



图灵原版计算机科学系列



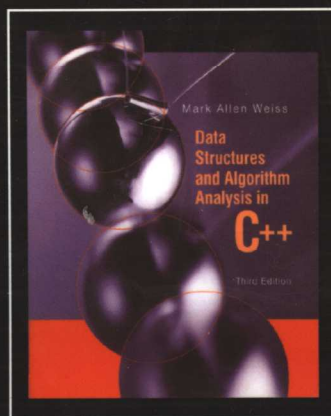
Data Structures and Algorithm Analysis in C++
Third Edition

数据结构与算法分析

—— C++ 描述

(英文版 · 第3版)

[美] Mark Allen Weiss 著



人民邮电出版社
POSTS & TELECOM PRESS

Data Structures and Algorithm Analysis in C++ Third Edition

数据结构与算法分析 —— C++ 描述 (英文版 · 第3版)

Mark Allen Weiss 教授撰写的数据结构与算法分析方面的著作曾被评为 20 世纪最佳的 30 部计算机著作之一, 已经成为公认的经典之作, 被全球数百所大学采用为教材, 广受好评。

本书秉承 Weiss 著作一贯的严谨风格, 同时又突出了实践。书中充分应用了现代 C++ 语言特性, 透彻地讲述了数据结构的原理和应用, 不仅使学生具备算法分析能力, 能够开发高效的程序, 而且让学生掌握良好的程序设计技巧。

本版特色

- 更新并测试了所有代码, 确保与 ANSI/ISO C++ 标准兼容。
- STL (标准模板库) 完全融合在全书之中。
- 完全修订了列表、栈和队列的内容。
- 简化了源代码, 避免了与类模板的分离编译有关的复杂语法。
- 每章结尾都提供了丰富的习题, 按难度划分, 强调了各章的关键概念。



Mark Allen Weiss 1987 年在普林斯顿大学获得计算机科学博士学位, 师从著名算法大师 Robert Sedgwick, 现任美国佛罗里达国际大学计算与信息科学学院教授。他曾经担任全美 AP (Advanced Placement) 考试计算机学科委员会的主席 (2000-2004)。他的主要研究方向是数据结构、算法和教育学。

本书相关信息请访问: 图灵网站 <http://www.turingbook.com>

读者 / 作者热线: (010) 88593802

反馈 / 投稿 / 推荐信箱: contact@turingbook.com



www.PearsonEd.com

上架建议 计算机 / 计算机科学 / 数据结构

ISBN 7-115-15233-0



9 787115 152336 >

This edition is licensed for distribution and sale in the People's Republic of China only, excluding Hong Kong, Macao and Taiwan, and may not be distributed and sold elsewhere.

此版本仅限在中华人民共和国境内 (中国香港、澳门、台湾地区除外) 销售发行。

ISBN7-115-15233-0/TP · 5674

定价: 59.00 元

人民邮电出版社网址 www.ptpress.com.cn

TURING 图灵原版计算机科学系列

**Data Structures and Algorithm Analysis in C++
Third Edition**

数据结构与算法分析

C++描述

(英文版·第3版)

[美] Mark Allen Weiss 著

 **人民邮电出版社**
POSTS & TELECOM PRESS

图书在版编目 (CIP) 数据

数据结构与算法分析: C++描述: 第3版 / (美) 维斯著. —北京: 人民邮电出版社, 2006.11
(图灵原版计算机科学系列)

ISBN 7-115-15233-0

I. 数... II. 维... III. ①数据结构—教材—英文②算法分析—教材—英文③C语言—
程序设计—教材—英文 IV. TP311.12

中国版本图书馆 CIP 数据核字 (2006) 第 104268 号

内 容 提 要

本书是数据结构和算法分析的经典教材, 书中使用主流的程序设计语言 C++ 作为具体的实现语言。书中内容包括表、栈、队列、树、散列表、优先队列、排序、不相交集算法、图论算法、算法分析、算法设计、摊还分析、查找树算法、k-d 树和配对堆等。本书适合作为计算机相关专业本科生的数据结构课程和研究生算法分析课程的教材。本科生的数据结构课程可以使用本书第 1 章~第 9 章, 多学时课程还可以讲解第 10 章; 研究生算法分析课程可以使用第 6 章~第 12 章的内容。

图灵原版计算机科学系列

数据结构与算法分析——C++ 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>
- 北京顺义振华印刷厂印刷
- 新华书店总店北京发行所经销
- ◆ 开本: 800 × 1000 1/16
- 印张: 37.75
- 字数: 750 千字
- 印数: 1—3 000 册
- 2006 年 11 月第 1 版
- 2006 年 11 月北京第 1 次印刷

著作权合同登记号 图字: 01-2006-5377 号

ISBN 7-115-15233-0/TP · 5674

定价: 59.00 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 Algorithm Analysis in C++*, Third Edition, 0321375319 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 © 2006.

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

目的/目标

本书全面论述了数据结构和算法分析，即组织大量数据的方法和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迫切。具有讽刺意味的是，由于在输入量很大时程序的效率明显降低，因此这又要求更加关注效率问题。通过在实际编程之前对算法进行分析，学生可以确定一个特定的解法是否可行。例如，在本书中学生可看到一些特定的问题，并了解精心的实现如何能够把处理大量数据的时间从 16 年减至不到 1 秒。因此，本书中论述的算法和数据结构均对其运行时间进行了分析。在某些情况下，还研究了影响实现运行时间的一些微小细节。

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越来越大，越来越复杂，这就要求开发更加复杂的程序。本书的目的是在教会学生使用良好的程序设计技巧的同时让学生具备算法分析能力，使得他们能够开发这种具有最高效率的程序。

本书适用于本科生的高级数据结构课程或是研究生的算法分析课程。使用本书的学生应该具有中等程度的程序设计方面的知识，包括指针、递归和面向对象程序设计，还要具有离散数学的某些知识。

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 C++。

C++已经成为系统编程的主流语言。除了修正了许多 C 语言的语法方面的缺陷之外，C++还提供了直接结构（类和模板）来实现抽象数据类型的通用数据结构。

撰写本书最困难的部分是确定 C++在书中所占的比例。使用太多的 C++的特性将使教材变得很晦涩，使用得太少又会和使用支持类的 C 语言撰写的教材区别不大。

我们采用的方法是以基于对象的方法来阐述文中的内容。这样，本书中几乎没有用到继承性。我们采用类模板来描述通用数据结构。通常情况下尽量避免使用深奥的 C++特性，而是使用标准 C++中的 `vector` 和 `string` 类。通过采用 C++的现代特性来取代在本书第 1 版中所用的次级特性简化了很多代码。本书前几版已经实现了类模板，方法是将类模板的接口与其实现分离开。毫无疑问，这是一种好方法，但是这种方法暴露出了编译的问题，读者事实上很难利用这些代码。本版中，在线的源代码将类模板作为一个单元来表示，而不再将接口和实现分离开。本书第 1 章对书中所用到的 C++特性进行了介绍，并阐述了对类模板的处理方法。附录 A 描述了如何重写

类模板，用于分离编译。

以 Java 和 C++ 两种语言描述的数据结构的完全版在因特网上可以得到。我们使用类似的编码习惯使得这两种语言间的相似性表现得更加明显。

第 3 版中的主要变化

第 3 版中包含了大量的错误修正，并且书中的大部分章节都经过了修订以提高其可读性。另外，本版还有以下几方面的变化：

- 书中的所有代码都做了更新以适应现代的 C++ 特性。
- 第 3 章进行了大量的修订，并且讨论了标准 `vector` 和 `list` 类的使用及其实现。标准 `vector` 类在其他数据结构的实现中被广泛使用。
- 第 4 章修订后包含了关于 `set` 和 `map` 类的讨论，并通过一个扩展的例子介绍了它们在有效算法设计中的应用。在第 9 章中也包含了一个使用标准 `map` 类来实现最短路径算法的例子。
- 第 7 章讨论了标准 `sort` 算法，其中包括一个关于实现重载的标准 `sort` 算法所涉及的技巧。
- 书中提供的源代码都已经进行了简化，从而避免了与类模板的分离编译有关的复杂语法。修订后的代码可以使读者更专心于算法本身，而不是过多地关注 C++。

内容概述

第 1 章包含离散数学和递归的一些内容。我相信熟悉递归的唯一办法是反复不断地看一些好的用法。因此，除第 5 章外，递归遍布本书每一章的例子中。第 1 章还介绍了一些 C++ 的内容，作为对 C++ 基础知识的了解，包含在 C++ 类设计中模板和重要结构的讨论。

第 2 章讨论算法分析。这一章阐述了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，对它们进行分析。这一章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。这一章较之以前的版本进行了大量的修订。现在包含了关于 STL `vector` 和 `list` 类的讨论，包括有关迭代的内容，并且提供了 STL `vector` 和 `list` 类的重要子集的实现。

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介绍的。本章还介绍了 AVL 树和伸展树。查找树实现细节的更详细讨论可在第 12 章找到。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。本版中新增的部分是对 STL `set` 和 `map` 类的讨论，包括一个讲解了如何使用三个分离的图来高效率地解决问题的例子。

第 5 章相对较短，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展散列。

第 6 章是关于优先队列的。这一章还讲解了二叉堆，还有一些附加内容，论述优先队列某些理论上很有趣的实现方法。斐波那契堆在第 11 章讨论，配对堆在第 12 章讨论。

第 7 章是排序。它是关于编程细节和分析的非常特殊的一章，讨论并比较了所有重要的通用排序算法。对插入排序、希尔排序、堆排序以及快速排序这四种算法进行了详细的分析。这一章末尾还讨论了外部排序。

第 8 章讨论不相交集算法并证明其运行时间。这一章短且特殊，如果不讨论 Kruskal 算法则这一章可跳过。

第 9 章讲授图论算法。图论算法之所以重要，不仅因为它们在实践中频繁出现，而且因为它们的运行时间特别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。为把这些问题放在一个适当的环境下讨论中，书中提供了对复杂性理论（包括 NP 完全性和不可判定性）的简短讨论。

第 10 章通过考查一般的问题求解技巧来介绍算法设计。这一章含有大量的实例。这里及后面各章使用了伪代码，使学生对一个示例算法的理解不受具体实现细节的困扰。

第 11 章处理摊还分析。对第 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，这一章给出了查找树和配对堆完全的仔细的实现。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。

第 1 章～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，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。在第 11 章分析的高级数据结构可以很容易地在前面各章中查到。第 9 章中对 NP 完全性的讨论太过简短，以至于不能用于算法分析课程。可以使用论述 NP 完全性的其他书籍来补充本书的这部分不足。

习题

每章末尾提供的习题与正文中讲授内容的顺序相匹配。最后的习题是把一章作为一个整体来处理而不是针对特定的一节来考虑的。较困难的习题标有一个星号，更难的习题标有两个星号。

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习题的解法。

补充材料

所有的读者都可以在 www.aw.com/cssupport 网站得到下面的补充材料：

- 书中例子的程序代码。

此外，以下材料可从 Addison-Wesley 的教师资源中心 (www.aw.com/irc) 获得，但是这部分内容仅针对有资质的教师。教师可以访问 Addison-Wesley 的教师资源中心或向当地的 Addison-Wesley 代表申请下面这些资料：

- 书中部分习题的答案。
- 书中的图。

致谢

在本丛书几部著作的准备过程中，我得到许多人的帮助。有些人在本书的其他版本中提到过，谢谢他们每一位。

同往常一样，Addison-Wesley 的专家们使得本书的写作过程更加轻松。我愿意借此机会感谢本书的编辑 Michael Hirsch 和文字编辑 Marilyn Lloyd。我还想感谢 Paul Anagnostopoulos 和他在 Windfall Software 的同事，感谢他们的出色工作使最后的书稿成书。特别感谢我的爱妻 Jill，感谢她所做的一切。

最后，我还想感谢广大的读者，他们发来电子邮件并指出前面各版中的一些错误和矛盾之处。我的网页 <http://www.cis.fiu.edu/~weiss> 将包含更新的（C++的、C 的以及 Java 的）源代码、勘误表以及指向提交问题报告的一个链接。

Mark Allen Weiss
于佛罗里达州迈阿密

CONTENTS

Chapter 1 Introduction

1

- 1.1 What's the Book About? 1
- 1.2 Mathematics Review 2
 - 1.2.1 Exponents 3
 - 1.2.2 Logarithms 3
 - 1.2.3 Series 4
 - 1.2.4 Modular Arithmetic 5
 - 1.2.5 The P Word 6
- 1.3 A Brief Introduction to Recursion 7
- 1.4 C++ Classes 11
 - 1.4.1 Basic class Syntax 12
 - 1.4.2 Extra Constructor Syntax and Accessors 12
 - 1.4.3 Separation of Interface and Implementation 15
 - 1.4.4 `vector` and `string` 17
- 1.5 C++ Details 19
 - 1.5.1 Pointers 19
 - 1.5.2 Parameter Passing 21
 - 1.5.3 Return Passing 22
 - 1.5.4 Reference Variables 23
 - 1.5.5 The Big Three: Destructor, Copy Constructor, `operator=` 23
 - 1.5.6 C-style Arrays and Strings 26
- 1.6 Templates 29
 - 1.6.1 Function Templates 29
 - 1.6.2 Class Templates 30
 - 1.6.3 `Object`, `Comparable`, and an Example 32
 - 1.6.4 Function Objects 34
 - 1.6.5 Separate Compilation of Class Templates 35
- 1.7 Using Matrices 37
 - 1.7.1 The Data Members, Constructor, and Basic Accessors 37
 - 1.7.2 `operator[]` 37
 - 1.7.3 Destructor, Copy Assignment, Copy Constructor 39



Summary	39
Exercises	39
References	41

Chapter 2 Algorithm Analysis

43

2.1	Mathematical Background	43
2.2	Model	46
2.3	What to Analyze	46
2.4	Running Time Calculations	49
2.4.1	A Simple Example	49
2.4.2	General Rules	50
2.4.3	Solutions for the Maximum Subsequence Sum Problem	52
2.4.4	Logarithms in the Running Time	58
2.4.5	Checking Your Analysis	62
2.4.6	A Grain of Salt	63
	Summary	63
	Exercises	64
	References	69

Chapter 3 Lists, Stacks, and Queues

71

3.1	Abstract Data Types (ADTs)	71
3.2	The List ADT	72
3.2.1	Simple Array Implementation of Lists	72
3.2.2	Simple Linked Lists	73
3.3	vector and list in the STL	74
3.3.1	Iterators	75
3.3.2	Example: Using <code>erase</code> on a List	77
3.3.3	<code>const_iterators</code>	77
3.4	Implementation of vector	79
3.5	Implementation of list	83
3.6	The Stack ADT	94
3.6.1	Stack Model	94
3.6.2	Implementation of Stacks	95
3.6.3	Applications	96
3.7	The Queue ADT	104
3.7.1	Queue Model	104
3.7.2	Array Implementation of Queues	104
3.7.3	Applications of Queues	106
	Summary	107
	Exercises	108

Chapter 4 Trees

113

- 4.1 Preliminaries 113
 - 4.1.1 Implementation of Trees 114
 - 4.1.2 Tree Traversals with an Application 115
- 4.2 Binary Trees 119
 - 4.2.1 Implementation 120
 - 4.2.2 An Example: Expression Trees 121
- 4.3 The Search Tree ADT—Binary Search Trees 124
 - 4.3.1 contains 125
 - 4.3.2 findMin and findMax 125
 - 4.3.3 insert 129
 - 4.3.4 remove 130
 - 4.3.5 Destructor and Copy Assignment Operator 132
 - 4.3.6 Average-Case Analysis 133
- 4.4 AVL Trees 136
 - 4.4.1 Single Rotation 139
 - 4.4.2 Double Rotation 142
- 4.5 Splay Trees 149
 - 4.5.1 A Simple Idea (That Does Not Work) 150
 - 4.5.2 Splaying 152
- 4.6 Tree Traversals (Revisited) 158
- 4.7 B-Trees 159
- 4.8 Sets and Maps in the Standard Library 165
 - 4.8.1 Sets 165
 - 4.8.2 Maps 166
 - 4.8.3 Implementation of `set` and `map` 167
 - 4.8.4 An Example That Uses Several Maps 168
- Summary 174
- Exercises 174
- References 181

Chapter 5 Hashing

185

- 5.1 General Idea 185
- 5.2 Hash Function 186
- 5.3 Separate Chaining 188
- 5.4 Hash Tables Without Linked Lists 192
 - 5.4.1 Linear Probing 193
 - 5.4.2 Quadratic Probing 195
 - 5.4.3 Double Hashing 199
- 5.5 Rehashing 200

5.6	Hash Tables in the Standard Library	204
5.7	Extendible Hashing	204
	Summary	207
	Exercises	208
	References	211

Chapter 6 Priority Queues (Heaps)

213

6.1	Model	213
6.2	Simple Implementations	214
6.3	Binary Heap	215
	6.3.1 Structure Property	215
	6.3.2 Heap-Order Property	216
	6.3.3 Basic Heap Operations	217
	6.3.4 Other Heap Operations	220
6.4	Applications of Priority Queues	225
	6.4.1 The Selection Problem	226
	6.4.2 Event Simulation	227
6.5	d -Heaps	228
6.6	Leftist Heaps	229
	6.6.1 Leftist Heap Property	229
	6.6.2 Leftist Heap Operations	230
6.7	Skew Heaps	235
6.8	Binomial Queues	239
	6.8.1 Binomial Queue Structure	240
	6.8.2 Binomial Queue Operations	241
	6.8.3 Implementation of Binomial Queues	244
6.9	Priority Queues in the Standard Library	251
	Summary	251
	Exercises	251
	References	257

Chapter 7 Sorting

261

7.1	Preliminaries	261
7.2	Insertion Sort	262
	7.2.1 The Algorithm	262
	7.2.2 STL Implementation of Insertion Sort	263
	7.2.3 Analysis of Insertion Sort	264
7.3	A Lower Bound for Simple Sorting Algorithms	265

7.4	Shellsort	266
7.4.1	Worst-Case Analysis of Shellsort	268
7.5	Heapsort	270
7.5.1	Analysis of Heapsort	272
7.6	Mergesort	274
7.6.1	Analysis of Mergesort	276
7.7	Quicksort	279
7.7.1	Picking the Pivot	280
7.7.2	Partitioning Strategy	282
7.7.3	Small Arrays	284
7.7.4	Actual Quicksort Routines	284
7.7.5	Analysis of Quicksort	287
7.7.6	A Linear-Expected-Time Algorithm for Selection	290
7.8	Indirect Sorting	292
7.8.1	<code>vector<Comparable*></code> Does Not Work	295
7.8.2	Smart Pointer Class	295
7.8.3	Overloading <code>operator<</code>	295
7.8.4	Dereferencing a Pointer with <code>*</code>	295
7.8.5	Overloading the Type Conversion Operator	295
7.8.6	Implicit Type Conversions Are Everywhere	296
7.8.7	Dual-Direction Implicit Conversions Can Cause Ambiguities	296
7.8.8	Pointer Subtraction Is Legal	297
7.9	A General Lower Bound for Sorting	297
7.9.1	Decision Trees	297
7.10	Bucket Sort	299
7.11	External Sorting	300
7.11.1	Why We Need New Algorithms	300
7.11.2	Model for External Sorting	300
7.11.3	The Simple Algorithm	301
7.11.4	Multiway Merge	302
7.11.5	Polyphase Merge	303
7.11.6	Replacement Selection	304
	Summary	305
	Exercises	306
	References	311

Chapter 8 The Disjoint Set Class

315

8.1	Equivalence Relations	315
8.2	The Dynamic Equivalence Problem	316
8.3	Basic Data Structure	317

8.4	Smart Union Algorithms	321
8.5	Path Compression	324
8.6	Worst Case for Union-by-Rank and Path Compression	325
8.6.1	Analysis of the Union/Find Algorithm	326
8.7	An Application	331
	Summary	334
	Exercises	335
	References	336

Chapter 9 Graph Algorithms

339

9.1	Definitions	339
9.1.1	Representation of Graphs	340
9.2	Topological Sort	342
9.3	Shortest-Path Algorithms	345
9.3.1	Unweighted Shortest Paths	347
9.3.2	Dijkstra's Algorithm	351
9.3.3	Graphs with Negative Edge Costs	360
9.3.4	Acyclic Graphs	360
9.3.5	All-Pairs Shortest Path	364
9.3.6	Shortest Path Example	365
9.4	Network Flow Problems	367
9.4.1	A Simple Maximum-Flow Algorithm	367
9.5	Minimum Spanning Tree	372
9.5.1	Prim's Algorithm	373
9.5.2	Kruskal's Algorithm	376
9.6	Applications of Depth-First Search	378
9.6.1	Undirected Graphs	379
9.6.2	Biconnectivity	381
9.6.3	Euler Circuits	385
9.6.4	Directed Graphs	388
9.6.5	Finding Strong Components	390
9.7	Introduction to NP-Completeness	392
9.7.1	Easy vs. Hard	392
9.7.2	The Class NP	393
9.7.3	NP-Complete Problems	394
	Summary	396
	Exercises	396
	References	404

Chapter 10 Algorithm Design Techniques 409

- 10.1 Greedy Algorithms 409
 - 10.1.1 A Simple Scheduling Problem 410
 - 10.1.2 Huffman Codes 413
 - 10.1.3 Approximate Bin Packing 419
- 10.2 Divide and Conquer 427
 - 10.2.1 Running Time of Divide and Conquer Algorithms 428
 - 10.2.2 Closest-Points Problem 430
 - 10.2.3 The Selection Problem 435
 - 10.2.4 Theoretical Improvements for Arithmetic Problems 438
- 10.3 Dynamic Programming 442
 - 10.3.1 Using a Table Instead of Recursion 442
 - 10.3.2 Ordering Matrix Multiplications 444
 - 10.3.3 Optimal Binary Search Tree 447
 - 10.3.4 All-Pairs Shortest Path 451
- 10.4 Randomized Algorithms 454
 - 10.4.1 Random Number Generators 455
 - 10.4.2 Skip Lists 459
 - 10.4.3 Primality Testing 461
- 10.5 Backtracking Algorithms 464
 - 10.5.1 The Turnpike Reconstruction Problem 465
 - 10.5.2 Games 469
- Summary 475
- Exercises 475
- References 485

Chapter 11 Amortized Analysis 491

- 11.1 An Unrelated Puzzle 492
- 11.2 Binomial Queues 492
- 11.3 Skew Heaps 497
- 11.4 Fibonacci Heaps 499
 - 11.4.1 Cutting Nodes in Leftist Heaps 500
 - 11.4.2 Lazy Merging for Binomial Queues 502
 - 11.4.3 The Fibonacci Heap Operations 506
 - 11.4.4 Proof of the Time Bound 506
- 11.5 Splay Trees 509
 - Summary 513
 - Exercises 513
 - References 515

Chapter 12	Advanced Data Structures and Implementation	517
12.1	Top-Down Splay Trees	517
12.2	Red-Black Trees	525
12.2.1	Bottom-Up Insertion	526
12.2.2	Top-Down Red-Black Trees	527
12.2.3	Top-Down Deletion	531
12.3	Deterministic Skip Lists	535
12.4	AA-Trees	540
12.5	Treaps	547
12.6	k -d Trees	549
12.7	Pairing Heaps	553
	Summary	558
	Exercises	558
	References	563
Appendix A	Separate Compilation of Class Templates	567
A.1	Everything in the Header	568
A.2	Explicit Instantiation	568
A.3	The export Directive	570
Index		571