

算法面试的评价逻辑

在和读者同学保持紧密互动的这两年里，我发现不少同学对面试都是有畏惧心理的，似乎面试官的心，就好像那镜中花、水中月，是非常虚无而不可捉摸的东西。所以在小册的最后，我想和大家聊聊算法面试的“评价逻辑”。

评价逻辑这东西，稍有不慎就会变成某一个面试官完全主观的个人脱口秀。为了使样本尽可能丰富、逻辑尽可能客观，笔者充分利用了每次面试结尾面试官抛出的“你还有什么想问我的吗？”这个问题，对超过10个考察算法的前端/全栈团队的不同角度的算法面试评价逻辑进行了梳理和抽象，这里作为彩蛋性质的知识点放送给大家。

涉及到软实力的内容，多说无益，贵在好记好执行。这里我就说两个点：1.应该做什么；2.不应该做什么

应该做什么

1. 面试前：大量做题，及时总结——给自己囤一个尽可能巨大的知识方法库。
2. 面试中：积极思考自己的知识储备和题目间的关系。没有关系，创造关系也要讲。充分相信“我是一个算法扎实的牛人”——面试官可以不信，但你必须信。
3. 面试后：保持思考，趁热打铁。面试已经结束，但你的职业生涯还在继续。一要珍惜向面试官提问的机会，尽可能从对方嘴里套话，理解他的命题意图和评价逻辑；二要抓住那些面试过程中不会做的题，把它当成你对象，就算再难搞也不要生气（你会随便跟你对象生气么？：）），抓紧它，搞懂它，下一次面试它就会从你的软肋变成你的底气。

不应该做什么

这里我着重说说面试中常见的两个大忌：

1. 畏于表达：面试官面你，不是为了难为你，他也是在给自己找未来的同事；面试官不是神，他可能技术还不如你。所以说遇到脸再臭的面试官，都不要怂。题目描述得不清楚，直接问；面试官点评的内容有问题，直接提；就算自己现有的解题思路并不能真的把题目做出来，也要试着用“虽然我没有找到最终解，但我想先说下我现在的想法xxxxx”这样的话术争取一个表达的机会。

答案的对错很多时候并不重要，重要的是你在回答的过程中展现了多少你的思考。

2. 过早地对新题目投降：我们前期大量做题、反复总结，不是为了在面试的时候能够遇到原题上去默写（当然这种情况也是有的），而是为了自己在面对新题目的时候能够有充分的知识储备和自信心去应对。

首先从态度上来说，大家千万不要畏惧新题目，他很可能只是一个化了妆的旧题目。然后就是**不要急于回答**，让这个问题在你脑子里多兜几圈，多试几条路，看看能不能和自己已知的解题技巧关联起来（就像我在前面20多个小节给大家示范的那样）。

如果你觉得这个过程可能会比较久，可以用“我来想一下哈”这样的话术来争取时间。在充分的思考之后，你做出这道题的概率将会大大提升；就算你还是不会做，但你至少已经经历了一个思考的过程，有了可以表达的东西。还是那句话，**不怕做不对，就怕不表达。**

前方的路

拓展阅读

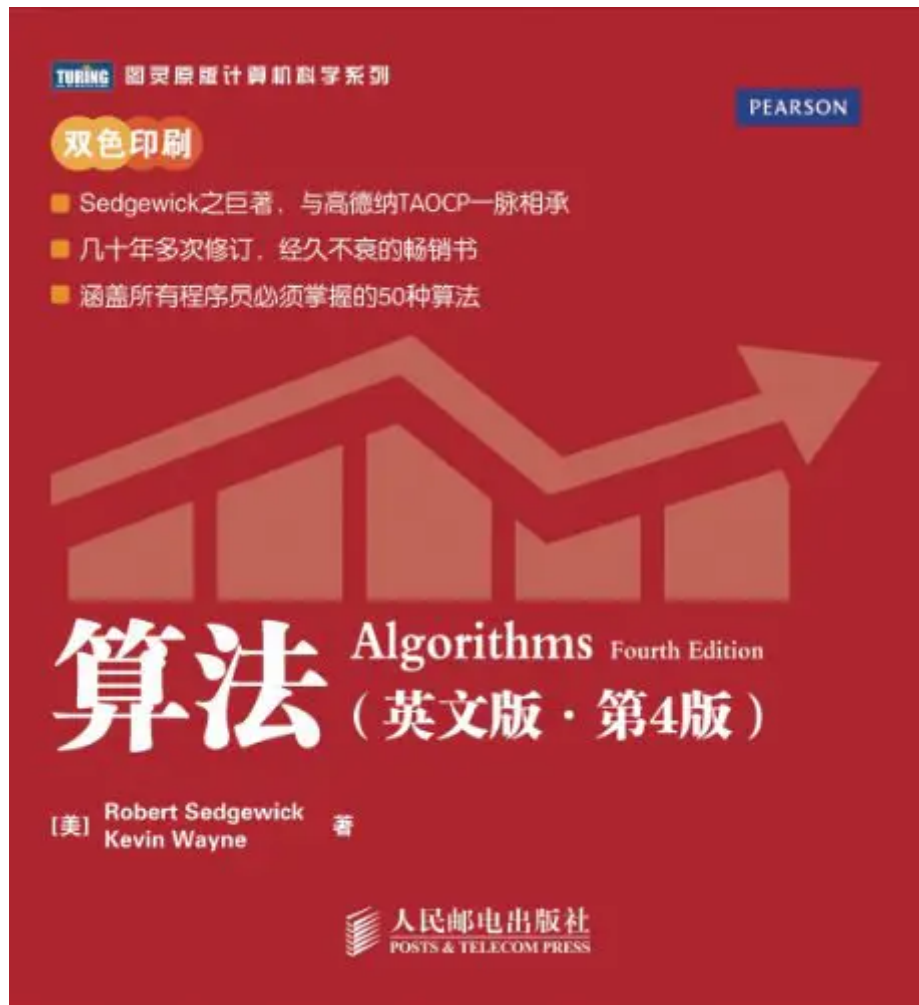
如果你在读完这本小册后，感到意犹未尽，想要在算法世界里更进一步的话，我除了想给你双击666，还想推荐你读读下面这几本书：

《剑指offer》



[非常经典的刷题书](#)，适合考前一个月突击复习找手感。这本书现在已经出到第二版了，笔者从第一版发行开始学习，书中的题目如今看来仍不过时。难度方面，相信对学完算法小册的你来说，已经无法构成威胁，哈哈~

《算法》



[内容翔实、绝佳的入门教材](#)。

这本书虽然不是用 JS 语言描述的，但在表达上对新手非常友好。我第一次读这本书是在大学时，至今仍然庆幸自己没有选错启蒙读物。

《学习JavaScript数据结构与算法》



图灵程序设计丛书

Packt

Learning JavaScript Data Structures
and Algorithms, Third Edition

学习JavaScript 数据结构与算法 (第3版)

[巴西] 洛伊安妮·格罗纳 著
吴双 邓钢 孙晓博 等译

用JavaScript学习常用的数据结构与算法
高效解决编程常见问题



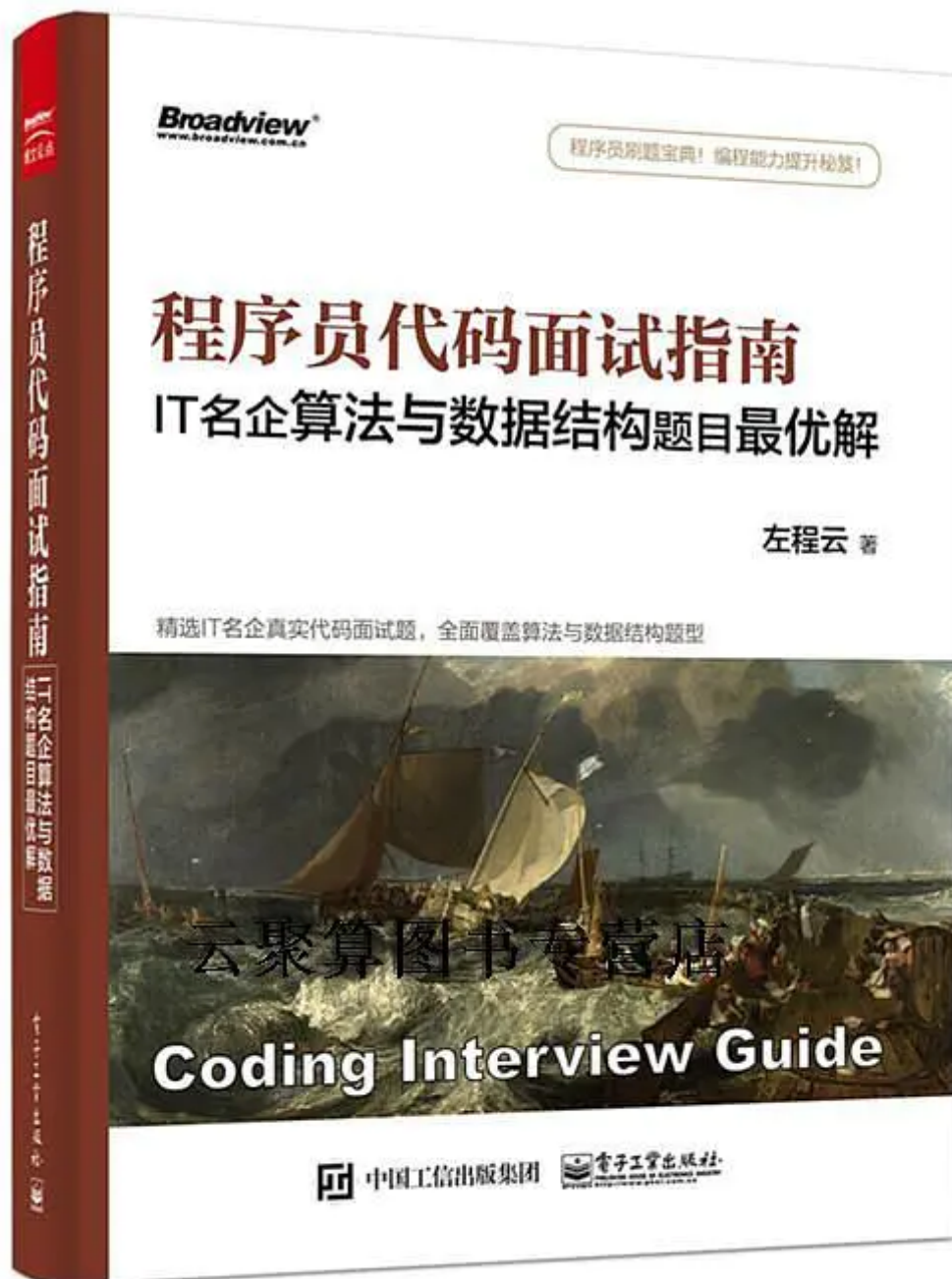
中国工信出版集团



人民邮电出版社
POSTS & TELECOM PRESS

如果你对《算法》的篇幅和非 JS 语言的表达感到望而生畏，我建议你读读这本[对新手更加友好的入门书](#)。这本书很薄，更加侧重于对数据结构的讲解，适合非科班的同学光速入门。

《程序员代码面试指南：IT名企算法与数据结构题目最优解》



这是一本[中国人写的书](#)。

目前来看这本书的评价比较两极化，笔者个人的感觉是大家可以在刷题后期用这本书来练手。书中会讲解一些力扣上没有覆盖的题目，作者解题的思路也时不时会让人觉得耳目一新。

工作/生活上的一些教训和建议

很多读我小册的同学喜欢叫我“老铁”，既然是“老铁”，这里就和大家唠唠嗑，分享一下自己过去这一年里学到的最重要的几个知识点。如果大家也能从中受益，那才真是我的福报了。

1. 身体健康应该是每个程序员优先级为 **P0** 的需求。每半年或一年做一次全面体检，非常有必要。
从这个角度看，学好算法相当于间接续命，因为当你修不动福报的时候，还可以在算法能力的支持下
去外企休息两年。
——病假休到第二周的修言如是说

2. 如果你没有做出过一份好的技术 PPT，那么不应该盲目地排斥做 PPT 这件事情。

过去我很反感 PPT 侠这号人物，觉得别一天到晚瞎比比，有种就 “**show me the code**”。但在过去的一年里，因为各种各样的契机，我自己也不得不频繁使用 PPT 来表达想法、完成宣讲。这才渐渐发现，做 PPT 其实是一个非常好的抽象和提炼自己思路的方式。具体到执行层面，我们更多是钻一个点。而 PPT 却可以帮助我们去把点串成线、进而去看一整个面的问题。

我个人在制作和表达 PPT 的过程中，发现了许多执行阶段所难以感知的问题，也挖掘出了执行阶段无暇思考的一些非常好的项目演进方向，最后收获了大量的实实在在可落地可执行的东西，可以说是受益良多。

看来，要做一个好的工程师，只有 **code** 可以 **show** 是远远不够的，我们还需要刻意训练自己的抽象能力和归纳能力。

PS：插播一个有趣的事情。前两年我看过一篇文章，标题叫做《累死累活干不过做 PPT 的！》。两年前的我深以为然（这帮 PPT 侠真讨厌！），两年后的我再次深以为然（做 PPT 牛逼！）。哈哈~ 如果你读到这里，仍然对 PPT 侠抱有偏见，那么也没关系，可以试试从写各种文档开始，养成记录和总结工作的好习惯。毕竟，PPT 也只是我们做总结的一种形式嘛~

3. 关注他人，做利他的事情，往往就是在做对的事情。

这话是写给和曾经的我一样讨厌做业务开发的同学们的。我相信不少同学讨厌做业务，无非是出于以下两方面的想法：

a. “这玩意儿不过是在堆代码，对我的技术提升一点卵用也没有”

b. “我不想和不懂技术的产品/运营/品牌商/各路业务方打交道”

无论是 a 还是 b，本质上都是因为我们不想去关注与技术无关的人和事，想做纯粹的程序员。

过去一年里，我因为秉持对“纯粹”的坚持，吃了不少工作上的苦头。后来在对 ~~3.75~~ 客户价值的渴求下，尝试转变思路，把一部分研究技术的时间拿出来做业务输入，发现很多事情都变得豁然开朗。

其实作为一个好的工程师，我们的任务确实不是只有 **coding** 这么简单，更多地还要去思考如何用技术去创造产品上的价值，进而去实现业务上的目标。工程师与科学家的区别，就在于他关注的不是技术本身，而是如何使用技术去创造实际的价值。这就要求我们去关注到更多的人，更多的事情，同时思考自己的技能和这些人、这些事之间的关联。这些思考，往往能为我们带来技术上的新思路、新方法。

致谢

正事说完了，接下来最最不能省略的，仍然是致谢。

过去我写致谢，喜欢说“感谢大家读完了这本小册”。确实，在职程序员主动扩展自己的知识边界本身就是一件不易的事情，能够学完28节算法的大家都是狠人，修言给你跪下点个赞！

但算法小册更特别，在这里，我还要谢谢最初那一波每日一push、坚持要看算法小册的读者。算法小册诞生于掘金的一个比较特殊的（没人管的）时期，命题、试读、上线，整个过程中很大程度上仰仗了读者老铁们的参与、支持和鼓励。总之，掘友牛逼！

尾声

前前后后说了这么多，最后还是要回到算法面试上来。

现在的你，有了硬基础，有了软实力。但这些都还不够，你需要做的，是在时间允许的前提下，是刷更多的题、做更多的总结。

当你不知道该用什么姿势刷题，不知道该从什么角度出去去总结的时候，欢迎你随时回到算法小册这里，再看看自己走过的路，相信一定会有别样的收获。

希望大家都能身体健康，学有所成。也欢迎大家添加我的微信 xyalinode，和我聊聊工作与生活。大家加油！