

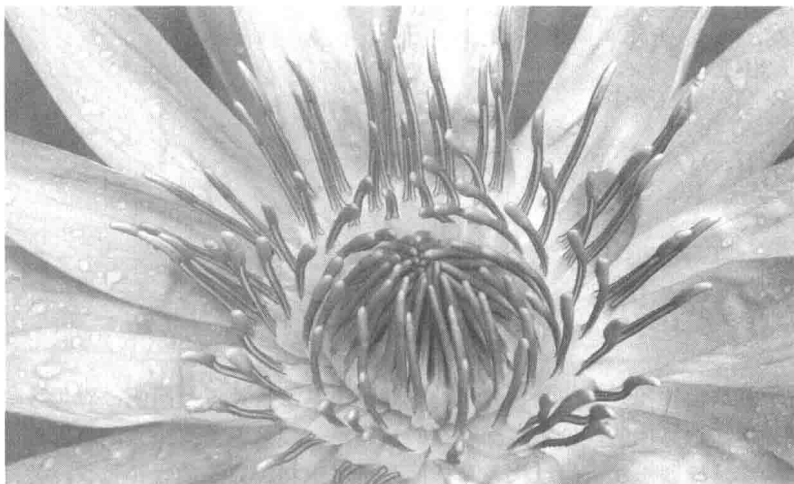


算法设计

(英文版)

[美] 乔恩·克莱因伯格 (Jon Kleinberg) 伊娃·塔多斯 (Éva Tardos) 著

Algorithm Design



算法设计

英文版

[美] 乔恩·克莱因伯格 (Jon Kleinberg) 伊娃·塔多斯 (Éva Tardos) 著

Algorithm Design

人民邮电出版社

北京

图书在版编目 (C I P) 数据

算法设计：英文版 = Algorithm Design / (美) 乔恩·克莱因伯格, (美) 伊娃·塔多斯著. — 北京：人民邮电出版社, 2019.5
ISBN 978-7-115-49592-1

I. ①算… II. ①乔… ②伊… III. ①算法设计—高等学校—教材—英文 IV. ①TP301.6

中国版本图书馆CIP数据核字(2018)第231755号

内 容 提 要

这是一本关于算法设计和分析的经典教材。本书围绕算法设计进行组织，对每种算法技术选择了多个典型范例进行分析，把算法的理论跟实际存在的问题结合起来，具有很大的启发性。本书侧重算法设计思路，每章都从实际问题出发，经过深入具体的分析引出相应算法的设计思想，并对算法的正确性和复杂性进行合理的分析和论证。本书覆盖面很宽，且含有 200 多道精彩的习题，最后还扩展了 PSPACE 问题、参数复杂性等内容。

本书适合作为计算机及相近专业本科高年级学生以及研究生算法课程的参考教材，也适合作为对信息学奥林匹克竞赛感兴趣的高中生的指导书籍。对算法分析和设计感兴趣的 IT 专业技术人员也可以将本书作为案头必备的参考书或工程实践手册。

-
- ◆ 著 [美] 乔恩·克莱因伯格 (Jon Kleinberg)
[美] 伊娃·塔多斯 (Eva Tardos)
 - 责任编辑 杨海玲
 - 责任印制 焦志炜
 - ◆ 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>
北京鑫正大印刷有限公司印刷
 - ◆ 开本：800×1000 1/16
印张：52.25
字数：1 239 千字 2019 年 5 月第 1 版
印数：1—2 000 册 2019 年 5 月北京第 1 次印刷
- 著作权合同登记号 图字：01-2017-9237 号
-

定价：138.00 元

读者服务热线：(010)81055410 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

Authorized Reprint from the English language edition, entitled ALGORITHM DESIGN, 1st Edition by KLEINBERG, JON; TARDOS, ÉVA, published by Pearson Education, Inc, Copyright © 2006 Pearson Education Inc.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from Pearson Education, Inc.

ENGLISH language edition published by POSTS & TELECOM PRESS, Copyright © 2019.

本书英文版由 Pearson Education Inc 授权人民邮电出版社独家出版。未经出版者书面许可，不得以任何方式复制本书内容。

本书封面贴有 Pearson Education（培生教育）出版集团激光防伪标签，无标签者不得销售。

版权所有，侵权必究。

前言

算法思想无处不在，在计算机科学和其他领域中的体现都很明显。因特网路由标准的一些主要变化，可以看成是人们对一种最短路径算法的不足和另一种算法的相对优势的争论。生物学家用于表示基因和基因组之间相似性的基本概念是用算法定义的。经济学家对组合拍卖在实践中的可行性所表达的担忧，部分原因是在一些特殊情况下，这些拍卖包含计算难解的搜索问题。而且，算法概念不仅限于众所周知的、长期存在的问题，在很多领域中出现的新问题中，人们也经常看到这些思想的身影。有一天，雅虎的一位科学家在午餐时告诉我们，他们向用户提供广告的系统可以归结为一系列问题，这些问题在很大程度上可以用网络流问题来建模。我们去纽约时碰巧遇到以前的一名学生，现在是管理顾问，负责为大型医院提供人力服务，也讲述了类似的事情。

关键不只是算法有很多应用。在更深的层面上，算法这一主题是一个强大的“镜头”，透过它可以查看计算机科学领域。算法问题构成了计算机科学的核心，但它们很少以整洁、精确的数学问题的形式出现。相反，它们往往有许多杂乱的、应用程序特定的细节，由一些至关重要和一些无关紧要的东西纠缠在一起。因此，算法工作由两个基本部分组成：得到数学上整洁的问题核心，然后根据问题结构确定适当的算法设计技术。这两个部分相互影响：越能自如地运用各种可能的设计技术，也就越开始认识到世上混乱问题中的整洁形式描述。在最有效的情况下，算法思想不仅能提供适当问题的解决方案，它们还构成了一种语言，让你可以清晰地表达基本问题。

本书的目标是将这种方法带入算法研究，作为一个设计过程，它始于各种计算应用程序中出现的问题，构建在对算法设计技术的理解之上，最终得到这些问题的有效解决方案。我们试图探讨算法思想在计算机科学中的作用，并将

这些思想与一些精确制定的问题联系起来，我们可以为它们设计算法并进行分析。换言之，导致这些问题的根本问题是什么？如何选择这些特定的方式来描述它们？如何认识到不同情况下适用哪些设计原则？

为此，我们的目标是，建议如何在不同计算领域的复杂问题中识别算法问题的清晰形式描述，并针对由此产生的问题，建议如何设计有效的算法。通过重新组织思路（包括错误的起点和死胡同），从最简单的初始方法到最终的解决方案，通常可以更好地理解复杂的算法。这导致了一种阐述风格，不是描述从问题陈述到算法的最直接路径，但我们认为它更好地反映了我们和同事真正思考这些问题的方式。

概述

本书适用于这样的学生：他们应该已经学完了基于编程的两学期计算机科学引导课程（标准的“CS1/CS2”课程），已编写过实现基本算法的程序，操作过离散结构（树和图），并应用过基本的数据结构（如数组、列表、队列和堆栈）。因为 CS1/CS2 与第一门算法课程之间的衔接并不是完全标准的，所以我们在本书开始介绍了一些自包含的主题，某些院校中学过 CS1/CS2 的学生会很熟悉这些内容，但其他院校可能将这些内容放在第一门算法课程的教学大纲中。因此，这些内容可以用于复习也可以当作新内容处理。我们加入这些内容，是希望本书适用于更广泛的课程，并在假设的预备知识方面更灵活。

按照上述思路，我们利用了许多计算机科学和相关领域的问题，来开发基本的算法设计技术。这里列出一些有代表性的例子，我们相当详细地讨论了来自下列领域的应用，包括系统和网络的应用程序（缓存、交换、互联网上的域间路由），人工智能（规划、博弈、霍普菲尔德网络），计算机视觉（图像分割），数据挖掘（变更点检测、聚类），运筹学（航线调度），以及计算生物学（序列比对、RNA 二级结构）。

计算难解性的概念，特别是 NP 完全性，在本书中起着重要作用。这与我们如何考虑算法设计的整个过程是一致的。有时候，在应用领域出现的有趣问题存在有效的解决方案，而有时候它可以被证明是 NP 完全的。为了全面考察新的算法问题，人们应该能够同样熟练地探索这两个方面。由于计算机科学中的许多自然问题都是 NP 完全的，因此开发处理难解问题的方法已成为算法研究中的一个关键问题。本书以很大篇幅反映了这一主题。不应将发现问题是 NP 完全的看作故事的结束，而应该将其看作是对我们开始寻找近似算法、启发式局部搜索技术或易解的特殊情况的鼓励。我们对这 3 种方法都进行了广泛的探讨。

问题和带解答的练习

本书的一个重要特征是问题集。本书各章共包含 200 多个问题，这是我们在康奈尔大学教学课程的一部分，几乎所有问题都在课外作业中被开发，或者在课堂测验进行了考试。我们将问题视为本书的一个重要组成部分，并且让它们的结构与我们对内容的整体方法保持一致。其中大部分内容包含了一些问题的详细文字描述，这些问题出现在计算机科学应用领域或世界其他地方。部分问题是实践我们在教材中讨论的问题：建立必要的符号和形式化，设计算法，然后分析这个算法并证明它是正确的。（我们认为这些问题的完整答案应该包括所有这些部分：带完整解释的算法、运行时间的分析和正确性的证明。）这些问题的想法很大程度上来自我们多年来与人们的讨论，他们在不同的领域工作。而且，在某些情况下，这些问题也记录了一个有趣的（虽然是容易的）算法的应用，我们没有在其他任何地方看到过。

为了帮助解决这些问题，我们在每章中都加入了一节，名为“带解答的练习”，讨论一个或多个问题，并描述了如何形式化一个解。因此，专门针对每个带解答的练习的讨论，要比编写完整、正确的解决方案所需的时间长得多（换言之，如果将这些解决方案指定为课外作业题，所用的时间明显要比获得完全学分所需的时间长）。实际上，与本教材的其余部分一样，这些节中的讨论应该看成是试图让人们了解一个更大的过程，通过这个过程可以考虑这种类型的问题，并最终形成精确解的详细说明。

关于在课程中使用这些问题作为作业，有两点值得一提。首先，问题大致按照难度增加的顺序排列，但这只是一个粗略的指导，建议不要过分看重它：因为大部分问题是专为本科生班的作业设计的，每章中的大部分问题在难度上实际上都差不多。其次，除编号最小的几个问题之外，其他问题都需要投入一些时间，既要问题描述与这一章中的算法技术联系起来，又要实际设计必要的算法。在我们的本科课程中，每周大约布置 3 道这样的问题。

教学特色和补充材料

除了问题和带解答的练习，本书还有一些其他教学特色和补充材料，以方便教学。

如前所述，本书中大量篇幅专门用于算法问题的形式描述（包括其背景和潜在动机），以及针对该问题的算法设计和分析。为了反映这种风格，这些部分始终围绕一系列小节进行组织：“问题”，描述问题并确定精确的形式定义；“设

“计算法”，采用适当的设计技术开发算法；“分析算法”，证明算法的性质并分析它的效率。这些节在正文中用羽毛图标突出显示。在包含问题扩展或进一步分析算法时，还有其他专门针对这些问题的节。使用这种结构是为了提供一种相对统一的表述风格，从最初讨论计算应用中出现的问题，直到对这些问题的解决方法的详细分析。

本书提供了许多补充材料作为教学的支持。配套的教师手册完成了书中所有练习，为每个问题提供完整的解决方案，还可以提供由普林斯顿大学的 Kevin Wayne 开发的一套教学用 PPT，它是按照本书章节的顺序组织的，可以作为以本书为教材的课堂教学的材料。这些文件可在培生教育出版集团的官方网站 (www.pearsoned.com) 上获得。

最后，我们很希望收到有关本书的反馈。特别是，就像任何这样篇幅的书一样，毫无疑问，本书最终版本中仍然会有错误。有关错误的建议和报告可通过电子邮件发送给我们，地址为 algbook@cs.cornell.edu，请在邮件的主题中注明“feedback”。

各章概要

第 1 章首先介绍一些有代表性的算法问题。一开始就是稳定匹配问题 (stable matching problem)，因为我们认为它比任何抽象的讨论更具体，也更优雅地提出了算法设计中的基本问题：稳定匹配是从自然但复杂的现实世界问题中提出的，从中可以抽象出一个有趣的问题陈述，以及一个令人惊讶的有效算法来解决这个问题。第 1 章的其余部分讨论 5 个“有代表性的问题”，这些问题预先展示了课程其余部分的主题。这 5 个问题是相互关联的，因为它们都是独立集问题 (independent set problem) 的变异或特殊情况。但一个可以通过贪心算法解决，一个可以通过动态规划解决，一个可以通过网络流解决，一个 (独立集问题本身) 是 NP 完全的，还有一个是 PSPACE 完全的。密切相关的问题在复杂性方面可能有很大差异，这一事实是本书的一个重要主题，这 5 个问题作为里程碑，随着本书的推进而再次出现。

第 2 章和第 3 章包含前面提到的 CS1/CS2 课程系列的衔接。第 2 章介绍用于分析算法的关键数学定义和符号，以及它们背后的发人深省的原理。这一章首先非正式地概述计算易解问题的含义，并用多项式时间的概念作为效率的正式概念。然后，更正式地讨论函数的增长率和渐近分析，并对算法分析中常见函数及其标准应用提供指导。第 3 章介绍使用图所需的基本定义和算法原语，这些是本书中许多问题的核心。在 CS1/CS2 课程系列的后期，学生通常会实现许多基本图算法，但在更广泛的算法设计环境中提供这些材料是很有价值的。

特别是，我们讨论了图的基本定义、图的遍历技术（如宽度优先搜索和深度优先搜索）以及有向图概念（包括强连通性和拓扑排序）。

第 2 章和第 3 章还介绍许多基本数据结构，它们将用于实现整本书中的算法。更高级的数据结构将在后续章中介绍。关于数据结构，我们会在实现本书中开发的算法需要时引入它们。因此，虽然这里介绍的许多数据结构对学过 CS1/CS2 系列的学生来说是很熟悉的，但我们的重点是在更广泛的算法设计和分析环境中使用这些数据结构。

第 4 章到第 7 章介绍 4 种主要的算法设计技术：贪心算法、分治法、动态规划和网络流。使用贪心算法的挑战在于判断它们何时好用，何时不好用。对这个主题的介绍，我们围绕对证明贪心算法正确的各种论据进行分类的方式来展开。这一章最后介绍贪心算法的一些主要应用，包括最短路径、无向生成树和有向生成树、聚类 and 压缩。对于分治法，我们首先讨论一些策略，将递推关系求解为运行时间的界限。然后，我们展示对这些递推的熟悉程度可以如何指导算法的设计，这些算法改进了对许多基本问题的直接方法，包括排名比较，平面上最近点对的计算，以及快速傅里叶变换。接下来，我们开始动态规划，从隐藏在它背后的递归直觉开始，然后通过它们自然产生的应用，构建越来越多有表现力的递推形式描述。这一章最后对两个基本问题的动态规划方法进行扩展讨论：序列比对及其在计算生物学中的应用；图中的最短路径及其与因特网路由协议的联系。最后，我们介绍网络流问题的算法，我们将这一章的重点放在讨论网络流的大量不同应用上。在算法课程中介绍网络流时，学生往往无法欣赏到它可以应用的广泛问题，我们试图通过介绍在负载均衡、调度、图像分割和许多其他问题中的应用，来公正地展示它的多种能力。

第 8 章和第 9 章介绍计算难解性。我们将大部分注意力集中在 NP 完全性上，按主题方式组织基本的 NP 完全问题，帮助学生在遇到新问题时识别用于归约的候选项。我们给出一些相当复杂的 NP 完全性证明，并指导学生如何构建有难度的归约。我们还考虑 NP 完全性之外的计算难度类型，特别是 PSPACE 完全性。我们发现这样做是有价值的，即强调难以解决的问题并非以 NP 完全性为终点，而 PSPACE 完全性也构成了人工智能的一些核心概念的基础（规划和博弈游戏），否则无法在我们考察的算法世界中找到它们的位置。

第 10 章到第 12 章介绍处理计算难解问题的 3 种主要技术，即识别结构上的特殊情况、近似算法和局部搜索启发式算法。关于易解特殊情况的一章强调，在实践中出现的 NP 完全问题的实例可能不像最坏情况那么难，因为它们通常包含一些结构，可以在设计有效算法时加以利用。我们将说明，当 NP 完全问题仅限于

树结构输入时，它们如何经常有效地解决。我们还对图的树分解进行扩展讨论。虽然这个主题更适合研究生课程而不是本科生课程，但这是一种相当实用的技术，很难为学生找到现成可用的参考资料。关于近似算法的一章，既讨论设计有效算法的过程，又讨论充分理解最优解，以明白算法的界限。作为近似算法的设计技术，我们专注于贪心算法、线性规划以及称为“定价法”的第三种方法，第三种方法结合前两种方法的思想。最后，我们讨论局部搜索启发式算法，包括 Metropolis 算法和模拟退火算法。本科算法课程中经常略过这个主题，因为对这些算法的可证性保证知之甚少。然而，鉴于它们在实践中的广泛应用，我们认为学生了解这些算法是有价值的，我们还包括了一些有可证性保证的案例。

第 13 章介绍随机化在算法设计中的应用。关于这个主题，已经有几本很好的研究生水平的书。我们的目标是提供一个更紧凑的介绍，让学生能够利用一般从本科离散数学课程中获得的概率知识来运用随机技术。

本书的用法

本书主要用于本科生的第一门算法课程，但它也可以作为研究生导论性课程的基础。

在本科阶段使用本书时，我们一节课大约讲一节。如果一节课讲不完一节的内容（例如，如果该节提供了进一步的应用作为附加示例），我们会将这些额外的材料作为学生可以在课外阅读的补充材料。我们跳过加星号的节。虽然这些节包含重要的主题，但它们对于课程的展开并不那么重要，而且有时它们也比较难。在本书的前半部分，我们也倾向于跳过每章的一两节（例如，我们倾向于跳过 4.3 节、4.7 节、4.8 节、5.5 节、5.6 节、6.5 节、7.6 节和 7.11 节）。第 11 章至第 13 章，我们大约每章只讲一半。

值得强调的最后一点是：不要将后面的章节看成是“高级章节”，从而认为它超出了本科算法课程的范围。我们设计这些章节的目标，是让每章的前几节都应该适合本科生。我们自己的本科生课程包括所有这些章的材料，因为我们认为所有这些主题在本科阶段都占有重要地位。

最后，我们主要将第 2 章和第 3 章作为对早期课程材料的回顾。但是，如前所述，这两章的使用在很大程度上取决于每门特定课程与其预备知识的关系。

由此产生的教学大纲大致如下：第 1 章，第 4 章至第 8 章（不包括 4.3 节、4.7 节、4.8 节、4.9 节、5.5 节、5.6 节、6.5 节、6.10 节、7.4 节、7.6 节、7.11 节和 7.13 节），第 9 章（简述），第 10 章的 10.1 节和 10.2 节，第 11 章的 11.1 节、11.2 节、11.6 节和 11.8 节，第 12 章的 12.1 节至 12.3 节，以及第 13 章的 13.1 节至 13.5 节。

本书自然也支持导论性的研究生算法课程。我们认为，对于有志投身于各种不同领域研究的学生，这种课程应该介绍算法设计中当前重要的主题。在这里，我们发现强调问题的形式描述也很有用，因为学生很快就会尝试在许多不同的子领域中定义他们自己的研究问题。对于这种类型的课程，我们包含第 4 章和第 6 章中后面的主题（4.5 节至 4.9 节和 6.5 节至 6.10 节），以及第 7 章的所有内容（前面几节讲得更快），还快速介绍第 8 章中的 NP 完全性（因为许多新研究生在本科学过），然后将剩下的时间花在第 10 章至第 13 章。虽然在研究生的导论性课程中，我们的重点是更高级的部分，但我们发现，对学生来说，有一本完整的书籍用于复习或补充背景知识是很有用的。因为在这门课的学生的本科背景有所不同。

最后，本书支持研究生、研究人员或计算机专业人员自学，他们希望了解如何能够在自己的工作环境中使用特定的算法设计技术。一些研究生和同事已经以这种方式使用了本书的部分内容。

致谢

本书源于我们在康奈尔大学教授的一系列算法课程。多年来，这些课程已经随着这个领域的发展而发展，反映了康奈尔大学教师对其的影响，包括 Juris Hartmanis、Monika Henzinger、John Hopcroft、Dexter Kozen、Ronitt Rubinfeld 和 Sam Toueg。我们要更广泛地感谢康奈尔大学的所有同事，感谢他们对本书中的材料以及有关该领域本质的更广泛问题进行了无数次讨论。

在教授这门课程时，所有的课程人员对这些材料的形成都提供了极大的帮助。感谢我们的本科生和研究生助教 Siddharth Alexander、Rie Ando、Elliot Anshelevich、Lars Backstrom、Steve Baker、Ralph Benzing、John Bicket、Doug Burdick、Mike Connor、Vladimir Dzhohor、Shaddin Doghmi、Alexan der Druyan、Bowe Du、Sasha Evfimievski、Ariful Gani、Vadim Grinshpun、Ara Hayrapetyan、Chris Jeuell、Igor Kats、Omar Khan、Mikhail Kobayakov、Alexei Kopylov、Brian Kulis、Amit Kumar、Yeongwee Lee、Henry Lin、Ash-win Machanavajjhala、Ayan Mandal、Bill McCloskey、Leonid Meyerguz、Evan Moran、Niranjana Nagarajan、Tina Nolte、Travis Ortogero、Martin Pa'l、Jon Peress、Matt Piotrowski、Joe Polastre、Mike Priscott、Xin Qi、Venu Ramasubramanian、Aditya Rao、David Richardson、Brian Sabino、Rachit Siamwalla、Sebastian Silgado、Alex Slivkins、Chaitanya Swamy、Perry Tam、Nadya Travinin、Sergei Vassilvitskii、Matthew Wachs、Tom Wexler、Shan-Leung Maverick Woo、Justin Yang 和 Misha Zatsman。他们中的许

多人都对本书提供了宝贵的见解、建议和评论。我们还要感谢上这些课的所有学生，他们多年来对本书的早期书稿提出了意见和反馈。

在过去几年中，本书的发展得益于使用本书出版前的初稿进行教学的同行的反馈和建议。Anna Karlin 大胆地在华盛顿大学采用本书的初稿作为她的课程教材，当时本书还处于早期阶段。随后有许多人将它用作课程教材或作为教学资源，他们是 Paul Beame、Allan Borodin、Devdatt Dubhashi、David Kempe、Gene Kleinberg、Dexter Kozen、Amit Kumar、Mike Molloy、Yuval Rabani、Tim Roughgarden、Alexa Sharp、Shanghua Teng、Aravind Srinivasan、Dieter van Melkebeek、Kevin Wayne、Tom Wexler 和 Sue Whitesides。我们非常感谢他们的投入和建议，我们对内容的许多修订就来源于此。我们要再一次感谢 Kevin Wayne，他制作了与本书相关的补充材料，这让后来的教师使用本书更为方便。

在其他许多时候，我们对书中特定主题的处理方法反映了特定同事的影响。其中的许多贡献我们无疑会有所遗漏，但我们特别感谢 Yuri Boykov、Ron Elber、Dan Huttenlocher、Bobby Kleinberg、Evie Kleinberg、Lillian Lee、David McAllester、Mark Newman、Prabhakar Raghavan、Bart Selman、David Shmoys、Steve Strogatz、Olga Veksler、Duncan Watts 和 Ramin Zabih。

在过去许多年里，与 Addison Wesley 出版社合作非常愉快。我们感谢 Matt Goldstein 在此过程中提供的所有建议和指导，并帮助我们将大量的评阅材料综合成改进本书的具体计划。最早我们与 Susan Hartman 关于写作本书的交流也非常有价值。我们感谢 Matt 和 Susan，以及 Addison Wesley 出版社的 Michelle Brown、Marilyn Lloyd、Patty Mahtani 和 Maite Suarez-Rivas，感谢 Windfall 软件公司的 Paul Anagnostopoulos 和 Jacqui Scarlott 所做的所有关于编辑、制作和整个项目管理方面的工作。我们要再一次感谢 Paul 和 Jacqui 的专业排版。我们感谢 Joyce Wells 为本书设计封面，感谢达特茅斯出版公司的 Nancy Murphy 对图表的处理，感谢 Ted Laux 为本书制作索引，感谢 Carol Leyba 和 Jennifer McClain 所做的文字编辑和校对工作。

我们感谢 Anselm Blumer（塔夫茨大学）、Richard Chang（马里兰大学巴尔的摩分校）、Kevin Compton（密歇根大学）、Diane Cook（德克萨斯大学阿灵顿分校）、Sariel Har-Peled（伊利诺伊大学厄巴纳香槟分校）、Sanjeev Khanna（宾夕法尼亚大学）、Philip Klein（布朗大学）、David Matthias（俄亥俄州立大学）、Adam Meyerson（加州大学洛杉矶分校）、Michael Mitzenmacher（哈佛大学）、Stephan Olariu（欧道明大学）、Mohan Paturi（加州大学圣地亚哥分校）、

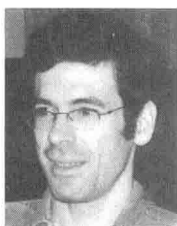
Edgar Ramos (伊利诺伊大学厄巴纳-香槟分校)、Sanjay Ranka (佛罗里达大学, 盖恩斯维尔)、Leon Reznik (罗彻斯特理工学院)、Subhash Suri (加州大学圣巴巴拉分校)、Dieter van Melkebeek (威斯康星大学麦迪逊分校) 和 Bulent Yener (伦斯勒理工学院) 慷慨地付出了他们的时间, 针对本书的书稿给出了详细而深入的评阅意见, 这些意见为本书的最终版本带来了无数大大小小的改进。

最后, 感谢我们的家人——Lillian 和 Alice, 以及 David、Rebecca 和 Amy。我们感谢他们的支持、耐心和许多其他贡献, 在此处的致谢中难以用言语表达。

本书是在 20 世纪 90 年代后期的非理性繁荣时期开始写的, 对我们许多人来说, 当时计算技术的光芒似乎短暂地划过天空, 这里传统上是由流行文化领域的名人和其他各色人物占据的 (这可能只是我们想象的)。现在, 在几年的炒作和股票价格回归现实之后, 人们能够意识到, 在某些方面, 计算机科学已经被这个时期永远地改变了, 而在其他方面, 它依然没变: 从早期开始, 这个领域就激动人心, 现在一样强烈而诱人, 公众对信息技术的迷恋仍然充满活力, 计算的范围继续扩展到新的学科。因此, 对于所有因各种原因受此主题吸引的学生, 我们希望无论你们的计算追求把你们带到什么地方, 你们都发现本书是令人愉快的、有用的指南。

Jon Kleinberg
Éva Tardos
2005 年于伊萨卡

作者简介



乔恩·克莱因伯格 (Jon Kleinberg) 是康奈尔大学计算机科学教授。他于 1996 年从麻省理工学院获得博士学位。他荣获过美国国家科学基金会事业奖 (NSF Career Award)、海军研究局青年研究员奖 (ONR Young Investigator Award)、IBM 杰出创新奖 (IBM Outstanding Innovation Award)、美国国家科学院创新研究奖 (National Academy of Sciences Award for Initiatives in Research)，还获得过帕卡德基金会 (Packard Foundations) 和斯隆基金会 (Sloan Foundations) 的研究基金以及康奈尔工程学院与计算机科学系教学奖。

克莱因伯格的研究集中在算法上，特别是与网络结构和信息相关的算法，以及这些算法在信息科学、优化、数据挖掘以及计算生物学等方面的应用。他利用信息中心和权威信息进行网络分析的工作，对形成最新一代因特网搜索引擎的基础起了很大的作用。



伊娃·塔多斯 (Éva Tardos) 是康奈尔大学计算机科学教授。她于 1984 年从匈牙利布达佩斯的厄特沃什大学获得博士学位。她是美国艺术与科学学院院士、ACM 会士，她荣获过美国国家科学基金会总统青年研究员奖 (NSF Presidential Young Investigator Award)、富尔克森奖 (Fulkerson Prize)，获得过古根海姆基金会 (Guggenheim Foundations)、帕卡德基金会 (Packard Foundations) 和斯隆基金会 (Sloan Foundations) 的研究基金以及康奈尔工程学院和计算机科学系教学奖。

塔多斯的研究兴趣主要集中在图和网络问题的算法设计和分析上。她因在网络流算法和网络问题的近似算法方面的工作而闻名。她最近的工作重点是算法博弈论，这是一个新兴领域，关注为自私用户设计系统和算法。

目录

1	Introduction: Some Representative Problems / 引言：一些典型问题	1
1.1	A First Problem: Stable Matching / 第一个问题：稳定匹配	1
1.2	Five Representative Problems / 五个典型问题	12
	Solved Exercises / 带解答的练习	19
	Exercises / 练习	22
	Notes and Further Reading / 注释和进一步阅读	28
2	Basics of Algorithm Analysis / 算法分析基础	29
2.1	Computational Tractability / 计算可解性	29
2.2	Asymptotic Order of Growth / 增长的渐近阶	35
2.3	Implementing the Stable Matching Algorithm Using Lists and Arrays / 用列表和数组实现稳定匹配算法	42
2.4	A Survey of Common Running Times / 常见运行时间综述	47
2.5	A More Complex Data Structure: Priority Queues / 更复杂的数据结构：优先队列	57
	Solved Exercises / 带解答的练习	65
	Exercises / 练习	67
	Notes and Further Reading / 注释和进一步阅读	70
3	Graphs / 图	73
3.1	Basic Definitions and Applications / 基本定义与应用	73
3.2	Graph Connectivity and Graph Traversal / 图连通性与图遍历	78

3.3	Implementing Graph Traversal Using Queues and Stacks / 用优先队列与栈实现图遍历	87
3.4	Testing Bipartiteness: An Application of Breadth-First Search / 二分性测试：广度优先搜索的应用	94
3.5	Connectivity in Directed Graphs / 有向图中的连通性	97
3.6	Directed Acyclic Graphs and Topological Ordering / 有向无环图和拓扑排序	99
	Solved Exercises / 带解答的练习	104
	Exercises / 练习	107
	Notes and Further Reading / 注释和进一步阅读	112
4	Greedy Algorithms / 贪心算法	115
4.1	Interval Scheduling: The Greedy Algorithm Stays Ahead / 区间调度：贪心算法保持领先	116
4.2	Scheduling to Minimize Lateness: An Exchange Argument / 最小延迟的调度：交换论证	125
4.3	Optimal Caching: A More Complex Exchange Argument / 最优缓存：更复杂的交换论证	131
4.4	Shortest Paths in a Graph / 图的最短路径	137
4.5	The Minimum Spanning Tree Problem / 最小生成树问题	142
4.6	Implementing Kruskal's Algorithm: The Union-Find Data Structure / 实现 Kruskal 算法：Union-Find 数据结构	151
4.7	Clustering / 聚类	157
4.8	Huffman Codes and Data Compression / 哈夫曼码和数据压缩	161
*4.9	Minimum-Cost Arborescences: A Multi-Phase Greedy Algorithm / 最小费用有向树：多阶段贪心算法	177
	Solved Exercises / 带解答的练习	183
	Exercises / 练习	188
	Notes and Further Reading / 注释和进一步阅读	205
5	Divide and Conquer / 分治	209
5.1	A First Recurrence: The Mergesort Algorithm / 第一个递推式：归并排序算法	210

5.2	Further Recurrence Relations / 进一步的递推关系	214
5.3	Counting Inversions / 计数逆序	221
5.4	Finding the Closest Pair of Points / 寻找最近点对	225
5.5	Integer Multiplication / 整数乘法	231
5.6	Convolutions and the Fast Fourier Transform / 卷积和快速傅里叶变换	234
	Solved Exercises / 带解答的练习	242
	Exercises / 练习	246
	Notes and Further Reading / 注释和进一步阅读	249
6	Dynamic Programming / 动态规划	251
6.1	Weighted Interval Scheduling: A Recursive Procedure / 加权区间调度: 递归过程	252
6.2	Principles of Dynamic Programming: Memoization or Iteration over Subproblems / 动态规划原理: 备忘录或子问题迭代	258
6.3	Segmented Least Squares: Multi-way Choices / 分段最小二乘: 多重选择	261
6.4	Subset Sums and Knapsacks: Adding a Variable / 子集和与背包: 加一个变量	266
6.5	RNA Secondary Structure: Dynamic Programming over Intervals / RNA 二级结构: 区间上的动态规划	272
6.6	Sequence Alignment / 序列比对	278
6.7	Sequence Alignment in Linear Space via Divide and Conquer / 通过分治在线性空间中的序列比对	284
6.8	Shortest Paths in a Graph / 图中的最短路径	290
6.9	Shortest Paths and Distance Vector Protocols / 最短路径和距离向量协议	297
6.10	Negative Cycles in a Graph / 图中的负环	301
	Solved Exercises / 带解答的练习	307
	Exercises / 练习	312
	Notes and Further Reading / 注释和进一步阅读	335
7	Network Flow / 网络流	337