



计 算 机 科 学 丛 书

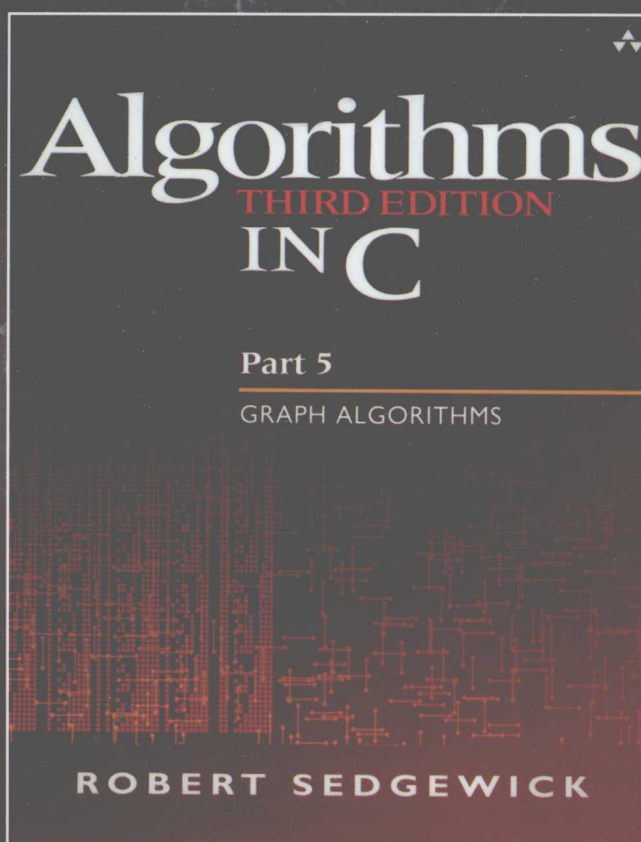
原书第3版

算法：C语言实现

(第5部分)

图算法

(美) Robert Sedgewick 著 霍红卫 译
普林斯顿大学 西安电子科技大学



Algorithms in C
Parts 5, Graph Algorithms
Third Edition



机械工业出版社
China Machine Press

计 算 机 科 学 丛 书

原书第3版

算法：C语言实现

(第5部分)

图算法

(美) Robert Sedgewick 著 霍红卫 译
普林斯顿大学 西安电子科技大学

Algorithms in C
Part 5, Graph Algorithms
Third Edition



机械工业出版社
China Machine Press

本书是深入论述算法的三卷本教程《算法：C 语言实现》（第 3 版）中的第二卷——图算法。作者在这次修订中重写了许多内容，增加了数千个新练习、数百个新图表、数十个新程序，并对图表和程序做了详尽的注释说明。新版中不仅涵盖了新的主题，而且还提供了对许多经典算法的更充分的解释，包括图的性质、图搜索、有向图、最小生成树、最短路径和网。本书涵盖了足够的基本内容及较详细的图算法高级主题，既可单独用作数据结构与算法课程的教材，也可与第一卷（第 1~4 部分）结合使用。

本书适合高等院校计算机专业师生参考，也可供软件开发人员参考。

Simplified Chinese edition copyright © 2010 by Pearson Education Asia Limited and China Machine Press.

Original English language title: *Algorithms in C: Part 5, Graph Algorithms; Third Edition* (ISBN 0-201-31663-3) by Robert Sedgewick. Copyright © 2002.

All rights reserved.

Published by arrangement with the original publisher, Pearson Education, Inc., publishing as Addison-Wesley.

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

本书法律顾问 北京市展达律师事务所

本书版权登记号：图字：01-2006-3990

图书在版编目（CIP）数据

算法：C 语言实现——第 5 部分，图算法（原书第 3 版）/（美）塞奇威克（Sedgewick, R.）著；霍红卫译. —北京：机械工业出版社，2009.11

（计算机科学丛书）

书名原文：Algorithms in C: Part 5, Graph Algorithms; Third Edition

ISBN 978-7-111-28505-2

I. 算… II. ①塞… ②霍… III. ①电子计算机—算法理论 ②C 语言—程序设计
IV. TP301.6 TP312

中国版本图书馆 CIP 数据核字（2009）第 183990 号

机械工业出版社（北京市西城区百万庄大街 22 号 邮政编码 100037）

责任编辑：刘立卿

北京市荣盛彩色印刷有限公司印刷

2010 年 1 月第 1 版第 1 次印刷

184mm × 260mm · 19.75 印张

标准书号：ISBN 978-7-111-28505-2

定价：59.00 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

客服热线：(010) 88378991；88361066

购书热线：(010) 68326294；88379649；68995259

投稿热线：(010) 88379604

读者信箱：hzjsj@hzbook.com

出版者的话

文艺复兴以降，源远流长的科学精神和逐步形成的学术规范，使西方国家在自然科学的各个领域中取得了垄断性的优势；也正是这样的传统，使美国在信息技术发展的六十多年间名家辈出、独领风骚。在商业化的进程中，美国的产业界与教育界越来越紧密地结合，计算机学科中的许多泰山北斗同时身处科研和教学的最前线，由此而产生的经典科学著作，不仅筹划了研究的范畴，还揭示了学术的源变，既遵循学术规范，又自有学者个性，其价值并不会因年月的流逝而减退。

近年，在全球信息化大潮的推动下，我国的计算机产业发展迅猛，对专业人才的需求日益迫切。这对计算机教育界和出版界都既是机遇，也是挑战；而专业教材的建设在教育战略上显得举足轻重。在我国信息技术发展时间较短的现状下，美国等发达国家在其计算机科学发展发展的几十年间积淀和发展的经典教材仍有许多值得借鉴之处。因此，引进一批国外优秀计算机教材将对我国计算机教育事业的发展起到积极的推动作用，也是与世界接轨、建设真正的世界一流大学的必由之路。

机械工业出版社华章分社较早意识到“出版要为教育服务”。自1998年开始，华章分社就将工作重点放在了遴选、移译国外优秀教材上。经过多年的不懈努力，我们与Pearson, McGraw-Hill, Elsevier, MIT, John Wiley & Sons, Cengage等世界著名出版公司建立了良好的合作关系，从他们现有的数百种教材中甄选出Andrew S. Tanenbaum, Bjarne Stroustrup, Brian W. Kernighan, Dennis Ritchie, Jim Gray, Alfred V. Aho, John E. Hopcroft, Jeffrey D. Ullman, Abraham Silberschatz, William Stallings, Donald E. Knuth, John L. Hennessy, Larry L. Peterson等大师名家的一批经典作品，以“计算机科学丛书”为总称出版，供读者学习、研究及珍藏。大理石纹理的封面，也正体现了这套丛书的品位和格调。

“计算机科学丛书”的出版工作得到了国内外学者的鼎力襄助，国内的专家不仅提供了中肯的选题指导，还不辞劳苦地担任了翻译和审校的工作；而原书的作者也相当关注其作品在中国的传播，有的还专程为其书的中译本作序。迄今，“计算机科学丛书”已经出版了近两百个品种，这些书籍在读者中树立了良好的口碑，并被许多高校采用为正式教材和参考书籍。其影印版“经典原版书库”作为姊妹篇也被越来越多实施双语教学的学校所采用。

权威的作者、经典的教材、一流的译者、严格的审校、精细的编辑，这些因素使我们的图书有了质量的保证。随着计算机科学与技术专业学科建设的不断完善和教材改革的逐渐深化，教育界对国外计算机教材的需求和应用都将步入一个新的阶段，我们的目标是尽善尽美，而反馈的意见正是我们达到这一终极目标的重要帮助。华章分社欢迎老师和读者对我们的工作提出建议或给予指正，我们的联系方式如下：

华章网站：www.hzbook.com

电子邮件：hzjsj@hzbook.com

联系电话：(010) 88379604

联系地址：北京市西城区百万庄南街1号

邮政编码：100037



华章教育

华章科技图书出版中心

译者序

本书是算法方面的优秀著作之一。它系统地阐述了算法学的特征以及算法的应用领域，讨论了算法分析在理论计算机科学中的作用。并通过实验数据和分析结果表明选择何种算法来解决实际问题。这卷书对图的性质和类型、图搜索、有向图、最小生成树、最短路径及网络流等内容进行了透彻的论述。通过归约的概念，建立了调度问题与最短路径问题之间的关系。

这本书不仅适合于软件设计人员和计算机科学专业的学生，而且也适合于那些想利用计算机并使它更有效或是想要解决更大问题的人们。这本书中的算法代表了图算法领域所研究的知识主体。对于大量的应用问题，这些知识主体已经成为有效使用计算机的不可缺少的部分，尤其体现在网络算法、电路设计、调度、事务处理、资源分配等领域，在此所描述的基本方法在这些领域的科学研究及应用中已日显重要。作为最有影响的搜索引擎之一 Google，其中最著名的 PageRank 算法就是图模型成功应用的一个典型代表。

本书主要内容及特点如下：

- 对于图的性质及类型做了完整的综述。
- 提供了有向图、最小生成树、最短路径及网络流方面的诸多算法，这些算法是计算机科学的核心基础。
- 对算法提供了很多可视化的信息，还有大量实验研究和基本分析研究，从而为选择算法解决实际问题提供了依据。
- 提供了 700 多个练习题，从而有助于深入理解算法的特征以及设计有效的算法。
- 本书以大量图例说明算法的工作过程，使得算法更加易于理解和掌握。
- 本书提供了读者易于实现和调试的 C 语言描述的算法的详尽信息，并通过示例程序展示了算法工作的详尽过程。
- 适合作为高等院校算法设计课程的教材，同时可作为从事软件开发和工程设计的专业人员的参考书。

由于时间较紧及译者水平有限，译文难免有错误及不妥之处，恳请读者批评指正。

最后感谢本书的编辑们所做的细致工作，使得本书的文字更加优美和流畅。

霍红卫

西安电子科技大学计算机学院

2009 年 9 月

中文版序

In the years since this book was first published in English, there has been a profound shift in the way that students can learn algorithms. Specifically, the widespread deployment of personal computers and workstations has meant that each student can have computational experience with each algorithm learned. Teachers, students, and readers in China who take advantage of this shift have the opportunity to gain deeper understanding of a broader range of material about algorithms than ever before.

Students can know that all of the code in this book has been implemented and tested. The code that appears in the book is available on the web, and each student should, from the beginning, get into the habit of running each algorithm, testing it on various inputs, developing basic experimental results, and comparing the performance of different algorithms for the same task. With this preparation, a student often can answer a question that comes to mind about an algorithm, by running it. If a modification that might improve the algorithm comes to mind, a student can study whether or not it does so. I am certain that new variations of these algorithms and new algorithms to solve the problems that we address will be discovered as hundreds of thousands (or even millions) of a new generation of students learn to work with algorithms in such a concrete manner.

Teachers can encourage students to learn properties of each algorithm in detail, with knowledge that the book builds layers of abstraction. Data structures such as priority queues and symbol tables might seem rather abstract when students first encounter them, but they play an essential role in developing efficient solutions to complicated real-world problems. With a solid understanding of the basic abstractions, students are prepared to appreciate their importance in solving the more difficult problems. By encouraging students to treat the understanding of each algorithm and data structure as an intellectual exercise, teachers often can later reward that effort by showing how to solve a real-world problem that could not be approached without proper use of a fundamental algorithm.

As mentioned in the Notes on Exercises in the original book, there are far too many exercises for anyone to attempt them all. The most important advice that I can give each student, teacher, and reader in China is to read all the exercises and use them (and the basic material in the text) as a starting point to expand our knowledge about algorithms.

自本书首次在英国出版以来, 学生们学习算法的方法一直在发生着深刻的转变。具体来说, 个人电脑和工作站的广泛使用, 意味着每个学生都可以实现所学的每个算法。中国的教师、学生以及读者谁利用了这种转变, 就有机会获得关于算法的更广泛素材的深层次的理解, 这是前所未有的。

学生们可能知道，书中的所有代码均已实现并测试过。出现在书中的代码可在网上得到，每个学生应该从一开始，就习惯性地运行每个算法，测试它的各种输入，开发基本的实验结果，并比较同一任务的不同算法的性能。有了这个准备，学生通过运行算法，往往能回答所想到的算法问题。如果学生想到改进算法的办法，可以进行深入的研究以确定是否会这样。我确信，随着成千上万（甚至百万）新一代的学生学会以这种具体方式所表示的算法，就会发现这些算法新的变型以及发现求解所涉及问题的新算法。

教师可以鼓励学生利用本书所构建的抽象层的知识来详细了解每个算法的性质。当学生首次遇到优先队列和符号表这样的数据结构时，它们看起来相当抽象，但这些数据结构在开发复杂实际现实问题的有效解决方案方面起着重要的作用。有了对于基本抽象的扎实理解，学生就可准备欣赏它们在解决更加困难问题中的重要性。教师要鼓励学生理解每个算法和数据结构，这样他们将能解决现实世界的问题，而这些是要通过合理地使用基本算法才能做到的。

正如原书练习中的注释中提到的，有太多的练习需要人们尝试。我能给中国学生、教师以及读者的最重要的建议，就是阅读所有的练习，并使用它们（和文中的基本素材）作为我们扩充关于算法知识的起点。

Robert Sedgewick

2009 年 10 月

前 言

清华大学出版社

图和图算法在现代计算机应用中颇为常见。对于在实际中出现的一些图处理问题，本书描述了目前最重要的解决方法。由于需要相关知识的人日渐增多，本书的主要目标是使读者了解这些方法及其所蕴含的基本原理。本书从基本原理展开，并从基本信息开始，从经典方法到现代仍在研发中的技术逐一展开讨论。精心选择的示例、详尽的图表以及完善实现的补充材料无一不体现在算法和应用的描述中。

算法

本书是对当前使用的最重要的计算机算法进行深度研究的三卷中的第二卷：图算法。第一卷（第1~4部分）覆盖了基本概念（第1部分）、数据结构（第2部分）、排序算法（第3部分）和搜索算法（第4部分）；本卷（第5部分）覆盖了图与图算法；（尚未出版的）第三卷（第6~8部分）覆盖了字符串（第6部分）、计算几何（第7部分）和高级算法及应用（第8部分）。

这些书可作为计算机科学低年级本科生的教材。学习本课程之前要求学生掌握基本程序设计技巧并熟知计算机系统，不过尚未选修计算机科学或计算机应用的高级领域的专业课程。这些书还可用作自学或作为从事计算机系统或应用程序开发的参考读本，因为它们包含了实用算法的实现以及关于这些算法性能特征的详尽信息。这些书包含内容广泛，适合作为这一领域的入门读物。

多年以来，这三卷书共同构成的《算法：C语言实现》（第3版）已经得到世界各地的学生和程序员的广泛使用。我完全重写了这一版的内容，并且增加了数千个新练习、数百个新图表、数十个新程序以及对图表和程序详尽的注释说明。这个新版本不仅涵盖了新的主题，而且还提供了对许多经典算法的更充分的解释。全书对抽象数据类型的强调使这些程序使用更为广泛，而且在现代面向对象的编程环境中也更为适用。对于已经阅读过以前版本的人来说，会从这一版找到更为丰富的信息；并且所有读者都会从中找到富有教益的内容，有效地学习本书提供的基本概念。

这些书适合于程序员和计算机科学专业的学生阅读。每一个使用计算机的人都希望它能运行得更快，或者可解决更大规模的问题。我们所考虑的算法代表了近50年发展起来的知识体系，该体系是在各种应用中有效地使用计算机解决问题不可缺少的部分。从物理学中的 N -体模拟问题到分子生物学中的基因序列问题，在此所描述的基本方法在科学研究中已日显重要；另外，对于从数据库系统到Internet搜索引擎，这些方法已经成为现代软件系统的重要组成部分。随着计算机应用的覆盖面越来越广，基本算法的影响也日益显著，特别是本书所涵盖的基本图算法，作用更为突出。本书的目标是要提供一种资源，使广大学生以及专业人士可以了解并明智地利用图算法来解决计算机应用中出现的问题。

本书范围

本书共6章，包含了图的性质及类型、图搜索、有向图、最小生成树、最短路径和网。希望本书描述的方法使读者能够尽可能广泛地理解图算法的基本属性。

如果读者已经学过算法设计与分析的基本原理，并且有利用 C、Java 或 C++ 等高级编程语言的经验，你将会更有效地学习本书的内容。当然，可以利用本书第一卷（第 1 ~ 4 部分）做充分的准备。本书假设读者有关于数组、链表和 ADT 设计的基本知识，并使用过优先队列、符号表以及合并 - 查找 ADT。它们都在前四部分中描述过（同时也在其他关于算法与数据结构的书中介绍过）。

图和图算法的基本性质根据基本原理即可确立，但要充分理解算法的性质却需要扎实的数学基础。尽管这里所讨论的高级数学概念是简明的、概括性的和描述性的，但相对于前四部分的主题，还是需要读者有较好的数学基础才能更深入地理解图算法。不过，有不同数学基础的读者都可从中受益。这是因为每个人应该理解并使用的基本图算法只是与并非所有人都理解的高级算法略有差异。在此的主要意图是把一些重要的算法与贯穿本书的其他一些方法放在一起讨论，而不是对所有数学知识做全面的介绍。但是，严谨性是好的数学基础所要求的，由此可以得到好的程序。因此我尽量在理论家所喜爱的形式化处理和实践家所需要的内容丰富性之间进行权衡，同时不失严谨性。

教学用法

在教学中如何使用本书内容有很大的灵活性，这取决于教师的偏好以及学生所做的准备。这里所描述的算法多年以来已经得到广泛应用，而且无论对于从事实际工作的程序员还是计算机科学专业的学生，这些算法都代表了基本的知识体系。书中涵盖了足够的基本内容可用作数据结构和算法课程的学习，也有足够详细的高级主题用于图算法课程。有些教师可能希望强调与实现和实践有关的内容，而另外一些教师则可能把重点放在分析和理论概念上。

也可把本书与前四部分的某些内容结合起来作为一个更为全面的课程教材。这样，教师就可以用一种一致的风格来讲授基础知识、数据结构、排序、搜索和图算法的内容。教学中使用的幻灯片、程序设计示例作业、为学生提供的交互式练习以及与课程有关的其他资料都可在本书的主页上找到。

书中的练习（几乎全都是在这一版中新增加的）可分为几类。一类练习的目的是为了测试对正文中内容的理解，要求读者能够完成某个例子或应用正文中描述的概念。另一类练习则涉及实现算法和把算法整理到一起，或者进行实验研究从而对各种算法进行比较以及了解其性质。还有一类练习则是一些重要信息的知识库，其详细程度本身不适合放在正文中。阅读并思考这些练习，会使每个读者受益匪浅。

实用算法

任何人若希望更有效地使用计算机，都可以把这本书用作参考书，或用于自学。有程序设计经验的人可以从本书中找到有关某个特殊主题的信息。从很大程序上说，尽管某些情况下某一章中的算法使用了前一章中的方法，但读者仍可以独立于本书的其他章节阅读本书的某个章节。

本书的定位是研究实用的算法。本书提供了算法的详尽信息，读者可以放心地实现和调试算法，并使算法能够用于求解某个问题，或者为某个应用提供相关功能。书中包括了所讨论的方法的完整实现，并在一系列一致的示例程序中给出了这些操作的描述。由于我们使用了实际代码，而不是伪代码，因而在实际中可以很快地使用这些程序。通过访问本书的主页可以得到程序的代码清单。

实际上，书中算法的一个实际应用会产生数百幅图表。正是这些图表提供的立体视觉直观地发现了许多算法。

本书详细讨论了算法的特征以及它们可能应用的场合。尽管并不强调，但是书中论述了算法分析与理论计算机科学的联系。在适当的时候，书中都给出了经验性的数据和分析结果用以说明为什么选择使用某些算法。如果有趣，书中还会描述所讨论的实际算法与纯理论结果之间的关系。关于算法性能特征和实现的特定信息的综合、概括和讨论都会贯穿本书的始终。

编程语言

书中所有实现所用的程序设计语言均为 C 语言（本书的 C++ 版本和 Java 正在开发）。任何特定语言都有优缺点；我们使用 C 语言是因为它是一种广泛使用的语言，并且能够为本书的实现提供所需的特征。由于没有多少结构是 C 所特有的，因而用 C 编写的程序可以很容易地变成用其他现代编程语言编写的程序。在适当的时候，我们会使用标准 C 中的术语，但本书并不打算成为 C 程序设计的参考手册。

我们力争编写优雅、简明和可移植的代码实现，但同时关注实现的效率，因而我们在开发的各个阶段都试图了解代码的性能特征。在这一版中有很多新的程序，老版的很多程序也已更新，主要的目的是使它们在用作抽象数据类型实现时更为实用。对程序所做的广泛的实验性比较研究贯穿在本书中。

本书的目标是以尽可能简单、直接的方式来展示算法。本书使用尽可能一致的风格，因而很多程序看起来是相似的。对于许多算法而言，无论使用哪种语言，算法都具有相似性：例如（选取一个著名的例子）Dijkstra 算法就是 Dijkstra 算法，无论采用 Algol-60、Basic、Fortran、Smalltalk、Ada、Pascal、C、C++、Modula-3、PostScript、Java 表示，还是任何一种其他编程语言表示，也不管其所在的环境，它都被证明是一种有效的图处理算法。

有关练习的说明

给练习分类是一件充满风险的事情，因为本书读者的知识背景和经验参差不齐。虽然如此，还应适当加上指导，所以许多练习都加了一个记号（共有 4 种不同记号），以帮助你判断如何加以解决。

测试你对内容理解程度的练习用空心三角符号标识，如下所示：

▷ 17.2 给出下图的所有连通子图。

0-1 0-2 0-3 1-3 2-3

通常，这样的练习是与正文中的例子直接相关的。它们的难度不大，但是完成这些练习会使你了解某个事实或掌握某个概念，它们可能是你在阅读正文时感到困惑不解的问题。

为正文内容添加新的且需要思考的信息的练习用空心圆符号标识，如下所示：

○ 18.2 在对图 18.2 和 18.3 中所示的迷宫进行 Trémaux 探索时，以下序列中，哪一个不能作为各交叉点开灯的顺序？

0-7-4-5-3-1-6-2
0-2-6-4-3-7-1-5
0-5-3-4-7-1-6-2
0-7-4-6-2-1-3-5

这种练习将鼓励你考虑与正文中内容相关的重要概念，或者回答出现在你阅读正文时遇

到的一个问题。即使你没有时间做这些练习，也会发现阅读这些练习是非常有价值的。

具有挑战性的练习用黑色圆点标识，如下所示：

- 19.2 在线找出一个大型 DAG 图，可以是一个大型软件系统中函数定义依赖关系所定义的 DAG 图，也可以是一个大型文件系统的目录连接所定义的 DAG 图。

这种练习可能需要花费大量时间才能完成，这取决于你的经验。一般而言，最有效的方法是分几个阶段来解决。

少数难度极大的练习（相对于其他大多数练习）用两个（甚至三个）黑色圆点标识，如下所示：

- 20.35 开发一个用于生成 V 个顶点、 E 条边的随机图的合理程序，使得 Prim 算法的 PFS 实现（程序 20.4）的运行时间是非线性的。

这种练习类似于研究文献上所讨论的问题，但本书中的内容可能会让你做好准备，乐于尝试解决这些问题（而且很可能成功）。

对于考察你的程序设计能力和数学能力的练习，则没有明确记号。这些要求程序设计能力或数学分析能力的练习是一种自我检查。我们鼓励所有的读者都通过实现算法来检测对算法的理解程度。

致 谢

对于本书早期的版本，很多人提供了有用的反馈信息。特别要提出的是普林斯顿大学和布朗大学成百上千的学生们在过去的多年里一直承受着初稿的粗糙。特别要感谢 Trina Avery 和 Tom Freeman 对于第一版的出版所给与的帮助；还要感谢 Janet Incerpi，是她的创造力和聪明才智使我们使用早期原始的数字排版硬件和软件来制作本书的第一版；要感谢 Marc Brown，他参与了算法可视化研究，而本书中的很多图表正来源于此；感谢 Dava Hanson，对于我提出的关于 C 语言方面的所有问题，他总是乐于回答。感谢 Kevin Wayne，它耐心回答了我关于网的基本问题。我还要感谢众多读者，他们对各个版本提供了详尽的评论，这其中包括：Guy Almes、Jon Bentley、Marc Brown、Jay Gischer、Allan Heydon、Kennedy Lemke、Udi Manber、Dana Richards、John Reif、M. Rosenfeld、Stephen Seidman、Michael Quinn 和 William Ward。

为了完成这一新版本，我有幸与 Addison-Wesley 的 Peter Gordon 和 Helen Goldstein 一起工作。他们耐心地指导这个项目的完成，使其经历了从一般性修改到大幅度重写的过程。同样我还有幸与 Addison-Wesley 的许多专业人员一起工作。这个项目的性质使得本书带给他们非同寻常的挑战，对于他们的忍耐力我致以诚挚的感谢。

在写这本书的过程中，我得到了两位良师益友。在此我要特别向他们表示感谢。首先，Steve Summit 从技术的角度对各版本的初稿都做了仔细的检查，并向我提出了数千条详尽的意见，尤其是关于程序方面的建议。Steve 很清楚我的目标是要提供优雅、有效且实用的实现代码，他的意见不仅帮助我给出实现一致性的度量标准，而且还帮助我对其中许多实现做了重大的改进。其次是 Lyn Dupre，他也对我的手稿提出了数千条的意见，这些建议对我来说是无价之宝，不仅帮助我改正和避免了许多语法错误，而且更重要的是，使我找到一种一致性的写作风格，由此才使我把如此繁多的技术资料整理在一起。能够有机会向 Steve 和 Lyn 学习，我万分感激。他们的投入对于本书的完成至关重要。

我在这里所写的内容大多受益于 Don Knuth 的授课和著作，他是我在斯坦福大学的导

师。尽管 Don 对本书没有直接的影响，但在本书中仍然能够感受到他的存在，因为正是他为算法研究奠定了科学基础，才使像本书这样的工作得以完成。我的朋友兼同事 Philippe Flajolet 是使算法分析发展成为一个成熟领域的主力，他对这本书具有同样的影响力。

非常感谢普林斯顿大学、布朗大学以及法国国立计算机与自动化研究所（Institute National de Recherche en Informatique et Automatique, INRIA）给予的支持，在这些地方，我完成了本书大部分的工作；还要感谢美国国防部防御分析研究所（Institute for Defense Analysis）以及施乐的帕洛阿尔托研究中心（Xerox Palo Alto Research Center），我在他们那里访问期间完成了本书的一些工作。本书的某些部分离不开国家自然科学基金（National Science Foundation）和海军研究中心（Office of Naval Research）的慷慨支持。最后，我要感谢 Bill Bowen、Aaron Lemonick 和 Neil Rudenstine，感谢他们对建立普林斯顿大学学术环境的支持，使我能够在这样良好的环境中，在承担众多其他事务的同时完成本书。

Robert Sedgewick

Marly-le-Roi, 法国, 1983 年 2 月

普林斯顿, 新泽西州, 1990 年 1 月

詹姆斯镇, 罗得岛, 2001 年 5 月

目 录

出版者的话
译者序
中文版序
前 言

第五部分 图 算 法

第 17 章 图的性质及类型	1
17.1 术语	3
17.2 图的 ADT	9
17.3 邻接矩阵表示	12
17.4 邻接表表示	15
17.5 变量、扩展和开销	18
17.6 图生成器	23
17.7 简单路径、欧拉路径和哈密顿 路径	30
17.8 图处理问题	40
第 18 章 图 搜 索	47
18.1 探索迷宫	47
18.2 深度优先搜索	50
18.3 图搜索 ADT 函数	54
18.4 DFS 森林的性质	57
18.5 DFS 算法	62
18.6 可分离性和双连通性	66
18.7 广度优先搜索	72
18.8 广义图搜索	79
18.9 图算法分析	85
第 19 章 有向图和有向无环图	90
19.1 术语和游戏规则	91
19.2 有向图中的 DFS 剖析	98
19.3 可达性和传递闭包	103
19.4 等价关系和偏序	112
19.5 有向无环图	114

19.6 拓扑排序	117
19.7 有向无环图中的可达性	124
19.8 有向图中的强连通分量	126
19.9 再论传递闭包	133
19.10 展望	136
第 20 章 最小生成树	139
20.1 表示	141
20.2 MST 算法的基本原理	144
20.3 Prim 算法和优先级优先搜索	149
20.4 Kruskal 算法	156
20.5 Boruvka 算法	160
20.6 比较与改进	163
20.7 欧几里得 MST	167
第 21 章 最短路径	169
21.1 基本原理	174
21.2 Dijkstra 算法	177
21.3 所有对最短路径	185
21.4 无环网中的最短路径	191
21.5 欧几里得网	197
21.6 归约	201
21.7 负权值	211
21.8 展望	223
第 22 章 网络流	225
22.1 流网络	228
22.2 增大路径最大流算法	236
22.3 预流 - 推进最大流算法	254
22.4 最大流归约	264
22.5 最小成本流	275
22.6 网络单纯形算法	282
22.7 最小成本流归约	293
22.8 展望	299
第五部分参考文献	302

第五部分 图 算 法

第 17 章 图的性质及类型

实际应用中的许多计算问题不仅仅涉及元素 (item) 的集合, 还会涉及元素之间连接 (connection) 的集合。这些连接所蕴含的关系很自然地引出大量的问题: 沿着连接是否存在从一个元素到另一个元素的路径? 从某一给定元素可以到达多少个其他元素? 从这个元素到另一个元素的最佳路径是什么?

为了建立这些情况的模型, 我们使用一种称为图 (graph) 的抽象对象。在这一章里, 我们会详细考察图的基本性质, 为学习多种有用算法打下基础, 这些算法可用于回答前面所提出的那些问题。这些算法充分利用了我们在本套书第 1~4 部分所讨论的计算工具。另外它们还可以作为解决重要应用问题的基础, 因为没有好的算法技术, 我们甚至无法考虑这些问题的解决方案。

图论是组合数学的一个主要分支。它在数百年来已得到深入研究。很多重要且有用的图的性质已经得到证明, 但仍有大量难题未被解决。尽管仍有很多内容需要学习, 但是在本书中我们将只从有关图的众多知识中抽取我们需要理解的知识, 并使用大量有用且基础的算法。

与我们研究的其他很多问题领域类似, 图的算法学的研究相对较新。尽管一些基本算法非常古老, 但多数算法都是在最近几十年发现的。即使是最简单的图算法, 也会得出很有用的计算机程序, 而我们考察的非平凡算法是已知的最为优雅和有趣的算法。

为了阐明涉及图处理的应用的多样性, 我们从几个例子来探索这个内容丰富的领域中的算法。

地图 某人计划一次旅行, 他需要回答诸如这样的几个问题: “如何用最小花费 (least expensive) 从普林斯顿到达圣何塞? ”。如果一个人更加关注时间而不是金钱, 他可能会问: “如何用最短时间 (fastest) 从普林斯顿到达圣何塞? ”。为了回答这些问题, 我们需要处理元素 (城镇) 之间的连接 (旅行路线) 信息。

超文本 当我们浏览 Web 时, 会遇到一些文档, 其中包含着引用其他文档的链接 (link)。通过单击这些链接, 我们就从一个文档转到另一个文档, 整个 Web 就是一个图, 其中元素就是文档, 连接是链接。图处理算法是搜索引擎的基本组成部分, 可以帮助我们在 Web 上定位信息。

电路 电路中包括诸如晶体管、电阻器和电容等元件, 它们复杂地连接在一起。我们使用计算机来控制组成电路的机器, 检查电路是否执行所需的功能。我们要回答一些简单的问题, “这是一条较短的路径吗? ”, 复杂的问题有: “我们是否可以把这个电路布在一个芯片上以使这些连线互不交叉? ”。在这种情况下, 对第一个问题的回答只取决于连接 (导线) 的性质, 而对第二个问题的回答则需要关于导线、这些导线连接的元素以及芯片的物理约束

条件等详细信息。

调度 制造过程中需要执行大量满足一些约束条件的任务，这些条件规定某个任务必须要在其他任务完成之后才能开始执行。我们把这些约束条件看作任务（元素）之间的连接，这样就得到一个经典的调度（scheduling）问题：如何调度任务，才能既满足约束条件，又能在最短时间内完成整个过程。

事务 某电话公司维护一个电话呼叫通信量的数据库。这里，连接表示为电话呼叫。我们对互连结构的特点很感兴趣，因为我们希望布线及设置开关使这个结构能够有效地处理通信量。另一个例子是金融机构跟踪市场中的买/卖订单状况。这种情况下连接表示两个客户之间的现金转移。在这个例子中有关连接结构特点的知识可以增强我们对于市场情况的了解。

匹配 学生们要在诸如社会团体、大学或医院等选择性的机构申请职位。这里的元素对应着学生和机构；连接对应着申请。我们希望找到一些方法使感兴趣的学生与某些可用职位相匹配。

网络 计算机网络由互联的站点组成，这些站点可以发送、转发和接收各种类型的消息。我们不仅关注可以得到从一个站点到其他站点的消息，同时关注在网络变化时维持站点之间的连接关系。例如，我们可能希望检查某个给定的网络，以确信其中没有一小部分站点或连接对网络是如此关键，以至于一旦它们出现问题将会造成其余站点之间无法连接。

程序结构 编译程序用构建图的方式来表示一个大型软件系统的调用结构。这里的元素是各种函数或组成系统的模块；连接要么关联一个函数调用另一个函数的可能性（静态分析），要么关联正在运行的系统的实际调用（动态分析）。我们需要分析这个图以确定如何最好地为程序分配资源以达到高效。

这些例子表明，在很多应用中，图是最合适的抽象模型，也表明了我们使用图时所遇到的计算问题。这样的问题将是我们本书的重点。在实际中遇到上述很多应用时，所涉及的数据量非常巨大，高效的算法则会影响到一个解决方案是否可行。

在这套书的第1部分我们已经遇到了图。实际上，我们在第1章详细介绍的第一个算法是合并-查找算法，它是图算法的一个基本例子。我们在第3章还使用图来说明二维数组和链表的应用。在第5章中用图说明了递归程序和基本数据结构之间的关系。任何链式数据结构都是一种图的表示，处理树和其他一些链表结构的常用算法都是图算法的特例。本章的目的是为理解所有图算法提供一种环境，包括从第1部分的简单算法到第18章至第22章的复杂算法。

像往常一样，我们很想知道解决某个特定问题使用哪种算法最为有效。研究图算法的性能特征具有很大的挑战性，这是因为：

- 算法的开销不仅依赖于元素集性质，而且依赖于连接集的很多性质（以及连接关系中所蕴含的图的全局性质）。
- 我们可能遇到的各类图的精确模型是难以建立的。

我们通常假设图算法的最坏情况下的性能界限，即使某些情况下的实际性能评价可能更高。幸运的是，正如我们将要看到的，大量的算法是最优的（optimal），并涉及极少不必要的工作。对于所有给定规模的图，其他一些算法消耗同量的资源。我们可以精确地预测这些算法在特定情况下是如何执行的。不能做出这样精确的预测时，我们需要特别关注各种图的性质，以便对实际情况做出估计，而且必须对这些性质对于算法性能的影响做出评估。

我们首先讨论图的定义和性质，考察用于描述它们的标准术语。接着将定义基本 ADT

(抽象数据类型) 接口, 使用这些接口我们可以研究图算法, 还将定义两个表示图的最重要的数据结构, 它们是邻接矩阵 (adjacency matrix) 表示和邻接表 (adjacency list) 表示。还要研究实现基本 ADT 函数的各种方法。然后, 我们考察可以生成随机图的客户程序, 可以使用这些程序来测试我们的算法和学习图的性质。所有这些材料为我们引入图处理算法提供了基础, 这些图处理算法可用于求解与在图中找路径相关的三个经典问题。它说明了图问题的难度有很大的差别, 即使这些问题看上去类似。我们在本章最后对本书中将要考察的最重要的图处理问题进行了综述, 并根据它们的解决难度进行了排列。

17.1 术语

有大量与图有关的术语。大多数术语有简明的定义。为了便于参考, 这里集中介绍这些术语。我们在第 I 部分讨论基本算法时已经使用过一些概念; 而另一些概念在第 18 章至第 22 章讨论高级算法时才会涉及。

定义 17.1 图 (graph) 由一个顶点 (vertex) 集和一个边 (edge) 集组成, 其中边将一对不同顶点连接在一起 (每对顶点之间至多有一条边连接)。

在 V 个顶点的图中, 我们用 0 到 $V-1$ 作为各顶点的名字。选择这种命名方法的主要原因是我们可以使用数组的下标快速地访问每个顶点对应的信息。在 17.6 节中, 我们考虑使用符号表的一个程序, 它建立了 V 个顶点名与 $0 \sim V-1$ 之间的 V 个整数的一一 (1-1) 对应。有了这个程序, 我们可以不失一般性地使用下标作为顶点名 (为表示方便)。有时假设隐含地定义了顶点集, 通过用边集来定义图, 且只考虑那些至少在一条边上包含的顶点。为了避免诸如“带有如下边集的 10 顶点图”这样的累赘用法, 当顶点个数可以从上下文中明显看出时, 我们并不会明确地指出顶点个数。根据约定, 我们总是用 V 表示一个给定的图中的顶点数, 用 E 表示其中的边数。

我们将图的这一定义 (在第 5 章中遇到过) 作为标准定义, 但在技术上有两点简化。首先, 该定义不允许有重复边 (数学上把这样的边称为平行边 (parallel edge), 包含这样边的图称为多重图 (multigraph)。其次, 该定义不允许出现顶点连接到自身的边; 这样的边称为自环 (self-loop)。有时把没有平行边或自环的图称为简单图 (simple graph)。

在形式定义中我们使用简单图, 因为这样更容易表达它们的基本性质, 而且在很多应用中不需要平行边或自环。例如, 在一个给定顶点数的简单图中, 可以限定它的边数。

性质 17.1 有 V 个顶点的图至多有 $V(V-1)/2$ 条边。

证明 在 V^2 个可能的顶点对中包含 V 个自环, 而对于不同顶点对的每条边都计算了两次, 因此, 边数至多为 $(V^2 - V)/2 = V(V-1)/2$ 。 ■

如果允许平行边, 这个限制就不能成立。因为如果一个图不是简单图, 那么它也可能由两个顶点及连接它们的数亿条边组成 (即使图中只有一个顶点, 也可能有数亿个自环)。

对于某些应用, 我们可能把消除平行边和自环看作实现中必须强调的数据处理问题。不论在本书的什么地方, 只要为方便起见使用包含平行边或自环的扩展定义来考虑一个应用或是研制一个算法时就要这样做。例如, 在 17.4 节讨论的一个经典算法中, 自环起着关键的作用; 而平行边在第 22 章所讨论的应用中也相当常见。一般来讲, 从上下文就能知道术语“图”是指“简单图”、“多重图”还是“带有自环的多重图”。

数学家们往往不加区分地交换使用顶点 (vertex) 和结点 (node) 这两个词, 不过我们一般在讨论图时使用顶点, 在讨论表示时使用结点。例如, 在 C 数据结构中就使用结点。我们一般假设顶点都有一个名字, 可以携带其他有关信息。类似地, 数学家们也用弧 (arc)、

边 (edge) 和链接 (link) 等词来描述两个顶点之间连接的抽象, 但我们统一在讨论图时使用边 (edge), 讨论 C 数据结构时使用链接 (link)。

如果一条边连接两个顶点, 就称这两个顶点是相互邻接的 (adjacent), 这条边依附于 (incident) 这两个顶点。顶点的度 (degree) 就是依附于它的边数。我们使用 $v-w$ 记法表示连接 v 和 w 的一条边; 用 $w-v$ 表示同一条边的另一种记法。

子图 (subgraph) 是组成图的边 (及相关顶点) 的一个子集。很多计算任务都涉及识别不同类型的子图。如果要识别一个图中顶点的子集, 则称该子集以及连接该子集中某两个成员的所有边为与这些顶点关联的导出子图 (induced subgraph)。

通过将顶点标记为点, 并用线连接这些点可以画出一个图。绘图能够带给我们关于图的结构直观认识; 但是这种直观性可能产生误导, 因为图的定义与表示无关。例如, 图 17-1 中的两种画法以及边的列表表示同一个图, 因为图只包含其顶点集和边 (顶点对) 集, 而且顶点集和边集均为无序的, 除此以外再无其他。虽然将图看作边集已足够, 但是我们还讨论其他的一些表示, 它们特别适合于作为 17.4 节的图数据结构的基础。

将一个给定图的顶点放在平面上, 并画出这些点和连接它们的边称为绘图 (graph drawing)。顶点的可能放置方式、边的绘制风格以及绘制中的美学约束不计其数。遵从各种自然约束的绘图算法已经得到深入研究, 而且已有很多成功的应用 (见第五部分参考文献)。例如, 最简单的约束是要求边不相交。平面图 (planar graph) 就是能在平面上画出的边不相交的图。确定一个图是否是平面的是一个很吸引人的算法学问题, 我们在 17.8 节中将会简略讨论。能够产生一种有助于可视化的表示也是一个很好的目标, 另外绘图是一个很吸引人的研究领域, 但是成功的绘图往往很难实现。顶点数和边数都有很多的图是抽象的对象, 要合适地绘制它们不太可行。

对于某些应用, 例如表示地图或电路的图, 所绘出的图可以带有很丰富的信息, 因为顶点即为平面上的点, 而且它们之间的距离是相关的。我们将这种图称为欧几里得图 (Euclidean graph)。对于其他一些应用, 例如表示关系或调度的图, 图中只含有连通性信息, 顶点的几何位置的特定信息并未提供。我们在第 20 章和第 21 章的欧几里得图中所讨论的算法将会探讨几何信息, 但我们的主要工作放在不使用任何几何信息的算法上, 并且强调图一般来说独立于绘制或计算机中的任何特定的表示。

如果从只关注连通性自身考虑, 我们可能希望顶点编号只是方便使用的标号, 而且对于两个图, 如果能够改变一个图的编号, 使得其边集等于另一个, 则认为它们是同

构的 (isomorphic)。确定两个图是否同构的问题是一个困难的计算问题 (见图 17-2 和练习 17.5), 它很有挑战性, 因为对顶点编号有 $V!$ 种可能的方式, 要穷尽所有可能性不太可能。因此, 尽管通过将同构图处理为相同的结构来减少所考虑的图结构种类的做法很诱人, 但我们很少这样做。

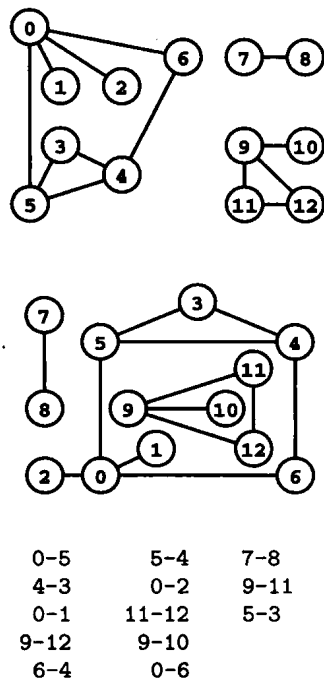


图 17-1 一个图的三种不同表示

图是由它的顶点和边来定义的, 而不是由所选择的绘图方法来定义。这两个绘图方式表示了同一个图, 下面的边列表也同样表示此图。给定图中有 13 个顶点, 编号从 0 到 12。