

Google工程师的技术面试指南



技术面试是一个工程师成长到一定阶段后必然要承担的一项工作，优秀的技术面试官能帮助公司筛选出优秀的工程师，并且潜移默化的吸引候选人选择加入公司。相反，糟糕的面试不仅会错失优秀候选人，甚至还会给公司招来大麻烦。尽管技术面试如此重要，我还是了解到，很多公司的技术面试官都是“无证上岗”，hr随便抓壮丁去面试，面试质量参差不齐。本文就这个问题，根据我自己的面试经验和思考，总结了一些面试技巧分享给大家，希望有所帮助。当然，即便你不是面试官，看看面试官的面试套路，也能做到知己知彼，更好地投其所好地发挥。

01

如何阅读候选人简历

阅读候选人的简历，这是招聘流程中摆在我们面前的第一项工作。候选人的简历各式各样，我曾经见过很多人的简历写超过10页，项目零零总总罗列几十个，也有见过个人评价可以像散文一样写半页纸的，工程师们一般都比较忙，如何快速的阅读简历又不失重点呢？我总结了两点：忽略掉主观性太强的描述，比如简历中的自我评价，重复性的项目经历，一些熟悉某某技术，精通某某语言的主观性很强的描述，这些都可以一眼带过。另一方面，抓住候选人的亮点，亮点就是那些非常有力的能证明候选人能力的经历，我根据自己的经验罗列了一些，供大家参考：

首先，对于候选人来讲，大公司的工作经历是很重要的能力背书，而且级别越高可以粗略等同认为越优秀，虽然也会有个例，但一般情况下，阿里P8要比P7技术能力优秀，百度的T7要比T6优秀，腾讯的T3.1要比T2.3优秀。但是这种情况只针对大公司，对于一些创业公司，小公司来讲，Title并不与能力划等号，小公司技术总监的技术能力不如大公司的一个普通的资深工程师的情况也是常有的事情。

其次，有很好的教育背景，GPA高也是很大的加分项，比如清北复交毕业，虽然都说做技术学历不重要，但好学校的学生对计算机基础知识 / 通识知识掌握的更好，计算机思维，逻辑思维更强，相对要聪明一些，而且在工作中发现成绩好的同学往往在工作中表现出很高的执行力和快速交付能力/沟通能力，工作中的表现更优秀。

再次，有比较有技术含量的项目经历。为什么强调是“有技术含量”的项目经历呢？因为我发现，有很多工作年限比较长的候选人，简历十几页，项目大小几十个，而很多项目都是3，5个月就做完，用到的技术也比较重复 / 浅显，对于候选人的技术积累来说，10年的经验跟一年的经验差不多，所以项目不在多，而在于能体现候选人的技术能力。

还有，有高质量的开源项目；项目背景比较切合；有在技术网站发表过文章或高质量的技术博客；做过一些业余项目等，都可以作为加分项。

02

如何设计面试题

一般来讲，大部分的面试流程都会先安排一轮电话面试，通过之后再安排现场面试。电话面试的主要作用就是简历挤水分筛选掉特别差的候选人，因为电话面试毕竟不便于沟通，对于一些复杂问题也没有其他辅助方式来帮助理解和表达，所以面试题目最好不要过于开放，能够有标准答案的最好，类似笔试，也以基础知识或过往项目的简单询问为主。而现场面试一般比电话面试在沟通上更有优势，在题目设计上可以更灵活。

一般来讲，公司里面是没有面试题库的，一般都是面试官自己想的，来源有很多，网上看到的，项目里面涉及到的，自己被面试时遇到的问题等。拿来主义得到的面试题最好要修改一下，特别是网上常见的，以免候选人也看到过。

面试题的设计一定要有同理心，简单的讲就是要学会换位思考，要站在候选人的角度去看待面试题是否合适，是不是难度太大？题目背景是否容易解释的清楚？是否能让

候选人对题目的理解跟面试官一致等等。这一点听起来比较简单，但做好比较难。张小龙说过最牛逼的产品经理要有能瞬间把自己变成小白用户来认识自己设计的产品的能力。我们在设计面试题目的时候，也要学会放空自己，努力做到忘记答案，再去考量这个问题的难度。不过我们还有一个侧面的方法了解面试题设计的是否得当：先把题目讲给周围的同事听，看看他们是否能很容易的理解，能够答出来。

面试题目最好能层层递进，先从简单的问起，如果候选人能很好的回答，再继续深挖，逐级递进，当面试者无法继续回答出来时，大体上就能知道面试者对此方面知识掌握的深度了。

举一个我经常用的面试题：“有一个数组，数组中存储的是Cat对象，每个Cat对象有多个成员变量，其中一个代表颜色color，有两个值白色和黑色，要求编写一个函数将数组中所有的白猫都放到黑猫前面。”

如果候选人能够顺利解答，我会继续加大难度：“如果猫的颜色有三种，白色、黑色、灰色，编写一个函数将数组中白猫放到最前面，灰猫放到中间，黑猫放到最后面，比如：原来数组为 黑白灰白白黑灰灰，经过排序之后白白白灰灰黑黑”。

还可以继续加大难度：“不仅白灰黑之间按顺序排列，而且白猫，灰猫，黑猫各自内部原来的先后顺序也不能变。”虽然解题思路差不多，但显然需要处理的细节变多了，难度比之前两个都要大了。

这样一层一层递进，难度逐渐加大，这样的面试题一方面有区分度，另一方面不至于太难导致候选人完全答不上来。

03

少问记忆性问题和太理论性问题

有些JAVA面试官逢人便问JVM几种垃圾回收算法优劣对比，这种题目在我看来是没有太大意义的，一方面没有区分度，另一方面容易突击准备，往往考察不出候选人的真实能力，所以我面试有个原则，不直接问记忆性问题，也不直接问理论性问题，比如Spring AOP，IOC的实现原理，Spring Transaction有哪几种事务传播方法，分布式一致性的解决方法有哪些啊？JVM垃圾回收算法？Zookeeper的应用场景有哪些啊？

并不是说这些技术不重要不需要考察，而是换个问法，将这些记忆性的、理论性的知识融入实践，比如考察刚刚提到的Spring中的AOP, IOC原理概念，我一般会这样子设计面试题：给一些包含Spring功能特性的代码片段，让候选人阐述一下从应用启动到代码执行都经过了哪些主要的操作？当然还会告诉候选人主要考察spring的AOP/IOC特性，并且提示候选人越详细越好，以免候选人不能理解面试官的意图，答非所问。这样的问法让候选人言之有物，而且避免机械记忆性的背诵，更能测试出候选人是否真正的理解。

再比如，有些面试官为了考察候选人多线程方面的知识，经常会问到的题目比如ConcurrentHashMap的实现原理，volatile关键词的作用等，我之前也多次拿这些题目当做过面试题，但总结下来发现，大部分候选人都能答得七七八八，区分度很低。我之后换了一种问法，要求候选人将一个线程不安全的类改写成线程安全的类，这期间涉及到volatile,lock,并发容器，Atomic原子操作，CAS无锁编程等，发现只有极少部分候选人给出锁粒度小，并发度高的代码，部分候选人在提示下可以解决，一些候选人则仅能写出一把synchronized大锁的并发度很低的代码。事实很明显，那些能够给出优秀答案的候选人，必定是有着实践经验，并且深入思考过，真正理解的人，而相反，其他人可能只是临时看了几篇技术博客而已。

04

白板编程真的有必要吗

白板编程就是在面试现场让候选人在白板上写一段代码，当然白板只是一个代指说法，也可能是白纸上，也有可能是Google Doc共享文档里。白板编程外企面试比较流行，国内有些候选人不怎么接受，特别是工作年限较长的，一说要写个代码，求职者就觉得是在“羞辱”他，觉得不应该从这么基础的问起。不过根据我的面试经验发现，这种拒绝写代码的大龄码农，满嘴架构，高可用，高性能，分布式，往往一写代码就抓瞎，代码写的惨不忍睹。所以只要是面试一线技术研发岗位，不管是资深工程师，架构师，还是开发leader，我都会要求候选人现场至少写一段代码。

哪种类型的题目适合白板编程呢？举一个我之前经常用的例子：“写一个函数将ipv4地址字符串(仅包含数字，点，空格)转化成32位整数，另外，数字和点之间的空格是合法的，其他情况均为非法地址，要求输出合法地址的32位整型结果。”

这个题目不需要任何的算法背景和技巧，纯粹考察候选人的基本编程素质：逻辑思维是否清晰，细节是否考虑全面，是否能写出bug free的代码，是否有计算机思维能关注时

间空间复杂度等。而且在候选人完成代码之后，我还会要求候选人将代码讲给我听，当然不是因为我不懂，而是这样还能顺带考察候选人的表达能力，沟通能力，毕竟讲给别人听让别人理解要比单纯自己理解难很多。

除此之外，我个人也不建议编程题目涉及需要记忆的算法，比如被很多人诟病的面试题：写个快排，没有人会天天背诵快排算法，写不出来也理所应当，如果换个问法，比如给候选人讲一下快排的思想，然后让候选人代码实现，测试候选人是否能写出bug free的代码，我觉得这反倒是一个比较好的编程题目。

05

智力问题为什么会收到青睐

我们经常在网上看到说谷歌，微软等大外企经常会面试智力题目，面试智力问题到底有没有意义呢？答案是肯定的，我认为智力问题不在于候选人最终是否能提出标准答案，而在于提供一个话题跟面试者讨论，考察候选人是否是一个有想法的人，是否跟面试官在一个思维层次上，沟通流畅；更重要的是考察候选人思路是否清晰，逻辑推理能力是否够强，信息挖掘能力是否强，总结能力是否够强等等基本素质。这就跟有些公司面试过程会安排优选人跟面试官中午一块吃个饭一样，都是看是否聊得来。

举一个我之前曾经用的面试题目，也是我面试谷歌时被问到的一个问题：新建了一栋100层的办公楼，设计一个电梯调度系统，能让大家上下楼都节省时间，这个问题没有标准答案，你会发现不同的候选人回答相差很大，优秀的候选人会不停的挖掘背景信息，定义清晰需求，理清楚思路，通过一步一步严密的逻辑推理计算，合理的假设，让问题变得清晰可解，而有些候选人则无从下手。

我个人认为智力问题最好是比较开放性的问题，一定不要太难的问题，也不要是什么刁钻的问题，得有得聊。有很多面试官拿数学难题考候选人，希望45分钟答出来标准答案，这本身就是不可能，除非之前候选人已经看过，这样的问题也就没有意义了。比如网上比较流行的“扔鸡蛋测楼高的问题”“沙漠如何背最多水问题”。

当然，智力问题也并不是适合所有的公司。一般成熟型的大公司，对候选人可以接受比较长的培养时间，而且默认聪明的人学习能力都很强，所以对过往技术经验并非特别的看中，所以一般喜欢面试算法，智力问题。对于一些创业型公司，更看重候选人的工作经验，青睐技术多面手，来了就能产出，所以就不适合在智力问题上浪费太多的面试时间。

把面试当做一场技术讨论

筛选候选人就是筛选将来与你共事的人，所以为了更准确的反应候选人在以后的工作中的表现，不妨把面试当做一场与未来同事的技术讨论，在讨论的过程中感受候选人的技术能力。

技术面试就好比打乒乓球，一来一往中感受彼此的技术实力，面试的过程切忌类似与笔试一样的一问一答单向沟通。特别是一些开放性问题，架构设计的问题，本身就没有标准答案，背景又过于复杂开放，如果只是丢给候选人回答，中间没有任何沟通交流和引导，候选人是很难抓住重点展现出面试官心里期望的表现。

比如我们面试过程中经常会让候选人介绍某个项目的架构设计，当候选人讲解完项目的架构设计，如果面试官一个问题都不提然后就跳到其他问题，这种体验对不管是候选人来说还是面试官来说都不是很好的。而相反，如果面试官能一语中的的提出设计中的缺陷，或者追问架构中的技术难点，深入的跟候选人讨论，这样一方面能给候选人充分发挥的机会，另一方面，也会赢来候选人对公司技术的认可。

最后，总结

相信很多工程师随着面试经验的积累，即便没有经过培训，面试工作也可以做的非常好，因为毕竟优秀的工程师都逻辑清晰思维敏锐，也希望本文能给大家在成长的过程提供一点帮助。

关注微信公众号：小争哥
跟着Google工程师学数据结构和算法

后台回复PDF获取独家算法资料

