TURING

Google · Apple · Amazon

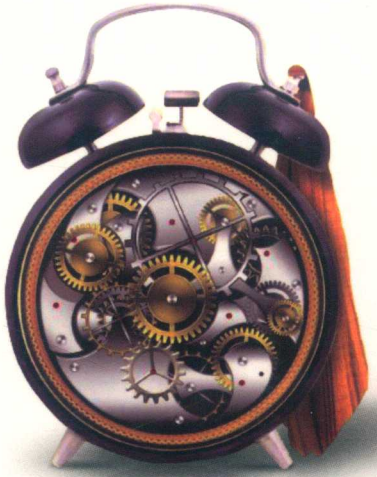
189道知名科技公司编程面试真题及解答

数十万程序员求职成功的敲门砖

程序员面试金典

（第6版）

[美] 盖尔·拉克曼·麦克道尔 ◎ 著　　刘博楠 赵鹏飞 李琳骁 漆犇 ◎ 译



CRACKING THE CODING INTERVIEW

6th Edition

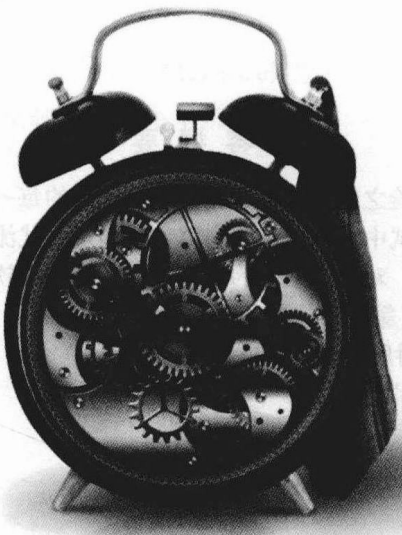
中国工信出版集团　人民邮电出版社 POSTS & TELECOM PRESS

TURING

程序员面试金典

（第6版）

[美] 盖尔·拉克曼·麦克道尔 ◎ 著　　刘博楠 赵鹏飞 李琳骁 漆犇 ◎ 译



CRACKING THE CODING INTERVIEW

6th Edition

人民邮电出版社
北　京

图书在版编目（CIP）数据

程序员面试金典 : 第6版 / (美) 盖尔·拉克曼·麦克道尔著 ; 刘博楠等译. -- 2版. -- 北京 : 人民邮电出版社, 2019.9
ISBN 978-7-115-51719-7

Ⅰ. ①程… Ⅱ. ①盖… ②刘… Ⅲ. ①程序设计－资格考试－自学参考资料 Ⅳ. ①TP311.1

中国版本图书馆CIP数据核字(2019)第155639号

内 容 提 要

本书是原谷歌资深面试官的经验之作，层层紧扣程序员面试的每一个环节，全面而详尽地介绍了程序员应当如何应对面试，才能在面试中脱颖而出。内容主要涉及面试流程解析，面试官的幕后决策及可能提出的问题，面试前的准备工作，对面试结果的处理，以及出自微软、苹果、谷歌等多家知名公司的189 道编程面试题及详细解决方案。第 6 版修订了上一版中一些题目的解法，为各章新增了介绍性内容，加入了更多的算法策略，并增添了对所有题目的提示信息。

本书适合程序开发和设计人员阅读。

◆ 著　　[美] 盖尔·拉克曼·麦克道尔
译　　刘博楠　赵鹏飞　李琳骁　漆　犇
责任编辑　张海艳
责任印制　周昇亮
◆ 人民邮电出版社出版发行　　北京市丰台区成寿寺路11号
邮编　100164　　电子邮件　315@ptpress.com.cn
网址　http://www.ptpress.com.cn
涿州市京南印刷厂印刷
◆ 开本：787×1092　1/16
印张：38
字数：991千字　　2019年 9 月第 2 版
印数：17 401 - 20 900册　　2019年 9 月河北第 1 次印刷
著作权合同登记号　图字：01-2015-8298号

定价：149.00元

读者服务热线：(010)51095183转600　印装质量热线：(010)81055316
反盗版热线：(010)81055315
广告经营许可证：京东工商广登字 20170147 号

版权声明

感谢 Davis 和 Tobin，以及生活中使我们赏心悦目的一切事物。

中文版推荐序一

拉勾招聘是中国优秀的互联网招聘平台之一。如今，有 50 多万互联网公司以及 2000 多万互联网人才在使用拉勾招聘找工作。拉勾招聘对互联网人才，尤其是程序员这个群体非常了解。人才是一家公司最重要的资产，优秀的程序员永远稀缺！面试时，公司看重的不仅仅是候选人的经验，对于其文化契合度、基础算法，以及未来潜力的考察也越来越细致和深入。对求职者来说，找到一份称心如意的工作极其不易，需要精心准备，这也是和公司缘分匹配的一个过程。

我本人也经历过找工作的艰辛，所以对此深有体会。不管是经验丰富的“老司机”“老鸟”，还是初入职场的“小白”“菜鸟”，面试准备都是必不可少的。求职者可以登录招聘网站（比如拉勾招聘）了解一家公司的发展历程、公司人数，体验其现有产品，掌握其他候选人对这家公司的面试评价，从而更加全面立体地了解面试公司。当然，求职者也会关心公司都有哪些福利，办公环境是否舒适，是否有健身房，是否有下午茶，等等。

绝大多数互联网公司面试程序员时都会考查算法和数据结构，对于每一个程序员来说，提升算法和数据结构等方面的技能至关重要。算法题目形式多样，通过这些题目，能考查求职者的基础知识是否扎实，是否有分析问题和解决问题的能力。《程序员面试金典》是一本经典求职面试书，书中涉及数量众多、质量上乘的算法和数据结构面试题，不仅有解题思路和原理的讲解，还有实例演示和不同难度题的多种解法，是程序员求职的好帮手，闲暇之余翻阅一下也会有助于日常编码能力的提升。

在此，祝每一位程序员都能找到自己满意的工作，斩获心仪的 Offer！

——马建春，拉勾招聘 CTO

中文版推荐序二

《程序员面试金典》是一本硅谷互联网公司技术面试经典图书。作者盖尔·拉克曼·麦克道尔结合自身丰富的面试经历，以及多年对互联网招聘行业形势的整理归纳，帮助许多想要加入Facebook、亚马逊、微软、苹果等互联网企业的求职者获得了心仪的工作机会。

算法和数据结构在现今技术面试环节中极为重要。通过力扣（LeetCode）相关数据我们发现，不论是国内一线互联网大厂还是创业公司，对程序员算法和数据结构的掌握程度越来越重视，甚至在技术面试中要求手写代码。面试过程中除了会出现一些常用的数据结构，比如树、栈、队列等问题，也会出现一些高级的数据结构，比如图、优先队列等问题。对于算法，从最基础的排序、搜索到动态规划，都是企业非常看重的考核点。技术栈每天在不断变化，越来越多的互联网企业看中的不再只是面试者的技术广度，掌握多门计算机语言、了解多种技术栈已经不是考核程序员最为重要的因素，更为重要的是其能适应这个行业的变化并不断成长。这背后，最为核心的要素便是计算机科学思维、算法思维以及逻辑思维能力。

对于程序员读者，当你仔细阅读后，会发现本书除了能给你带来算法和数据结构等相关知识以及互联网企业招聘模式，还能帮你掌握如何将知识转化为职业成长的技能，有效应对互联网企业人才招聘模式的转变，从而将日常解决技术问题的能力提升一个层面。如果你缺乏相关工作经验，那么本书能帮你在专业技能上查缺补漏。通过阅读，你将能够整理出一个成系统的学习方向，掌握互联网企业面试流程、考点，以及一些很难了解到的注意事项，做到提前避“坑”。

对于面试官读者，判断求职者的上手速度以及未来成长空间格外重要，但更需要考察其将思路快速转化为代码的能力。借鉴硅谷成熟的模式，适当地为白板面试做些准备，能够帮助你寻找到支撑业务长久发展并有巨大成长空间的优秀工程师。

职业技能提升非一日之功，静下心来仔细阅读，你将收获巨大。Have fun coding!

张云浩，力扣（LeetCode）CEO

序

亲爱的读者：

我先做个自我介绍。

我不是招聘人员，而是软件工程师。正因如此，我深知要在面试现场迅速想出精妙算法并在白板上写下完美代码的感受。之所以能感同身受，是因为我与你有过同样的经历：我参加过谷歌、微软、苹果、亚马逊以及其他诸多公司的面试。

我也当过面试官，让求职者做过同样的事情。我还筛选过成千上万份简历，在其中“上下求索”，希望挑出那些或许能在面试中脱颖而出的工程师。当求职者解出或者试图解出那些具有挑战性的题目时，我评估着他们的表现。在谷歌时，就某位求职者是否达到了录用要求，我曾与招聘委员会的同事有过激烈争辩。因为我反复地经历过整个流程，所以对招聘的各个环节了如指掌。

亲爱的读者，你也许要在明天、下周或明年去迎接面试挑战。我撰写本书，旨在帮助你加深对计算机科学基础知识的理解，并在此之后学会该如何运用这些基础知识，成功闯过技术面试这一关。

第 6 版在第 5 版的基础上增加了 70% 的内容：添补了更多的面试题，修订了部分原有题目的解法，为各章新增了介绍性内容，加入了更多的算法策略，增添了对所有题目的提示信息，等等。欢迎访问我们的网站（http://www.CrackingTheCodingInterview.com），你可以跟其他求职者互通有无，发现新天地。

与此同时，我也感到无比兴奋，你一定能从本书中学到新的技能。充分的准备将会使你拥有各种技术技能和沟通技巧。不管最终结果如何，只要拼尽全力，便无怨无悔！

请务必用心研读本书前面的介绍性章节，其中的要点和启示也许可以决定你的面试结果，“录用”与“拒绝”就在一线之间。

此外，**切记：面试非易事！**根据我在谷歌多年面试的经历，我留意到有些面试官会问一些“简单”的问题，有些则会专挑难题来问。但是你知道吗？面试中碰到简单的问题，不见得就能轻松过关。完美解决问题（只有极少数求职者才能做到）不是公司录用你的关键，只有把题答得比其他求职者更出色才能让你脱颖而出。所以，碰到棘手的难题不要惊慌，或许其他人一样觉得很难。解答得不够完美是没有问题的。

请努力学习，不断实践。祝你好运！

盖尔·拉克曼·麦克道尔

CareerCup.com 创始人兼 CEO

前　言

招聘中的问题

讨论完招聘事宜，我们又一次沮丧地走出会议室。那天，我们重新审查了 10 位“过关”的求职者，但是全都不堪录用。我们很纳闷，是自己太过苛刻了吗？

我尤为失望，因为由我推荐的一名求职者也被拒了。他是我以前的学生，以高达 3.73 的 GPA 毕业于华盛顿大学，这可是世界上最棒的计算机专业院校之一。此外，他还完成了大量的开源项目工作。他精力充沛、富于创新、头脑敏锐、踏实能干。无论从哪方面来看，他都堪称真正的极客。

但是，我不得不同意其他招聘人员的看法：他还是不够格。就算我的强力推荐可以让他侥幸过关，但他在后续的招聘环节可能还是会失利，因为他的硬伤太多了。

他尽管十分聪明，但答起题来总是磕磕巴巴的。大多数成功的求职者都能轻松搞定第一道题（这一题广为人知，我们只是略作调整而已），可他却没能想出合适的算法。虽然他后来给出了一种解法，但没有提出针对其他情形进行优化的解法。最后，开始写代码时，他草草地采用了最初的思路，可这个解法漏洞百出，最终还是没能搞定。他算不上表现最差的求职者，但与我们的“录用底线”相去甚远，结果只能铩羽而归。

几个星期后，他给我打电话，询问面试结果。我很纠结，不知该怎么跟他说。他需要变得更聪明些吗？不，他其实智力超群。做个更好的程序员？不，他的编程技能和我见过的一些最出色的程序员不相上下。

与许多积极上进的求职者一样，他准备得非常充分。他研读过 Brian W. Kernighan 和 Dennis M. Ritchie 合著的《C 程序设计语言》，也学习过麻省理工学院出版的《算法导论》等经典著作。他可以细数很多平衡树的方法，也能用 C 语言写出各种花哨的程序。

我不得不遗憾地告诉他：光是看这些书还远远不够。这些经典学院派著作能够教会你错综复杂的研究理论，帮助你成为出类拔萃的软件工程师，但是对程序员的面试助益不多。为什么呢？容我稍稍提醒你一下：即使从学生时代起，**你的面试官**其实都没怎么接触过所谓的红黑树算法。

要顺利通过面试，就得“**真枪实弹**”地做准备。你必须演练**真正的**面试题，并掌握它们的解题模式。你必须学会开发新的算法，而不是死记硬背见过的题目。

本书就是我根据自己在顶尖公司积累的第一手面试经验和随后在辅导求职者面试过程中提炼而成的精华。我曾经与数百名求职者有过“交锋”，本书可以说是我面试过几百位求职者后的结晶。同时，我还从成千上万求职者与面试官提供的问题中精挑细选了一部分。这些面试题出自许多知名的高科技公司。可以说，本书囊括了 189 道世界上最好的程序员面试题，它们都是从数以千计的好问题中挑选出来的。

我的写作方法

本书重点关注算法、编程和设计问题。为什么呢？尽管面试中也会有行为面试题，但是答案会随个人的经历而千变万化。同样，尽管许多公司也会考问细节（例如，“什么是虚函数”），但通过演练这些问题而取得的经验非常有限，更多的是涉及非常具体的知识点。本书只会述及其中一些问题，以便你了解它们“长”什么样。当然，对于那些可以拓展技术技能的问题，我会给出更详细的解释。

我的教学热情

我特别热爱教学。我喜欢帮助人们理解新概念，并提供一些学习工具，从而充分激发他们的学习热情。

我第一次正式的教学经历是在美国宾夕法尼亚大学就读期间，那时我才读大二，同时担任本科计算机科学课程的助教。我后来还在其他一些课程中担任过助教，并最终在宾夕法尼亚大学推出了自己的计算机科学课程。该课程专注于教授一些实际的“动手”技能。

在谷歌担任工程师时，培训和指导新的工程师是我最喜欢的工作之一。后来，我还利用“20%时间”[①]在华盛顿大学教授两门计算机科学课程。

多年之后，我仍然继续在教授计算机科学的相关课程，但是这次我的目标是帮助创业公司的工程师准备收购面试。我看到他们犯了不少错误，经历了很多困难，而我正好拥有帮助他们解决这些问题的技巧和策略。

《程序员面试金典》《产品经理面试宝典》《金领简历：敲开苹果、微软、谷歌的大门》和CareerCup都能充分体现我的教学热情。即便是现在，你也会发现我经常出现在CareerCup.com上为用户答疑解惑。

请加入我们的行列吧！

电子书

扫描如下二维码，即可购买本书电子版。

① 在谷歌，“20%时间”是指公司允许工程师每个星期花一天时间做与工作无关的项目，详见谷歌官方博客。

——译者注

目　录

第 1 章

面试流程

在大多数顶尖科技公司和许多其他公司的面试中，算法和编程问题占最大一部分。这些问题可以归类为问题解决型题目（problem-solving question）。面试官希望测试你解答未见过的算法题目的能力。

很多时候，你或许只能够在一场面试中完成一道题。45 分钟并不长，在这样短的时间内很难解决几个不同的问题。

整个解题过程中，你应该尽可能地大声讲解你的思考过程。有时面试官或许会中途打断你，想给你一些提示。没关系，这十分常见，而且这并不意味着你表现得很糟糕（当然不需要提示则会更好）。

面试结束后，面试官会对你的表现有一个基本的印象。或许，他会为你的表现打一个分数，但是，分值实际上并不代表一个定量的评价。从来没有一个表格能列出不同的表现应该获得多少分，面试成绩并不是这样得出的。

其实，面试官一般会根据以下几个方面对你的表现做出评价。

- 分析能力：你在解决问题的过程中是否需要很多帮助？你的解决方案优化到了什么程度？你用多长时间得出了解决方案？如果不得不设计或者架构一个新的解决方案，你是否能够很好地组织问题，并且全面考虑不同决策的取舍？
- 编程能力：你是否能够成功地将算法转化为合理的代码？代码是否整洁且结构清晰？你是否思考过潜在的错误？你是否有良好的编程风格？
- 技术知识、计算机科学基础知识：你是否有扎实的计算机科学以及相关技术的基础知识？
- 经验：你在过去是否做出过良好的技术决策？你是否构建过有趣且具有挑战性的项目？你是否展现出魄力、主动性或者其他的重要品质？
- 文化契合度、沟通能力：你的个人品质和价值观是否与公司和团队相契合？你和面试官是否沟通顺畅？

这些方面的权重会根据不同的题目、面试官、职位、团队和公司有所变化。对于一个标准的算法题目，面试的表现基本上完全取决于前三个方面。

1.1 为什么

这是求职者在开始准备面试时最常见的问题之一。面试流程为什么是这样的？现实中可能存在以下情况。

(1) 许多出色的候选人在这些面试中表现不佳。

(2) 如果真的遇到这样的问题，你可以查找答案。

(3) 在现实世界中，你很少会使用诸如二叉搜索树之类的数据结构。如果你确实需要，肯定可以学习。

(4) 白板编程是模拟的环境。显然，在现实世界中，你永远不会在白板上编写代码。

这些抱怨并不是没有依据的。事实上，我至少在一定程度上赞同这些说法。

同时，对于一些职位（并不是所有职位），有理由以这种方式进行面试。你是否同意这样的逻辑并不重要，但是你最好能够了解在面试中为什么会问及这些问题，这有助于你理解面试官的想法。

1.1.1　错过了优秀人才是可以的

虽然令人难过（也令求职者沮丧），但这是真的。

对公司而言，优秀的求职者被拒实际上是可以接受的。公司的目的是组建强大的员工队伍，因此可以接受错过优秀求职者这一事实。当然，公司并不希望出现这样的情况，因为这样会增加招聘的成本。尽管如此，只要仍然可以拥有足够多的优秀员工，这是一个可以接受的折中方法。

公司更担心的是“错误肯定”：一些人在面试中表现得很好，但实际上并不是非常优秀。

1.1.2　解决问题的技能很宝贵

你如果能够独自或在一些提示下解决几个难题，那么你很可能擅长于开发最优算法。你是个聪明人。

聪明的人往往能够出色地完成工作，这对公司来说是很有价值的。当然，这不是唯一重要的事情，但是这是个非常重要的亮点。

1.1.3　基础数据结构和算法知识很有用

许多面试官认为，计算机科学的基础知识实际上非常有用。树、图、链表、排序等经常会在工作当中出现，所以应该掌握这些知识。

你可以根据需要学习这些知识吗？当然可以。但是，如果你不知道二叉搜索树的存在，就很难知道何时应该使用它。而如果你知道它的存在，那么也就基本上掌握了它的基础概念。

另外一些面试官认为，依靠数据结构和算法来判断求职者的表现是一种很好的“替代”手段。即使这些知识学起来并不是很难，但是他们认为，是否掌握这些技能和能否成为优秀的开发人员有很强的相关性。掌握这些知识往往意味着你已经完成了计算机科学专业的学历教育（在这个过程中，你已经学到并掌握了相当广泛的技术知识）或者自学了这些知识。无论哪一种情况，这都是一个好的信号。

数据结构和算法知识出现在面试中的另一个原因是：很难问一个不涉及这些知识的问题解决型题目。事实证明，绝大多数问题解决型题目都涉及一些相关的基础知识。当有足够多的求职者掌握这些基础知识时，考查有关数据结构和算法的问题则很容易形成一种模式。

1.1.4　白板让你专注于重要的事情

要在白板上编写完美的代码，确实十分困难。不过面试官并不期望你能够做到完美。绝大多数人的代码中会出现一些 bug 或小的语法错误。

白板的好处在于，你可以在某种程度上专注于整体结构。你并没有编译器，所以不需要使代码能够通过编译。你也不需要写出整个类的定义和样板代码。你应该专注于代码中有趣、关键的部分，即题目所要求的核心功能。

这并不是说你应该只写一些伪代码，也不是说代码的正确性无关紧要。大多数面试官并不接受伪代码，而且代码中的错误越少越好。

另外，使用白板会鼓励求职者多交流、多解释他们的思考过程。而如果给求职者一台计算机，则会大大减少与他们的交流。

1.1.5 但这并不适用于每个人、每家公司和每种场合

上述内容旨在帮助你了解公司的想法。

我个人怎么看？在适当的场合，当这样的面试流程有效时，可以对求职者的问题解决能力进行合理的判断，因为表现出色的人往往比较聪明。

然而，这样的面试流程并不总是奏效的。你或许会遇到不称职的面试官，或者面试官会问及不合适的题目。

另外，这样的方法也并不适合所有的公司。一些公司会更重视以前的经验，或者需要求职者具有特定的技术能力。而这些数据结构和算法问题并没有考虑到这些方面。

这样的过程也不会衡量求职者的职业道德或者专注力。然而，几乎没有任何一种面试流程可以评估这方面的能力。

该面试流程并不是完美的，但是又有什么样的面试流程是完美的呢？所有的方法都有缺点。

我的结论是：现实既然如此，只需尽力而为，做到最好。

1.2 面试问题的来源

求职者经常会问某个公司最近使用的面试问题是什么。会这样问，表示求职者对于面试问题的来源存在着根本性误解。

在大部分公司，并不存在面试问题的清单。实际上，每个面试官会挑选自己的面试问题。

因为使用哪些问题在某种程度上是完全自由的，所以并不会有一道面试题成为“谷歌最新面试题”——这只不过是因为就职于谷歌的一位面试官恰巧最近问了这道题目罢了。

今年谷歌使用的面试题和三年前使用的面试题其实并没有什么区别。实际上，谷歌和类似的公司（亚马逊、Facebook 等）所使用的面试题一般说来也没有什么不同。

不同公司的面试风格存在着一些差异。一些公司专注于算法（有时会涉及一些系统设计的内容），另一些公司则喜欢基础知识题目。但是在同一类别的题目中，很少会出现一道题属于一家公司而不属于另一家公司的现象。一道谷歌算法面试题和一道 Facebook 算法面试题基本上是一样的。

1.3 一切都是相对的

如果没有评分体系，如何评估你？一位面试官怎样才能确定对你应该有怎样的期望？

问得好。搞清楚这个问题的答案，实际上很有意义。

同一位面试官会使用同一道面试题来比较你和其他的求职者，这是一个对比的过程。

例如，假设你想出了一个很不错的脑筋急转弯或者是数学题目。你问好朋友 Alex 这道题目，他花了 30 分钟求解出了答案。你问 Bella 这道题目，她用了 50 分钟。Chris 一直都没有解出这道题。虽然 Dexter 只用了 15 分钟，但是你不得不给他一些关键的提示信息，否则他花费的时间要远多于此。Ellie 花了 10 分钟，并且她提出了一个你从没有想到过的解题方法。Fred 花了 35 分钟。