

国外计算机科学经典教材

THOMSON

Data Structures Using C++

数据结构

—— C++ 版

(美) D. S. Malik 著
王海涛 丁炎炎 译



清华大学出版社

国外计算机科学经典教材

数 据 结 构

——C++版

(美) D. S. Malik 著

王海涛 丁炎炎 译

清华大学出版社

北 京

D. S. Malik

Data Structures Using C++

EISBN: 0-619-15907-3

Copyright © 2003 by Course Technology, a division of Thomson Learning.

Original language published by Thomson Learning (a division of Thomson Learning Asia Pte Ltd).

All Rights reserved.

本书原版由汤姆森学习出版集团出版。版权所有，盗印必究。

Tsinghua University Press is authorized by Thomson Learning to publish and distribute exclusively this simplified Chinese edition. This edition is authorized for sale in the People's Republic of China only (excluding Hong Kong, Macao SAR and Taiwan). Unauthorized export of this edition is a violation of the Copyright Act. No part of this publication may be reproduced or distributed by any means, or stored in a database or retrieval system, without the prior written permission of the publisher.

本中文简体字翻译版由汤姆森学习出版集团授权清华大学出版社独家出版发行。此版本仅限在中华人民共和国境内(不包括中国香港、澳门特别行政区及中国台湾地区)销售。未经授权的本书出口将被视为违反版权法的行为。未经出版者预先书面许可，不得以任何方式复制或发行本书的任何部分。

981-254-346-5

北京市版权局著作权合同登记号 图字：01-2003-4332

本书封面贴有清华大学出版社激光防伪标签，无标签者不得销售。

图书在版编目(CIP)数据

数据结构——C++版/(美)马立克(Malik, D.S.)著；王海涛，丁炎炎译.—北京：清华大学出版社，2003

书名原文：Data Structures Using C++

(国外计算机科学经典教材)

ISBN 7-302-07491-7

I. 数… II. ①马…②王…③丁… III. ①数据结构—教材②C语言—程序设计—教材

IV. ①TP311.12②TP312

中国版本图书馆 CIP 数据核字(2003)第 097460 号

出 版 者：清华大学出版社 地 址：北京清华大学学研大厦

<http://www.tup.com.cn> 邮 编：100084

社 总 机：010-62770175 客户服务：010-62776969

组稿编辑：曹 康

文稿编辑：杜一民

封面设计：康 博

版式设计：康 博

印 装 者：清华大学印刷厂

发 行 者：新华书店总店北京发行所

开 本：185×260 印张：50.25 字数：1286 千字

版 次：2004 年 1 月第 1 版 2004 年 1 月第 1 次印刷

书 号：ISBN 7-302-07491-7/TP·5522

印 数：1~4000

定 价：92.00 元

前 言

欢迎阅读本书。作为 CS2 课程(计算机科学 C++第二课程)的教材,本书将为您和您的学生带来一些新意。CS2 课程要求达到计算机科学课程对编程的要求。在 50 多个学期中,我成功地向计算机科学专业的学生教授了编程和数据结构的课程,本书正是我长期教学经验的精髓和升华。

本书是对由 Course Technology 出版的 CS1 书籍,即《C++ Programming: From Problem Analysis to Program Design》一书的延续。本书采用与 CS1 相同的方法,内容以清楚易懂为宗旨。我们对本书进行了反复的修改,直至符合学生的阅读要求。书中大部分示例来自于学生的课堂交流。

本书假定读者熟知 C++的基本知识,如数据类型、控制结构、函数与参数及数组。如果您需要复习这些概念,或者将 Java 作为第一编程语言,那么附录 G 中的相关内容将对您十分有帮助。如果您需要更详细地了解附录 G 以外的 CS1 的主题,请参考作者上面提到的 C++编程书籍以及附录 H 中列出的资料。另外,本书还要求读者具备一些适当的数学知识背景,如大学代数。

本书主要内容

由于要作为学习计算机编程的第二门课程,因此本书将把重点放在数据结构和面向对象设计(object-oriented design, OOD)上。书中提供的编程示例有效地使用了 OOD 技术,用程序解决某个特定的问题。

第 1 章介绍了软件工程原理。首先描述了软件的生命周期,然后讨论了算法分析之所以重要的原因,并介绍了算法分析中使用的大 O 符号。OOD 有 3 个基本原则:封装性、继承性和多态性。C++中通过类实现封装。第 1 章的后半部分讨论了用户定义的类。如果您熟悉如何创建并使用自己的类,可以略过这部分内容。这一章还讨论了解决特定问题的基本 OOD 技术。

第 2 章继续讨论 OOD 原则,并介绍了继承性和两类多态性(第三类多态性将在附录 F 中介绍)。如果读者熟悉 C++中的继承、运算符重载和模板的知识,则可以跳过这一章。

C++有 3 种基本数据类型:简单类型、结构化类型和指针类型。本书假定读者熟悉简单数据类型和数组(一种结构化数据类型)。结构化数据类型在第 1 章中介绍;第 3 章将讨论 C++中指针数据类型的工作原理以及指针和类的关系。这一章用指针和模板进行阐述并开发了一个通用代码来实现使用动态数组的表。

C++配备了标准模板库(Standard Template Library, STL)。STL 另外还提供了处理表(连续或者链接的表)、堆栈和队列的代码。第 4 章讨论了 STL 的一些重要功能,并说明了一些 STL 工具在程序中的使用方法。这一章还特别介绍了顺序容器 vector(向量)和 deque(双端队列)。接着,本章介绍了如何开发自己的代码实现并操作数据,以及如何使用专业代码进行编程。

第 5 章讨论了链表。本章首先解释了链表的基本属性，如项的插入和删除，如何创建一个链表。接着，开发了一个通用代码，用来处理单独的链表中的数据。这一章还讨论了双重链表，带有头和尾节点的链表以及循环链表。此外，本章还讨论了 STL 类 list(表)。

第 6 章介绍了递归并用多种示例演示如何以递归的方式讨论并解决问题。

第 7 章和第 8 章讨论了堆栈和队列。除了演示如何开发您自己的通用代码并实现堆栈和队列外，这些章节还解释了 STL 类的堆栈和队列如何工作。这些章节开发的编程代码都是通用的。

第 9 章描述了搜索算法。在分析了顺序搜索算法之后，本章讨论了二叉树搜索算法并对这种算法进行了简要的分析。在对基于比较的搜索算法初步探讨之后，本章还讨论了散列算法。

第 10 章介绍了排序算法如选择排序、插入排序、快速排序、合并排序和堆排序。第 11 章讨论了二叉树。第 12 章介绍了图并讨论了图的算法，如最短路径，最小生成树和拓扑排序。

第 13 章继续第 4 章关于 STL 的讨论。特别介绍了 STL 的关联容器和算法。

附录 A 列出了 C++ 中的保留字。附录 B 列出了 C++ 运算符的优先级和关联性。附录 C 列出了美国信息交换标准码(American Standard Code for Information Interchange, ASCII)和扩充二进制代码的十进制交换码(Extended Binary Code Decimal Interchange, EBCDIC)的字符集。附录 D 列出了可被重载的 C++ 运算符。附录 E 描述了一些广泛使用的库例程。附录 F 讨论了其他的 C++ 主题——虚函数以及运算符和类的地址。附录 G 回顾了 C++ 的基本元素，并比较了 C++ 和 Java 语言的基本概念，如数据类型，控制结构，函数和参数以及数组。如果您的第一编程语言是 Java，那么附录 G 将有助于您熟悉这些 C++ 基本元素。附录 H 为进一步学习 C++ 提供参考，这里包括了附录 G 中没有提到的关于 C++ 的论题。附录 I 提供章节中部分练习题的答案。

本书的使用方法

本书的主要目的在于用 C++ 和 OOD 介绍数据结构来解决特定问题。为此，本书讨论了一些数据结构，如链表、堆栈、队列和二叉树。C++ 的 STL 也提供了必要的代码来实现这些数据结构。无论如何，本书强调的是教您如何开发自己的代码。同时，我们也希望您能学会使用专业代码编写程序。

第 4 章介绍了 STL。在接下来的章节中，介绍了如何开发您自己的代码后，举例说明了如何使用现有的 STL 代码。因此本书可以有多种使用方法。如果在首次阅读本书时您对 STL 不感兴趣，可以跳过第 4 章和接下来讨论特定 STL 组件的章节部分。

第 6 章讨论了递归。但是第 6 章并不是阅读第 7 章和第 8 章的先决条件。如果您在阅读了这些章节之后再阅读第 6 章，那么您可以跳过第 7 章的“消除递归”一节的内容，而是在阅读了第 6 章之后再阅读这一节内容。尽管学习第 6 章不要求理解第 9 章，但是也应该看一下第 9、10 章。因此，我们建议您在阅读第 9 章之前先阅读第 6 章。

图 0-1 说明了这些章节的关系。

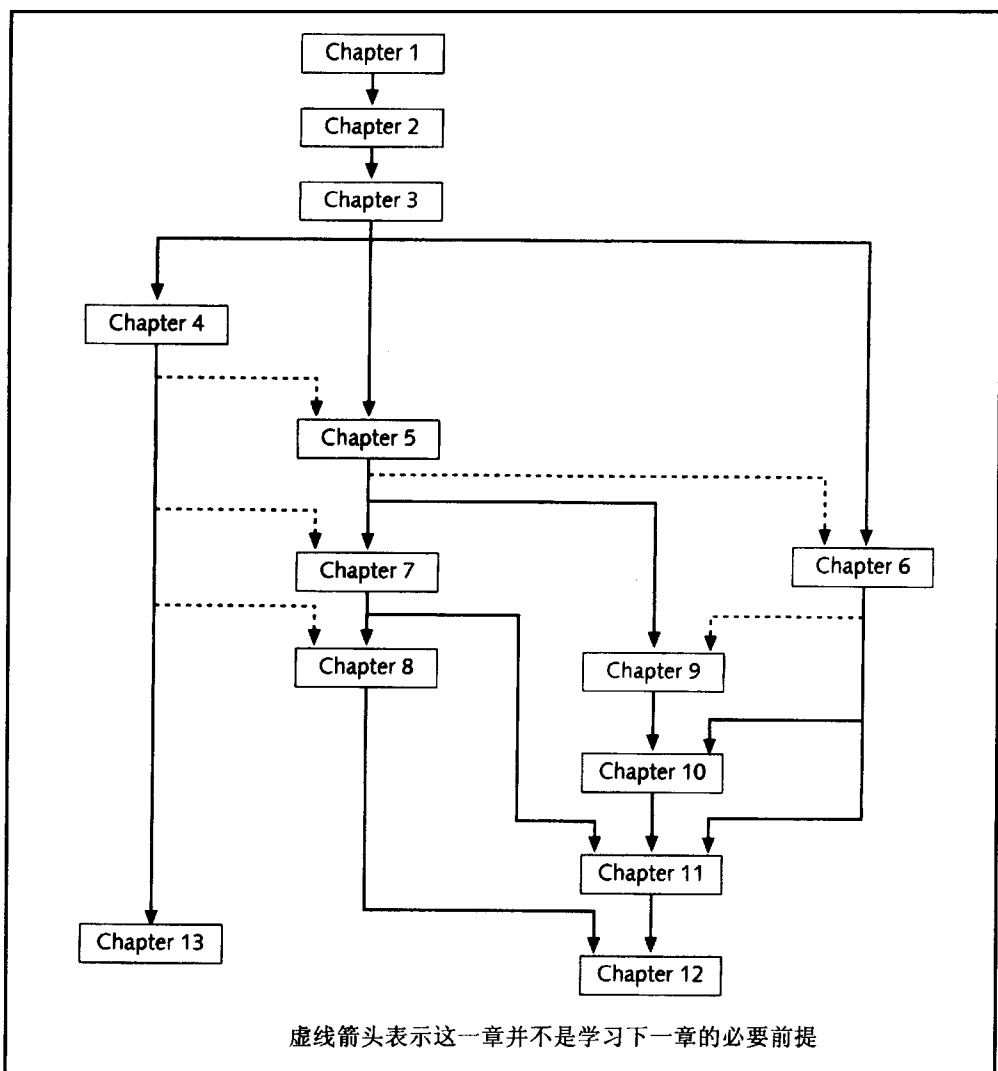


图 0-1 章节间关系图

本书特点

本书中的各章都具有如下特点。它们不仅有助于学习，而且使读者可以自由安排学习进度。

- 各章开头的“本章主要内容”概述了该章中所讨论的 C++ 编程概念。
- “注意”部分强调了与章节中介绍的概念有关的重要内容。
- 丰富而详尽的可视图形(visual diagram)说明了概念难点。本书包含了 350 多幅图。
- 各章中带有编号的示例(Numbered Example)是用相关代码解释关键概念的小程序。这些示例中编程代码的每一行都有编号。每一个程序都通过示例的运行进行说明并逐行解释，同时详细讨论每一行的理论基础。
- “编程示例”(Programming Example)是各章最后着重介绍的程序。这些示例包括了输入、输出、问题分析和算法设计的全过程以及程序清单。另外，这些编程示例中的问题均使用 OOD 来解决。这些示例构成了本书的核心部分。这些程序很有条理且用户友好。编

程示例首先着眼于问题分析,接着研究算法设计,每一步算法都使用 C++ 编码。除了教授用户解决问题的技巧以外,这些程序还介绍如何在一个实际的 C++ 程序中实现概念。我强烈建议学生认真学习程序示例,以便有效地学习 C++。

- 各章最后的“快速回顾”(Quick Review)部分通过总结章节中提及的概念来巩固读者新学的知识。在阅读章节之后,读者可快速回顾本章的重点并通过书后练习进行自我测试。很多读者将快速回顾作为测试前对章节进行快速复习的方法。
- “练习题”(Exercises)部分进一步强化所学知识,确保学生真正掌握章节内容。
- “编程练习”(Programming Exercises)部分要求学生按特定要求编写 C++ 程序。

本书自始至终以有助于学习的步调来介绍概念。本书简单易懂。在介绍关键概念之前,我们先解释某些元素必要的原因,然后用示例和小程序来解释该概念。

所有的源代码和解决方案都经历了编写、编译和质量保证测试的过程。可以用 Microsoft Visual C++ .NET 或 Metrowerks Code Warrior 来编译程序。

教学工具

当本书用于教学时可以利用下列补充材料。请参阅书后的“教辅材料申请表”索取相关材料。电子教学手册。与本书对应的教学手册包括:

- 有助于备课的附加教学资料,包括对授课主题的建议。
- 所有章节结尾资料的解决方案,包括编程练习。

ExamView。教辅材料中有一个功能强大的测试软件包——ExamView,它允许教师进行笔试、机试(基于 LAN)和基于 Internet 的测验。ExamView 包括上百个与课文论题配套的问题,使学生能够为进一步复习而设计包括页面索引在内的学习指南。基于计算机和 Internet 的测验允许学生在他们的电脑上进行测验,并能够对试卷自动评分,节省了教师的时间。

PowerPoint 演示。本书各章都有 Microsoft PowerPoint 幻灯片。它们作为课堂讲解的辅助部分,使学生可以通过网络进行章节复习,也可以打印出来发放给学生。教师可以将他们介绍的附加论题添加进这些幻灯片中。

远程教学。Course Technology 得意之作就是在 WebCT、Blackboard 以及它自己的课程增强工具 MyCourse.com 中提供了在线课程,这些课程提供了最完善的动态教学环境。当您在自己的课程中添加在线内容时,同时也添加了自我测试、链接、词汇。更为重要的是,这是通向 21 世纪最为重要的信息来源。希望您充分利用无论是在线的或是不在线的课程。如果您要知道更多关于如何将远程教学引入课程的信息,请与当地 Course Technology 的销售代表联系。

源代码。可从 www.course.com 获得源代码。这些源代码同时包括了运行程序时所需的文件。但首先要将文件存储在软盘中。

解决方案文件。本书中所有编程练习的源代码都可以在 www.course.com 找到。运行编程练习的文件也包含在源代码中。然而,应首先将文件存储在软盘中。

目 录

第 1 章 软件工程基本原理和 C++ 类	1
1.1 软件的生命周期	1
1.2 软件开发阶段	2
1.2.1 分析阶段	2
1.2.2 设计阶段	2
1.2.3 实现阶段	3
1.2.4 测试和调试	5
1.3 算法分析：大 O 表示法	6
1.4 类	12
1.4.1 统一建模语言图	15
1.4.2 变量(对象)的声明	15
1.4.3 访问类的成员	16
1.4.4 类的内置运算	17
1.4.5 赋值运算符和类	17
1.4.6 类的作用域	17
1.4.7 函数和类	18
1.4.8 引用参数和类对象(变量)	18
1.4.9 成员函数的实现	19
1.4.10 构造函数	23
1.4.11 调用构造函数	26
1.4.12 构造函数和默认参数	26
1.4.13 析构函数	27
1.4.14 结构	27
1.5 数据抽象、类和抽象数据类型	27
1.6 编程示例：糖果机	36
1.6.1 问题分析和算法设计	36
1.6.2 收银机	36
1.6.3 控制装置	38
1.6.4 主程序	40
1.7 标识类、对象和操作	45
1.8 快速回顾	45
1.9 练习题	47
1.10 编程练习	51

第 2 章 面向对象的设计方法和 C++	54
2.1 继承	54
2.1.1 重新定义基类的成员函数	57
2.1.2 派生类与基类的构造函数	58
2.1.3 派生类的头文件	64
2.1.4 头文件的多重包含	64
2.1.5 类的保护成员	66
2.1.6 三种继承方式: 公有继承, 保护继承或私有继承	66
2.2 聚合	67
2.3 多态: 运算符和函数重载	71
2.4 运算符重载	71
2.4.1 为什么要重载运算符	71
2.4.2 运算符重载	72
2.4.3 运算符函数的语法	73
2.4.4 重载运算符的限制	73
2.4.5 this 指针	73
2.4.6 类的友元函数	77
2.4.7 定义友元函数	77
2.4.8 运算符函数的两种形式: 成员函数和非成员函数	80
2.5 重载二元运算符	81
2.5.1 重载二元运算符(算术运算符和关系运算符)为成员函数	81
2.5.2 重载二元运算符(算术运算符和关系运算符)为非成员函数	83
2.5.3 重载输出(<<)和输入(>>)运算符	84
2.5.4 重载输出运算符(<<)	84
2.5.5 重载输入运算符(>>)	85
2.5.6 重载运算符形式的选择: 成员函数和非成员函数	87
2.6 编程示例: 复数	87
2.7 函数重载	92
2.8 模板	92
2.8.1 函数模板	93
2.8.2 类模板	95
2.8.3 头文件和类模板的实现文件	97
2.9 快速回顾	97
2.10 练习题	99
2.11 编程练习	109
第 3 章 指针和基于数组的表	113
3.1 指针数据类型和指针变量	113

3.1.1	声明指针变量	113
3.1.2	取地址运算符(&).....	114
3.1.3	取值运算符(*).....	115
3.1.4	类、结构和指针变量	120
3.1.5	初始化指针变量	123
3.1.6	动态变量	123
3.1.7	指针变量的运算	125
3.2	动态数组	126
3.2.1	函数和指针	128
3.2.2	指针和函数返回值	128
3.3	浅复制、深复制与指针	129
3.4	类和指针：一些特例	131
3.4.1	析构函数	132
3.4.2	赋值运算符	133
3.4.3	重载赋值运算符	135
3.4.4	复制构造函数	138
3.5	重载数组索引(下标)运算符([]).....	144
3.6	编程示例：newString	145
3.7	基于数组的表	151
3.7.1	复制构造函数	160
3.7.2	重载赋值运算符	161
3.7.3	搜索	161
3.7.4	插入	163
3.7.5	删除	163
3.7.6	各种表操作的时间复杂度	164
3.8	编程示例：多项式的运算	168
3.9	快速回顾	175
3.10	练习题	177
3.11	编程练习	182
第 4 章	标准模板类库	185
4.1	STL 的组成部分	185
4.1.1	容器类型	186
4.1.2	顺序容器	186
4.2	顺序容器：向量容器	186
4.2.1	声明 vector 对象	187
4.2.2	为向量容器声明一个迭代器	189
4.2.3	容器以及 begin 和 end 函数	190

4.2.4 对所有容器通用的成员函数	193
4.2.5 顺序容器公共的成员函数	194
4.2.6 copy 算法	194
4.2.7 ostream 迭代器和 copy 函数	196
4.3 顺序容器：双端队列	198
4.4 迭代器	201
4.4.1 迭代器的类型	202
4.4.2 输入迭代器	202
4.4.3 输出迭代器	202
4.4.4 前向迭代器	203
4.4.5 双向迭代器	203
4.4.6 随机访问迭代器	203
4.4.7 流迭代器	205
4.5 编程示例：成绩报告单	206
4.5.1 问题分析与算法设计	208
4.5.2 主程序	219
4.5.3 程序清单	221
4.6 快速回顾	226
4.7 练习题	227
4.8 编程练习	231
第 5 章 链表	234
5.1 链表	234
5.2 链表的属性	236
5.3 项的插入和删除	239
5.3.1 插入	239
5.3.2 删除	241
5.4 构建链表	243
5.4.1 正向构建链表	243
5.4.2 反向构建链表	247
5.5 ADT 链表	248
5.5.1 默认构造函数	251
5.5.2 销毁表	252
5.5.3 初始化表	252
5.5.4 重载输出运算符	253
5.5.5 表的长度	253
5.5.6 检索第一个节点的数据	253
5.5.7 检索最后一个节点的数据	254

5.5.8 搜索表	254
5.5.9 在表头插入节点	255
5.5.10 在表尾插入节点	256
5.5.11 删除节点	256
5.5.12 复制表	261
5.5.13 析构函数	262
5.5.14 复制构造函数	262
5.5.15 重载赋值运算符	263
5.6 有序链表	264
5.6.1 搜索表	265
5.6.2 插入节点	266
5.6.3 删除节点	270
5.6.4 有序链表的头文件	271
5.7 双向链表	273
5.7.1 默认构造函数	276
5.7.2 isEmptyList	276
5.7.3 销毁表	277
5.7.4 初始化表	277
5.7.5 表的长度	277
5.7.6 重载输出运算符	278
5.7.7 反向打印表	278
5.7.8 搜索表	279
5.7.9 第一个和最后一个元素	279
5.7.10 插入节点	280
5.7.11 删除节点	282
5.8 STL 顺序容器: list	284
5.9 带有头节点和尾节点的链表	289
5.10 循环链表	290
5.11 编程示例: Video Store	291
5.12 快速回顾	309
5.13 练习题	310
5.14 编程练习	315
第 6 章 递归	319
6.1 递归的定义	319
6.1.1 直接递归和间接递归	321
6.1.2 无穷递归	321
6.2 递归法解决问题	322

6.3	递归还是迭代	337
6.4	递归和回溯： n -皇后问题	338
6.4.1	回溯	338
6.4.2	回溯和4皇后问题	340
6.4.3	8皇后问题	341
6.5	快速回顾	345
6.6	练习题	346
6.7	编程练习	348
第7章	堆栈	353
7.1	堆栈	353
7.2	使用数组实现堆栈	355
7.2.1	初始化堆栈	358
7.2.2	销毁堆栈	359
7.2.3	空堆栈	359
7.2.4	满堆栈	359
7.2.5	入栈	359
7.2.6	返回栈顶元素	361
7.2.7	出栈	361
7.2.8	复制堆栈	362
7.2.9	构造函数和析构函数	363
7.2.10	复制构造函数	364
7.2.11	重载赋值运算符(=)	364
7.2.12	堆栈的头文件	365
7.3	编程示例：求最高 GPA	366
7.4	堆栈的链表实现	370
7.4.1	默认构造函数	372
7.4.2	销毁堆栈	373
7.4.3	初始化堆栈	373
7.4.4	入栈	374
7.4.5	返回栈顶元素	377
7.4.6	出栈	377
7.4.7	由类 <code>LinkedListType</code> 派生而来的堆栈	379
7.5	堆栈应用：后缀表达式计算器	381
7.5.1	主算法	385
7.5.2	完整的程序清单	387
7.6	消除递归：反向打印一个链表的非递归算法	391
7.7	STL 堆栈类(堆栈容器适配器)	396

7.8 快速回顾	398
7.9 练习题	398
7.10 编程练习	402
第 8 章 队列	404
8.1 队列	404
8.1.1 队列操作	404
8.1.2 队列的数组实现	405
8.2 队列的链式实现	416
8.3 从类 <code>linkedListType</code> 派生而来的队列	420
8.4 STL 类 <code>queue</code> (队列容器适配器)	423
8.5 优先级队列	425
8.6 队列的应用: 模拟	426
8.6.1 设计队列系统	427
8.6.2 客户	427
8.6.3 服务器	430
8.6.4 服务器表	433
8.6.5 等待客户的队列	438
8.6.6 主程序	440
8.7 快速回顾	447
8.8 练习题	447
8.9 编程练习	451
第 9 章 搜索算法	453
9.1 搜索算法	453
9.1.1 顺序搜索	456
9.1.2 顺序搜索算法分析	457
9.1.3 有序表	458
9.1.4 折半搜索	459
9.1.5 折半搜索算法的性能	462
9.1.6 将数据项插入到一个有序表	463
9.2 基于比较的搜索算法的下限	465
9.3 散列算法	466
9.3.1 散列函数: 示例	467
9.3.2 冲突解决	467
9.3.3 冲突解决: 开型寻址法	467
9.3.4 二次探测	469
9.3.5 删除: 开型寻址法	472
9.3.6 散列法: 使用二次探测来实现	473

9.3.7 冲突解决: 链地址法(开散列方法).....	476
9.3.8 散列法性能分析	477
9.4 快速回顾	478
9.5 练习题	480
9.6 编程练习	481
第 10 章 排序算法	484
10.1 排序算法	484
10.2 选择排序: 基于数组的表	484
10.3 插入排序: 基于数组的表	490
10.4 插入排序: 基于链表的表	495
10.5 基于比较的排序算法的下限	499
10.6 快速排序: 基于数组的表	500
10.7 归并排序: 基于链表的表	505
10.7.1 划分	507
10.7.2 归并	509
10.7.3 分析: 归并排序	512
10.8 堆排序: 基于数组的表	512
10.8.1 构建堆	513
10.8.2 分析: 堆排序	520
10.9 再论优先级队列	520
10.9.1 在优先级队列中插入一个元素	521
10.9.2 从优先级队列删除一个元素	521
10.10 编程示例: 选举结果	521
10.10.1 问题分析和算法设计	522
10.10.2 主程序	528
10.10.3 对姓名排序	530
10.10.4 处理投票数据	532
10.10.5 计算选票数的总和	534
10.10.6 打印标题和结果	534
10.11 快速回顾	538
10.12 练习题	539
10.13 编程练习	540
第 11 章 二叉树	543
11.1 二叉树	543
11.2 二叉树的遍历	549
11.2.1 中序遍历	549
11.2.2 前序遍历	549

11.2.3	后序遍历	549
11.2.4	二叉树的实现	552
11.3	二叉搜索树	560
11.3.1	search 函数	562
11.3.2	Insert 函数	564
11.3.3	Delete 函数	565
11.4	二叉搜索树分析	572
11.5	二叉树的非递归遍历算法	573
11.5.1	非递归中序遍历	573
11.5.2	非递归前序遍历	574
11.5.3	非递归后序遍历	576
11.6	二叉树遍历和作为参数的函数	577
11.7	AVL(平衡)树	580
11.7.1	AVL 树的插入操作	582
11.7.2	AVL 树的旋转	587
11.7.3	AVL 树的删除操作	598
11.7.4	AVL 树的性能分析	599
11.8	编程示例: Video Store	600
11.9	快速回顾	608
11.10	练习题	609
11.11	编程练习	612
第 12 章	图	614
12.1	初识图	614
12.2	图的定义和符号	615
12.3	图的表示方法	617
12.3.1	邻接矩阵	617
12.3.2	邻接表	618
12.4	图的操作	619
12.5	回顾模板	620
12.6	图的 ADT 定义	620
12.7	图的遍历	624
12.7.1	深度优先遍历	624
12.7.2	广度优先遍历	626
12.8	最短路径算法	628
12.9	最小生成树	634
12.10	拓扑排序	641
12.11	快速回顾	647

12.12	练习题	648
12.13	编程练习	650
第 13 章	标准模板库(STL)II	652
13.1	pair 类	652
13.1.1	比较 pair 类型的对象	654
13.1.2	pair 类型和 make_pair 函数	654
13.2	关联容器	656
13.2.1	关联容器: 集合和多重集合	657
13.2.2	关联容器: 映射和多重映射	661
13.3	容器、相关头文件和迭代器支持	666
13.4	算法	666
13.5	STL 算法分类	667
13.5.1	非修改算法	667
13.5.2	修改算法	667
13.5.3	数值算法	668
13.5.4	堆算法	668
13.5.5	函数对象	669
13.5.6	谓词	674
13.5.7	插入迭代器	674
13.6	STL 算法	676
13.6.1	fill 和 fill_n 函数	676
13.6.2	generate 和 generate_n 函数	678
13.6.3	find、find_if、find_end 和 find_first_of 函数	679
13.6.4	remove、remove_if、remove_copy 和 remove_copy_if 函数	681
13.6.5	replace、replace_if、replace_copy 和 replace_copy_if 函数	685
13.6.6	swap、iter_swap 和 swap_ranges 函数	687
13.6.7	search、search_n、sort 和 binary_search 函数	690
13.6.8	adjacent_find、merge 和 inplace_merge 函数	694
13.6.9	reverse、reverse_copy、rotate、rotate_copy 函数	696
13.6.10	count、count_if、max、max_element、min、min_element 和 random_shuffle 函数	699
13.6.11	for_each 和 transform 函数	702
13.6.12	includes、set_intersection、set_union、set_difference 和 set_symmetric_difference 函数	705
13.6.13	accumulate、adjacent_difference、inner_product 和 partial_sum 函数	713
13.7	快速回顾	718
13.8	练习题	720
13.9	编程练习	722