



Programming Interviews Exposed: Coding Your Way
Through the Interview, 4th Edition

斩获Offer: IT名企面试策略与编程笔试精解

约翰·摩根 (John Mongan)
[美] 诺亚·金德勒 (Noah Kindler) 著
埃里克·吉格尔 (Eric Giguère)
程 钢 等译



清华大学出版社

斩获 Offer: IT 名企面试 策略与编程笔试精解

约翰·摩根(John Mongan)
[美] 诺亚·金德勒(Noah Kindler) 著
埃里克·吉格尔(Eric Giguère)

程 钢 等译

清华大学出版社

北 京

John Mongan, Noah Kindler, Eric Giguère

Programming Interviews Exposed: Coding Your Way Through the Interview, 4th Edition

EISBN: 978-1-119-41847-4

Copyright © 2018 by John Wiley & Sons, Inc.

All Rights Reserved. This translation published under license.

Trademarks: Wiley, the Wiley logo, Wrox, the Wrox logo, Programmer to Programmer, and related trade dress are trademarks or registered trademarks of John Wiley & Sons, Inc. and/or its affiliates, in the United States and other countries, and may not be used without written permission. Visual Studio is a registered trademark of Microsoft Corporation. All other trademarks are the property of their respective owners. John Wiley & Sons, Inc., is not associated with any product or vendor mentioned in this book.

本书中文简体字版由 Wiley Publishing, Inc. 授权清华大学出版社出版。未经出版者书面许可, 不得以任何方式复制或抄袭本书内容。

北京市版权局著作权合同登记号 图字: 01-2018-5171

Copies of this book sold without a Wiley sticker on the cover are unauthorized and illegal.

本书封面贴有 Wiley 公司防伪标签, 无标签者不得销售。

版权所有, 侵权必究。侵权举报电话: 010-62782989 13701121933

图书在版编目(CIP)数据

斩获 Offer: IT 名企面试策略与编程笔试精解 / (美) 约翰·摩根(John Mongan), (美) 诺亚·金德勒(Noah Kindler), (美) 埃里克·吉格尔(Eric Giguère) 著; 程钢 等译. —北京: 清华大学出版社, 2019

书名原文: Programming Interviews Exposed: Coding Your Way Through the Interview, 4th Edition
ISBN 978-7-302-52671-1

I. ①斩… II. ①约… ②诺… ③埃… ④程… 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 053597 号

责任编辑: 王 军

封面设计: 孔祥峰

版式设计: 思创景点

责任校对: 牛艳敏

责任印制: 杨 艳

出版发行: 清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址: 北京清华大学学研大厦 A 座

邮 编: 100084

社 总 机: 010-62770175

邮 购: 010-62786544

投稿与读者服务: 010-62776969, c-service@tup.tsinghua.edu.cn

质量反馈: 010-62772015, zhiliang@tup.tsinghua.edu.cn

印 装 者: 北京鑫海金澳胶印有限公司

经 销: 全国新华书店

开 本: 170mm×240mm

印 张: 20.25

字 数: 444 千字

版 次: 2019 年 4 月第 1 版

印 次: 2019 年 4 月第 1 次印刷

定 价: 59.80 元

产品编号: 080802-01

译者序

程序员是互联网时代的主要建设者，其薪资待遇遥遥领先于其他行业。虽然我国各项 IT 技术的人才缺口较多，但是由于人口基数大的客观现实和信息技术教育的日益普及，岗位竞聘显得极其激烈。如今连小学生都在学习 Python 编程，不难想象，日后会有多少个程序员！若要在应聘中脱颖而出，除了有过硬的技术本领，还应做好哪些准备呢？答案就在本书中！

从内容上来看：

本书是面试的指南针，无论读者处在什么阶段——胸有成竹或一脸茫然、高校新晋或职场老手、兴趣使然或宏图大志，本书从求职之前的准备工作开始，事无巨细地在每一个分岔点做好守候。读者适合什么岗位、知识如何储备、包装如何到位、心态如何控制、与人事管理人员如何交流、在面试官面前如何表现，均有建议。

本书是编程知识的百宝箱，无论读者是闲来无事还是临时烧香，所喜或所需的门道——基本数据结构、经典算法、编程语言特征、设计模式、当下主流技术，甚至活跃解题思维的脑筋急转弯，本书都为读者逐一备好，开篇引进门，再拓宽知识领域，以字里行间对面试解题的刨根究底，激发读者对编程的兴趣和热情。

本书是事业的助推器，书中描述的面试过程或编程知识能与读者的切身经历回响共鸣。本书从第 1 版至今，已逾十年。时光荏苒，有什么编程知识还依旧重要，有什么经验教训还依旧实用，本书积淀下来，就是为了对读者的事业再添一把力。

另外，通过翻译本书，我还发现了本书具有如下特色：

(1) 知识涵盖面广。除了剖析求职准备、求职过程、电话面试的关键步骤和关注重点外，还分别讲解了二十余项面试重点，包括链表、树、图、数组和字符串、递归、排序、并发、面向对象编程、设计模式、数据库、图形和位运算、数据科学、随机数、人工智能、数理统计、脑筋急转弯、知识方面和非技术方面的问题等。

(2) 题型具有典型性。若读者掌握了应对书中题型的技巧，就能在面试时触类旁通，处变不惊。

(3) 讲解具有启发性。与一般教科书不同，本书不是给出问题后马上提供并解释答案，而是通过模拟真实面试的场景，引导读者获得线索，找到问题切入点，最终得到答案。在引导过程中锻炼读者的编程思维。

本书久经打磨，是多种 IT 技术职位竞聘不可多得的好书。尽管如此，我还是敬请读者在掌握本书要义之后，继续扩大知识面，特别是要关注和钻研与应聘职位相关的专业知识，方能稳操胜券。

本书主要由程钢翻译，在翻译过程中力求“信、达、雅”，但是鉴于译者水平有限，错误和失误在所难免，如有任何意见和建议，请不吝指正，感激不尽！

关于作者

John Mongan 是一位自学成才的程序员，担任多家软件公司和制药公司的顾问，具有专业经验。他在软件测试技术方面拥有三项专利，并拥有加州大学圣地亚哥分校的生物信息学硕士学位和博士学位，在学校的研究方向为蛋白质动力学的超级计算机模拟。目前是加州大学旧金山分校放射学和生物医学影像学院信息系的助理教授和副主席，研究重点是机器学习应用于放射学数据和计算机化临床决策支持。

Noah Kindler 是安全技术公司 Avira 的技术副总裁。他领导多个产品的软件设计和开发团队，用户群超过 1 亿人。

Eric Ciguère(很久以前)在 Commodore VIC-20 上开始 BASIC 编程，并且迷上了计算机编程。他拥有滑铁卢大学计算机科学的 BMath 和 MMath 学位，有着丰富的专业编程经验，并且出版了几本编程书籍。目前是 Google 的一名资深软件工程师。

关于技术编辑

Wayne Heym 博士是俄亥俄州立大学工程学院计算机科学与工程系的高级讲师。他还与学院的可重用软件研究组(Reusable Software Research Group, RSRG)合作。他对 RSRG 的开发原则和语言——具有可验证性和效率的可重用软件语言(Reusable Software Language with Verifiability and Efficiency, RESOLVE)——保持着浓厚的兴趣。他喜欢将计算机编程艺术美好的一面和计算机编程科学神奇的一面传授给初级程序员，还喜欢将程序员引入计算机科学理论基础这一富足王国。

Dan Hill 是一名软件工程师和软件开发经理，拥有十五年以上的编程经验，从事的项目包括 Web 开发、用户界面设计、后台系统架构、数据库、安全和加密以及移动应用开发。他曾在硅谷创业公司和大型科技公司工作，并组织了很多编程面试，拥有斯坦福大学计算机科学学士和硕士学位。

致 谢

我们非常感谢 Wiley 和 Serendipity23 Editorial Services 的同事们为第 4 版修订付出的努力。感谢项目编辑 Adaobi Obi Tulton 的奉献，他熟练的编辑能力、干练的组织水平和专业的坚持态度使我们的工作保持正轨。执行编辑 Jim Minatel 的个人关注特别关键，感谢他们的参与、工作和相助。

技术编辑 Wayne Heym 和 Dan Hill 的工作大大提高了《斩获 Offer: IT 名企面试策略与编程笔试精解》一书的质量，他们曾为之前的版本做出过重要贡献。他们深思熟虑的评论和细致的审查消除了许多错误和疏漏，并且不可估量地提升了本书的清晰度。感谢他们方方面面的工作。还要感谢 Andrew Taylor 对新数据科学材料的进一步审查，以及 Tom Mongan 在校对方面的帮助。

没有前三版，就没有第 4 版，很多人为前三版做出了贡献。特别感谢 Michael J. Mongan 在第 3 版的参与和帮助。感谢第 3 版的编辑 Maureen Sparsse。此外，感谢编辑 Margaret Hendrey 和 Marjorie Spencer，感谢他们的耐心和乐于助人。也非常感谢曾经的审稿人和顾问 Dan Hill、Elise Lipkowitz、Charity Lu、Rob Maguire 和 Tom Mongan。Dan 的贡献尤其大——他的细致而详细的审查极大地提高了第 1 版的质量。

序 言

解答编程面试中的题目需要掌握一系列技能，与成为优秀程序员的要求不尽相同。当你首次接触编程面试题目时，也许不太擅长，但可以培养和提高这些技能。本书作为培养和提高面试技能的第一步，将利用你的编程专业技能，迅速让你精通编程面试技能。

自第1版出版以来，本系列图书有效地建立了编程书籍的一个新主题领域，现在众多网站、博客和论坛都在为编程面试给出相关建议和问题样例。有这么多手段，为什么《斩获 Offer：IT 名企面试策略与编程笔试精解》还能够吸引你为之投入时间和金钱呢？

本书的重点是授予你在编程面试中取得成功所需的技巧和方法。在给出每个问题后，通过阐述导向解决方案的思维过程来强化相应技能，指示你在遇到问题时可以如何向前迈进。这些技能与常见的编程技能重叠，但它们不是一回事。有些优秀的程序员就是活生生的例子，由于没有锻炼面试技巧，他们在编程面试中败得焦头烂额。我们在早期职业生涯中曾经好几次焦头烂额，但你可以通过本书筹谋准备，以避免这种情况发生。一旦掌握了本书讲授的技能，就可以将其应用于其他书籍和网络中出现的各种题目。

要想成为编程面试的解题高手，有件事情必须永远不变，即不能只注重被动阅读解题技巧，更重要的是练习实践。如果在阅读本书题目解答之前已经尽可能多地独自研究了多种应对策略，那么你将从本书中获得更多信息。

尽管自第1版以来该系列图书的内容已经大大扩充，并且所使用的编程语言发生了变化，但我们仍然坚持第1版前言中所描述的目标和方法，如后所附。

第1版前言

编程、开发、技术咨询及其他岗位需要进行编程面试，本书有助于你在相应的求职中做好准备。编程面试与传统的求职和面试书籍中所述的几乎没有相似之处。该面试几乎完全由编程问题、智力难题和计算机技术问题组成。本书讨论了其中可能覆盖的各种问题，并以真实面试问题为例说明需要如何给出这些问题的最优方案。

此时此刻，你可能想知道我们是谁，凭什么我们有权写这本书。我们都是刚毕业的学生，在过去几年中经历了很多面试。从为初创公司编写设备驱动程序，到老牌大公司的技术咨询，我们参加了各种工作的面试。本书基于我们从这些面试中获得的经验和观察——做什么会被录用，做什么不会被录用。就面试而言，我们相信像本书这样的书有可能是最佳的基础读物。我们将告诉大家在美国顶级软件和计算机公司面试时的真实情况，以及需要做些什么来取得成功，而不是给出一些人力资源主管关于应该如何进行面试的想法，或者专注于介绍猎头会如何看待面试。



说实话，我们认为现在的面试模式并不理想。目前的模式过于强调解决难题和熟悉相对有限的知识体系的能力，而且通常无法衡量对企业成功至关重要的许多技能。

为此，我们没有在本书中编造任何问题。每一道题都出自实际面试，力求最新。问题的类型设置和难度分布类似于你在面试中会遇到的实际情况。我们必须强调，本书中提出的问题是面试题目的代表性样题，而不是全集。死记硬背本书答案将完全不能奏效。也许本书中出现的一些题目会出现在面试中，但请不要坚持这么想。大量题目将不断变化地出现，所有看过这本书的聪明面试官都不会再使用本书出现的任何问题。另一方面，面试问题包含较少的主题领域和问题类型，而这些问题很少发生变化。如果你不只是学习并解决本书提出的具体问题，而是能解答本书所列的题型，你就能处理面试过程中被询问的问题。

本书将分几个步骤来提高你的解题能力，帮助你靠近目标。首先，在提出有关这些话题的题目之前，本书会酌情回顾重要概念。其次，本书不仅仅是回答问题，而是从开始到方案来说明问题解决过程。我们发现，对于示例而言，大多数教科书和几乎所有的智力难题都采用了另一种方法：从一个问题入手，接着立即给出答案，然后解释为什么答案是正确的。根据我们的经验，这样做的结果是，读者可能理解特定答案及其正确的原因，但缺少关于作者是如何想出该解决方案或类似问题可能如何解决的

线索。我们希望本书逐步获得解决方案的方法能够解决对应问题，既能帮助大家了解答案，还能帮助大家掌握如何得出答案。

根据观察来学习绝不如边做边学有效。如果想充分利用本书，你将不得不自己解决书中问题。我们建议你采用以下方法：

(1) 题目阅读完毕，请放下本书并尝试制定解决方案。

(2) 如果被问题卡住，请阅读解决方案。由于我们不在一开始就将答案和盘托出，因此你在开始时不必担心我们要放弃告诉你完整的答案。

(3) 阅读书中内容以获得所需的提示，读到刚好够即可，然后放下书并继续解题。

(4) 根据需要重复以上步骤。

自己找到的解决方案越多，对本书题目解答过程的理解就越好。此外，这种方法非常类似于实际的面试体验，你必须自己解决问题，但面试官会在你遇到困难时给予提示。

编程是一个具有挑战性的技术行业。只通过一本书传授所有相关的计算机和编程知识是不可能的。因此，我们不得不对你的角色做出一些假设。我们假设你的计算机背景至少相当于攻读计算机科学学位的第一年或第二年学术水平。具体来说，我们希望了解 C 语言编程，已经具备了使用 C++ 或 Java 进行面向对象编程的一些经验，并且理解计算机体系结构和计算机科学理论的基础知识。这些实际上是一般开发工作的最低要求，因此大多数面试官也有类似的期望。如果发现自己在这些领域稍有欠缺，在开始求职和面试之前，请大家认真考虑接受更多的教育。

也许你拥有比刚才描述的最低要求更多的计算机知识和经验。如果是这样，可能有些更高级的主题对你会特别有吸引力。但是，无论你有什么经验，都不要忽视基本主题和问题。无论简历上有什么，面试官都倾向于从基础知识开始。

我们已尽一切努力确保本书中的所有信息都是正确的，所有代码都已经过编译和测试。然而，作为程序员，大家都有切身体会，程序的漏洞和错误是不可避免的。只要意识到问题的存在，我们就会发布更正。

我们相信你会发现本书对于获得期待的工作有促进作用。希望你在求职时翻看本书，会觉得本书涵盖的某些智力难题是一场愉悦的探索。如果想告诉我们对本书的反馈，分享对任何特定问题或主题的看法，或者在最近的一次面试中遇到的题目，我们很乐意收到你的来信。

去追求羡慕旁人的职位吧！

前言

编写《斩获 Offer: IT 名企面试策略与编程笔试精解》一书是为了让大家为编程面试做好准备,这样就可以证明自己是一名优秀的程序员。本书不讲授如何编程。本书展示了如何在编程面试中发挥自己的编程技巧。在阅读本书时,请记住,编程面试(在大多数情况下)不是事实回忆测试,因此本书不是面试时需要了解的各种事实的备忘录。相反,它通过示例引出成功所需的技巧和思维过程。真正掌握这些内容的最佳方法是花时间解决并理解问题。

为什么有编程面试

为什么软件公司会使用编程面试?他们希望聘请能与他人合作的优秀程序员,以成功生产出优秀的产品。然而,痛苦的经历告诉雇主,大部分编程工作的申请人根本不能编码。你可能会认为通过仔细审查简历、经验、课程作业和学位就可以筛选出这些申请人,但在实践中,这样做的效果不好。具有闪亮简历和突出的多年行业相关经验的申请者大有人在,数量多得惊人,但他们甚至连最简单的编程任务也无法完成。他们中的很多人已经掌握了足够的术语,使他们能够在有关编程和技术的对话中表现出色。雇用这种不能编码的“开发人员”可以轻松弄垮一个部门(甚至一个小公司)。

认识到传统面试在辨别不能编码的申请人方面无效,雇主采取了合乎逻辑的步骤:要求申请人在面试期间进行编码,因此编程面试诞生了。编程面试非常有效地将那些能编码的人与不能编码的人分开,这就是为什么编程面试往往属于技术面试流程的普遍组成部分。

编程面试的挑战在于雇主不仅想要筛选出不能编码的人,还希望将最优秀的程序员与那些勉强胜任的程序员区分开来。通常情况下,面试官会尝试通过提出有难度的编程挑战来衡量申请人的能力,并注意申请人解决问题的速度和准确程度。

这种方法的问题在于,由于面试固有的时间限制,可以在编程面试中测试的技能仅与实际开发工作中相关的技能部分重叠。不可避免的是,编程面试会当场评估候选人解决问题的能力,有人在一旁观察,但你手上没有任何平常可用的参考资料。没有时间编写大量代码,因此问题必须有简短的解决方案。大多数方案简短的问题都是微不足道的,所以为了避免这种情况,许多面试问题涉及不寻常的算法技巧、荒谬的限制或不常用的语言特性。因为在实际的开发工作中通常不会出现这些类型的问题,所以对于面试经历的特殊性,没有准备的优秀程序员有可能会面试不合格。

相反,对于职业环境中的开发所需的很多技能在编程面试中没有得到很好的评估(或根本没有)——包括的方面诸如团队沟通和团队工作、大型代码库的架构和管理、按

计划一致地生产可靠代码的时间管理和纪律,以及处理大型项目、识别所有组成部分以及完成项目的能力。

显然,编程面试并不能很好地衡量申请人作为未来员工的价值。但重要的是,编程面试是雇主选择雇用谁的方式,因此无论这个手段是否为理想的评估形式,我们都需要在其中表现良好。本书致力于教你如何使自己的编程技巧适应面试问题的特殊性,并提供所需的准备和练习,以便你在面试中表现卓越,获得期待的工作。

如何使用本书

准备是主导编程面试过程的关键。以下是有关如何有效使用本书准备编程面试的一般准则:

- 给自己足够的时间做好准备。尽早开始准备,最好在面试前几周甚至几个月。需要时间来理解本书提出的概念(如果没有那么多时间,试着留出一些不间断的时间来研究这些材料)。
- 练习回答问题。不要只阅读解决方案。被题目卡住时,根据答案提示推进解决问题并验证答案。尝试模拟面试经历。面试时在纸上或白板上写代码的机会很多,多练练这个!要想通过笔而非键盘来引导大脑编程,确实需要一定练习才可以。
- 确保了解基本概念。理解问题背后的概念是成功的关键。不要跳过或掩盖自己不理解的材料。本书提供了足够的解释来刷新你以前学过的主题,但如果遇到一些自己已经完全忘记或从未学过的东西,你可能需要在其他参考资料中阅读更多相关内容。
- 不必记住问题答案。面试官不太可能提出本书中的任何问题。即使他们这样做,他们也可能以任何方式改变问题。如果通过死记硬背来作答,那么答案可能无法正确。
- 不断练习。完成本书的学习后,准备工作不能停止。继续研究编程问题。这些题目很容易在互联网上找到。查找其他参考资料(特别是在专业领域方面),继续阅读。

就是现在,我们开始吧!

目 录

第 1 章 求职之前..... 1	
1.1 了解自己..... 1	
1.2 了解市场..... 3	
1.2.1 基本的市场情况..... 3	
1.2.2 关于外包..... 3	
1.3 培养顺应市场的技能..... 4	
1.4 完成任务..... 5	
1.5 准备好网上个人资料..... 6	
1.6 小结..... 7	
第 2 章 求职过程..... 9	
2.1 寻找和联系公司..... 9	
2.1.1 寻找公司..... 9	
2.1.2 获得推荐..... 10	
2.1.3 与猎头合作..... 10	
2.1.4 直接联系公司..... 11	
2.1.5 招聘会..... 11	
2.1.6 技术驱动的网站..... 12	
2.2 面试过程..... 12	
2.2.1 筛选面试..... 12	
2.2.2 现场面试..... 13	
2.2.3 着装..... 13	
2.3 招聘人员的角色..... 13	
2.4 录用和谈判..... 14	
2.4.1 应对招聘人员的施压..... 15	
2.4.2 商量薪资..... 15	
2.4.3 接受和拒绝录用..... 16	
2.5 小结..... 16	
第 3 章 电话面试..... 19	
3.1 了解电话面试..... 19	
3.1.1 软件工程师主持的电话 面试..... 19	
3.1.2 非技术人员主持的电话 面试..... 20	
3.2 如何进行电话面试..... 21	
3.3 电话面试问题样例..... 22	
3.3.1 C 中的内存分配..... 22	
3.3.2 权衡递归的利弊..... 22	
3.3.3 移动编程..... 22	
3.3.4 FizzBuzz..... 23	
3.3.5 字符串翻转..... 24	
3.3.6 删除重复项..... 24	
3.3.7 括号嵌套..... 25	
3.4 小结..... 26	
第 4 章 编程解题方法..... 27	
4.1 面试过程..... 27	
4.1.1 面试场景..... 27	
4.1.2 面试问题..... 28	
4.1.3 选用哪种编程语言..... 28	
4.1.4 互动是关键..... 29	
4.2 解决问题..... 29	
4.2.1 基本步骤..... 30	
4.2.2 被题目困住时..... 31	
4.3 分析解决方案..... 32	
4.3.1 大 O 分析法..... 32	
4.3.2 大 O 分析的原理..... 33	
4.3.3 最好情况、平均情况和 最坏情况..... 34	
4.3.4 优化和大 O 分析..... 35	
4.3.5 如何进行大 O 分析..... 35	
4.3.6 哪种算法更好..... 35	
4.3.7 内存占用分析..... 36	
4.4 小结..... 37	

第 5 章 链表	39	第 7 章 数组和字符串	93
5.1 为什么考链表	39	7.1 数组	93
5.2 各种链表	40	7.1.1 C 和 C++	94
5.2.1 单链表	40	7.1.2 Java	95
5.2.2 双向链表	42	7.1.3 C#	95
5.2.3 循环链表	42	7.1.4 JavaScript	96
5.3 基本链表操作	42	7.2 字符串	96
5.3.1 跟踪头元素	42	7.2.1 C	97
5.3.2 遍历链表	43	7.2.2 C++	97
5.3.3 插入和删除元素	44	7.2.3 Java	98
5.4 有关链表的面试问题	46	7.2.4 C#	99
5.4.1 栈实现	46	7.2.5 JavaScript	99
5.4.2 维护链表尾指针	51	7.3 有关数组和字符串的面试问题	99
5.4.3 removeHead 中的错误	57	7.3.1 找到第一个不重复的字符	99
5.4.4 链表的倒数第 m 个元素	58	7.3.2 删除指定的字符	102
5.4.5 链表展平	61	7.3.3 反转单词	105
5.4.6 链表还原	64	7.3.4 整数/字符串转换	108
5.4.7 空或循环	66	7.3.5 UTF-8 字符串验证	113
5.5 小结	69	7.4 小结	116
第 6 章 树和图	71	第 8 章 递归	117
6.1 树	71	8.1 理解递归	117
6.1.1 二叉树	73	8.2 与递归有关的面试问题	120
6.1.2 二叉搜索树	74	8.2.1 二分搜索	120
6.1.3 堆	76	8.2.2 字符串的排列	122
6.1.4 常见搜索	76	8.2.3 字符串的组合	125
6.1.5 遍历	77	8.2.4 电话单词	127
6.2 图	77	8.3 小结	132
6.3 与树和图有关的面试问题	78	第 9 章 排序	133
6.3.1 树的高度	78	9.1 排序算法	133
6.3.2 先序遍历	79	9.1.1 选择排序	134
6.3.3 先序遍历(不使用递归)	80	9.1.2 插入排序	135
6.3.4 最近共同祖先	82	9.1.3 快速排序	136
6.3.5 从二叉树到堆	83	9.1.4 归并排序	137
6.3.6 不平衡二叉搜索树	86	9.2 与排序有关的面试问题	139
6.3.7 Kevin Bacon 的六度	87		
6.4 小结	91		

9.2.1 最佳排序算法	139	12.1.2 面试中的设计模式	178
9.2.2 稳定的选择排序	142	12.2 常见的设计模式	178
9.2.3 多键排序	145	12.2.1 单例模式	178
9.2.4 使排序稳定	146	12.2.2 构建器模式	179
9.2.5 经过优化的快速排序	147	12.2.3 迭代器模式	180
9.2.6 煎饼排序	150	12.2.4 观察者模式	180
9.3 小结	152	12.2.5 装饰器模式	181
第 10 章 并发	153	12.3 关于设计模式的面试 问题	181
10.1 基本线程的概念	153	12.3.1 实现单例模式	181
10.1.1 线程	153	12.3.2 装饰器模式与继承	184
10.1.2 系统线程与用户 线程	154	12.3.3 高效的观察者更新	185
10.1.3 监视器和信号量	154	12.4 小结	185
10.1.4 死锁	155	第 13 章 数据库	187
10.1.5 线程示例	155	13.1 数据库基础知识	187
10.2 与并发有关的面试问题	158	13.1.1 关系数据库	187
10.2.1 忙等	158	13.1.2 SQL	188
10.2.2 生产者/消费者	160	13.1.3 NoSQL	192
10.2.3 哲学家进餐	162	13.1.4 对象数据库	192
10.3 小结	166	13.1.5 混合键-值/列数据库	192
第 11 章 面向对象编程	167	13.1.6 数据库事务	193
11.1 基础知识	167	13.1.7 分布式数据库	194
11.1.1 类与对象	167	13.2 有关数据库的面试问题	195
11.1.2 构造与析构	168	13.2.1 简单的 SQL	195
11.1.3 继承与多态	169	13.2.2 公司和员工数据库	195
11.2 有关面向对象的编程 问题	170	13.2.3 不采用聚合求最 大值	198
11.2.1 接口和抽象类	170	13.2.4 三值逻辑	199
11.2.2 虚方法	172	13.2.5 课程学习模式	200
11.2.3 多重继承	173	13.3 小结	204
11.2.4 资源管理	174	第 14 章 图形和位操作	205
11.3 小结	176	14.1 图形	205
第 12 章 设计模式	177	14.2 位操作	206
12.1 设计模式是什么	177	14.2.1 二进制补码表示法	206
12.1.1 为何使用设计模式	177	14.2.2 位运算符	207
		14.2.3 通过移位实现优化	208

14.3	与图形有关的面试问题	208
14.3.1	八分之一圆	208
14.3.2	矩形重叠	210
14.4	与位操作有关的面试问题	213
14.4.1	高位优先或低位优先	213
14.4.2	1 的个数	215
14.5	小结	218
第 15 章	数据科学、随机数和统计学	219
15.1	概率和统计	220
15.1.1	描述性统计和推断性统计	221
15.1.2	置信区间	221
15.1.3	统计检验	222
15.2	人工智能与机器学习	223
15.3	随机数生成器	224
15.4	与数据科学、随机数和统计学有关的面试问题	225
15.4.1	不可重复的结果	226
15.4.2	学得越多, 懂得越少	227
15.4.3	掷骰子	229
15.4.4	计算 π	232
15.5	小结	236
第 16 章	计数、测量和排序难题	237
16.1	解决脑筋急转弯问题	237
16.1.1	注意假设	238
16.1.2	不要被吓倒	239
16.1.3	小心简单的问题	239
16.1.4	估算问题	240
16.2	脑筋急转弯问题	240
16.2.1	数一数打开的储物柜	240

16.2.2	三个开关	242
16.2.3	过桥	243
16.2.4	重弹珠	246
16.2.5	美国加油站数量	250
16.3	小结	251
第 17 章	图形和空间难题	253
17.1	先画出来	253
17.2	有关图形和空间的问题	254
17.2.1	船和码头	254
17.2.2	数立方体	256
17.2.3	狐狸和鸭子	259
17.2.4	燃烧保险丝	261
17.2.5	躲开火车	262
17.3	小结	263
第 18 章	知识方面的问题	265
18.1	相关准备	265
18.2	面试问题举例	266
18.2.1	C++ 与 Java	266
18.2.2	友元类	267
18.2.3	参数传递	268
18.2.4	宏和内联函数	269
18.2.5	继承	270
18.2.6	垃圾收集	271
18.2.7	32 位应用与 64 位应用	272
18.2.8	网络性能	272
18.2.9	Web 应用安全	273
18.2.10	密码技术	275
18.2.11	哈希表与二叉搜索树	275
18.2.12	MapReduce	276
18.3	小结	276
第 19 章	非技术问题	277
19.1	为何要出非技术问题	277
19.2	面试问题举例	278

19.2.1 你想做什么工作.....	278	附录 A 简历	285
19.2.2 你最喜欢的编程语言 是什么.....	279	A.1 技术简历	285
19.2.3 你的工作风格是什么	279	A.1.1 糟糕的示例	285
19.2.4 你能告诉我你有什么 经验吗.....	280	A.1.2 推销自己	288
19.2.5 你的职业目标是什么	280	A.1.3 保持简短	288
19.2.6 你为什么要换工作	280	A.1.4 列出正确的信息.....	289
19.2.7 你希望的薪水是多少.....	281	A.1.5 简明清晰	290
19.2.8 你过去的薪水是多少.....	283	A.1.6 仅限相关信息	291
19.2.9 我们为什么应该录 用你	284	A.1.7 使用反向时间顺序 排序	292
19.2.10 你为什么想为这 家公司工作.....	284	A.1.8 保持校对的习惯.....	293
19.2.11 有什么问题要问 我吗.....	284	A.1.9 改进后的示例	293
19.3 小结	284	A.1.10 经理和高级开发 人员	294
		A.1.11 针对职位裁剪简历.....	299
		A.2 简历样本	299