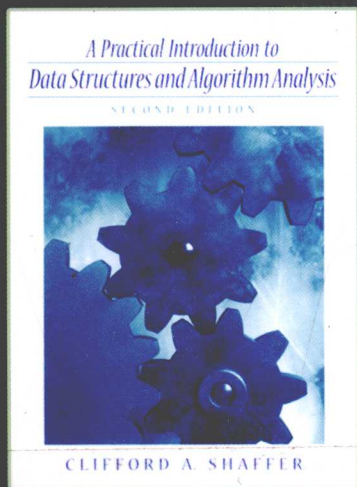


数据结构与算法分析 (C++版)(第二版)

A Practical Introduction to Data Structures and Algorithm
Analysis (C++ Edition), Second Edition



[美] Clifford A. Shaffer 著

张铭 刘晓丹 等译



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

<http://www.phei.com.cn>

国外计算机科学教材系列

数据结构与算法分析

(C++ 版)

(第二版)

A Practical Introduction to Data Structures and
Algorithm Analysis
Second Edition

[美] Clifford A. Shaffer 著

张 铭 刘晓丹 等译

电子工业出版社
Publishing House of Electronics Industry
北京 · BEIJING

内 容 简 介

本书采用程序员最爱用的面向对象C++语言来描述数据结构和算法,并把数据结构原理和算法分析技术有机地结合在一起,系统介绍了各种类型的数据结构和排序、检索的各种方法。作者非常注意对每一种数据结构的存储方法及有关算法进行分析比较。书中还引入了一些比较高级的数据结构与先进的算法分析技术,并介绍了可计算性理论的一般知识。本版的重要改进在于引入了参数化的模板,从而提高了算法中数据类型的通用性,支持高效的代码重用。

本书适合作为大专院校计算机软件专业与计算机应用专业学生的教材和参考书,也适合计算机工程技术人员参考。

Authorized translation from the English language edition, entitled A Practical Introduction to Data Structures and Algorithm Analysis (C++ Edition), Second Edition, ISBN: 9780130284464 by Clifford A. Shaffer, published by Pearson Education, Inc, publishing as Prentice Hall, Copyright © 2001 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.

CHINESE SIMPLIFIED language edition published by PEARSON EDUCATION ASIA LTD., and PUBLISHING HOUSE OF ELECTRONICS INDUSTRY. Copyright © 2010.

本书中文简体字翻译版由电子工业出版社和Pearson Education培生教育出版北亚洲有限公司合作出版。未经出版者预先书面许可,不得以任何方式复制或抄袭本书的任何部分。

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。

版权贸易合同登记号 图字:01-2001-5093

图书在版编目(CIP)数据

数据结构与算法分析: C++版: 第2版 / (美) 谢弗 (Shaffer, C. A.) 著; 张铭等译.

北京: 电子工业出版社, 2010.1

(国外计算机科学教材系列)

书名原文: A Practical Introduction to Data Structures and Algorithm Analysis (C++ Edition), Second Edition

ISBN 978-7-121-08958-9

I. 数… II. ①谢… ②张… III. ①数据结构-教材 ②算法分析-教材 ③C语言-程序设计-教材
IV. TP311.12 TP312

中国版本图书馆CIP数据核字(2009)第089186号

责任编辑: 马 岚

印 刷: 北京东光印刷厂

装 订: 三河市皇庄路通装订厂

出版发行: 电子工业出版社

北京市海淀区万寿路173信箱 邮编: 100036

开 本: 787 × 1092 1/16 印张: 21.5 字数: 550千字

印 次: 2010年1月第1次印刷

定 价: 35.00元

凡所购买电子工业出版社的图书有缺损问题, 请向购买书店调换; 若书店售缺, 请与本社发行部联系。联系及邮购电话: (010) 88254888。

质量投诉请发邮件至 zltts@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线: (010) 88258888。

出版说明

21世纪初的5至10年是我国国民经济和社会发展的关键时期,也是信息产业快速发展的关键时期。在我国加入WTO后的今天,培养一支适应国际化竞争的一流IT人才队伍是我国高等教育的重要任务之一。信息科学和技术方面人才的优劣与多寡,是我国面对国际竞争时成败的关键因素。

当前,正值我国高等教育特别是信息科学领域的教育调整、变革的重大时期,为使我国教育体制与国际化接轨,有条件的高等院校正在为某些信息学科和技术课程使用国外优秀教材和优秀原版教材,以使我国在计算机教学上尽快赶上国际先进水平。

电子工业出版社秉承多年来引进国外优秀图书的经验,翻译出版了“国外计算机科学教材系列”丛书,这套教材覆盖学科范围广、领域宽、层次多,既有本科专业课程教材,也有研究生课程教材,以适应不同院系、不同专业、不同层次的师生对教材的需求,广大师生可自由选择和自由组合使用。这些教材涉及的学科方向包括网络与通信、操作系统、计算机组织与结构、算法与数据结构、数据库与信息处理、编程语言、图形图像与多媒体、软件工程等。同时,我们也适当引进了一些优秀英文原版教材,本着翻译版本和英文原版并重的原则,对重点图书既提供英文原版又提供相应的翻译版本。

在图书选题上,我们大都选择国外著名出版公司出版的高校教材,如Pearson Education培生教育出版集团、麦格劳-希尔教育出版集团、麻省理工学院出版社、剑桥大学出版社等。撰写教材的许多作者都是蜚声世界的教授、学者,如道格拉斯·科默(Douglas E. Comer)、威廉·斯托林斯(William Stallings)、哈维·戴特尔(Harvey M. Deitel)、尤利斯·布莱克(Uyless Black)等。

为确保教材的选题质量和翻译质量,我们约请了清华大学、北京大学、北京航空航天大学、复旦大学、上海交通大学、南京大学、浙江大学、哈尔滨工业大学、华中科技大学、西安交通大学、国防科学技术大学、解放军理工大学等著名高校的教授和骨干教师参与了本系列教材的选题、翻译和审校工作。他们中既有讲授同类教材的骨干教师、博士,也有积累了几十年教学经验的老教授和博士生导师。

在该系列教材的选题、翻译和编辑加工过程中,为提高教材质量,我们做了大量细致的工作,包括对所选教材进行全面论证;选择编辑时力求达到专业对口;对排版、印制质量进行严格把关。对于英文教材中出现的错误,我们通过与作者联络和网上下载勘误表等方式,逐一进行了修订。

此外,我们还将与国外著名出版公司合作,提供一些教材的教学支持资料,希望能为授课老师提供帮助。今后,我们将继续加强与各高校教师的密切联系,为广大师生引进更多的国外优秀教材和参考书,为我国计算机科学教学体系与国际教学体系的接轨做出努力。

电子工业出版社

教材出版委员会

- | | | |
|----|-----|---|
| 主任 | 杨芙清 | 北京大学教授
中国科学院院士
北京大学信息与工程学部主任
北京大学软件工程研究所所长 |
| 委员 | 王 珊 | 中国人民大学信息学院院长、教授 |
| | 胡道元 | 清华大学计算机科学与技术系教授
国际信息处理联合会通信系统中国代表 |
| | 钟玉琢 | 清华大学计算机科学与技术系教授
中国计算机学会多媒体专业委员会主任 |
| | 谢希仁 | 中国人民解放军理工大学教授
全军网络技术研究中心主任、博士生导师 |
| | 尤晋元 | 上海交通大学计算机科学与工程系教授
上海分布计算技术中心主任 |
| | 施伯乐 | 上海国际数据库研究中心主任、复旦大学教授
中国计算机学会常务理事、上海市计算机学会理事长 |
| | 邹 鹏 | 国防科学技术大学计算机学院教授、博士生导师
教育部计算机基础课程教学指导委员会副主任委员 |
| | 张昆藏 | 青岛大学信息工程学院教授 |

译者序

数据结构与算法分析是一门计算机专业十分重要的基础课,计算机科学各领域及各种应用软件都要使用相关的数据结构和算法。

当面临一个新的设计问题时,设计者需要选择适当的数据结构,并设计出满足一定时间和空间限制的有效算法。本书作者把数据结构和算法分析有机地揉合在一本教材中,有助于读者根据问题的性质选择合理的数据结构,并对时间空间复杂性进行必要的控制。

本书采用当前流行的面向对象的 C++ 语言来描述数据结构和算法,因为 C++ 语言是程序员最广泛使用的语言。因此,程序员可以把本书中的许多算法直接应用于将来的实际项目中。尽管数据结构和算法在设计本质上还是很底层的東西,并不像软件工程大型项目设计那样,对面向对象方法具有直接的依赖性,因此有人会认为并不需要采用面向对象的高级技术来描述底层的算法,但是采用 C++ 语言能够更好地体现抽象数据类型的概念,从而更本质地描述数据结构和算法。为了使本书清晰易懂,作者有意回避了 C++ 的某些重要特性。

这个版本的重要改进是引入了参数化的模板,从而提高了算法中数据类型的通用性,支持高效的代码重用。

本书包括四大部分内容,第一部分是准备工作,介绍了一些基本概念和术语,以及基本的数学知识。

第二部分介绍了最基本的数据结构,依次为线性表(包括栈和队列)、二叉树和树。对每种数据结构的讲解都从其数学特性入手,先介绍抽象数据类型,然后再讨论不同的存储方法,并且研究不同存储方法的可能算法。值得赞赏的是,作者结合算法分析来讨论各种存储方法和算法的利弊,摒弃那些不适宜的方法,这样就调动了读者思维,使其可从中学到考虑问题的方法。这种“授人以渔”的策略使读者在今后设计和应用数据结构时能够全面地考虑各种因素,并选择最佳方案。

作为最常用的算法,排序和检索历来是数据结构讨论的重点问题。这在第三部分的第 9 章和第 10 章中进行了详尽的讨论。排序算法最能体现算法分析的魅力,它的算法速度要求非常高:其中内排序主要考虑的是怎样减少关键码之间的比较次数和记录交换次数,以提高排序速度;而外排序则考虑外存的特性,尽量减少访问操作,以提高排序速度。第 8 章证明了所有基于比较的排序算法的时间代价是 $\Theta(n \log n)$,这也是排序问题的时间代价。检索则考虑怎样提高检索速度,这往往与存储方法有关。书中介绍了几种高效的数据结构,如自组织线性表、散列表、B 树和 B⁺ 树等,都具有极好的检索性能。

第四部分介绍了数据结构的应用与一些高级主题,其中包括图、跳跃表、广义表和稀疏矩阵等更复杂的线性表结构,还包括 Trie 结构、AVL 树等复杂树结构,以及 k-d 树、PR 四分树等空间数据结构。另外,第四部分还简单介绍了求和、递归关系分析和均摊分析等高级算法分析技术,这些技术对于提高程序员的算法分析能力具有重要作用。

本书的前言及第 1 章至第 7 章由张铭翻译,第 8 章至第 15 章由刘晓丹翻译。另外,肖毅、柴雯、肖之屏、刘堃、赵培翔、李丽、王蜀安、张海东、刘振飞和李健等人也参加了本书的翻译工作,在此对他们的辛勤劳动表示感谢。由于水平有限,难免有不妥之处,欢迎批评指正。

前 言

我们研究数据结构的目的是为了学会编写更高效的程序。既然现在的计算机速度一年比一年快,为什么还会需要高效率的程序呢?这是由于人类解决问题的雄心与能力是同步增长的。现代计算技术在计算能力和存储容量上的革命,仅仅提供了计算更复杂问题的有效工具,而程序的高效性要求永远也不会过时。

程序高效性的要求不会,也不应该与合理的设计和简明清晰的编码相矛盾。高效程序的设计基于良好的信息组织和优秀的算法,而不是基于“编程小伎俩”。如果一个程序员没有掌握程序设计简明清晰的基本原理,就不可能编写出有效的程序。反过来讲,简洁的程序需要合理的数据组织和清晰的算法。大多数计算机科学系的课程设置都已意识到,要培养良好的程序设计技能,首先应该强调基本的软件工程原理。因此,一旦程序员学会了程序设计和实现简明清晰的原理,下一步就应该学习有效的数据组织和算法,以提高程序的效率。

途径

本书描述了许多用于表示数据的技术。这些技术体现在以下的原则中:

1. 每一种数据结构和每一个算法都有其时间、空间的代价和效率。当面临一个新的设计问题时,设计者要透彻地掌握权衡时间、空间代价和算法有效性的方法,以适应问题的需要。这就要懂得算法分析的原理,而且还需要了解所使用的物理介质的特性(例如,数据存储存储在磁盘上与存储在主存中时,就有不同的考虑)。
2. 与代价和效率有关的是时空权衡。例如,人们通常增加空间代价来减少运行时间,反之亦然。程序员所面对的时空权衡问题普遍存在于软件设计和实现的各个阶段,因此必须把这个概念牢记在心。
3. 程序员应该充分了解一些现成的方法,以免进行不必要的重复开发工作。因此,学生们需要了解经常使用的数据结构和相关算法。
4. 数据结构服从于应用需求。学生们必须把分析应用需求放在第一位,然后再寻找一种与实际应用相匹配的数据结构。要做到这一点,需要应用上述三条原则。

教学建议

数据结构和算法设计的书籍往往囿于下面两种情形之一:一种是教材,一种是百科全书。有些书籍试图融合这两种编排,但通常是二者都没有组织好,而本书是作为教材来编写的。我相信了解那些用于选择或设计可解决问题的数据结构的基本原理十分重要,会比死记硬背书本内容重要得多。因此,本书中涵盖了大多数(但不是所有的)标准数据结构。为了阐述一些重要原理,也包括了某些并非广泛使用的数据结构。另外,本书中还介绍了一些相对较新但即将得到广泛应用的数据结构。

本书可作为本科生一个学期的教学内容,也可作为专业技术人员的自学教材。读者应该具备编程经验,最好学过相当于两个学期的结构化程序设计语言(如 Pascal 或 C 语言),并且最好懂得一些 C++ 的基本知识。早已熟悉递归的读者具有一定的优势。先修完一门好的离散

数学课程对于数据结构的学习也大有裨益。第2章中给出了一些理解本书所必备的数学预备知识。读者遇到不熟悉的数学问题时,可以查阅这一章中的相关内容。

尽管本书应该一个学期完成,但书中超过了一个学期的内容,这样就可以为教师提供一些选择的余地。二年级学生的基本数据结构和算法分析背景不太多,可以给他们详细地讲解第1~12章的内容,再从第13章中选择一些专题来讲解,我就是这样来给二年级学生讲课的。背景知识更丰富的学生,可以先读第1章,跳过第2章中除“深入学习导读”之外的内容,简要地浏览第3章和第4章(请着重阅读4.1.3小节和4.2小节),然后详细阅读其余的章节。另外,教师可以根据程序设计实习的需要,选择第13章中的某些专题内容。

第13章是针对进行较大的程序设计练习而编写的。我建议所有选修数据结构的学生,都应该做一些高级树结构或其他较复杂的动态数据结构的上机实习,如第12章中的跳跃表或稀疏矩阵。所有这些数据结构都没有二叉检索树难,学完第5章的学生都有能力来实现它们。

本书尽量合理地安排内容顺序。教师可以根据需要自由地重新组织内容。读者掌握了第1至第6章后,以下的内容就相对独立了。显然,外排序依赖于内排序和磁盘文件系统。Kruskal 最小支撑树算法使用了6.2小节关于 UNION/FIND 的算法。9.2小节的自组织线性表谈到了8.3小节讨论的缓冲区置换技术。第14章的讨论基于本书的例题。15.2小节依赖于图论知识。一般情况下,大多数主题都只依赖于同一章中讨论过的内容。

关于 C++

本书中所有示例程序都是用 C++ 来编写的,但也并不想难倒那些对 C++ 不熟悉的读者。在努力保持 C++ 优点的同时,我尽量使示例程序简明、清晰。C++ 在本书中只是作为阐释数据结构的工具,而且实际上只用了 C++ 的一个小子集。特别是书中用到了 C++ 隐蔽实现细节的特性,如类(class)、私有成员(private class member)、构造函数(constructor)、析构函数(destructor)。这些特性支持着一个关键的概念,即体现于抽象数据类型(abstract data type)中的逻辑设计与体现于数据结构中的物理实现的分离。

为了使本书清晰易懂,我完全回避了 C++ 的某些重要特性,并有意排除或尽量少用一些经验丰富的 C++ 程序员常用的特性,如类的层次(class hierarchy)、继承(inheritance)和虚函数(virtual function)。运算符和函数的重载(overloading)也很少使用。C 的原始语义比 C++ 所提供的一些类似功能要好一些。

当然,上述 C++ 的特性在实际程序中是合理的设计基础,但是它们只能掩盖而并没有加强本书中所阐述的原理。例如,类的继承在避免重复编码和降低程序错误率方面很重要,但是从教育学的标准观点来看,类的继承在若干类中分散了数据元素的描述,从而使得程序更难理解。因此,在本书中,当类的继承对阐述观点有明显作用时才使用继承来定义类(例如第5.3.1小节)。但这并不意味着程序员都应该遵从类似的原则,避免代码重复和减少错误是很重要的目标,不要把本书中的示例程序直接复制到你自己的程序中,只把它们看做是对数据结构原理的阐释即可。

我需要做出的最痛苦的选择是:在示例代码中是否使用模板(template)。写本书的第一版时,我决定不使用模板,这是因为考虑到它们的语义对于不熟悉 C++ 语言的人来说掩盖了代码的含义。在随后的几年中,使用 C++ 的计算机科学课程急剧地增加,我和编辑都感到现在的读者比从前的读者更熟悉模板的语义,因此本书在示例代码中大量地使用了模板。

本书中的程序提供了有关数据结构原理的真实阐释,而不是完整实现了用于商用软件的标准数据结构。本书中的示例代码是为了清晰地阐述数据结构是怎样运作的,是对文字阐述的补充。不宜脱离相关的文字阐述而孤立地阅读或使用示例程序,因为大量的背景信息包含在文字阐述中,而不是在代码中。代码是对文字阐述的完善,反之则不然。

这些例子中进行的参数检查,比起从事商业软件设计的程序员在编程中进行的参数检查要少得多。某些参数检查是以调用 Assert 的形式完成的,这是 C 库函数 assert 的改进函数。assert 的输入是一个布尔表达式,一旦这个表达式的值为假(false),程序就立即终止。在实际程序中通常认为这种功能是不必要的,但是该功能对于理解一个数据结构怎样运作十分有用。

在示例程序中,我严格区分了“C++ 实现”和“伪码”。标明“C++ 实现”的示例程序已在一个以上的编译器中真正编译过。“伪码”的示例通常具有与 C++ 接近的语法,但是典型地包含一行以上的更高级描述。当我发现尽管并不十分精确但很简单的描述具有更好的教学效果时,就使用伪码。

几乎每一章都是以“深入学习导读”一节来结束的,这些小节并不是所在那一章的综合参考索引,而是为了通过这些导读书籍或文章给读者提供更广泛的信息和乐趣。有些情况下我也提供了某个知名计算机科学家的重要背景文章。

习题和项目设计

只靠读书是不能学会灵活使用数据结构的。一定要通过编写实际的程序、比较不同的数据结构技术,以观察在一种给定条件下哪种结构更有效。另外,学生也需要通过编写程序来提高他们的算法分析与设计能力。这里提供了 350 多个练习题和项目设计的程序实习题,希望读者能够很好地利用它们。

与作者联系的方法以及相关资料的获取

欢迎读者对本书可能存在的错误提出宝贵意见,作者的邮箱为 shaffer@cs.vt.edu。或者可以通信联系,地址如下:

Clifford Shaffer
Department of Computer Science
Virginia Tech
Blacksburg, VA 24061

读者还可以通过如下网址获取本书配套资料:

<http://people.cs.vt.edu/~shaffer/Book/>

目 录

第一部分 预备知识

第 1 章 数据结构和算法	2
1.1 数据结构的原理	3
1.1.1 学习数据结构的必要性	3
1.1.2 代价与效益	4
1.2 抽象数据类型和数据结构	5
1.3 问题、算法和程序	8
1.4 深入学习导读	9
1.5 习题	10
第 2 章 数学预备知识	13
2.1 集合和关系	13
2.2 常用数学术语	15
2.3 对数	16
2.4 递归	17
2.5 级数求和与递归	19
2.6 数学证明方法	22
2.6.1 反证法	22
2.6.2 数学归纳法	22
2.7 评估	26
2.8 深入学习导读	26
2.9 习题	27
第 3 章 算法分析	31
3.1 概述	31
3.2 最佳、最差和平均情况	34
3.3 换一台更快的计算机,还是换一种更快的算法	35
3.4 渐近分析	37
3.4.1 上限	37
3.4.2 下限	38
3.4.3 Θ 表示法	39
3.4.4 化简法则	40
3.5 程序运行时间的计算	40

3.6 问题的分析	44
3.7 容易混淆的概念	44
3.8 多参数问题	45
3.9 空间代价	46
3.10 实际操作中的一些因素	48
3.11 深入学习导读	49
3.12 习题	49
3.13 项目设计	52

第二部分 基本数据结构

第4章 线性表、栈和队列	54
4.1 线性表	54
4.1.1 顺序表的实现	57
4.1.2 链表	59
4.1.3 线性表实现方法的比较	66
4.1.4 元素的表示	67
4.1.5 双链表	68
4.2 字典 ADT	72
4.3 栈	77
4.3.1 顺序栈	77
4.3.2 链式栈	78
4.3.3 顺序栈与链式栈的比较	79
4.3.4 递归的实现	80
4.4 队列	82
4.4.1 顺序队列	82
4.4.2 链式队列	85
4.4.3 顺序队列与链式队列的比较	86
4.5 深入学习导读	86
4.6 习题	86
4.7 项目设计	89
第5章 二叉树	90
5.1 定义及主要特性	90
5.1.1 满二叉树定理	91
5.1.2 二叉树的抽象数据类型	92
5.2 周游二叉树	93
5.3 二叉树的实现	95
5.3.1 使用指针实现二叉树	95
5.3.2 空间代价	99

5.3.3 使用数组实现完全二叉树	101
5.4 二叉查找树	102
5.5 堆与优先队列	108
5.6 Huffman 编码树	112
5.6.1 建立 Huffman 编码树	113
5.6.2 Huffman 编码及其用法	118
5.7 深入学习导读	120
5.8 习题	120
5.9 项目设计	122
第 6 章 树	124
6.1 树的定义与术语	124
6.1.1 树结点的 ADT	124
6.1.2 树的周游	126
6.2 父指针表示法	126
6.3 树的实现	131
6.3.1 子结点表表示法	131
6.3.2 左子结点/右兄弟结点表示法	132
6.3.3 动态结点表示法	132
6.3.4 动态左子结点/右兄弟结点表示法	134
6.4 K 叉树	134
6.5 树的顺序表示法	135
6.6 深入学习导读	137
6.7 习题	137
6.8 项目设计	139

第三部分 排序和检索

第 7 章 内排序	142
7.1 排序术语及记号	142
7.2 三种代价为 $\Theta(n^2)$ 的排序方法	143
7.2.1 插入排序	143
7.2.2 起泡排序	144
7.2.3 选择排序	145
7.2.4 交换排序算法的时间代价	146
7.3 Shell 排序	147
7.4 快速排序	149
7.5 归并排序	152
7.6 堆排序	155
7.7 分配排序和基数排序	156

7.8 对各种排序算法的实验比较	160
7.9 排序问题的下限	162
7.10 深入学习导读	164
7.11 习题	165
7.12 项目设计	167
第8章 文件管理和外排序	168
8.1 主存储器和辅助存储器	168
8.2 磁盘	170
8.2.1 磁盘结构	170
8.2.2 磁盘访问代价	173
8.3 缓冲区和缓冲池	175
8.4 程序员的文件视图	178
8.5 外部排序	179
8.6 外部排序的简单方法	181
8.7 置换选择排序	182
8.8 多路归并	185
8.9 深入学习导读	187
8.10 习题	187
8.11 项目设计	189
第9章 检索	190
9.1 检索已排序的数组	190
9.2 自组织线性表	191
9.3 集合的检索	195
9.4 散列方法	196
9.4.1 散列函数	196
9.4.2 开散列方法	199
9.4.3 闭散列方法	200
9.5 深入学习导阅读	208
9.6 习题	209
9.7 项目设计	211
第10章 索引技术	212
10.1 线性索引	213
10.2 ISAM	215
10.3 树形索引	216
10.4 2-3 树	218
10.5 B 树	222
10.5.1 B ⁺ 树	223
10.5.2 B 树分析	228

10.6	深度学习导读	229
10.7	习题	229
10.8	项目设计	231

第四部分 应用与高级话题

第 11 章	图	234
11.1	术语和表示法	234
11.2	图的实现	237
11.3	图的周游	241
11.3.1	深度优先搜索	242
11.3.2	广度优先搜索	243
11.3.3	拓扑排序	244
11.4	最短路径问题	246
11.4.1	单源最短路径	246
11.4.2	每对顶点间的最短路径	249
11.5	最小支撑树	250
11.5.1	Prim 算法	251
11.5.2	Kruskal 算法	253
11.6	深度学习导读	254
11.7	习题	254
11.8	项目设计	256
第 12 章	线性表和数组高级技术	257
12.1	跳跃表	257
12.2	广义表	261
12.3	矩阵的表示方法	263
12.4	存储管理	265
12.4.1	动态存储分配	266
12.4.2	失败处理策略和无用单元收集	271
12.5	深度学习导读	274
12.6	习题	274
12.7	项目设计	275
第 13 章	高级树形结构	277
13.1	Trie 结构	277
13.2	平衡树	280
13.2.1	AVL 树	280
13.2.2	伸展树	282
13.3	空间数据结构	285

13.3.1	k-d 树	286
13.3.2	PR 四分树	289
13.3.3	其他空间数据结构	292
13.4	深入学习导读	293
13.5	习题	293
13.6	项目设计	293
第 14 章	分析技术	295
14.1	求和技术	295
14.2	递归关系	297
14.2.1	估计上下限	297
14.2.2	扩展递归	298
14.2.3	分治法递归	299
14.2.4	快速排序平均情况分析	300
14.3	均摊分析	301
14.4	深入学习导读	303
14.5	习题	303
14.6	项目设计	306
第 15 章	计算的限制	307
15.1	归约	307
15.2	难解问题	311
15.2.1	NP 完全性	311
15.2.2	绕过 NP 完全性问题	315
15.3	不可解问题	316
15.3.1	不可数性	316
15.3.2	停机问题的不可解性	318
15.3.3	确定程序行为是不可解的	320
15.4	深入学习导读	320
15.5	习题	320
15.6	项目设计	321
附录 A	实用函数	323
参考文献		324

第一部分

预备知识

第 1 章 数据结构和算法

第 2 章 数学预备知识

第 3 章 算法分析

第1章 数据结构和算法

得克萨斯州达拉斯市方圆 500 英里之内有多少个城市的人口超过 25 万？一个公司里有多少人每年的收入超过 10 万美元？用不超过 1000 英里长的电缆是否能把所有的电话用户都连接起来？现在就正确回答诸如这样的问题是不可能的，因为这里还没有给出回答这些问题所必须要了解的信息。必须以能够及时找到答案的方式来组织信息，才能满足我们的需要。

信息的表示是计算机科学的基础。大多数计算机程序的主要目标与其说是完成运算，不如说是存储信息和尽快地检索信息。因此，研究数据结构和算法就成了计算机科学的核心问题。本书的目的就是帮助读者理解怎样组织信息，以便支持高效的数据处理。

本书有三个主要目的。第一个目的是介绍常用的数据结构，这些数据结构形成了一个程序员的基本数据结构工具箱(toolkit)。对于许多问题，工具箱里的数据结构正是理想的选择。

第二个目的是引入并加强“权衡”(tradeoff)的概念，每一个数据结构都有其相关的代价和效益的权衡。本书通过列举出不同的数据结构，并指出它们应用于实际问题时的代价和效益来讨论“权衡”的概念。

第三个目的是讲解如何评估一个数据结构或算法的有效性。只有通过这样的分析，才能确定工具箱中的哪一个数据结构对于一个新问题是最合适的。这种技术也使得程序员能够判断自己或别人发明的新数据结构的价值。

对于一个问题通常有多种解法，应该选择哪一种呢？计算机程序设计的核心有两个目标（有时它们互相冲突）：

1. 设计一种容易理解、编码和调试的算法。
2. 设计一种能有效利用计算机资源的算法。

在理想情况下，最终的程序都能实现这两个目标，有时把这样的程序称为是“完美的”。本书给出的算法和程序实例在这种意义上讲是趋于完美的。与目标 1 有关的问题不是本书的目的，它们主要涉及到的是软件工程原理；本书主要讲的是与目标 2 有关的问题。

怎样度量效率呢？第 3 章将给出估算一种算法或者一个计算机程序的效率的方法，称为算法分析(algorithm analysis)。算法分析还可以度量一个问题的内在复杂程度。以后的章节中只要介绍算法就会用到算法分析方法。用这种方法可以清楚地看到在解决同一个问题时不同算法在效率上的差异。

本章通过提出与选择和使用数据结构相关的话题，为后面的内容做准备。本章将先审视一个设计者为所要完成的任务选择一个合适数据结构的过程，然后讨论在程序设计中如何进行抽象，最后将探索问题、算法和程序三者之间的关系。