



作者:Yura Shevchenko 来源: skypixel.com

关于一线码农的面试，我想说

求职面试在绝大部分人来说都是必不可少的，自己作为求职者也参与了不少面试（无论成功或者失败），作为技术面试官参与面试也有四五年的经验，在面试过程中也见识到了各种各样的人(有厉害的，也有奇葩的)。在这里也只想谈谈自己的一些看法，我说的不一定对，有不同的意见可以留言参与讨论。

面试本来就是一个双向选择的过程，面试官和候选人的地位本应该是一个平等的位置，面试官希望通过简单的交流沟通可以对候选人的技术，沟通等有一定了解进而确定候选人是否匹配相应的职位。个人认为一场成功的面试最好是能够让求职者和面试官都有一定的收获（曾经也遇到过在某次面试后，HR 告诉我有候选人特意跟她反馈要表达对面面试官的感谢，因为让他很有收获，这当然还是让我感到非常高兴的），每次参与面试，也希望自己能达到这个目标。对于候选人来说能从面试过程了解自己的不足或者交流探讨面试问题；对于面试官来说能了解候选人的技术和项目，在交流探讨中也是一次学习和巩固。另外面试能否通过最终强调是职位匹配，一个萝卜一个坑，萝卜太大或太小都不一定合适。所以有时候面试没通过并不是候选人不够优秀，也有可能是候选人过于优秀（例如本来只想招聘 P6，结果来了一个 P8的候选人肯定不合适）。

因为面试时间有限，1个小时(一般情况)的时间很难去全面了解候选人的技术实力，因此在面试过程中很难做到绝对的公平。举个简单的例子，面试官出一道题目，候选人 A 可能曾经做过或见过，所以能够比较轻松地回答出这个问题，而候选人 B 没有做过，虽然不能答出让面试官满意的答案，但 B 提供了一些解题的思路，虽然最终并没有答出这道题目，这就一定说明候选人 B 比 A 差么？并不是吧。

下面就从这道题目说起，这道题目是我在过往的面试中经常考察的一道题目。

动笔试试吧

实现一个函数，完成 开根号 的操作，方法签名如下：

```
double sqrt(int v, double t)
```

要求：

- 1. 不能调用系统库函数，诸如 `Math.sqrt(v)` 之类的；
- 2. 假设计算出的结果为 `r`，要求满足这个条件： $|r - \sqrt{v}| \leq t$ ，其中 $\sqrt{v}$ 是真实的值，`t` 为给定的一个误差范围，例如0.1等，即你计算出的值要在给定的误差范围内。
- 3. 实现语言不限，你条件可以比上述更加苛刻，但不能宽松。例如调用你的接口 `sqrt(9, 0.21)` 返回值属于 `[2.79, 3.21]` 这个区间的任意一个都满足条件。

看到这里，其实你可以 **拿出笔和纸，尝试解答一下**，需要注意的是答案要满足给定的误差条件，欢迎沟通交流。其实这个题目就是 leetcode 上原题稍加变化得到，做过的肯定觉得 leetcode 其他题目来说相对比较简单。但没做过也没关系，如果在面试官的提示下能够最终把这道题目解出来，在我看来也 OK 的，甚至有可能比刷过题记住解题答案的更好（当然刷过题目本身的肯定会围绕这个题目穿插其他小问题的）。



分割线

开始解答

其实刚开始，我认为这道题目比较简单，至少在给予提示后，理想情况下大部分一线 coding 的程序员都可以给出实实在在 `code` 的。然而“理想很丰满，现实很骨感”，事实并非如此，然而在面试很多人之后，发现此道题目并不简单，如果你能写出来，说明你已经比很多人优秀了（至少在我过往的社招面试经历中）。

当被问起这道题目之后，可能遇到的答案如下。

直接放弃

题目给出后，我一般首先明确候选人弄清楚了题目的含义然后会给一两分钟让候选人先思考一下。

面试官：你有什么思路吗？  
求职者：没有啊。

可能候选人内心OS是：“你出这样的题目是不是有病啊，明明有 `lib` 函数可以直接用的”。(之前同组有其他小伙伴确实有遇到这样的候选人，语言虽没这样夸张，大意是：实际工作中会出现这样的问题吗？我直接给你百度一个就行了)

在此强调，面试这道题目并不是想强调这个题目本身，期望以这道题目为契机，考察候选人在分析问题和解决问题的能力，在交流过程中所体现的逻辑推理和思维方式等，当然最后也会看看实实在在的 **Code**，从编码过程中看候选人的编程习惯，风格等等。

也有候选人刚开始抱着那个约束误差范围的不等式研究 N 久，然后没有然后了的。刚开始看这个条件当然好，但如果这个不等式没有思路可以先放一放，没必要在那苦熬。

面试官：这样吧，如果我问题 根号10 等于多少？你怎么回答？  
求职者：3点几吧。  
面试官：你怎么知道是3 点几，因为你知道9开根号是3，想象一下，你也可以完全用程序帮忙模拟你大脑思考的过程。  
求职者：我再想想.....

其实这里是希望提醒候选人，我们首先是要解题，然后才考虑效率。即不管用什么方法能够给出一个答案的，这个时候候选人可能进入下一个阶段了。

在实际工作场景中其实也是一样，遇到一个问题，首先我们要想到的是如何解决这个实际问题，有了最基础的解决方案之后再谈优化。

暴力搜索

实际面试过程中也有人是直接到这个阶段的。

先用一个循环找到 `r`，使得 `r^2` 是离给定 `v` 最近的平方数，即你希望算根号10，先找到3，因为3^2=9。

然后再用一个循环，每次 `r++t`，直到 `r^2 > v` 为止。

面试官：这个方法从理论上讲，是一个可行的方案，设想一下，如果我的精度要求很高，希望计算的 `v` 也很大，如 `sqrt(v = 10000000, t = 0.000001)` 之类的，调用你这个方法效率是不是很低，这个时候应该怎么优化？  
求职者：这样的话，我这个方法效率确实比较低，不过可以这样优化，比如设置一个步长，一次迭代后，如果没有达到预期，可以不断修改这个步长来增大逼近真实值的速度，比如10倍误差，100倍误差等。

其实，在与候选人的不断交流中可以看出候选人的 **Problem Solving** 的能力，这也是面试考察中的一点。例如关于上面问题的优化，也可能用于在实际工作中遇到的问题。

例如，我们在实际工作中可能经常会写一些异步的回调通知接口等，这个接口可能是其他团队维护的，有可能由于网络问题等回调接口可能会失败进而需要重试，对于重试的机制其实就可以借鉴上面的“步长”机制，第一次回调失败，我等待 1s 后重试，失败再重试，也许间隔 1s 不太恰当，是否可以修改等待的步长，等待比如 5s, 10s? 等等再重试直到成功。为什么要这样做？也许对方 `server` 本来现在就处于峰值，你不停的重试不但没有增加你接口调用成功的机会，反而增加对方 `server` 的负担。

额，跑题了，回到这个问题本身，继续

面试官：恩，这样做确实可以优化。但从本质上讲，假设我们不考虑误差的话，这个题目其实就是在在一个有序的列表里面去搜索满足条件的特定的值。刚刚你的方法是一个线性的搜索方案。常见的搜索还有什么吗？

求职者：二分搜索？

面试官：对呀，你可以尝试想想能否借用一下这个思路来解决这个问题。

二分搜索

当然，部分候选人提示二分后，就直接能够 `get` 到点，并且能够写出二分大体框架，但可能结束条件写的有点问题。

如果候选人还没有思路，就会继续

面试官：这样理解吧，你刚刚的搜索整数点的过程其实是线性的，一个数一个数去暴力穷举。借助二分的意思就是，比如算 根号10，你搜索范围是 [0, 10] (其实除了几个数之外范围可以更小[0, v/2])，你能证明么？。

- 因为5^2 = 25 > 10，所以 `r` 属于[0, 5)
- 因为(5/2)^2 = 6.25 < 10，所以 `r` 属于 (2.5, 5)
- 因为(3.75^2 = 14.0625 > 10)，所以 `r` 属于 (2.5, 3.75)
- 继续，如果你结束条件不太确定，可以暂时不管...

提示到这里，感觉已经相对比较明显了。

面试官：你现在明白了吗？  
求职者：明白了。  
面试官：那你写一下代码吧。

一个二分查找，算法思路都结合例子讲了一遍了，在候选人回答明白的情况下，理想当中，作为一线开发者写出来应该不会有问题吧。然而...理想和现实还是有差距的。

很多人都喜欢用递归写，可是很多人递归里面的最重要的结束条件都木有，一些边界条件等等都木有。所以一般情况下，代码写完后，我会让候选人自己写测试用例。

面试官：写好了是吧，你写几个测试用例吧，假设这个接口是别人写的，你应该从哪个角度去测试。  
求职者：`sqrt(-4, 0.21)`，哎呀，我这里忘了判断了。改一下代码。  
求职者：`sqrt(0, 0.21)`，`sqrt(4, 0.21)`... 还有问题，再改改。  
面试官：.....

为什么要别人提示要测试用例，才去 `check` 自己写的代码的正确性呢。有的候选人写的代码，就不拿一些异常情况去 `check`，就用上面讲的 `sqrt(10, 0.21)` 的例子都得不到预期结果。

能够到达这一个步骤的人已经较少了，如果你有较全测试用例和边界条件的判断，再加上后面的结束条件能够正确，基本上这道题目就算满意了。

关于结束条件

本质上讲，这个算法就是一个迭代逼近的过程，用二分的思路后，关键在于什么时候结束。题目中已经给了误差条件  $|r - \sqrt{v}| \leq t$ ，难点在于其中的 $\sqrt{v}$ 不知道，不太方便直接进行计算判断。不少人用一个另外的结束条件来进行了判断即： $|r^2 - v| \leq t$ ，其实这两个条件是不一样的。

对于这个结束条件，你有什么看法吗？能证明你的想法吗？

其他解法

当然本题还有一些其他的数学解法，例如用牛顿迭代法，梯度下降法(最速下降法)，泰勒公式展开等等。如果候选人能想到这些，说明他还是有一定数学基础的，如果愿意可以让他讲讲。（考察这道题目本意并不是期望候选人用这些数学方法解的。）

对于这道题目，你有什么比较好的思路吗？欢迎留言参与讨论。

常见问题

- 1. 问：为什么题目中的 `v` 的类型是 `int`？  
答：还真没有理由，`double` 也无所谓，可能仅仅是因为 leetcode 上原题计算的数是 `int` 吧。
- 2. 问：我能正确答对这道题目就一定能够通过这次面试吗？  
答：强调一下，面试中考察这样一个题目，这并不是仅仅考察这道题目本身，不是说你将这道题做对了，就能通过面试，反之也不是说你没做对这道题目就一定不能通过我们的面试。我们通过这道题目为契机，希望考察的是候选人在分析问题解决问题的能力，在交流过程中所体现的逻辑推理和思维方式等，当然也有最后实实在在的 `code`。
- 3. 问：这不是一道数学题目吗，为什么程序员面试需要考察这样的数学问题？  
答：同上，不是考察这道题目本身。另外，这也可以说这不是一道数学题目，当然能用数学的方式解答。候选人能用数学的方式解答也算正确。
- 4. 问：二分是这道题目的标准答案吗？我能用其他解法吗？  
答：同上，题目没有标准答案，就算你用最暴力的算法搜索出来也是正确的解法，其他数学方法也对。
- 5. 问：这道题目这么简单，牛顿迭代分分钟秒掉，是不是太简单了？  
答：给你点赞👍。
- 6. 问：这题目在说什么，我搞了半天没看懂，这TM是啥？  
答：如果确实认真看完整篇文章或跟面试官交流了那么久，还是根本不明白这到底说的是一个什么问题。那就罢了吧，跟他去吧，可能不是目标用户而已。
- 7. 问：我在实际工作中根本就不会遇到这样的问题，你问这个有什么用？  
答：同第2条答案。
- 8. 问：你们公司还缺人么，面试会考察哪些点？  
答：有兴趣或者有其他问题可以戳我邮箱，邮箱地址：aUB0YW5nbGVpLm1l。面试考察可能会涉及：CS 基础/Code/数据结构和算法/解决问题/项目经验/系统设计/沟通团队协作等等。

结语

本文题目是“从一道面试题谈谈一线大厂码农应该具备的基本能力”，其实，上面大部分内容只谈到了这道题目本身(也穿插了一些对这道题目的分析和理解)。上述题目的场景是社招面试中的，对于这样的题目来说招校招的反馈会更好。因为在校生可能对于工作经验，项目经验等比较欠缺，所以只好用一些比较固定的算法来面试进行筛选（本质上跟学校考试没有太大的区别）。

但这种类似的题目在社招场景下就完全不适用吗？社招的同学写不出来就有很充分的理由吗？或许你在工作场景中不会遇到实际这种题目，但我其实想表达的是，作为在最新线写代码的码农，在别人讲解了二分算法且自己也能理解的情况下，能写出这个二分算法应该不算太难？相当于一个需求，大家讨论了算法实现和解法，需要你把它变成能跑的 `code` 而已。

其实这篇文章最开始叫“从一道面试题谈谈一线码农应该具备的基本能力”，几年前发出来被喷了，后来想似乎被喷也有点道理，因为在日常有些场景下，“复制粘贴”工程师貌似也够用了，遇到问题有更高水平的人来帮你解决就行，大家都一样的话，怎么体现高手水平呢？但从用人单位角度想，当然是更希望招聘更加优秀的选手，怎样体现优秀呢？候选人基数太大，怎么筛选，其实也就“高考”一样嘛，通过“考试”择优录取而已。

我们就不去讨论是否每个写代码的人都需要有这样的能力（好像答案也是显而易见并不是）。但我建议咱们一线的程序员们（特指有上进心的一线程序员）应该对一些基本的数据结构和算法有所了解，对常见的算法复杂度有所了解？或者至少应该有这样的追求吧？比如二分搜索复杂度为什么是  $\log_2 N$ 。之前遇到过比如有有的候选人，Java 开发七八年经验了，简历描述精通 Java，但不清楚 `ArrayList`, `HashMap` 内部大概是怎么样做的（我理解，不管什么都需要知道大致的实现原理才有可能去优化遇到的各种实际问题吧？）。还有什么熟练掌握 Vim，结果其实就是熟练掌握如何打开和关闭 vim。还有的候选人口头表达达头头是道，结果落实到写代码就根本下不了笔。

有时候感觉大部分程序员都被大量的需求压迫着，被产品经理催着，仓促地码着繁琐的业务代码，不断的改着 Bug 又引入新的 Bug。业务产品经理重要么，当然重要（代码就是服务于具体业务的），但同时也还是希望我们不要抛弃一些基础的东西，多培养一下我们的编程素养。我们在用编程语言，利用各种工具来实现我们想要达到的目的的时候，能做到“知其然，知其所以然”岂不更好？

注：本文为旧文整理重发。

长按扫码更多面经分享



扫码回复 程序员 有彩蛋