

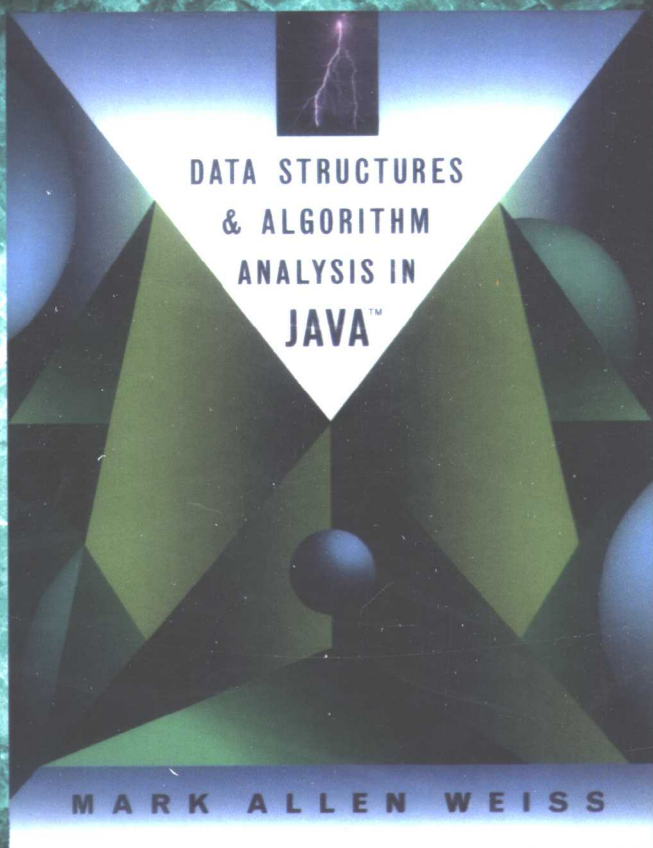


计 算 机 科 学 丛 书

数据结构与算法分析

——Java语言描述

(美) Mark Allen Weiss 著 冯舜玺 译



Data Structures
& Algorithm Analysis in Java



机械工业出版社
China Machine Press

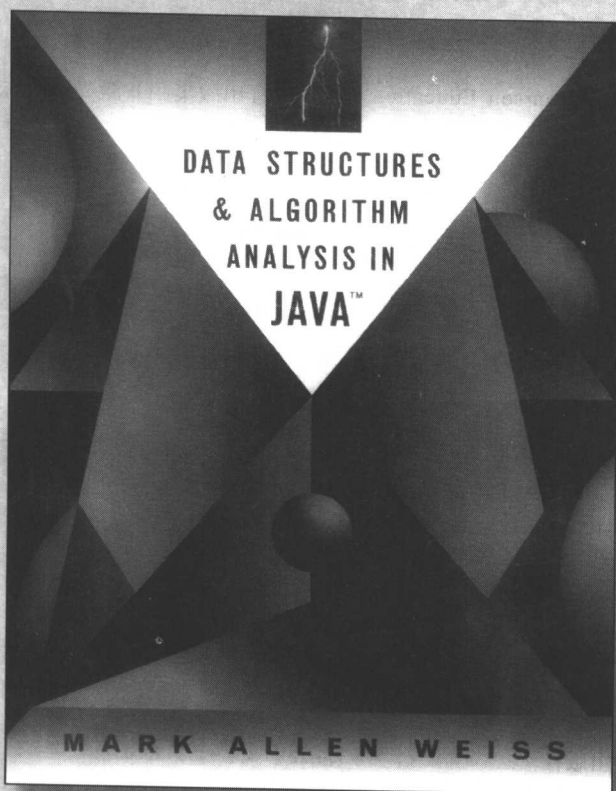


计 算 机 科 学 丛 书

数据结构与算法分析

——Java语言描述

(美) Mark Allen Weiss 著 冯舜玺 译



**Data Structures
& Algorithm Analysis in Java**



机械工业出版社
China Machine Press

本书是国外数据结构与算法分析方面的标准教材，用最卓越的Java编程语言作为实现工具讨论了数据结构（组织大量数据的方法）和算法分析（对算法运行时间的估计）。书中着重阐述了抽象数据类型的概念，并对算法的效率、性能和运行时间做了全面的分析，为读者开发高效率的程序奠定了基础。

本书可作为高级数据结构课程或者高等院校本科生、研究生算法分析课程的教材。

Simplified Chinese edition copyright © 2004 by Pearson Education Asia Limited and China Machine Press.

Original English language title: *Data Structures and Algorithm Analysis in Java* (ISBN: 0-201-35754-2) by Mark Allen Weiss, Copyright © 1999.

All rights reserved.

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

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。

本书版权登记号：图字：01-2002-6552

图书在版编目（CIP）数据

数据结构与算法分析：Java语言描述 /（美）维斯（Weiss, M. A.）著；冯舜玺译。
—北京：机械工业出版社，2004.8

（计算机科学丛书）

书名原文：Data Structures and Algorithm Analysis in Java

ISBN 7-111-14404-X

I. 数… II. ① 维… ② 冯… III. ① 数据结构 ② 算法分析 ③ Java语言—程序设计
IV. TP311.12

中国版本图书馆CIP数据核字（2004）第041900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：迟振春

北京中兴印刷有限公司印刷·新华书店北京发行所发行

2004年8月第1版第1次印刷

787mm × 1092mm 1/16 · 29.25印张

印数：0 001-5 000册

定价：40.00元

凡购本书，如有倒页、脱页、缺页，由本社发行部调换
本社购书热线：（010）68326294

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，不仅肇划了研究的范畴，还揭开了学术的源变，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、从业人员较少的现状下，美国等发达国家在其计算机科学发展的几十年间积淀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图文信息有限公司较早意识到“出版要为教育服务”。自1998年开始，华章公司就将工作重点放在了遴选、移译国外优秀教材上。经过几年的不懈努力，我们与Prentice Hall, Addison-Wesley, McGraw-Hill, Morgan Kaufmann等世界著名出版公司建立了良好的合作关系，从它们现有的数百种教材中甄选出Tanenbaum, Stroustrup, Kernighan, Jim Gray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、研究及度藏。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，有的还专诚为其书的中译本作序。迄今，“计算机科学丛书”已经出版了近百个品种，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，为进一步推广与发展打下了坚实的基础。

随着学科建设的初步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步入一个新的阶段。为此，华章公司将加大引进教材的力度，在“华章教育”的总规划之下出版三个系列的计算机教材：除“计算机科学丛书”之外，对影印版的教材，则单独开辟出“经典原版书库”；同时，引进全美通行的教学辅导书“Schaum's Outlines”系列组成“全美经典学习指导系列”。为了保证这三套丛书的权威性，同时也为了更好地为学校和老师服务，华章公司聘请了中国科学院、北京大学、清华大学、国防科技大学、复旦大学、上海交通大学、南京大学、浙江大学、中国科技大学、哈尔滨工业大学、西安交通大学、中国人民大学、北京航空航天大学、北京邮电大学、中山大学、解放军理工大学、郑州大学、湖北工学院、中国国家信息安全测评认证中心等国内重点大学和科研机构在计算机的各个领域的著名学者组成“专家指导委员会”，为我们提供选题意见和出版监督。

这三套丛书是响应教育部提出的使用外版教材的号召，为国内高校的计算机及相关专业的教学度身订造的。其中许多教材均已为M. I. T., Stanford, U.C. Berkeley, C. M. U. 等世界名牌大学所采用。不仅涵盖了程序设计、数据结构、操作系统、计算机体系结构、数据库、编译原理、软件工程、图形学、通信与网络、离散数学等国内大学计算机专业普遍开设的核心课程，而且各具特色——有的出自语言设计者之手、有的历经三十年而不衰、有的已被全世界的几百所高校采用。在这些圆熟通博的名师大作的指引之下，读者必将在计算机科学的宫殿中由登堂而入室。

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，但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。教材的出版只是我们的后续服务的起点。华章公司欢迎老师和读者对我们的工作提出建议或给予指正，我们的联系方法如下：

电子邮件: hzedu@hzbook.com

联系电话: (010) 68995264

联系地址: 北京市西城区百万庄南街1号

邮政编码: 100037

专家指导委员会

(按姓氏笔画顺序)

尤晋元	王 珊	冯博琴	史忠植	史美林
石教英	吕 建	孙玉芳	吴世忠	吴时霖
张立昂	李伟琴	李师贤	李建中	杨冬青
邵维忠	陆丽娜	陆鑫达	陈向群	周伯生
周立柱	周克定	周傲英	孟小峰	岳丽华
范 明	郑国梁	施伯乐	钟玉琢	唐世渭
袁崇义	高传善	梅 宏	程 旭	程时端
谢希仁	裘宗燕	戴 葵		

秘 书 组

武卫东

温莉芳

刘 江

杨海玲

译者序

数据结构和算法分析的重要性不仅体现在理论研究和计算机软件的设计之中，而且它和我们的日常生活及广泛的社会实践密切相关。应该说，只满足于设计实现自己意愿的程序而忽略其效率的做法是非常不可取的。人们常常发现，随着实践规模的扩大、科技水平的提高，计算机处理的数据量剧增，原本只需要几分钟或几小时就能算出结果的程序却变得成倍地耗费机时，甚至慢到令人无法容忍的地步。于是不得不回过头来重新考虑大量数据的组织和算法运行时间的评估，这实际上就是重新进行本应在设计程序之初考虑的数据结构和算法分析的工作。它说明，提高程序设计技巧和算法分析能力，培养时刻注意数据结构和算法效率的良好习惯，是开发具有高效程序的关键。可以看出，数据结构和算法分析是程序设计的重要理论基础，是计算机科学领域的核心内容之一。

本书为Mark Allen Weiss教授所著《数据结构与算法分析——C语言描述》的Java语言版。众所周知，Java语言在因特网上既畅通无阻又安全可靠，已经成为网络时代最重要的语言之一，它彻底改变了应用软件的开发模式。显而易见，使用这样一种语言解释数据结构和实现算法势在必行。此外，本书还将Java语言的基础部分和与数据结构相关的部分概要地以附录的形式列于书末，虽然简单，但却对各种Java水平的读者都具有参考价值。从内容上看，本书基本上与C语言版相同，不过，读者可以发现一些细节、实例和练习还是颇有新意的。

该书的翻译工作有幸得到姜炳坤、陶惠民、张秋华、党文运和刘玉珍诸位教授、专家的真诚关怀和热心帮助，译者表示衷心的感谢。

希望本书像其广为流传的原著一样，能够得到我国广大读者的关爱，对它的任何批评和指正都是译者热切期盼并衷心感谢的。

译者

前 言

目的/目标

本书讨论数据结构和算法分析。数据结构主要研究组织大量数据的方法，而算法分析是对算法运行时间的估计。随着计算机的速度越来越快，对于能够处理大量输入数据的程序的需求变得日益急切。可是，由于在输入量很大的时候程序的低效率现象变得非常明显，因此这又要求对效率问题给予更仔细的关注。通过在实际编程之前对算法的分析，学生可以确定一个特定的解法是否可行。例如，在本书中学生可查阅一些特定的问题并看到精心的实现方法是如何把处理大量数据的时间限制从16年减至不到1秒的。因此，若无运行时间的阐释，就不会有算法和数据结构的提出。在某些情况下，对于影响实现的运行时间的一些微小细节都需要认真地探究。

一旦确定了解法，接着就要编写程序。随着计算机功能的日益强大，它们必须解决的问题也变得更加巨大和复杂，这就要求开发更加复杂的程序。本书的目的是同时教授学生良好的程序设计技巧和算法分析能力，使得他们能够开发具有最高效率的程序。

本书适合作为高级数据结构（CS7）课程或是第一年研究生算法分析课程的教材。学生应该具有中等程度的程序设计知识，包括面向对象程序设计和递归这样一些内容，还要具有离散数学的某些知识。

处理方法

虽然本书中的内容大部分都是与语言无关的，但是，程序设计还是需要使用某种特定的语言。正如书名指出的，我们为本书选择了Java。

Java是相对较新的语言，常常用来和C++进行比较。Java具有许多优点，编程人员常常把Java看成是一种比C++更加安全、更加方便并且更容易使用的语言。因此，这使得它成为讨论和实现基础数据结构的一种优秀的核心语言。Java的其他方面，诸如线程和GUI（图形用户界面），虽然很重要，但是本书并不需要，因此也就不再讨论。

和每一种程序设计语言一样，Java也有一些缺点。它不直接支持一般的程序设计，因此需要一些变通方法，我们将在第1章讨论。Java对I/O的支持是很有限的，不过现在在Internet上有许多现成的I/O类，因此这也就不成问题了。在任何情况下，本书中的例子都最小限度地使用Java的I/O工具。

Java的优点是简化了C++的（有时是混乱的）语法，这也是一种趋势。最终用C++编码的程序员将会发现直接将Java代码转换成C++代码所暗藏的一些陷阱。为了摆脱这个问题，本书的配套网站包含有一章来描述C++并阐述C++的许多精妙之处，这对于许多读者来说并不是显而易见的。

以Java和C++二者描述的数据结构的完全版可以在因特网上得到。我们使用类似的编码方法使得这两种语言间平行的结构表现得更加明显。

内容提要

第1章包含离散数学和递归的一些复习材料。我相信掌握递归的惟一办法是反复不断地看一些好的用法。因此，除第5章外，递归遍及本书每一章的例子之中。第1章还介绍了一些材料，作为对基本Java的复习和回顾，包括对一般算法和数据结构所需的Java变通方法的讨论。

第2章讨论算法分析。这一章阐述渐近分析和它的主要弱点。这里提供了许多例子，包括对对数运行时间的深入解释。通过直观地把一些简单递归程序转变成迭代程序而对它们进行分析。此外，还介绍了更复杂的分治程序，不过有些分析（求解递归关系）要推迟到第7章再详细地进行。

第3章包括表、栈和队列。重点放在使用ADT对这些数据结构编程、这些数据结构的快速实现以及介绍它们的某些用途上。书中几乎没有完整的程序，但程序设计的许多思想体现在练习之中。

第4章讨论树，重点在查找树，包括外部查找树（B树）。UNIX文件系统和表达式树是作为例子来介绍的。本章还介绍了AVL树和伸展树。查找树的详细实现细节见第12章。树的另外一些内容，如文件压缩和博弈树，延迟到第10章讨论。外部介质上的数据结构作为几章中的最后论题来考虑。

第5章是相对较短的一章，主要讨论散列表。这里进行了某些分析，本章末尾讨论了可扩散列。

第6章是关于优先队列的。二叉堆也在这里讲授，还有些附加的材料论述优先队列某些理论上有趣的实现方法。斐波那契堆在第11章讨论，配对堆在第12章讨论。

第7章讨论排序。本章特别关注编程细节和分析，讨论并比较了所有通用的排序算法。对以下四种算法详细地进行了分析：插入排序、希尔排序、堆排序以及快速排序。本章末尾讨论了外部排序。

第8章讨论不相交集算法并证明其运行时间。这是简短且特殊的一章，如果不讨论Kruskal算法则该章可跳过。

第9章讲授图论算法。图论算法的重要性不仅因为它们在实践中经常用到，而且还因为它们的运行时间强烈地依赖于数据结构的恰当使用。实际上，所有标准算法都是和相应的数据结构、伪代码以及运行时间的分析一起介绍的。为把这些问题放进一本适当的教材中，我们对复杂性理论（包括NP完全性和不可判定性）进行了简短的讨论。

第10章通过考察一般的问题求解技巧讨论算法设计。这一章添加了大量的实例。本章及后面各章使用的伪代码可使得学生更好地理解例子，而避免被实现的细节所困扰。

第11章讨论摊还分析。对来自第4章和第6章的三种数据结构以及本章介绍的斐波那契堆进行了分析。

第12章讨论查找树算法、 k -d树和配对堆。不同于其他各章，本章给出了查找树和配对堆完整的仔细的实现。材料的安排使得教师可以把一些内容纳入到其他各章的讨论之中。例如，第12章中的自顶向下红黑树可以和（第4章的）AVL树一起讨论。

附录A列出了一些常用的Java类。附录B讨论Java的Collections包，这个标准的包（从Java 1.2起）包括了本书中讨论的几种数据结构的实现。

第1章到第9章为大多数一学期的数据结构课程提供了足够的材料。如果时间允许，那么第10章也可以包括进来。研究生的算法分析课程可以使用第7章到第11章的内容。第11章所分析的高级数据结构可以容易地在前面各章中查到。第9章中对NP完全性的讨论对这门课来说太过简单，Garey和Johnson的论NP完全性的书（有张立昂等翻译的中文译本《计算机和难解性》，科学出版社，1987——译者注）可以补充本书的不足。

练习

每章末尾提供的练习与书中讲授的内容的顺序相匹配。最后的一些练习是针对整个一章而不是针对特定的某一节来考虑的。较难的练习标以一个星号，更难的练习标有两个星号。

参考文献

参考文献位于每章的最后。一般说来，这些参考文献或者是历史性质的，代表着书中材料的原始来源，或者阐述对书中给出的结果的扩展和改进。有些文献提供了一些练习的解法。

代码的获得

书中例子的程序代码可通过匿名ftp在aw.com网站得到。这个网站也可以通过万维网来访问；其URL为<http://www.awl.com/cseng/>（从此处继续链接）。该资料的准确位置可能变化。

鸣谢

在本丛书几部著作的准备过程中，作者得到许多朋友的帮助。有些人在本书的其他版本中提到过；谢谢所有诸位。

如同往常一样，Addison-Wesley出版公司专家们的努力使得本书的写作过程更加轻松。借此机会感谢编辑Susan Hartman；副编辑Katherine Harutunian以及生产编辑Pat Unubun。我还要感谢Kris Engberg和她在Publication Services的同事，感谢她们出色地完成了本书最后的定稿任务。

我还想感谢本书的诸多审核者，他们提供了有价值的评论，其中许多已经纳入到本书之中。对于这一版，他们是：Chris Brown（罗切斯特大学），Thang N. Bui（宾夕法尼亚州立大学），G. H. Hedrick（俄克拉荷马州立大学），Sampath Kannan（宾夕法尼亚大学）以及Gurdip Singh（堪萨斯州立大学）。

最后，感谢广大的读者，他们发来电子邮件指出前面各版中的一些错误和矛盾之处。我的万维网页<http://www.cs.fiu.edu/~weiss>包含更新的（Java、C以及C++的）源代码、勘误表以及用于提交问题报告的一个链接。

Mark Allen Weiss

目 录

出版者的话	
专家指导委员会	
译者序	
前言	
第1章 引论	1
1.1 本书讨论的内容	1
1.2 数学知识复习	2
1.2.1 指数	2
1.2.2 对数	2
1.2.3 级数	3
1.2.4 模运算	4
1.2.5 证明方法	4
1.3 递归简论	6
1.4 Java中的一般对象	9
1.4.1 IntCell类	10
1.4.2 MemoryCell类	12
1.4.3 实现一般的findMax方法	13
1.5 异常	14
1.6 输入和输出	17
1.6.1 基本的流操作	17
1.6.2 StringTokenizer对象	17
1.6.3 顺序文件	18
1.7 代码的组织	20
1.7.1 包	20
1.7.2 MyInteger类	21
1.7.3 关于效率的考虑	21
小结	22
练习	22
参考文献	24
第2章 算法分析	25
2.1 数学基础	25
2.2 模型	27
2.3 要分析的问题	27
2.4 运行时间计算	29
2.4.1 一个简单的例子	30
2.4.2 一般法则	30
2.4.3 最大子序列和问题的解	32
2.4.4 运行时间中的对数	37
2.4.5 检验你的分析	40
2.4.6 分析结果的准确性	41
小结	41
练习	42
参考文献	46
第3章 表、栈和队列	47
3.1 抽象数据类型 (ADT)	47
3.2 表ADT	47
3.2.1 表的简单数组实现	48
3.2.2 链表	48
3.2.3 程序设计细节	49
3.2.4 双链表	54
3.2.5 循环链表	54
3.2.6 例子	55
3.2.7 链表的游标实现	59
3.3 栈ADT	64
3.3.1 栈模型	64
3.3.2 栈的实现	64
3.3.3 应用	69
3.4 队列ADT	75
3.4.1 队列模型	75
3.4.2 队列的数组实现	75
3.4.3 队列的应用	79
小结	80
练习	80
第4章 树	85

4.1 预备知识	85	6.2 一些简单的实现	156
4.1.1 树的实现	86	6.3 二叉堆	156
4.1.2 树的遍历及应用	87	6.3.1 结构性质	156
4.2 二叉树	89	6.3.2 堆序性质	157
4.2.1 实现	90	6.3.3 基本的堆操作	158
4.2.2 一个例子: 表达式树	90	6.3.4 其他的堆操作	162
4.3 查找树ADT——二叉查找树	93	6.4 优先队列的应用	164
4.3.1 find	95	6.4.1 选择问题	165
4.3.2 findMin和findMax	95	6.4.2 事件模拟	166
4.3.3 insert	96	6.5 d -堆	167
4.3.4 remove	97	6.6 左式堆	167
4.3.5 平均情形分析	99	6.6.1 左式堆性质	167
4.4 AVL树	101	6.6.2 左式堆操作	168
4.4.1 单旋转	102	6.7 斜堆	173
4.4.2 双旋转	105	6.8 二项队列	175
4.5 伸展树	110	6.8.1 二项队列结构	175
4.5.1 一个简单的想法 (不能直接使用)	111	6.8.2 二项队列操作	176
4.5.2 展开	112	6.8.3 二项队列实现	178
4.6 树的遍历	117	小结	182
4.7 B树	118	练习	183
小结	122	参考文献	187
练习	123	第7章 排序	189
参考文献	128	7.1 预备知识	189
第5章 散列	131	7.2 插入排序	189
5.1 一般想法	131	7.2.1 算法	189
5.2 散列函数	131	7.2.2 插入排序的分析	190
5.3 分离链接法	133	7.3 一些简单排序算法的下界	190
5.4 开放定址法	136	7.4 希尔排序	191
5.4.1 线性探测法	137	7.5 堆排序	194
5.4.2 平方探测法	138	7.6 归并排序	197
5.4.3 双散列法	143	7.7 快速排序	202
5.5 再散列	144	7.7.1 选取枢纽元	202
5.6 可扩散列	145	7.7.2 分割策略	204
小结	148	7.7.3 小数组	206
练习	149	7.7.4 实际的快速排序例程	206
参考文献	152	7.7.5 快速排序的分析	208
第6章 优先队列 (堆)	155	7.7.6 选择问题的线性期望时间算法	210
6.1 模型	155	7.8 排序算法的一般下界	212

7.9 桶式排序	214	9.6.3 欧拉回路	275
7.10 外部排序	214	9.6.4 有向图	278
7.10.1 为什么需要新算法	214	9.6.5 查找强分支	279
7.10.2 外部排序模型	214	9.7 NP完全性介绍	281
7.10.3 简单算法	215	9.7.1 难与易	281
7.10.4 多路合并	216	9.7.2 NP类	282
7.10.5 多相合并	217	9.7.3 NP完全问题	283
7.10.6 替换选择	217	小结	284
小结	218	练习	285
练习	219	参考文献	291
参考文献	223	第10章 算法设计技巧	295
第8章 不相交集ADT	227	10.1 贪婪算法	295
8.1 等价关系	227	10.1.1 一个简单的调度问题	296
8.2 动态等价性问题	227	10.1.2 哈夫曼编码	298
8.3 基本数据结构	229	10.1.3 近似装箱问题	302
8.4 灵巧求并算法	232	10.2 分治算法	309
8.5 路径压缩	233	10.2.1 分治算法的运行时间	309
8.6 按秩求并和路径压缩的最坏情形	235	10.2.2 最近点问题	311
8.7 一个应用	240	10.2.3 选择问题	314
小结	242	10.2.4 一些算术问题的理论改进	317
练习	242	10.3 动态规划	320
参考文献	243	10.3.1 用表代替递归	320
第9章 图论算法	245	10.3.2 矩阵乘法的顺序安排	322
9.1 若干定义	245	10.3.3 最优二叉查找树	325
9.2 拓扑排序	247	10.3.4 所有点对最短路径	327
9.3 最短路径算法	250	10.4 随机化算法	329
9.3.1 无权最短路径	251	10.4.1 随机数发生器	330
9.3.2 Dijkstra算法	254	10.4.2 跳跃表	333
9.3.3 具有负边值的图	259	10.4.3 素性测试	335
9.3.4 无圈图	260	10.5 回溯算法	337
9.3.5 所有顶点对最短路径	262	10.5.1 收费公路重建问题	338
9.4 网络流问题	262	10.5.2 博弈	341
9.5 最小生成树	266	小结	347
9.5.1 Prim算法	267	练习	347
9.5.2 Kruskal算法	269	参考文献	353
9.6 深度优先搜索的应用	271	第11章 摊还分析	359
9.6.1 无向图	271	11.1 一个无关的智力问题	359
9.6.2 双连通性	272	11.2 二项队列	360

11.3 斜堆	363	12.2.2 自顶向下红黑树	386
11.4 斐波那契堆	365	12.2.3 自顶向下删除	390
11.4.1 切除左式堆中的节点	366	12.3 确定性跳跃表	391
11.4.2 二项队列的懒惰合并	368	12.4 AA树	396
11.4.3 斐波那契堆操作	370	12.5 treap树	401
11.4.4 时间界的证明	371	12.6 k-d树	404
11.5 伸展树	373	12.7 配对堆	406
小结	376	小结	411
练习	376	练习	411
参考文献	377	参考文献	414
第12章 高级数据结构及其实现	379	附录A 一些库例程	417
12.1 自顶向下伸展树	379	附录B Collections 类库	429
12.2 红黑树	383	索引	445
12.2.1 自底向上插入	385		

第1章 引 论

在这一章，我们阐述本书的目的和目标并简要复习离散数学以及程序设计的一些概念。我们将要：

- 看到程序在合理的大量输入下的运行性能与其在适量输入下的运行性能具有同等重要性。
- 概括本书其余部分所需要的基本的数学基础。
- 简要复习递归。
- 概括用于本书的Java语言的某些重要特点。

1.1 本书讨论的内容

设有一组 N 个数而要确定其中第 k 个最大者，我们称之为选择问题 (selection problem)。大多数学习过一两门程序设计课程的学生写一个解决这种问题的程序不会有什么困难，“显而易见的”解决方法是相当多的。

该问题的一种解法就是将这 N 个数读进一个数组中，再通过某种简单的算法，比如冒泡排序法，以递减顺序将数组排序，然后返回位置 k 上的元素。

稍微好一点的算法可以先把前 k 个元素读入数组并对其排序（以递减的顺序）。接着，将剩下的元素再逐个读入。当新元素被读到时，如果它小于数组中的第 k 个元素则忽略，否则就将其放到数组中正确的位置上，同时将数组中的一个元素挤出数组。当算法终止时，位于第 k 个位置上的元素作为答案返回。

这两种算法的编码都很简单，建议读者试一试。此时我们自然要问：哪个算法更好？哪个算法更重要？还是两个算法都足够好？使用含有100万个元素的随机文件，在 $k=500\ 000$ 的条件下进行模拟发现，两个算法在合理的时间量内均不能结束；每种算法都需要计算机处理若干天才能算完（虽然最后还是给出了正确的答案）。在第7章将讨论另一种算法，该算法将在1秒钟左右给出问题的解。因此，虽然我们提出的两个算法都能算出结果，但是它们不能被认为是好的算法，因为对于第三种算法能够在合理的时间处理的输入数据量而言，这两种算法是完全不切实际的。

第二个问题是解决一个流行的字谜。输入由一些字母和单词的二维数组组成。目标是要找出字谜中的单词，这些单词可能是水平、垂直或沿对角线以任何方向放置的。作为例子，图1-1所示的字谜由单词this、two、fat和that组成。单词this从第一行第一列的位置即(1, 1)处开始并延伸至(1, 4)；单词two从(1, 1)到(3, 1)；fat从(4, 1)到(2, 3)；而that则从(4, 4)到(1, 1)。

	1	2	3	4
1	t	h	i	s
2	w	a	t	s
3	o	a	h	g
4	f	g	d	t

图1-1 字谜示例

现在至少有两种直观的算法来求解这个问题。对单词表中的每个单词，我们检查每一个

有序三元组（行，列，方向），验证是否有单词存在。这需要大量嵌套的for循环，但它基本上是直观的算法。

也可以这样，对于每一个尚未进行到字谜最后的有序四元组（行，列，方向，字符数），我们可以测试所指的单词是否在单词表中。这也导致使用大量嵌套的for循环。如果任意单词中的最大字符数已知，那么该算法有可能节省一些时间。

上述两种方法相对来说都不难编码并可求解通常发表于杂志上的许多现实的字谜游戏。这些字谜通常有16行16列以及40个左右的单词。然而，假设我们把字谜变成为只给出迷板（puzzle board）而单词表基本上是一本英语词典，则上面提出的两种解法需要相当可观的时间来解决这个问题，因此这两种方法都是不可接受的。不过，这样的问题还是有可能在数秒内解决的，甚至单词表可以很大。

在许多问题中，一个重要的观念是：写出一个可以工作的程序并不够。如果这个程序在巨大的数据集上运行，那么运行时间就变成了重要的问题。我们将在本书中看到对于大量的输入，如何估计程序的运行时间，尤其是在如何在尚未具体编码的情况下比较两个程序的运行时间。我们还将看到彻底改进程序速度以及确定程序瓶颈的方法。这些方法将使我们能够找到需要着重优化的那些代码段。

1.2 数学知识复习

这一节列出一些需要记忆或是能够推导出的基本公式，并复习基本的证明方法。

1.2.1 指数

$$X^A X^B = X^{A+B}$$

$$\frac{X^A}{X^B} = X^{A-B}$$

$$(X^A)^B = X^{AB}$$

$$X^N + X^N = 2X^N \neq X^{2N}$$

$$2^N + 2^N = 2^{N+1}$$

1.2.2 对数

在计算机科学中，除非有特别的声明，否则所有的对数都是以2为底的。

定义 $X^A = B$ 当且仅当 $\log_X B = A$ 。

由该定义可以得到几个方便的等式。

定理1.1

$$\log_A B = \frac{\log_C B}{\log_C A}; A, B, C > 0, A \neq 1$$

证明 令 $X = \log_C B$, $Y = \log_C A$ 以及 $Z = \log_A B$ 。此时由对数的定义得， $C^X = B$, $C^Y = A$ 以及 $A^Z = B$ 。联合这三个等式则产生 $(C^Y)^Z = C^X = B$ 。因此， $X = YZ$ ，这意味着 $Z = X/Y$ ，定理得证。

□

定理1.2

$$\log AB = \log A + \log B; A, B > 0$$

证明 令 $X = \log A$, $Y = \log B$ 以及 $Z = \log AB$ 。此时由于假设默认的底为2, $2^X = A$, $2^Y = B$ 及 $2^Z = AB$ 。联合最后的三个等式则有 $2^X 2^Y = 2^Z = AB$ 。因此 $X + Y = Z$, 这就证明了该定理。□

其他一些有用的公式如下, 它们都能够用类似的方法推导。

$$\log A/B = \log A - \log B$$

$$\log(A^B) = B \log A$$

$$\log X < X \text{ (对所有的 } X > 0 \text{ 成立)}$$

$$\log 1 = 0, \log 2 = 1, \log 1024 = 10, \log 1048576 = 20$$

1.2.3 级数

最容易记忆的公式是:

$$\sum_{i=0}^N 2^i = 2^{N+1} - 1$$

和

$$\sum_{i=0}^N A^i = \frac{A^{N+1} - 1}{A - 1}$$

在第二个公式中, 如果 $0 < A < 1$, 则

$$\sum_{i=0}^N A^i < \frac{1}{1 - A}$$

当 N 趋于 ∞ 时该和趋向于 $1/(1-A)$, 这些公式是“几何级数”公式。

可以用下面的方法推导关于 $\sum_{i=0}^{\infty} A^i$ ($0 < A < 1$) 的公式。令 S 是其和, 此时

$$S = 1 + A + A^2 + A^3 + A^4 + A^5 + \dots$$

于是

$$AS = A + A^2 + A^3 + A^4 + A^5 + \dots$$

如果将这两个等式相减 (这种运算只能对收敛级数进行), 等号右边所有的项相消, 只留下1:

$$S - AS = 1$$

这就是说

$$S = \frac{1}{1 - A}$$

可以用相同的方法计算 $\sum_{i=0}^{\infty} i/2^i$, 它是一个经常出现的和。我们写成

$$S = \frac{1}{2} + \frac{2}{2^2} + \frac{3}{2^3} + \frac{4}{2^4} + \frac{5}{2^5} + \dots$$

用2乘之得到