么？
- 29、你对工作有什么要求？
- 30、你希望与怎样的领导共事？
- 31、如果我录用你，你将怎样开展工作？
- 32、如果我录用你，你觉得是什么原因？
- 33、如果我没有录用你，你觉得是什么原因？
- 34、如果你在这次面试中没有被录用，你怎么打算？
- 35、兴趣爱好是什么？你的优点是什么？你的缺点是什么？
- 36、和其他同学相比，你觉得你的优势在哪里？
- 37、你认为其他同学如何评价你？/ 你觉得周围人眼中的你是什么样的？
- 38、如果我问他们会得到一个什么样的评价？
- 39、你最有成就感的事是什么？
- 40、生活 or 实习？你遇到的最大的挫折是什么 / 最遗憾的事？
- 41、大学期间，最大的收获是什么？
- 42、有哪些帮助别人的经历？
- 43、你实习、生活中遇到的最难忘的人是谁？TA 有哪些特质？
- 44、TA 身上的这些优秀特质？
- 45、你最想提升自己的哪一方面？对加班的看法 对工资的要求

十二、Q&A

Q：我只会 Go，能不能报 Java、C++ 的岗位？

可以，做算法题时，可以选择自己熟悉的语言。跨语言投递岗位时，必须好好准备基础课程和算法，问得肯定会更深入。此外，最好能够对目标语言有一些基本了解，比如 Java 的锁/Map、C++ 的引用/指针/函数重载/虚函数、Golang 的协程/Channel 等，防止面试官问到语言相关的题目时无话可说。

Q：手里有一个 offer，要不要继续面试？

有余力的情况下，多面试几家，得知自己的真实水平，也有和 HR 谈价的筹码。

Q：一二面主要问什么？

一二面主要考察候选人的专业能力和潜力。整体采用结构化面试，考察以下几项：

基础课程：计算机网络、操作系统等，15~20 分钟 职能专项：语言、框架、工具等，15~20 分钟 算法能力：两道题，30~60 分钟 实习/项目经历：15 分钟 一二面是互相补充的关系，不是每轮面试都严格遵循上面的时间分配。一二面不是平行面。一面过了才能进二面，二面过了才能进三面。

Q：三面/最后一面主要问什么？

三面一般是经理面，包括专业能力和软素质两部分。专业能力是对前几轮面试的补充，主要是查漏补缺。更多的时间会考察候选人的软素质，包括学习能力/合作能力/解决问题的能力等，见附录 - 综合面试问题。这些问题完全可以提前准备，打好草稿。

Q：我不喜欢的部门捞了我的简历怎么办？

可以在约面试的时候拒绝，说明你想去另一个部门；同时想办法联系另一个部门的 HR 来捞你的简历。

Q：转正是否真的都是白菜评级？

“实习生转正一般都是白菜价”是一个常见的误区，身边有很多反例。影响转正评级的因素主要有两个：(1) 实习期间的产出；(2) 手里其他 offer 的水平。

实习生转正是根据转正答辩来评级，而转正答辩主要是介绍自己实习期间的工作内容，所以转正评级实际上取决于实习期间的产出。影响产出的因素：

- 1、部门水平。实习生过去大多是承担一些“杂活”，但核心部门的任务显然比普通部门质量更高，也更容易有可罗列的产出。
- 2、个人水平。学习能力强、动手能力强，mentor / leader 就会分配一些相对更有挑战的工作。
- 3、Mentor。实习生的工作基本上都是 mentor 分配的，有一个水平高且负责任的 mentor，工作的目标感和成就感会更强，产出事半功倍。
- 4、主观能动性。主动要活儿，不划水；为团队做一些日常工作以外的贡献。“贡献”不一定非得是提升业务指标。如果入职后安装环境、熟悉业务、阅读源码的过程中踩了很多坑，那就整理一份新人手册吧，把经验和教训分享给后续新入职的同学。

转正答辩后的评级并不是固定的。从收到转正意向书 (7~9 月) 到拿到带薪校招 offer (10~11 月)，中间还有很长一段时间。HR 最终给出的薪资会综合转正评级与你手上的其他 offer 进行考虑。所以如果感觉自己转正评级不是很理想的话，可以尽可能多地争取几个其他公司的 offer。

如果觉得有用，就请分享到朋友圈吧！



极市平台

为计算机视觉开发者提供全流程算法开发训练平台，以及大咖技术分享、社区交流、竞...
765篇原创内容

公众号

▲点击卡片关注极市平台，获取最新CV干货

公众号后台回复“CVPR21检测”获取CVPR2021目标检测论文下载~

极市干货

深度学习环境搭建：如何配置一台深度学习工作站？

实操教程：OpenVINO2021.4+YOLOX目标检测模型测试部署 | 为什么你的显卡利用率总是0%？

算法技巧（trick）：图像分类算法优化技巧 | 21个深度学习调参的实用技巧

30⁺极市新项目需求来袭

持续更新

丰厚的项目报酬 | 持续的分成收益+免费算力 | 真实场景数据集

项目需求

安全帽检测

抽烟检测

井盖缺失检测

非机动车违停检测

人员闯入行为检测

人员未穿工服识别

楼内电动车检测

重型机械识别

道路事故识别

人员未穿工服

楼内电动车

重型机械

安全帽

非机动车违停

道路事故

抽烟

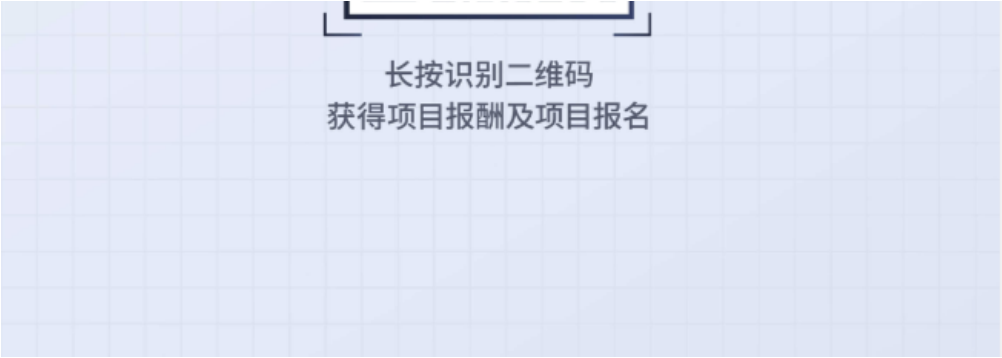
井盖缺失

人员闯入

More

More

More



CV技术社群邀请函

△长按添加极市小助手

添加极市小助手微信 (ID : cvmart4)

备注：姓名-学校/公司-研究方向-城市（如：小极-北大-目标检测-深圳）

即可申请加入极市目标检测/图像分割/工业检测/人脸/医学影像/3D/SLAM/自动驾驶/超分辨率/姿态估计/ReID/GAN/图像增强/OCR/视频理解等技术交流群

每月大咖直播分享、真实项目需求对接、求职内推、算法竞赛、干货资讯汇总、与 10000+来自港科大、北大、清华、中科院、CMU、腾讯、百度等名校名企视觉开发者互动交流~

觉得有用麻烦给个在看啦~

阅读原文

喜欢此内容的人还喜欢

我是如何从头开始写一篇顶级论文的
计算机视觉与机器学习



网络安全业务竞赛题库-(汇总1000题)
安全info

单项选择题 (1-4)
选择题 (606-78)
题: (790-1000)

有趣大会 · ACL2022 (长论文篇)
看个通俗理解吧



