



我的人缘 0

2 17 2499
主题 帖子 积分

发消



分享帖子到朋友圈

intelliui 2018-7-12 06:43:12 | 只看该作者 ▾



1楼 电梯直达 ☐

本楼: 97% (235) 2% (6)
全局: 98% (781) 1% (8)

最近准备换工作来地里潜水，看了很多刷题经验和刷题交流，这里我简单谈谈我从面试官角度出发，考察的重点与我的心理活动。说的不一定完全正确，大家自己对号入座。发帖目的第一是帮助新人，第二是求点大米去去看别人的帖子。我就不隐藏了，大家随便发表意见，不要拍砖。

本人简介：曾经微软dev, 35+, 10年经验，有FLG offer. 去年加入一个start up 公司，最近前景不明，在犹豫要不要去个稳定点的大公司。我从sde开始面试其他人，到现在估计面试过100+人次的面试和debrief. 我面过coding, problem solving, design, behavior. 本帖子只谈论纯粹coding, 视情况讨论要不要再开帖子讨论其他方面。

本文涉及下面几个问题：

- 1) 我刷过这个题目，还要不要伪装
- 2) 我觉得这题很简单，但是不知道为什么就挂了
- 3) 我觉得面试官不是很友好，没提示
- 4) 我一定要bug free才能被录取吗
- 5) leet code的hard问题真的会被问到吗？考起来有什么意义？

我们从一个非常经典的，大家可能都刷过的题目开始。序列化 / 反序列化 二叉树。先说个背景，能面到我这里的，基本需要面试者有3-5年的面试经验。做为应聘任何微软或者flg的高级dev (63 and above, T4/E4 and above), 面试官其实都是假设你刷过不少题目的。我假设你刷过这个题目，所以我并不关心你写的到底有多快，写的是不是完全bug free，我更关心的是你做事的方式和沟通问题的能力。具体请看下面。

1. 提出问题，请序列化 / 反序列化二叉树。

什么？面试者不知道什么是序列化，反序列化？那我就问个多线程爬虫，timing LRU 一类的。如果多线程，锁也不会，那说明这个面试者的项目经验很不足。为了和其他有经验的人相match, 往往我会给一个很open的问题，或者一个很难的hard coding (取决于他已经被考察了哪个方面)。我个人觉得很少有面试官上来会考你 very hard coding question, 只是当某个面试者与其他面试人比显得相当缺少项目经验，那他除非能在聪明敏捷或者下苦功方面有突出表现，否则很难击败其他候选人。

2. 交流阶段

如果面试者马上开始写程序，或者马上给出他的想法，我会觉得面试者太过于着急了。在实际项目中，你很熟悉的地方可能成为项目中的滑铁卢，因为你熟悉的地方可能有变化，可能有个big change你不知道，可能很快你熟悉的api就不工作了。你总是要有很好的沟通能力，确保大家知道你在做什么，及早的发现你的错误。在code review甚至是test 阶段被人发现问题对项目来说就是个灾难。

我期望面试者能够在这个阶段提出一些问题，例如，有什么限制条件吗？二叉树的value是什么类型？这个api谁来使用，内部的还是public的？这个api更注重speed还是space？我需要多线程吗？如果面试者没问任何问题马上开始照着leetcode的signature开始coding, 甚至class 名字也叫solution(很多人)，我会觉得面试者做事的方式不够成熟。可能以后工作中会很毛躁，需要人来指导。

3. 无论你提出的建议是什么，例如你说你觉得speed更重要，我可能会说我更期待space. 这样做是避免陷入你最熟悉的套路。假设我说，我更需要序列化之后的空间占用最小。这时候一般的候选人不会刷到这么深，开始思考。

如果候选人根本没有办法优化空间，那我会认为他give up too early. 我希望候选人能安静的思考，提出几种方案，哪怕方案不成立。

如果候选人提出过多的方案，没有问help, 这些方案也不工作，我就认为候选人沟通有问题，无法把握好度。

举个leetcode的例子来说明问题的深入，leetcode的序列化格式中，','是必要的吗？ '[' 是必要的吗？有没有办法用byte直接序列化？byte还能变长吗？base64是什么？zip会有帮助吗？

4. 候选人选定方案后，我希望他能和我沟通，看看是不是在有限时间内能够写完。项目中，很多时候谁都知道什么design是正确的，什么是bad smell, 最聪明的人不是能做出最好design的，而是在有限时间内能给出大家都接受的solution的人。如果空间优化非常好，但是代码将超级复杂，无法写完，那么面试者应该及时和我交流。我也会偶尔提醒一下这样能不能在40分钟内写完。如果面试者坚持，我不会坚持，我会看他是不是能写完。这就是either strong hire or fail.

5. 候选人可能这时候说太复杂了，我们简化一下，简化到他熟悉的，刷过的coding.

6. coding 开始

到这一步，如果你做了2, 3, 4, 5中的大部分，并且code看起来似乎是work的，没有什么致命的问题，我不会纠结于比如正负数，nullpointer, int.min, coding是不是整洁一类的无聊问题。一般就开7. 如果你2, 3, 4, 5都skip, 那么一般我就希望你bug free, 否则你没办法和其他候选者做比较。往往bug free又要求你code组织很好，否则谁都很难以一眼看出你有没有问题。

7. follow up

还有什么能改进的吗？这是考察面试者的知识面，也看你是不是耐心。很多时候，很多面试者做出题目高兴的得意忘形，到这里就开始语无伦次的乱说，这不会影响是不是hire，但是可能会影响是不是strong hire, 也可能影响到你的

level.

举例子：我觉得还需要写点java doc啊， unit test, regression test, performance tuning, benchmarking, A/B testing, 等等。

总结：如果你能想到1-7中的所有点，谁会在乎你是不是刷过这个题目？即使你刷过这个题目，我也十分确定你刷的比别人好很多，你就是我心中想要的那个同事。反之，如果你只做到了6#，但是别人做了1，2，3，4，5，6，7，那你又怎么能面试成功呢？

有感而发，组织的不是很严密，大家去粗取精，也说说你们的想法。