

数据结构、 算法与应用

—— C++ 语言描述

Data Structures,
Algorithms,
and Applications
in C++

(美) Sartaj Sahni 著

汪诗林 孙晓东 等译

王广芳 审校



机械工业出版社
China Machine Press



WCB
McGraw-Hill

计算机科学丛书

数据结构、算法与应用 ——C++语言描述

(美) Sartaj Sahni 著

汪诗林 孙晓东 等译

王广芳 审校



机械工业出版社
China Machine Press

本书是关于计算机科学与工程领域的基础性研究科目之一——数据结构与算法的专著。

本书在简要回顾了基本的C++ 程序设计概念的基础上,全面系统地介绍了队列、堆栈、树、图等基本数据结构,以及贪婪算法、分而治之算法、分枝定界算法等多种算法设计方法,为数据结构与算法的继续学习和研究奠定了一个坚实的基础。更为可贵的是,本书不仅仅介绍了理论知识,还提供了50多个应用实例及600多道练习题。

本书内容广博权威,结构清晰合理,是一本全新的有关数据结构与算法的教材,对于计算机科学与工程领域的从业人员也是一本很好的参考书。

Sartaj Sahni: Data Structures, Algorithms, and Applications in C++.

Original edition copyright © 1998 by McGraw-Hill. All rights reserved.

Chinese edition copyright © 2000 by China Machine Press. All rights reserved.

本书中文简体字版由美国麦格劳-希尔公司授权机械工业出版社独家出版。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。

版权所有,侵权必究。

本书版权登记号: 图字: 01-1999-0113

图书在版编目(CIP)数据

数据结构、算法与应用: C++ 语言描述 / (美) 萨尼 (Sahni, S.) 著; 汪诗林等译.
- 北京: 机械工业出版社, 2000.1

(计算机科学丛书)

书名原文: Data Structures, Algorithms, and Applications in C++

ISBN 7-111-07645-1

I. 数… II. ①萨… ②汪… III. C语言-程序设计 IV. TP312

中国版本图书馆CIP数据核字(1999)第65279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: 陈 谊

北京牛山世兴印刷厂印刷: 新华书店北京发行所发行

2000年1月第1版·2001年3月第3次印刷

787mm×1092mm 1/16·34.5印张

印数: 10 001-13 000册

定价: 49.00元

凡购本书,如有倒页、脱页、缺页,由本社发行部调换

译者序

在可视化程序设计平台广泛流行和应用的今天，程序设计不再是一件神秘的、专业性的工作，很多非计算机专业的人员都可以亲自动手设计应用程序。这似乎让人觉得，只要掌握了一门可视化程序设计语言，人人都可以成为编程高手，但事实并非如此。要想成为一个熟练的专业化程序设计人员，他（她）至少应该满足以下三个条件：一是能够熟练地选择和设计各种数据结构和算法，二是熟练掌握一门程序设计语言，三是熟知应用领域的相关知识。其中后两个条件比较容易实现，而第一个条件则需要花相当的时间和精力才能达到，它是区分一个程序设计人员水平高低的重要标志。之所以如此，是因为在绝大多数应用程序中都需要广泛使用各种各样的数据结构和算法。缺少数据结构和算法的深厚功底，很难设计出高水平的具有专业水准的应用程序。

本书的作者Sartaj Sahni 博士，多年来一直从事数据结构和算法方面的研究和教育工作，具有丰富的教学经验，曾获得IEEE计算机分会1997年Taylor L.Booth教育奖。他撰写了多部有关数据结构和算法方面的著作，本书是他在该领域为广大读者奉献的又一力作。

全书共包含三个部分，第一部分主要回顾一些重要的C++ 程序设计概念以及算法分析与评价的方法。第二部分首先对各种数据描述方法进行了精辟地概括，然后依次介绍了数组、矩阵、堆栈、队列、字典、二叉树、优先队列、竞赛树和图等基本数据结构，对于每一种数据结构都分别采用若干个来自不同领域的应用实例进行了具体的演示。第三部分重点介绍了一些常用的算法设计方法及应用，如贪婪算法、分而治之算法、动态规划方法、回溯算法和分枝定界算法。

本书的最大特色就是强调应用。通过现实生活中的许多应用实例具体演示了书中所介绍的各种数据结构和算法设计方法。根据实例，读者不但可以印证许多基本概念，而且能加深对它们的理解，从而更好地掌握相应的数据结构和算法并能达到熟练应用。通过把应用与理论知识紧密结合，极大地激发了读者学习数据结构和算法的兴趣。

如果你是一名程序设计新手，本书可以为你架起一座桥梁，使你如愿以偿地跨入专业程序设计人员的行列；如果你已经是一名专业程序设计人员，本书可以使你的程序设计水平更上一层楼。当然，如果你是一名计算机专业的在校学生，本书将是一本非常理想的关于数据结构和算法课程的教材或参考书。

参加本书翻译工作的有汪诗林、孙晓东、蒋艳凤、孙海燕、刘娜、郑倩冰、史军慧、陈海燕，由王广芳教授和汪诗林博士完成全书的审校工作。

因时间仓促，翻译若有不妥之处，恳请读者批评指正。

译者

1999年9月于长沙

译者介绍



汪诗林，1968年3月生，国防科技大学计算机学院在职博士。近年来主要从事计算机软件、数据库、多媒体及虚拟现实等领域的教学和科研工作，独立完成多项软件研制任务，共发表教学和科研论文近30篇，获部委级科技进步成果二等奖2项，三等奖4项，获校级优秀教学成果三等奖1项。编写及编译教材各1部（《数字逻辑》、《最新人工智能语言——CommonLisp及CLOS的系统开发方法》）。1997年参加全国第一届863高级技术人才培训班。



王广芳，1938年2月生，国防科技大学计算机学院教授。多年来从事计算机软件的教学工作和科研工作，特别是数据结构、操作系统的教学与研究工作。编著出版《数据结构》、《操作系统原理与方法》、《操作系统原理》等教材。曾获国家级优秀教学成果一等奖1项，部委级优秀教学成果二等奖1项。参加多项有关计算机软件的研制工作，特别是有关操作系统的研制工作，曾获部委级科技进步一等奖2项、二等奖3项、三等奖2项。

前言

有关数据结构和算法的研究是计算机科学与工程的基础性研究之一，掌握该领域的知识对于我们利用计算机资源高效地开发计算机程序非常必要。因此，目前所有的计算机专业都开设了一到两门相关的课程来讲授这方面的知识。一般来说，第一门程序设计课程主要向学生介绍基本的数据结构（如堆栈和队列）和基本的算法（如排序和矩阵运算），第二门程序设计课程进一步介绍更多的数据结构和算法，其后可设置一到两门课程专门致力于对数据结构和算法的研究。

计算机本科专业课程设置过多的问题已迫使许多院校对专业内容进行了大量合并，以尽量减少课程的数量。比如在佛罗里达大学，本科学士生仅开设了一门数据结构和算法课程（一个学期）。在学习本课程以前，要求学生们已经学习过一门C++程序设计课程（一个学期）和其他有关离散数学/结构方面的课程。

本书的编写目的有两个，一是作为相关专业课程的教材，二是作为参考材料，为其他进一步研究数据结构和算法的课程服务。全书共分三个部分。第一部分包含第1章和第2章，主要帮助读者回顾一下C++程序设计的概念以及算法分析与测量方法。对C语言程序设计很熟悉的学生应能通过第1章来跨越C与C++之间的鸿沟。第1章并不是C++的入门介绍，它给出了大多数容易为学生们所忽视的C++概念，如参数传递方式、模板函数、递归、动态存储分配、类、异常的引发和捕获等。其他更高级的C++概念（如继承、虚拟函数和抽象类等）则在首次出现的章节内加以介绍。第2章首先回顾了各种分析程序性能的方法——操作计数、执行步数和渐进符号（ O 、 Ω 、 Θ 、 o ），然后回顾了实际测量程序性能的方法。在第2章所给出的应用程序中，解决了许多在程序设计入门课程中所研究的基本问题（也是很经典的问题），如简单的排序算法（冒泡排序、选择排序、插入排序、计数排序）；简单的搜索算法（顺序搜索、折半搜索）；利用Horner法则进行多项式求值以及各种矩阵运算（如矩阵加、矩阵转置、矩阵乘）。尽管第2章的主要目的是研究程序性能分析和测量的方法，但同时也让每个学生都能熟悉一些基本的算法。

本书的第二部分包含第3~12章，它们对数据结构进行了深入的研究。第3章架设起数据结构研究的骨架，主要介绍了各种描述数据的方法——公式化描述、链表描述、模拟指针和间接寻址，同时根据每一种方法，分别设计了一个相应的C++类来描述线性表数据结构。在第3章结束的时候，我们根据各种不同数据描述方法在描述线性表结构时的性能表现，对这些方法进行了具体比较。第二部分的其他章节分别对其他各种数据结构进行了描述（采用第3章中所介绍的数据描述方法），如数组和矩阵（第4章）、堆栈（第5章）、队列（第6章）、字典（第7章和第11章）、二叉树（第8章）、优先队列（第9章）、竞赛树（第10章）和图（第12章）。

本书的第三部分包含第13~17章，主要研究了一些常用的算法设计方法，如贪婪算法（第13章）、分而治之算法（第14章）、动态规划（第15章）、回溯算法（第16章）和分枝定界算法（第17章）。此外，14.4节给出了两种下限的证明（最小最大问题和排序问题）；9.5.2节给出了机器调度的近似算法；10.5节给出了箱子装载算法；13.3.2节给出了0/1背包问题求解算法。此外，9.5.2节还介绍了NP-复杂问题。

本书的一个唯一特性就是强调应用，通过多个应用实例来演示各种数据结构和算法设计方法。在每一章的最后一节都针对该章所介绍的数据结构或算法设计方法给出了相应的具体应用。也有不少章节一开始就介绍了相关的应用。我们给出了大量来自不同领域的应用：排序（冒泡排序、选择排序、插入排序、计数排序、堆排序、归并排序、快速排序、箱子排序、基数排序和拓扑排序）；矩阵运算（矩阵加、矩阵转置、矩阵乘）；电路设计自动化（搜索电路网组、布线路由、元件折叠、电子布线、设置信号放大器、交叉分布、电路板排列）；压缩编码（LZW压缩、霍夫曼编码、可变位长编码）；计算几何（凸包和最近点对）；仿真（工厂仿真）；图像处理（图元标注）；趣味数学（汉诺塔、残缺棋盘、迷宫）；调度（LPT调度）；优化（箱子装载、货箱装船、0/1背包、矩阵乘法链）；统计（直方图、寻找最大值和最小值、寻找第k个最小值）；图论（生成树、图元、最短路径、最大完备子图、二分覆盖和旅行商）。在引入这些应用时，不需要学生具有相关领域的预备知识，因为本书中所提供的材料已经覆盖了相应的知识。相信这些应用实例可以为学生的学习提供不少趣味。

通过把应用实例与数据结构和算法设计方法紧密结合，我们希望能更好地激发学生的学习兴趣。另外，通过本书所提供的大约600个练习以及相关的Web站点，还可以使所学知识更进一步地巩固和丰富。

WEB 站点

本书所在站点的URL地址为：www.cise.ufl.edu/~sahni/dsac。访问该站点，你可以得到本书中的所有程序以及示例数据和所产生的输出。示例数据并不是特意设计的测试数据，而是仅供程序所用，以便比较给定的输出和实际产生的输出。

本书中的所有程序都已经在Borland C++ 5.01和GNU C++ 2.7.2.1下编译通过并能够正确运行。相应的文件都已经被压缩，并以两个zip文件的形式（一个对应于Borland C++，一个对应于GNU C++）放置在站点上。书中的程序编号与文件名的对应关系见readme文件（该文件也包含在相应的zip文件中）。

在Web站点中还提供了以下材料：每章练习的解答；一些测试样本以及相应的测试结果；一些附加的应用实例以及对本书中部分材料的进一步讨论。

怎样使用本书

可以采取多种方式来使用本书讲授数据结构和/或算法。教师应该根据学生的背景知识、希望强调哪些应用以及课程的学时数来做出决定。下面我们给出了几个可能的课程安排。我们建议作业的形式是让学生编写并调试一些程序，开始时程序可以比较短，随着课程的深入，程序将逐渐变大。学生应根据课堂上所讲授的主题同步阅读书中的相关内容。

两季度课程安排——第一季度
(一周回顾, 数据结构与算法课程)

周	主 题	内 容
1	回顾C++及程序性能	第1章和第2章布置作业1
2	公式化描述和链表描述	3.1 ~ 3.4 节, 完成作业1
3	链表, 间接寻址和模拟指针	3.4 ~ 3.7 节, 布置作业2
4	箱子排序和等价类	3.8 节, 完成作业2
5	数组和矩阵	第4章, 测试
6	堆栈和队列	第5章和第6章, 布置作业3
7	跳表和散列	第7章, 完成作业3
8	二叉树和其他树	8.1 ~ 8.9 节, 布置作业4
9	合并/搜索应用, 堆和堆排序	8.10.2 节, 9.1 ~ 9.3, 9.51 节, 完成作业4
10	左高树, 霍夫曼编码和竞赛树	9.4 ~ 9.5 节和第10章

两季度课程安排——第二季度
(数据结构与算法课程)

周	主 题	内 容
1	搜索树, AVL树或红黑树, 直方图	第11章, 布置作业1
2	图	12.1 ~ 12.7 节, 完成作业1
3	图	12.8 ~ 12.11 节, 布置作业2
4	贪婪算法	13.1 ~ 13.3.5 节, 完成作业2
5	贪婪算法和分而治之算法	13.3.6和14.1 节, 布置作业3
6	分而治之算法应用	14.2 节, 测试
7	求解递归式, 下限和动态编程	14.3, 14.4和15.1 节, 完成作业3
8	动态编程应用	15.2.1和15.2.2 节, 布置作业4
9	动态编程应用	15.2.3 ~ 15.2.5 节, 完成作业4
10	回溯和分枝定界算法	第16章和第17章

一学期课程安排
(两周回顾, 数据结构课程)

周	主 题	内 容
1	回顾C++	第1章、布置作业1
2	回顾程序性能	第2章
3	公式化描述和链表描述	3.1 ~ 3.4 节、完成作业1
4	链表、间接寻址和模拟指针	3.4 ~ 3.7 节、布置作业2
5	箱子排序和等价类	3.8 节
6	数组和矩阵	第4章、完成作业2、第1次测试
7	堆栈及其应用	第5章、布置作业3
8	队列及其应用	第6章
9	跳表和散列	第7章、完成作业3
10	二叉树和其他树	8.1 ~ 8.9 节、布置作业4
11	合并/搜索应用	8.10.2 节、第2次测试
12	优先队列、堆排序和霍夫曼编码	第9章、完成作业4
13	竞赛树和箱子装载	第10章、布置作业5
14	搜索树, AVL树或红黑树、直方图	第11章
15	图	12.1 ~ 12.7 节、完成作业5
16	图、归并排序和快速排序	12.8 ~ 12.11, 14.2.2 ~ 14.2.3 节

一学期课程安排
(一周回顾, 数据结构与算法课程)

周	主 题	内 容
1	回顾程序性能	第1章和第2章
2	公式化描述和链表描述	3.1 ~ 3.4 节, 布置作业1
3	链表、间接寻址和模拟指针	3.4 ~ 3.8 节
4	数组和矩阵	第4章, 完成作业1
5	堆栈和队列及其应用	第5章和第6章, 布置作业2
6	跳表和散列	第7章, 测试, 完成作业2
7	二叉树和其他树	8.1 ~ 8.9 节, 布置作业3
8	合并/搜索应用, 堆和堆排序	8.10.2, 9.1 ~ 9.3和9.5.1 节, 第2次测试
9	左高树、霍夫曼编码和竞赛树	9.4 ~ 9.5 节, 第10章, 完成作业3
10	搜索树, AVL树或红黑树, 直方图	第11章, 布置作业4
11	图	12.1 ~ 12.7 节
12	图和贪婪算法	12.8 ~ 12.11, 13.1 ~ 13.2 节, 完成作业4
13	货箱装载, 0/1背包, 最短路径和生成树	13.3 节, 布置作业5
14	分而治之算法	第14章
15	动态编程	第15章, 完成作业5
16	回溯和分枝定界算法	第16章和第17章

Sartaj Sahni
Gainesville
1997年10月

目 录

译者序
前言

第一部分 预备知识

第1章 C++程序设计	1
1.1 引言	1
1.2 函数与参数	2
1.2.1 传值参数	2
1.2.2 模板函数	3
1.2.3 引用参数	3
1.2.4 常量引用参数	4
1.2.5 返回值	4
1.2.6 递归函数	5
1.3 动态存储分配	9
1.3.1 操作符new	9
1.3.2 一维数组	9
1.3.3 异常处理	10
1.3.4 操作符delete	10
1.3.5 二维数组	10
1.4 类	13
1.4.1 类Currency	13
1.4.2 使用不同的描述方法	18
1.4.3 操作符重载	20
1.4.4 引发异常	22
1.4.5 友元和保护类成员	23
1.4.6 增加#ifdef, #define和#endif语句	24
1.5 测试与调试	24
1.5.1 什么是测试	24
1.5.2 设计测试数据	26
1.5.3 调试	28
1.6 参考及推荐读物	29
第2章 程序性能	30
2.1 引言	30
2.2 空间复杂性	31

2.2.1 空间复杂性的组成	31
2.2.2 举例	35
2.3 时间复杂性	37
2.3.1 时间复杂性的组成	37
2.3.2 操作计数	37
2.3.3 执行步数	44
2.4 渐进符号 (O 、 Ω 、 Θ 、 o)	55
2.4.1 大写 O 符号	56
2.4.2 Ω 符号	58
2.4.3 Θ 符号	59
2.4.4 小写 o 符号	60
2.4.5 特性	60
2.4.6 复杂性分析举例	61
2.5 实际复杂性	66
2.6 性能测量	68
2.6.1 选择实例的大小	69
2.6.2 设计测试数据	69
2.6.3 进行实验	69
2.7 参考及推荐读物	74

第二部分 数据结构

第3章 数据描述	75
3.1 引言	75
3.2 线性表	76
3.3 公式化描述	77
3.3.1 基本概念	77
3.3.2 异常类NoMem	79
3.3.3 操作	79
3.3.4 评价	83
3.4 链表描述	86
3.4.1 类ChainNode 和Chain	86
3.4.2 操作	88
3.4.3 扩充类Chain	91
3.4.4 链表遍历器类	92

3.4.5 循环链表	93	5.2 派生类和继承	162
3.4.6 与公式化描述方法的比较	94	5.3 公式化描述	163
3.4.7 双向链表	95	5.3.1 Stack的效率	164
3.4.8 小结	96	5.3.2 自定义Stack	164
3.5 间接寻址	99	5.4 链表描述	166
3.5.1 基本概念	99	5.5 应用	169
3.5.2 操作	100	5.5.1 括号匹配	169
3.6 模拟指针	102	5.5.2 汉诺塔	170
3.6.1 SimSpace的操作	103	5.5.3 火车车厢重排	172
3.6.2 采用模拟指针的链表	106	5.5.4 开关盒布线	176
3.7 描述方法的比较	110	5.5.5 离线等价类问题	178
3.8 应用	111	5.5.6 迷宫老鼠	180
3.8.1 箱子排序	111	5.6 参考及推荐读物	188
3.8.2 基数排序	116	第6章 队列	189
3.8.3 等价类	117	6.1 抽象数据类型	189
3.8.4 凸包	122	6.2 公式化描述	190
3.9 参考及推荐读物	127	6.3 链表描述	194
第4章 数组和矩阵	128	6.4 应用	197
4.1 数组	128	6.4.1 火车车厢重排	197
4.1.1 抽象数据类型	128	6.4.2 电路布线	201
4.1.2 C++数组	129	6.4.3 识别图元	204
4.1.3 行主映射和列主映射	129	6.4.4 工厂仿真	206
4.1.4 类Array1D	131	6.5 参考及推荐读物	217
4.1.5 类Array2D	133	第7章 跳表和散列	218
4.2 矩阵	137	7.1 字典	218
4.2.1 定义和操作	137	7.2 线性表描述	219
4.2.2 类Matrix	138	7.3 跳表描述	222
4.3 特殊矩阵	141	7.3.1 理想情况	222
4.3.1 定义和应用	141	7.3.2 插入和删除	223
4.3.2 对角矩阵	143	7.3.3 级的分配	224
4.3.3 三对角矩阵	144	7.3.4 类SkipNode	224
4.3.4 三角矩阵	145	7.3.5 类SkipList	225
4.3.5 对称矩阵	146	7.3.6 复杂性	229
4.4 稀疏矩阵	149	7.4 散列表描述	229
4.4.1 基本概念	149	7.4.1 理想散列	229
4.4.2 数组描述	149	7.4.2 线性开型寻址散列	230
4.4.3 链表描述	154	7.4.3 链表散列	234
第5章 堆栈	161	7.5 应用——文本压缩	238
5.1 抽象数据类型	161	7.5.1 LZW压缩	239

7.5.2 LZW压缩的实现	239	9.4.5 初始化最大HBLT	289
7.5.3 LZW解压缩	243	9.4.6 类MaxHBLT	289
7.5.4 LZW解压缩的实现	243	9.5 应用	293
7.6 参考及推荐读物	247	9.5.1 堆排序	293
第8章 二叉树和其他树	248	9.5.2 机器调度	294
8.1 树	248	9.5.3 霍夫曼编码	297
8.2 二叉树	251	9.6 参考及推荐读物	302
8.3 二叉树的特性	252	第10章 竞赛树	303
8.4 二叉树描述	253	10.1 引言	303
8.4.1 公式化描述	253	10.2 抽象数据类型WinnerTree	306
8.4.2 链表描述	254	10.3 类WinnerTree	307
8.5 二叉树常用操作	256	10.3.1 定义	307
8.6 二叉树遍历	256	10.3.2 类定义	307
8.7 抽象数据类型BinaryTree	259	10.3.3 构造函数、析构函数及Winner 函数	308
8.8 类BinaryTree	260	10.3.4 初始化赢者树	308
8.9 抽象数据类型及类的扩充	263	10.3.5 重新组织比赛	310
8.9.1 输出	263	10.4 输者树	311
8.9.2 删除	264	10.5 应用	312
8.9.3 计算高度	264	10.5.1 用最先匹配法求解箱子装载 问题	312
8.9.4 统计节点数	265	10.5.2 用相邻匹配法求解箱子装载 问题	316
8.10 应用	265	第11章 搜索树	319
8.10.1 设置信号放大器	265	11.1 二叉搜索树	320
8.10.2 在线等价类	268	11.1.1 基本概念	320
8.11 参考及推荐读物	275	11.1.2 抽象数据类型BSTree和 IndexedBSTree	321
第9章 优先队列	276	11.1.3 类BSTree	322
9.1 引言	276	11.1.4 搜索	322
9.2 线性表	277	11.1.5 插入	323
9.3 堆	278	11.1.6 删除	324
9.3.1 定义	278	11.1.7 类DBSTree	326
9.3.2 最大堆的插入	279	11.1.8 二叉搜索树的高度	327
9.3.3 最大堆的删除	279	11.2 AVL树	328
9.3.4 最大堆的初始化	280	11.2.1 基本概念	328
9.3.5 类MaxHeap	281	11.2.2 AVL树的高度	328
9.4 左高树	285	11.2.3 AVL树的描述	329
9.4.1 高度与宽度优先的最大及最小 左高树	285	11.2.4 AVL搜索树的搜索	329
9.4.2 最大HBLT的插入	287		
9.4.3 最大HBLT的删除	287		
9.4.4 合并两棵最大HBLT	287		

11.2.5 AVL搜索树的插入	329	12.7.5 链接类	382
11.2.6 AVL搜索树的删除	332	12.8 图的遍历	386
11.3 红-黑树	334	12.8.1 基本概念	386
11.3.1 基本概念	334	12.8.2 邻接矩阵的遍历函数	387
11.3.2 红-黑树的描述	336	12.8.3 邻接链表的遍历函数	388
11.3.3 红-黑树的搜索	336	12.9 语言特性	389
11.3.4 红-黑树的插入	336	12.9.1 虚函数和多态性	389
11.3.5 红-黑树的删除	339	12.9.2 纯虚函数和抽象类	391
11.3.6 实现细节的考虑及复杂性分析	343	12.9.3 虚基类	391
11.4 B-树	344	12.9.4 抽象类和抽象数据类型	393
11.4.1 索引顺序访问方法	344	12.10 图的搜索算法	394
11.4.2 m叉搜索树	345	12.10.1 宽度优先搜索	394
11.4.3 m序B-树	346	12.10.2 类Network	395
11.4.4 B-树的高度	347	12.10.3 BFS的实现	395
11.4.5 B-树的搜索	348	12.10.4 BFS的复杂性分析	396
11.4.6 B-树的插入	348	12.10.5 深度优先搜索	397
11.4.7 B-树的删除	350	12.11 应用	399
11.4.8 节点结构	353	12.11.1 寻找路径	399
11.5 应用	354	12.11.2 连通图及其构件	400
11.5.1 直方图	354	12.11.3 生成树	402
11.5.2 用最优匹配法求解箱子装载 问题	357		
11.5.3 交叉分布	359	第三部分 算法设计方法	
11.6 参考及推荐读物	363	第13章 贪婪算法	405
第12章 图	365	13.1 最优化问题	405
12.1 基本概念	365	13.2 算法思想	406
12.2 应用	366	13.3 应用	409
12.3 特性	368	13.3.1 货箱装船	409
12.4 抽象数据类型Graph和Digraph	370	13.3.2 0/1背包问题	410
12.5 无向图和有向图的描述	371	13.3.3 拓扑排序	412
12.5.1 邻接矩阵	371	13.3.4 二分覆盖	415
12.5.2 邻接压缩表	373	13.3.5 单源最短路径	421
12.5.3 邻接链表	374	13.3.6 最小耗费生成树	424
12.6 网络描述	375	13.4 参考及推荐读物	433
12.7 类定义	376	第14章 分而治之算法	434
12.7.1 不同的类	376	14.1 算法思想	434
12.7.2 邻接矩阵类	377	14.2 应用	440
12.7.3 扩充Chain类	380	14.2.1 残缺棋盘	440
12.7.4 类LinkedBase	381	14.2.2 归并排序	443
		14.2.3 快速排序	447

14.2.4 选择	452	第16章 回溯	492
14.2.5 距离最近的点对	454	16.1 算法思想	492
14.3 解递归方程	462	16.2 应用	496
14.4 复杂性的下限	463	16.2.1 货箱装船	496
14.4.1 最小最大问题的下限	464	16.2.2 0/1背包问题	503
14.4.2 排序算法的下限	465	16.2.3 最大完备子图	506
第15章 动态规划	467	16.2.4 旅行商问题	508
15.1 算法思想	467	16.2.5 电路板排列	510
15.2 应用	469	第17章 分枝定界	516
15.2.1 0/1背包问题	469	17.1 算法思想	516
15.2.2 图像压缩	471	17.2 应用	519
15.2.3 矩阵乘法链	476	17.2.1 货箱装船	519
15.2.4 最短路径	480	17.2.2 0/1背包问题	526
15.2.5 网络的无交叉子集	483	17.2.3 最大完备子图	528
15.2.6 元件折叠	486	17.2.4 旅行商问题	529
15.3 参考及推荐读物	491	17.2.5 电路板排列	532

第一部分 预备知识

第1章 C++程序设计

大家好！现在我们将要开始一个穿越“数据结构、算法和程序”这个抽象世界的特殊旅程，以解决现实生活中的许多难题。在程序开发过程中通常需要做到如下两点：一是高效地描述数据；二是设计一个好的算法，该算法最终可用程序来实现。要想高效地描述数据，必须具备数据结构领域的专门知识；而要想设计一个好的算法，则需要算法设计领域的专门知识。

在着手研究数据结构和算法设计方法之前，需要你能够熟练地运用C++编程并分析程序，这些基本的技能通常是从C++课程以及其他分散的课程中学到的。本书的前两章旨在帮助你回顾一下这些技能，其中的许多内容你可能已经很熟悉了。

本章我们将回顾C++的一些特性。因为不是针对C++新手，因此没有介绍诸如赋值语句、if语句和循环语句（如for和while）等基本结构，而是主要介绍一些可能已经被你忽略的C++特性：

- 参数传递方式（如传值、引用和常量引用）。
- 函数返回方式（如返回值、引用和常量引用）。
- 模板函数。
- 递归函数。
- 常量函数。
- 内存分配和释放函数：new与delete。
- 异常处理结构：try、catch和throw。
- 类与模板类。
- 类的共享成员、保护成员和私有成员。
- 友元。
- 操作符重载。

本章中没有涉及的其他C++特性将在后续章节中在需要的时候加以介绍。本章还给出了如下应用程序的代码：

- 一维和二维数组的动态分配与释放。
- 求解二次方程。
- 生成 n 个元素的所有排列方式。
- 寻找 n 个元素中的最大值。

此外，本章还给出了如何测试和调试程序的一些技巧。

1.1 引言

在检查程序的时候我们应该问一问：

- 它正确吗？

- 它容易读懂吗?
- 它有完善的文档吗?
- 它容易修改吗?
- 它在运行时需要多大内存?
- 它的运行时间有多长?
- 它的通用性如何? 能不能不加修改就可以用它来解决更大范围的问题?
- 它可以在多种机器上编译和运行吗? 或者说需要经过修改才能在不同的机器上运行吗?

上述问题的相对重要性取决于具体的应用环境。比如, 如果我们正在编写一个只需运行一次即可丢弃的程序, 那么主要关心的应是程序的正确性、内存需求、运行时间以及能否在一台机器上编译和运行。不管具体的应用环境是什么, 正确性总是程序的一个最重要的特性。一个不正确的程序, 不管它有多快、有多么好的通用性、有多么完善的文档, 都是毫无意义的(除非它变正确了)。尽管我们无法详细地介绍提高程序正确性的技术, 但可以为大家提供一些程序正确性的验证方法以及公认的一些良好的程序设计习惯, 它们可以帮助你编写正确的代码。我们的目标是教会你如何开发正确的、精致的、高效的程序。

1.2 函数与参数

1.2.1 传值参数

考察函数 Abc (见程序1-1)。该函数用来计算表达式 $a+b+b*c+(a+b-c)/(a+b)+4$, 其中 a 、 b 和 c 是整数, 结果也是一个整数。

程序1-1 计算一个整数表达式

```
int Abc(int a, int b, int c)
{
    return a+b+b*c+(a+b-c)/(a+b)+4;
}
```

在程序1-1中, a 、 b 和 c 是函数 Abc 的形式参数(formal parameter), 类型均为整型。如果在如下语句中调用函数 Abc :

```
z = Abc(2,x,y)
```

那么, 2, x 和 y 分别是对应于 a 、 b 和 c 的实际参数(actual parameter)。当 $Abc(2,x,y)$ 被执行时, a 被赋值为2; b 被赋值为 x ; c 被赋值为 y 。如果 x 和 y 不是int类型, 那么在把它们值赋给 b 和 c 之前, 将首先对它们进行类型转换。例如, 如果 x 是float类型, 其值为3.8, 那么 b 将被赋值为3。在程序1-1中, 形式参数 a 、 b 和 c 都是传值参数(value parameter)。

运行时, 与传值形式参数相对应的实际参数的值将在函数执行之前被复制给形式参数, 复制过程是由该形式参数所属数据类型的复制构造函数(copy constructor)完成的。如果实际参数与形式参数的数据类型不同, 必须进行类型转换, 从实际参数的类型转换为形式参数的类型, 当然, 假定这样的类型转换是允许的。

当函数运行结束时, 形式参数所属数据类型的析构函数(destructor)负责释放该形式参数。当一个函数返回时, 形式参数的值不会被复制到对应的实际参数中。因此, 函数调用不会修改实际参数的值。