

算 法



Jeff Erickson

第 0 版（出版前的草稿）—— 2018 年 12 月 29 日

0 1 2 3 4 5 6 7 8 9 — 27 26 25 24 23 22 21 20 19 18

© Copyright 2019 Jeff Erickson



本书受署名 4.0 国际许可保护，该许可见：<http://creativecommons.org/licenses/by/4.0/>。

在此处可以下载本书：<http://jeffe.cs.illinois.edu/teaching/algorithms/>

或 <http://algorithms.wtf>

请在此处提交错误：<https://github.com/jeffgerickson/algorithms>

我们编程的一部分是机械复制的，现在我们开始播出。

向 *Kim, Kate, and Hannah*
表示爱与敬意

对 *Erin* 的违约
表示感谢

Incipit prologus in libro alghoarismi de practica arismetrice.

—— Ioannis Hispalensis [John of Seville?],
《Liber 算法的实用性》

要我告诉你，我的朋友，你该如何理解呢？
去写一本关于它的书。

—— Henry Home, Lord Kames (1696 - 1782),
在一封给 Gilbert Elliot 先生的信中

个人总是错的。一个人可以设计许多东西，吸引其他人一起合作，与一些或
所有人吵架，有很多失误，做一些事情：集体更为先进，但个人总是错的。
—— Ralph Waldo Emerson, "Experience", *Essays, Second Series* (1844)

我在上面概述的是一本书的内容，即实现其基本计划和纳入其细节也许是不可
能的；我什么

—— Michael Spivak, 《微分几何 第一卷》第一版 前言 (1970)

前言

关于这本书

这本教科书来源于我为伊利诺伊大学香槟分校的各种算法课程写的一系列课堂讲稿，从 1999 年 1 月起，我每年都要使用这些讲稿一次。受本科生理论课程改革的推动，我于 2016 年对我的笔记进行了重大修订；这本书包括我修订的关于最基础课程材料的笔记的一个部分，主要反映了我们对必修的初级理论课程的算法的最新要求。

先修课程

我在伊利诺伊大学教的算法课程有两个重要的先决条件：学习一门关于离散数学的课程和一门关于基本数据结构的课程。因此，本教材可能不适合大多数学生作为数据结构和算法的第一门课程。特别的，我假设大家至少熟悉以下的一些内容：

- **离散数学：**高中代数、对数恒等式、朴素集合论、布尔代数、一阶谓词逻辑、集合、函数、等价关系、序关系、模运算、递归定义、树（作为抽象对象，而不是数据结构）、图（顶点和边，而不是函数图）。
- **证明技术：**直接、间接、矛盾、详尽的案例分析和归纳（尤其是强化归纳法和结构化归纳）。第 0 章使用归纳法，当第 $n-1$ 章使用归纳法时，第 n 章也使用归纳法。
- **迭代编程概念：**变量、条件、循环、记录、间接（地址/指针/引用）、子程序、递归。我认为自己并不精通任何特定的编程语言，但我认为至少有一种语言支持接口和递归。
- **基本抽象数据类型：**标量、序列、向量、集合、堆栈、队列、映射/字典、有序映射/字典、优先级队列。
- **基本数据结构：**数组、链表（单链表、双链表与循环链表）、二叉搜索树、至少一种形式的平衡二叉搜索树（例如 AVL 树、红黑树、treap 树、跳表、伸展树）、哈希表、二叉堆，最重要的是，本条与上一条的差异。
- **基本计算问题：**基本算术、排序、搜索、枚举、树遍历（前序、中序、后序、级序等）。
- **基本算法：**基本算法、顺序搜索、折半搜索、排序（选择、插入、合并、堆、快速、基数等）、树的广度和深度优先搜索（至少是二叉树），最重要的是，本条与上一条的差异。
- **基本算法分析：**渐近符号 ($o, O, \Theta, \Omega, \omega$)，将循环转换为和，递归调用转换为递推，计算简单和与递归。
- **数学能力：**具有抽象性、形式定义（特别是递归定义）和证明（特别是归纳证明）的能力；书写和遵循数学观点；识别和避免句法、语义和/或逻辑上的无稽之谈。

这本书在上下文中简要介绍了一些先修课程的内容，但更多的是作为一个提示，而不是一个好的介绍。为了更全面的概述，我强烈推荐以下免费提供的参考资料：

- Margaret M. Fleck. 理论计算机科学的基石，一本未出版的书，最近修订于 2013 年 1 月。可以在这里获取：<http://mfleck.cs.illinois.edu/building-blocks/>.
- Eric Lehman, F. Thomson Leighton, Albert R. Meyer. 计算机科学数学，未发表的讲稿，最新的版本（出版）是 2018 年 6 月。可以在这里获取：
<https://courses.csail.mit.edu/6.042/spring18/>. (I strongly recommend searching for the most recent revision.)
- Pat Morin. 开放数据结构，最新修订于 2016 年 1 月（0.1G β 版本）。免费开放内容的教科书，Pat 维护并定期更新。可以在这里获取：<http://opendatastructures.org/>.

额外的参考资料

请不要局限于此或任何其他单一的参考资料。作者和读者把他们自己的观点带到任何一个材料上；每个学生，甚至每个非常厉害的学生都没有导师的指导。找到一个最有效地让他们的直觉进入你头脑的作者需要一些努力，但从长远来看，这项努力付出了巨大的代价。

以下参考文献是直觉、例子、练习和灵感的特别有价值的来源；这并不是一个完整的列表。

- Alfred V. Aho, John E. Hopcroft, and Jeffrey D. Ullman. The Design and Analysis of Computer Algorithms. Addison-Wesley, 1974. (I used this textbook as an undergraduate at Rice and again as a masters student at UC Irvine.)
- Thomas Cormen, Charles Leiserson, Ron Rivest, and Clifford Stein. Introduction to Algorithms, third edition. MIT Press/McGraw-Hill, 2009. (I used the first edition as a teaching assistant at Berkeley.)
- Sanjoy Dasgupta, Christos H. Papadimitriou, and Umesh V. Vazirani. Algorithms. McGraw-Hill, 2006. (Probably the closest in content to this book, but considerably less verbose.)
- Jeff Edmonds. How to Think about Algorithms. Cambridge University Press, 2008.
- Michael R. Garey and David S. Johnson. Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman, 1979.
- Michael T. Goodrich and Roberto Tamassia. Algorithm Design: Foundations, Analysis, and Internet Examples. John Wiley & Sons, 2002.
- Jon Kleinberg and Éva Tardos. Algorithm Design. Addison-Wesley, 2005. Borrow it from the library if you can.
- Donald Knuth. The Art of Computer Programming, volumes 1–4A. Addison-Wesley, 1997 and 2011. (My parents gave me the first three volumes for Christmas when I was 14. Alas, I didn't actually read them until much later.)
- Udi Manber. Introduction to Algorithms: A Creative Approach. Addison-Wesley, 1989. (I used this textbook as a teaching assistant at Berkeley.)
- Ian Parberry. Problems on Algorithms. Prentice-Hall, 1995 (out of print). Downloadable from <https://larc.unt.edu/ian/books/free/license.html> after you agree to make a small charitable donation. Please honor your agreement.
- Robert Sedgewick and Kevin Wayne. Algorithms. Addison-Wesley, 2011.
- Robert Endre Tarjan. Data Structures and Network Algorithms. SIAM, 1983.
- Class notes from my own algorithms classes at Berkeley, especially those taught by Dick Karp and Raimund Seidel.
- Lecture notes, slides, homeworks, exams, video lectures, research papers, blog posts, and full-fledged MOOCs made freely available on the web by innumerable colleagues around the world.

关于练习

每章的结束都会有几个练习，其中大部分我在家庭作业、讨论/实验部分或考试中至少使用过一次。这些练习不是通过增加难度来进行的，而是（通常）通过共同的技术或主题进行的。一些问题用符号注释如下：

- ♥红心表示特别具有挑战性的问题；其中许多问题都出现在伊利诺伊州的博士生资格考试中。少数真正困难的问题以较大的♥红心为标志，少数开放性问题以较大♥红心为标志。
- ◆蓝色菱形表示需要熟悉后面章节的材料的问题，但在主题上属于它们所在的位置。然而，需要熟悉早期材料的问题并没有被标记出来；这本书和《生活》一样，是累积的。
- ♣绿色梅花指出需要熟悉本书范围之外的材料的问题，例如有限状态机、线性代数、概率或平面图。这些是罕见的。

- ♠黑桃表示需要大量体力劳动和/或编码的问题。这些是罕见的。
- ★橙色的星星表明你正在吃 1998 年前制造的幸运符。

这些练习被设计成实践的机会，而不是为了他们自己的目的；这些问题的目的不是为了解决这些特定的问题，而是为了实践一项特定的技能，或者解决一类问题。部分是因为这个原因，我不提供练习的答案；答案不是重点。特别的，没有“教师手册”；如果你自己不能解决问题，你或许不应该把它发给你的学生。也就是说，你可以在我教的任何课程的网页上找到我这学期布置的任何家庭作业问题的答案。没有什么能阻止你写一本教员手册！

偷这本书!

这本书是在一个知识共享许可下出版的，在没有我的许可下，允许你使用，重新分配，改编和混合它的内容，只要你标注原始的来源。本书的完整电子版可在我的网站 <http://jeffe.cs.illinois.edu/teaching/algorithms/>（或助记快捷方式 <http://algorithms.wtf>）、bug 报告网站 <https://github.com/jeffgerickson/algorithms> 和互联网档案（请访问 Google）上免费获得。

这本书的网站还包含数百页有关的和更高级的附加讲义，以及一个几乎完整的以前的家庭作业、考试、讨论/实验问题和其他教学资源的档案。每当我教一门算法课时，我都会修改、更新，有时还会挑选我的教材，这样无论我现在教什么课程，你都可以在网页上找到最新的修改。

无论你是学生还是老师，我们都欢迎你在自己的课堂上使用这本教科书的任何一个子集或我的其他课堂讲稿，而无需征得我的同意，这就是为什么我要将它们放到网上！但是，也请引用这本书，无论是通过名字还是链接返回 <http://algorithms.wtf>；如果你是一名学生，这一点尤其重要，你可以使用我的课程材料来帮助你完成家庭作业。（请与您的导师核实。）

但是，如果你是一名老师，我强烈建议你补充一些你自己写的材料。你自己写材料将加强你对材料的掌握和课堂演示，从而提高你学生对材料的掌握。它也会让你克服处理这本书中你不喜欢的部分的挫折感。所有的教科书都不完善，这本也不例外。

最后，请在开放的网络上自由、轻松和全球范围内使用您所写的任何内容，而不要隐藏在学习管理系统的大门后，这样其他地方的学生和教师就可以从您独特的见解中获益。特别是，如果您开发了直接补充本教材的有用资源，如幻灯片、视频或解决方案手册，请让我知道，以便我可以从图书网站添加到您的资源的链接。

致谢

这本教科书充分利用了无数学生、教师和研究人员对算法的贡献。特别的，我非常感谢三千多名伊利诺伊大学学生，他们把我的讲稿作为主要参考，提出了有用（有时很痛苦）的批评，并修改了一些真正糟糕的早期草案。感谢世界各地的许多同事和学生，他们在自己的课堂上使用了这些笔记，并发送了有用的反馈和错误报告。

我特别感谢我出色的教学助理的反馈和贡献（尤其是练习）：

Aditya Ramani, Akash Gautam, Alex Steiger, Alina Ene, Amir Nayyeri, Asha Seetharam,
Ashish Vulimiri, Ben Moseley, Brad Sturt, Brian Ensink, Chao Xu, Charlie Carlson, Chris
Neihengen, Connor Clark, Dan Bullok, Dan Cranston, Daniel Khashabi, David Morrison, Ekta

Manaktala, Erin Wolf Chambers, Gail Steitz, Gio Kao, Grant Czajkowski, Hsien-Chih Chang, Igor Gammer, Jacob Laurel, John Lee, Johnathon Fischer, Junqing Deng, Kent Quanrud, Kevin Milans, Kevin Small, Konstantinos Koiliaris, Kyle Fox, Kyle Jao, Lan Chen, Mark Idleman, Michael Bond, Mitch Harris, Naveen Arivazhagen, Nick Bachmair, Nick Hurlburt, Nirman Kumar, Nitish Korula, Patrick Lin, Phillip Shih, Rachit Agarwal, Reza Zamani-Nasab, Rishi Talreja, Rob McCann, Sahand Moza'ari, Shalan Naqvi, Srihita Vatsavaya, Shripad Thite, Spencer Gordon, Subhro Roy, Tana Wattanawaroon, Umang Mathur, Vipul Goyal, Yasu Furakawa, and Yipu Wang.

我还得到了伊利诺伊大学同事巨大帮助: Alexandra Kolla, Cinda Heeren, Edgar Ramos, Herbert Edelsbrunner, Jason Zych, Kim Whittlesey, Lenny Pitt, Madhu Parasarathy, Mahesh Viswanathan, Margaret Fleck, Shang-Hua Teng, Steve LaValle, and especially Chandra Chekuri, Ed Reingold, and Sariel Har-Peled.

当然, 这本书要归功于那些在第一时间教给我这种算法的人: Bob Bixby and Michael Pearlman at Rice; David Eppstein, Dan Hirschberg, and George Lueker at Irvine; and Abhiram Ranade, Dick Karp, Manuel Blum, Mike Luby, and Raimund Seidel at Berkeley.

我从 Herbert Edelsbrunner 那里偷了整个课程结构的第一次迭代, 从他那里偷了一个想法, 写下我自己的演讲笔记; 从 Steve LaValle 那里偷了一个想法, 把我的笔记的一个子集变成了一本书; 从 Robert Ghrist 那里偷了书籍设计的几个组成部分。

课程告诫

当然, 这些人都不应该为这本书的任何缺陷而受到指责。尽管经过了多轮的修改和编辑, 这本书还是包含了许多错误、bug、出丑、遗漏、混乱、拼错、算错、语法、认识错误、思维错误、糟糕的设计决策、历史错误、时代错误、不一致、夸大、犹豫不决、乱说、扭曲、过于简单化、胡说八道、结痂以及全篇的谎言, **所有这些都是 Steve Skiena 的错。**

我在 <https://github.com/jeffgerickson/algorithms> 上维护了一个问题跟踪程序, 在这里, 像你这样的读者可以提交 bug 报告、功能请求和关于这本书的一般反馈。如果您发现任何类型的错误, 无论是数学错误、语法错误、历史错误、排版错误、文化错误还是其他错误, 无论是在正文、练习中还是在我的其他课程材料中。(Steve 不太可能关心) 当然, 所有其他的反馈也是受欢迎的!

享受吧!

—— Jeff

对于作者来说, 传统的做法是宽宏大量地接受对仍然存在的任何问题的指责。我不知道。这本书中的任何错误、不清楚或问题都是别人的错, 但我很高兴知道这些错误、不清楚或问题, 以便确定谁该受到责备。

—— Steven S. Skiena, 《算法设计手册》(1997)

毫无疑问, 这一声明后面将是所有教科书的注释清单, 以及为什么每一本都是废话。

—— Adam Contini, MetaFilter, 2010 年 1 月 4 日