

TURING

图灵原版计算机科学系列

Addison
Wesley

Data Structures and Problem Solving Using Java

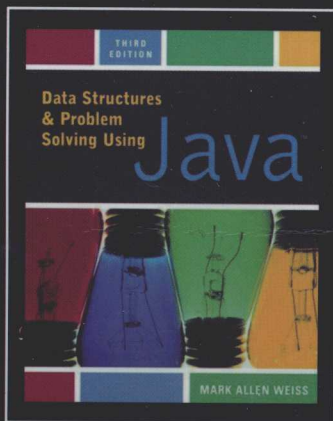
Third Edition

数据结构与问题求解

Java 语言描述

(英文版 · 第3版)

[美] Mark Allen Weiss 著



人民邮电出版社
POSTS & TELECOM PRESS



图灵原版计算机科学系列

Data Structures and Problem Solving Using Java
Third Edition

数据结构与问题求解

Java语言描述

(英文版·第3版)

[美] Mark Allen Weiss 著

江苏工业学院图书馆
藏书章

人民邮电出版社
北京

图书在版编目 (CIP) 数据

数据结构与问题求解: Java 语言描述: 第 3 版: 英文 / (美) 维斯 (Weiss, M.A.) 著.
—北京: 人民邮电出版社, 2007.8
(图灵原版计算机科学系列)

ISBN 978-7-115-16245-8

I. 数... II. 维... III. ①数据结构—教材—英文②JAVA 语言—程序设计—教材—英文
IV. TP311.12 TP312

中国版本图书馆 CIP 数据核字 (2007) 第 071336 号

内 容 提 要

本书从讲解什么是数据结构开始, 延伸至高级数据结构和算法分析, 强调数据结构和问题求解技术。本书的目的是从抽象思维和问题求解的观点提供对数据结构的实用介绍, 试图包含有关数据结构、算法分析及其 Java 实现的所有重要的细节。作者采用了独特的方法将数据结构分成说明和实现两部分, 并充分利用了已有的数据结构库 (Java 集合类 API)。本书分为 4 个部分: 第一部分讨论适合大多数应用的集合类 API 的一个子集, 并覆盖基本的算法分析技术、递归和排序算法; 第二部分包含了一组集合类 API 的应用实例; 第三部分讨论数据结构的实现; 第四部分描述了高级的数据结构, 如伸展树、偶堆和不相交集数据结构。

本书适合作为本科生数据结构课程或研究生算法分析课程的教材。教师可以灵活地选择本书的内容, 选择最适合对应课程的内容授课。

图灵原版计算机科学系列

数据结构与问题求解: Java 语言描述 (英文版·第 3 版)

- ◆ 著 [美] Mark Allen Weiss
责任编辑 杨海玲
- ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
三河市海波印务有限公司印刷
新华书店总店北京发行所经销
- ◆ 开本: 800 × 1000 1/16
印张: 40.75
字数: 845 千字 2007 年 8 月第 1 版
印数: 1—3 000 册 2007 年 8 月河北第 1 次印刷

著作权合同登记号 图字: 01-2007-2464 号

ISBN 978-7-115-16245-8/TP

定价: 65.00 元

读者服务热线: (010)88593802 印装质量热线: (010)67129223

版 权 声 明

Original edition, entitled *Data Structures and Problem Solving Using Java, Third Edition*, 0321322134 by Mark Allen Weiss, published by Pearson Education, Inc., publishing as Addison-Wesley, Copyright © 2006 by 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.

China edition published by POSTS & TELECOM PRESS Copyright © 2007.

This edition is manufactured in the People's Republic of China, and is authorized for sale only in the People's Republic of China excluding Hong Kong, Macao and Taiwan.

本书英文版由 Pearson Education Asia Ltd. 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或抄袭本书内容。

仅限于中华人民共和国境内（香港、澳门特别行政区和台湾地区除外）销售发行。

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

本书是为计算机专业的两学期课程而设计的，从讲述什么是数据结构开始，延伸至高级数据结构和算法分析。

数据结构课程的内容已经经过了若干年的演变，尽管对于它所覆盖的内容有一些共识，但在细节问题上还有大量的分歧。大家都接受的一个主题是软件开发的原理，最主要的是封装和信息隐藏的概念。从算法方面来看，所有的数据结构课程都趋向于包括运行时间分析、递归、基本排序算法和基本数据结构的介绍。许多大学还提供高级的课程，在更高的层次上讨论数据结构、算法和运行时间分析的问题。本书的内容是为这两个层次的课程设计的，这样就不必要购买第二本教材。

虽然在数据结构领域最激烈的争论都围绕着程序设计语言的选择，但还需要做出如下一些基本选择。

- 是否要尽早引入面向对象的设计或基于对象的设计。
- 数学上的严密程度。
- 在数据结构的实现和使用之间的适当平衡。
- 与所选语言相关的程序设计细节问题（如是否应该较早地使用 GUI）。

我写本书的目的是从抽象思维和问题求解的观点提供对数据结构的实用介绍。我试图覆盖有关数据结构、它们的分析以及它们的Java实现的所有重要细节，而不是停留在理论上很有趣但没有被广泛使用的数据结构。想在一个学期的课程中讲完本书中所有不同数据结构的使用与分析是不可能的，因此本书允许教师灵活地选择所教授的内容。教师需要确定在实践和理论之间的适当平衡，然后选择最适合课程的内容。正如前言后面讨论的那样，我按照尽量减少各章之间依赖性的原则来组织本书。

独特的方法

我最基本的前提是所有语言的软件开发工具都含有较大的库，许多数据结构都是这些库的一部分。我预测数据结构课程的重点最终将会从实现转向使用。本书采用独特的方法将数据结构分成说明和实现两部分，并利用已有的数据结构库：Java 集合类（Collections）API。

第一部分的第2章中完整地讨论了集合类API中适合于大多数应用的一个子集。第一部分也覆盖了基本的分析技术、递归和排序。第二部分包含了集合类API中数据结构的一组应用。直到第三部分才讨论集合类API的实现，而这时这些数据结构早已用到了。因为集合类API是Java的一部分，因此学生可以尽早用现成的软件组件设计大的项目。

尽管本书中有许多集合类 API 的应用，但它既不是关于集合类 API 的书，也不是关于集合类 API 具体实现的初级教材，它仍然是一本强调数据结构和基本问题求解技术的书。当然，数据结构的设计中所用的通用技术在集合类 API 的实现中都可用，因此第三部分中有几章包括了集合类 API 的实现。不过教师可以在第三部分中选择较简单的实现，以避免讨论集合类 API 的协议。第 2 章介绍了集合类 API，它是理解第二部分中的代码的基础。本书试图只用集合类 API 的基本部分。

许多教师宁愿采用更传统的方法，即首先定义、实现每种数据结构，然后才是应用。因为第二部分和第三部分的材料之间没有依赖性，因此传统的课程也能采用本书。

预备知识

使用本书的学生应该具备面向对象或面向过程的程序设计语言的知识。基本的知识包括基本数据类型、运算符、控制结构、函数（方法）以及输入和输出（但不一定要知道数组和类）。

离散数学的知识也是非常有用的，但不是绝对的预备知识。书中给出了一些数学证明，但更复杂的证明要通过复习数学知识来实现。第 3 章和第 15 章~第 20 章需要一定程度的数学水平。教师可以简单地跳过数学证明部分，而只说明结果。本书的所有证明都清晰地标记出来，与正文分开。

第 3 版所做修改小结

(1) 本书的代码使用泛型（generics）进行了完全重写，泛型是 Java 5 引入的一种技术。代码还大量使用了增强的 for 循环和自动装箱。

(2) 在 Java 5 中，优先级队列是标准集合类 API 的一部分。这个修改反映在第 17 章的讨论和第二部分的某些代码中。

本书的结构

第一部分讨论大 O 表示法和算法范型，包括递归和随机化。我用一章介绍排序，另一章描述基本数据结构。用集合类 API 讲述数据结构的接口和运行时间。至此，教师可以有几种方法介绍剩余的内容，具体包括下面两种。

(1) 在第三部分中，当描述每个数据结构时讨论对应的实现（集合类 API 的版本或者较简单的版本）。教师可以要求学生以各种方式扩展类，如习题中所建议的。

(2) 先说明如何使用集合类 API 的每个类，在课程的稍后部分再介绍实现。第二部分中的实例研究可以用来支持这种方法。因为完整的实现在每个较新的 Java 编译器上都可使用，教师可以在程序设计项目中使用集合类 API。这种方式的详细介绍将在稍后讨论。

第四部分描述了高级的数据结构，如伸展树、偶堆和不相交集数据结构。如果时间允许的话，可以讲述这些内容，或者将其作为下一门课程的内容。

逐章内容的组织

第一部分集中讲基本的算法和构件块。第1章提供了时间复杂度和大O表示法的完整讨论，还讨论和分析了二分搜索。第2章非常重要，因为它覆盖了集合类API，并直观地论证了每个数据结构所支持的操作的运行时间是多少（在第三部分讨论这些数据结构的实现，包括集合类API风格的版本和简化的版本）。这一章还讨论了迭代器模式以及嵌套、局部和匿名类。内部类推迟到第三部分，作为实现技术讨论。第3章讨论递归，首先介绍归纳法证明，然后讨论分治、动态规划和回溯。有一节专门描述一些递归的数值计算算法，这些算法可用来实现RSA密码系统。对许多学生来讲，第3章后半部分的材料更适合在后续课程中学习。第4章对一些基本的排序算法进行了描述、编码和分析，包括插入排序、谢尔排序、归并排序和快速排序，以及间接排序。它也证明了排序算法的标准下界，并讨论了与选择相关的问题。第5章讨论了随机数，包括它们的生成和在随机化算法中的使用。

第二部分提供了一些实例研究，每一章围绕一个主题。第6章通过游戏介绍了一些重要技术。第7章通过平衡符号检查算法和标准运算符优先级解析算法讨论了计算机语言中栈的使用，而且提供了两个算法的完整实现代码。第8章讨论了文件压缩和交叉引用生成的基本实用程序。第9章泛泛地介绍了模拟的问题，首先介绍了一个可视为模拟的问题，然后介绍了更一般的事件驱动模拟。最后，第10章说明如何用数据结构来有效地实现图中的一些最短路径算法。

第三部分介绍数据结构的实现。第11章讨论了作为一种实现技术的内部类，并说明它们在ArrayList实现中的使用。在第三部分的剩余章节中提供了使用简单协议（各种insert、find和remove）的实现。在某些情况下，提供了使用更复杂Java语法的集合类API实现（复杂性还源自所需操作的数量较大）。在这一部分用到了一些数学知识，特别是第15章~第17章，教师可以根据教学情况跳过这些内容。第12章提供了栈和队列的实现：首先用扩展数组来实现，然后用链表来实现，最后讨论集合类API的版本。第13章讨论通用链表。单链表用一个简单的协议来说明，使用双向链表的更复杂的集合类API版本的实现在该章的最后讨论。第14章描述了树，并介绍了基本的遍历方式。第15章非常具体，它提供了二叉查找树的一些实现。一开始说明了基本的二叉查找树，然后延伸出支持顺序统计量的二叉查找树。该章讨论了AVL树但没有给出其实现，不过实现了更实用的红黑树和AA树。然后是集合类API中TreeSet和TreeMap的实现，最后研究了B树。第16章讨论了散列表，并在研究了一个较简单的可选方案后，介绍了作为HashSet和HashMap一部分的二次探测法的实现。第17章描述了二叉堆，并研究了堆排序和外排序。Java 5的集合类API中有优先级队列，因此我们实现了标准的版本。

第四部分的内容适用于更高级的课程或作为一般的参考，其中的算法对一年级的学生也适用。但为了完备性，这一部分包含了超出一年级学生理解范围的复杂数学分析。第18章描述了伸展树，它是一种在实际应用中效果非常好的二叉查找树，在某些需要优先级队列的应用中，它是二叉堆的竞争者。第19章描述了支持归并操作的优先级队列，并提供了偶堆的实现。最后，第20章研究了不相交集的标准数据结构。

章之间的依赖性

一般来讲,大多数章是互相独立的,下面是一些需要注意的依赖关系。

- 第 1 章 这一章应该在第 2 章和第 4 章之前介绍。递归(第 3 章)可以在这一章前面讲,但是,这样教师就要忽略掉某些避免低效率递归的细节问题。
- 第 2 章 这一章可以在第二部分或第三部分之前讲,也可以和它们结合在一起讲。
- 第 3 章 3.1 节~3.3 节的内容应该在讨论递归排序算法、树、Tic-Tac-Toe 实例研究和最短路径算法之前讲,而 RSA 密码系统、动态规划和回溯(须先讨论 Tic-Tac-Toe)可以选讲。
- 第 4 章 这一章应该在第 1 章和第 3 章之后讲,不过没有第 1 章和第 3 章也能介绍谢尔排序。谢尔排序不是递归的(因此不需要第 3 章),其运行时间的严格分析太复杂了,超出了本书的范围(因此几乎不需要第 1 章)。
- 第 11 章 这部分材料应该在集合类 API 的实现之前讨论。
- 第 12 章和第 13 章 这两章可以交换次序讲授。不过我宁愿先讲第 12 章,因为我相信它们比链表简单。
- 第 14 章和第 15 章 这两章可以交换次序讲授或同时讲授。

独立部分

其余部分几乎没有或完全没有依赖关系。

- 第 5 章 有关随机数的部分可以在需要时介绍。
- 第二部分 第 6 章~第 10 章可以与集合类 API 一起讲,也可以在集合类 API 后面讲,这几章本身可以是任意次序。这一部分有一些对前面几章的引用,包括 6.2 节参考了 3.7 节的讨论,8.2 节参考了 7.1 节中类似的词法分析代码。
- 第 16 章和第 17 章 这两章可以在任何时候讲。
- 第四部分 第 18 章~第 20 章的内容是独立的,通常应包含在后续课程中。

数学

我试图为强调理论的数据结构课程和需要更多分析的后续课程提供数学上的严密性。这些内容以独立的定理的形式(在某些情况下,是以独立的节或小节的形式)与书的正文相区分。在不强调理论的课程中,教师可以跳过这些内容。

在所有的情况下,定理的理解并不需要定理的证明。这是接口(定理的陈述)和它的实现(证明)分开的另一个实例。某些数学内容(如 3.4 节)可以跳过,不会影响其他部分的理解。

课程的组织

教授这门课的一个关键问题是确定如何使用第一部分到第三部分的材料。第 1 章讨论大 O 表示法。有一个习题要求学生写一个短程序并将运行时间与分析结果相比较,它可以用来测试学生的

理解程度。

第 2 章的关键的思想是以分类的方式说明不同的数据结构能够以不同的效率支持不同的访问模式。任何实例研究（除了使用递归的 Tic-Tac-Toe）都可以用来说明数据结构的应用。这样学生能了解数据结构以及它们是如何使用的，但不知道它们是如何有效地实现的。这就是本书与传统方法的区别所在。以这种方式观察事物将极大地提高学生抽象思维的能力。学生还可以提供某些集合类 API 组件的简单实现（第 2 章中的习题给出了一些建议），并看到已有的集合类 API 的高效数据结构实现和他们自己写的低效数据结构实现之间的区别。也可以要求学生扩充实例研究，但他们同样不需要知道数据结构的任何细节。

数据结构的有效实现可以在后面讨论，递归可以在教师认为合适的任何时候介绍，但是必须在二叉查找树以前。排序的细节可以在递归之后的任何时候讨论。这时，可以用同样的实例研究，根据对数据结构实现的修改进行实验，继续我们的课程。例如，学生可以用不同形式的平衡二叉查找树做实验。

选择更传统的方法的教师可以在讨论了第三部分中数据结构的实现后，再讨论第二部分中的实例研究。再重申一次，本书各章的安排尽可能设计成了互相独立的。



习题

习题有各种形式，我提供了四种。基本的简答题是一些简单的问题，或需要对本书描述的算法做手工的模拟。理论题考查一些需要数学分析或者理论上具有比较有趣的解的问题。实践题包含一些简单的程序设计问题，包括有关语法或某些有技巧的代码行的问题。最后，程序设计题包含一些扩展性习题。

教学特色

- 边栏的注释（加灰底的段落）用来突出要点。
- 关键概念列出重要的术语及它们的定义。
- 每章最后的常见错误列出了经常容易犯的错误。
- 在大部分章的最后提供了可供进一步阅读的参考文献。

补充材料

本书有各种可用的补充材料。对本书的所有读者，下面的资源可以在 <http://www.aw-bc.com/cssupport> 找到。



- 本书的源代码文件。（在每章后面的网上资源列出了该章代码的文件名。）¹

1. 本书对英文原版进行了改编及删减，删掉了原版中的前 4 章讲述 Java 语言的内容，所以，读者下载源代码文件时要注意本书与原版各章的对应关系：第 1 章对应原版第 5 章，第 2 章对应原版第 6 章，依次类推。

- 本书中所有图的 PowerPoint 幻灯片。

此外，通过验证的教师可以使用以下的补充材料。访问 <http://www.aw-bc.com/computing>，并按英文书名 *Data Structures and Problem Solving Using Java* 进行搜索，一旦到达了本书的目录页，选择 Instructor Resources 链接即可。

- *Instructor's Guide* 给出了获得材料的几种途径。材料包括测试题的实例、作业、教学大纲以及部分习题的答案。

致谢

在完成本书的过程中，许许多多人都给予了我帮助。在以前版本和相关 C++ 版本中我已向他们表示了谢意。但还有很多人给我发过电子邮件指出错误或解释的不一致，这些问题我已经在这个版本中改正了，但无法一一向他们致谢。

对本版，我要感谢 Addison-Wesley 的朋友们，他们是本书的编辑 Michael Hirsch、项目编辑 Juliet Silveri 和 Gillian Hall，以及市场部的 Michelle Brown。同时也感谢文字编辑 Penelope Hull、校对 Holly McLean-Aldis，以及封面设计 Joyce Wells。

本书的某些材料是根据我的教材 *Efficient C Programming: A Practical Approach* (Prentice Hall, 1995) 改编的，并得到了出版者的允许。我已经在章后面的适当位置包含了该参考文献。

我的主页 <http://www.cs.fiu.edu/~weiss> 将包含更新的源代码、勘误表以及用来接受错误报告的链接。

M.A.W

于佛罗里达州迈阿密市

contents

part one Algorithms and Building Blocks

chapter 1 algorithm analysis

3

1.1	what is algorithm analysis?	4
1.2	examples of algorithm running times	7
1.3	the maximum contiguous subsequence sum problem	8
1.3.1	the obvious $O(N^3)$ algorithm	9
1.3.2	an improved $O(N^2)$ algorithm	12
1.3.3	a linear algorithm	13
1.4	general big-oh rules	14
1.5	the logarithm	19
1.6	static searching problem	21
1.6.1	sequential search	21
1.6.2	binary search	21
1.6.3	interpolation search	24
1.7	checking an algorithm analysis	25
1.8	limitations of big-oh analysis	26
	summary	27
	key concepts	27
	common errors	28
	on the internet	28
	exercises	29
	references	34

chapter 2 the collections api**37**

- 2.1 introduction** 38
- 2.2 the iterator pattern** 39
 - 2.2.1 basic iterator design 40
 - 2.2.2 inheritance-based iterators and factories 41
- 2.3 collections api: containers and iterators** 43
 - 2.3.1 the Collection interface 43
 - 2.3.2 Iterator interface 46
- 2.4 generic algorithms** 48
 - 2.4.1 Comparator function objects 49
 - 2.4.2 the Collections class 49
 - 2.4.3 binary search 52
 - 2.4.4 sorting 52
- 2.5 the List interface** 52
 - 2.5.1 the ListIterator interface 54
 - 2.5.2 LinkedList class 55
- 2.6 stacks and queues** 58
 - 2.6.1 stacks 58
 - 2.6.2 stacks and computer languages 59
 - 2.6.3 queues 60
 - 2.6.4 stacks and queues in the collections api 60
- 2.7 sets** 61
 - 2.7.1 the TreeSet class 62
 - 2.7.2 the HashSet class 63
- 2.8 maps** 68
- 2.9 priority queues** 70
- summary** 73
- key concepts** 74
- common errors** 75
- on the internet** 75
- exercises** 76
- references** 79

chapter 3 recursion**81**

- 3.1 what is recursion?** 81
- 3.2 background: proofs by mathematical induction** 82
- 3.3 basic recursion** 84
 - 3.3.1 printing numbers in any base 86
 - 3.3.2 why it works 88
 - 3.3.3 how it works 89

3.3.4	too much recursion can be dangerous	90
3.3.5	preview of trees	91
3.3.6	additional examples	92
3.4	numerical applications	96
3.4.1	modular arithmetic	97
3.4.2	modular exponentiation	98
3.4.3	greatest common divisor and multiplicative inverses	99
3.4.4	the rsa cryptosystem	101
3.5	divide-and-conquer algorithms	103
3.5.1	the maximum contiguous subsequence sum problem	104
3.5.2	analysis of a basic divide-and-conquer recurrence	107
3.5.3	a general upper bound for divide-and-conquer running times	110
3.6	dynamic programming	112
3.7	backtracking	116
	summary	119
	key concepts	120
	common errors	121
	on the internet	121
	exercises	122
	references	126

chapter 4 sorting algorithms

127

4.1	why is sorting important?	128
4.2	preliminaries	129
4.3	analysis of the insertion sort and other simple sorts	129
4.4	shellsort	132
4.4.1	performance of shellsort	134
4.5	mergesort	135
4.5.1	linear-time merging of sorted arrays	136
4.5.2	the mergesort algorithm	137
4.6	quicksort	139
4.6.1	the quicksort algorithm	140
4.6.2	analysis of quicksort	142
4.6.3	picking the pivot	145
4.6.4	a partitioning strategy	146
4.6.5	keys equal to the pivot	148
4.6.6	median-of-three partitioning	148
4.6.7	small arrays	149
4.6.8	java quicksort routine	150
4.7	quickselect	152

4.8 a lower bound for sorting 154

summary 155

key concepts 155

common errors 156

on the internet 156

exercises 156

references 160

chapter 5 randomization**161****5.1 why do we need random numbers? 161****5.2 random number generators 162****5.3 nonuniform random numbers 167****5.4 generating a random permutation 169****5.5 randomized algorithms 170****5.6 randomized primality testing 173**

summary 176

key concepts 176

common errors 177

on the internet 178

exercises 178

references 180

part two Applications**chapter 6 fun and games****183****6.1 word search puzzles 183**

6.1.1 theory 184

6.1.2 java implementation 185

6.2 the game of tic-tac-toe 189

6.2.1 alpha-beta pruning 189

6.2.2 transposition tables 192

6.2.3 computer chess 197

summary 198

key concepts 198

common errors 198

on the internet 198

exercises 199

references 200

chapter 7 stacks and compilers**201****7.1 balanced-symbol checker 201**

7.1.1 basic algorithm 202

7.1.2 implementation 203

7.2 a simple calculator 212

7.2.1 postfix machines 213

7.2.2 infix to postfix conversion 213

7.2.3 implementation 216

7.2.4 expression trees 222

summary 224**key concepts 225****common errors 225****on the internet 225****exercises 226****references 227****chapter 8 utilities****229****8.1 file compression 229**

8.1.1 prefix codes 230

8.1.2 huffman's algorithm 232

8.1.3 implementation 235

8.2 a cross-reference generator 248

8.2.1 basic ideas 248

8.2.2 java implementation 248

summary 252**key concepts 252****common errors 253****on the internet 253****exercises 253****references 256****chapter 9 simulation****257****9.1 the josephus problem 257**

9.1.1 the simple solution 258

9.1.2 a more efficient algorithm 260

9.2 event-driven simulation 261

9.2.1 basic ideas 261

9.2.2 example: a modem bank simulation 263

summary 270**key concepts 270**

common errors	271
on the internet	271
exercises	271

chapter 10 **graphs and paths**

273

10.1 definitions	273
10.1.1 representation	275
10.2 unweighted shortest-path problem	285
10.2.1 theory	286
10.2.2 java implementation	289
10.3 positive-weighted, shortest-path problem	289
10.3.1 theory: dijkstra's algorithm	290
10.3.2 java implementation	294
10.4 negative-weighted, shortest-path problem	296
10.4.1 theory	296
10.4.2 java implementation	297
10.5 path problems in acyclic graphs	297
10.5.1 topological sorting	297
10.5.2 theory of the acyclic shortest-path algorithm	299
10.5.3 java implementation	301
10.5.4 an application: critical-path analysis	303
summary	305
key concepts	306
common errors	307
on the internet	307
exercises	308
references	311

part three Implementations

chapter 11 **inner classes and implementation of ArrayList** **315**

11.1 iterators and nested classes	315
11.2 iterators and inner classes	318
11.3 the AbstractCollection class	321
11.4 StringBuilder	322
11.5 implementation of ArrayList with an iterator	325
summary	330
key concepts	331

common errors	331
on the internet	331
exercises	331

chapter 12 stacks and queues

335

12.1 dynamic array implementations	335
12.1.1 stacks	335
12.1.2 queues	339
12.2 linked list implementations	344
12.2.1 stacks	345
12.2.2 queues	347
12.3 comparison of the two methods	351
12.4 the java.util.Stack class	351
12.5 double-ended queues	351
summary	353
key concepts	353
common errors	353
on the internet	353
exercises	353

chapter 13 linked lists

355

13.1 basic ideas	355
13.1.1 header nodes	357
13.1.2 iterator classes	358
13.2 java implementation	359
13.3 doubly linked lists and circularly linked lists	365
13.4 sorted linked lists	367
13.5 implementing the collections api LinkedList class	369
summary	379
key concepts	379
common errors	380
on the internet	380
exercises	380

chapter 14 trees

385

14.1 general trees	385
14.1.1 definitions	385
14.1.2 implementation	387
14.1.3 an application: file systems	387
14.2 binary trees	391