

清华大学

计算机系列教材

邓俊辉 编著

数据结构

(C++语言版)



清华大学出版

清华大学 计算机系列教材

邓俊辉 编著

数据结构

(C++语言版)

清华大学出版社
北京

内 容 简 介

本书按照面向对象程序设计的思想,根据作者多年的教学积累,系统介绍各类数据结构的功能、表示和实现,对比各类数据结构适用的应用环境;结合实际问题展示算法设计的一般性模式与方法、算法实现的主流技巧,以及算法效率的评判依据和分析方法;以高度概括的体例为线索贯穿全书,并通过对比和类比揭示数据结构与算法的内在联系,帮助读者形成整体性认识。

书中穿插大量验证型、拓展型和反思型习题,以激发读者的求知欲,培养自学能力和独立思考习惯;近 300 幅插图结合简练的叙述,200 多段代码配合详尽而简洁的注释,使深奥抽象的概念和过程得以具体化并便于理解和记忆。

针对学生基础、教学目标及课时总量的不同,本书提供了若干种典型的教学进度及学时分配方案,授课教师可视具体情况灵活选用。勘误表、插图、代码、部分习题解答以及讲义等相关教学资料均向公众开放,可从本书主页(<http://166.111.138.40/~deng/dsacpp/>)或出版社主页(<http://www.tup.com.cn>)下载。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

数据结构: C++ 语言版/邓俊辉编著. —北京:清华大学出版社, 2011.10

(清华大学计算机系列教材)

ISBN 978-7-302-26883-3

I. ①数… II. ①邓… III. ①数据结构—高等学校—教材 ②C 语言—程序设计—高等学校—教材 IV. ①TP311.12 ②TP312

中国版本图书馆 CIP 数据核字(2011)第 183911 号

责任编辑:龙啟铭

责任校对:焦丽丽

责任印制:王秀菊

出版发行:清华大学出版社

地 址:北京清华大学学研大厦 A 座

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

邮 编:100084

社 总 机:010-62770175

邮 购:010-62786544

投稿与读者服务:010-62795954, jsjic@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

印 刷 者:北京四季青印刷厂

装 订 者:三河市兴旺装订有限公司

经 销:全国新华书店

开 本:185×260 印 张:27.5

字 数:755 千字

版 次:2011 年 10 月第 1 版

印 次:2011 年 10 月第 1 次印刷

印 数:1~4000

定 价:39.00 元

产品编号:043322-01

丛书序

“清华大学计算机系列教材”已经出版发行了 30 余种，包括计算机科学与技术专业的基础数学、专业技术基础和专业等课程的教材，覆盖了计算机科学与技术专业本科生和研究生的主要教学内容。这是一批至今发行数量很大并赢得广大读者赞誉的书籍，是近年来出版的大学计算机专业教材中影响比较大的一批精品。

本系列教材的作者都是我熟悉的教授与同事，他们长期在第一线担任相关课程的教学工作，是一批很受本科生和研究生欢迎的任课教师。编写高质量的计算机专业本科生（和研究生）教材，不仅需要作者具备丰富的教学经验和科研实践，还需要对相关领域科技发展前沿的正确把握和了解。正因为本系列教材的作者们具备了这些条件，才有了这批高质量优秀教材的产生。可以说，教材是他们长期辛勤工作的结晶。本系列教材出版发行以来，从其发行的数量、读者的反映、已经获得的国家级与省部级的奖励，以及在各个高等院校教学中所发挥的作用上，都可以看出本系列教材所产生的社会影响与效益。

计算机学科发展异常迅速，内容更新很快。作为教材，一方面要反映本领域基础性、普遍性的知识，保持内容的相对稳定性；另一方面，又需要跟踪科技的发展，及时地调整和更新内容。本系列教材都能按照自身的需要及时地做到这一点。如王爱英教授等编著的《计算机组成与结构》、戴梅萼教授等编著的《微型计算机技术及应用》都已经出版了第四版，严蔚敏教授的《数据结构》也出版了三版，使教材既保持了稳定性，又达到了先进性的要求。

本系列教材内容丰富，体系结构严谨，概念清晰，易学易懂，符合学生的认知规律，适合于教学与自学，深受广大读者的欢迎。系列教材中多数配有丰富的习题集、习题解答、上机及实验指导和电子教案，便于学生理论联系实际地学习相关课程。

随着我国进一步的开放，我们需要扩大国际交流，加强学习国外的先进经验。在大学教材建设上，我们也应该注意学习和引进国外的先进教材。但是，“清华大学计算机系列教材”的出版发行实践以及它所取得的效果告诉我们，在当前形势下，编写符合国情的具有自主版权的高质量教材仍具有重大意义和价值。它与国外原版教材不仅不矛盾，而且是相辅相成的。本系列教材的出版还表明，针对某一学科培养的要求，在教育部等上级部门的指导下，有计划地组织任课教师编写系列教材，还能促进对该学科科学、合理的教学体系和内容的研究。

我希望今后有更多、更好的我国优秀教材出版。

清华大学计算机系教授，中国科学院院士

张钹

序

为适应快速发展的形势,计算机专业基础课的教学必须走内涵发展的道路,扎实的理论基础、计算思维能力和科学的方法论是支撑该学科从业人员进行理性思维和理性实践的重要基础。“程序设计基础”、“面向对象技术”、“离散数学”以及“数据结构”等相关课程,构成了清华大学计算机系专业基础课程体系中的一条重要脉络。近年来为强化学生在计算思维和实践能力方面的训练力度,课程组通过研究,探索和实践,着力对该课程系列的教学目标、内容、方法和各门课的分工,以及如何衔接等进行科学而系统的梳理,进一步明确了教学改革的方向。在这样的背景下,由邓俊辉撰写的《数据结构(C++语言版)》正式出版了。

为了体现教材的先进性,作者研读并参考了计算学科教学大纲(ACM/IEEE Computing Curricula),结合该课程教学的国际发展趋势和对计算机人才培养的实际需求,对相关知识点做了精心取舍,从整体考虑加以编排,据难易程度对各章节内容重新分类,给出了具体的教学方案。

为了不失系统性,作者依据多年的教学积累,对各种数据结构及其算法,按照分层的思想精心进行归纳和整理,并从数据访问方式、数据逻辑结构、算法构成模式等多个角度,理出线索加以贯穿,使之构成一个整体,使在学习数据结构众多知识点的同时,获得对这门学问相关知识结构的系统性和全局性的认识。

计算机学科主张“抽象第一”,这没有错,但是弄不好会吓倒或难倒学生。本书从具体实例入手,运用“转换-化简”、“对比-类比”等手法,借助大量的插图和表格,图文并茂地展示数据结构组成及其算法运转的内在过程与规律,用形象思维帮助阐释抽象过程,给出几乎所有算法的具体实现,并通过多种版本做剖析和对比,引领读者通过学习提升抽象思维能力。

计算机学科实践性极强,不动手是学不会的。为了强化实践,本书除了每章都布置人人必做的习题和思考题外,还有不少于授课学时的上机编程要求,旨在培养学生理性思维和理性实践的动脑动手能力。

中国计算机科学与技术学科教程 2002 曾批评国内有关程序设计类的课,一是淡化算法,二是“一开始就扎进程序设计的语言细节中去”。本书十分重视从算法的高度来讲述数据结构与算法的相互依存关系,在书的开篇就用极其精彩的例子讲清了算法效率和算法复杂度度的基本概念和方法,这就给全书紧密结合算法来讲数据结构打下了很好的基础。

这本书是精心策划和撰写的,结构严整,脉络清晰,行文流畅,可读性强。全书教学目标明确,内容丰富,基本概念和基本方法的阐述深入浅出,最大的特点是将算法知识、数据结构和编程实践有机地融为一体。我以为,引导学生学好本书,对于奠定扎实的学科基础,提高计算思维能力能够起到良好的作用。

清华大学计算机系教授



2011年9月

前言

背景

伴随着计算学科 (Computing Discipline) 近二十多年来的迅猛发展, 相关专业方向不断细化和分化, 相应地在计算机教育方面, 人才培养的定位与目标呈现明显的多样化趋势, 在知识结构与专业素养方面对人才的要求也在广度与深度上拓展到空前的水平。以最新版的计算学科教学大纲 (ACM/IEEE Computing Curricula, 以下简称 CC 大纲) 为例, 2001 年制定的 CC2001 因只能覆盖狭义的计算机科学方向而更多地被称作 CS2001。所幸的是, CC2001 的意义不仅在于针对计算机科学方向的本科教学提出了详细的指导意见, 更在于构建了一个开放的 CC2001 框架 (CC2001 Model)。按照这一规划, 首先应该顺应计算学科总体发展的大势, 沿着计算机科学 (CS)、计算机工程 (CE)、信息系统 (IS)、信息技术 (IT) 和软件工程 (SE) 以及更多潜在的新学科方向, 以不同分卷的形式制定相应的教学大纲计划, 同时以综述报告的形式做一概括统领; 另外, 不宜仍拘泥于十年的周期, 而应更为频繁地调整和更新大纲, 以及时反映计算领域研究的最新进展, 满足应用领域对人才的现实需求。

饶有意味的是, 无论从此后发表的综述报告还是各分卷报告都可看出, 作为计算学科知识结构的核心理念与技术体系的基石, 数据结构与算法的基础性地位不仅没有动摇, 反而得到进一步的强化和突出, 依然是计算学科研究开发人员的必备素养, 以及相关应用领域专业技术人员的看家本领。以 CC 大纲的综述报告 (Computing Curricula 2005 - The Overview Report) 为例, 在针对以上五个专业方向本科学位所归纳的共同要求中, 数据结构与算法作为程序设计概念与技能的核心, 紧接在数学基础之后列第二位。这方面的要求可进一步细分为五个层次: 对数据结构与算法核心地位的充分理解与认同, 从软件视角对处理器、存储器及显示器等硬件资源的透彻理解, 通过编程以软件方式实现数据结构与算法的能力, 基于恰当的数据结构与算法设计并实现大型结构化组件及其之间通讯接口的能力, 运用软件工程的原理与技术确保软件鲁棒性、可靠性及其面向特定目标受众的针对性的能力。

自 20 世纪末起, 我有幸参与和承担清华大学计算机系以及面向全校“数据结构”课程的教学工作, 在学习和吸收前辈们丰富而宝贵教学经验的同时, 通过悉心体会与点滴积累, 逐步摸索和总结出一套较为完整的教学方法。做为数据结构与算法一线教学工作者中的一员, 我与众多的同行一样, 在为这类课程的重要性不断提升而欢欣鼓舞的同时, 更因其对计算学科人才培养决定性作用的与日俱增而倍感责任重大。尽管多年来持续推进的教学改革已经取得巨大的进展, 但面对新的学科发展形势和社会发展需求, 为从根本上提高我国计算机理论及应用人才的培养质量, 我们的教学理念、教学内容与教学方法仍然有待于进一步的突破。

与学校“高素质、高层次、多样化、创造性”人才培养总体目标相呼应, 我所在的清华大学计算机系长期致力于培养“面向基础或应用基础的科学技术问题, 具备知识创新、技术创新或集成创新能力的研究型人才”。沿着这个大方向, 近年来我与同事们从讲授、研讨、作业、实践、考核和教材等方面入手, 在系统归纳已有教学资源和成果的基础上, 着力推进数据结构的课程建设与改革。其中, 教材既为所授知识提供了物化的载体, 也为传授过程指明了清晰的脉络, 更为教师与学生之间的交流建立了统一的平台, 其中重要性不言而喻。继 2006 年出版《数据结构与算法 (Java 语言描述)》之后, 本教材的出版也是编者对自己数据结构与算法教学工作的又一次系统总结与深入探索。

原则

在读者群体的定位、体例结构的编排以及环节内容的取舍等方面，全书尽力贯彻了以下原则。

■ 兼顾基础不同、目标不同的多样化读者群体

全书 12 章按四大部分组织，既相对独立亦彼此呼应，难度较大的章节以星号标注，教员与学生可视具体情况相应地灵活取舍。其中第 1 章绪论旨在尽快地将背景各异的读者引导至同一起点，为此将系统地引入计算与算法的一般性概念，确立时空复杂度的度量标准，并以递归为例介绍算法设计的一般模式；第 2~7 章为基础部分，涵盖序列、树、图、初级搜索树等基本数据结构及其算法的实现方法及性能分析，这也是多数读者在实际工作中最常涉及的内容，属于研读的重点；第 8~10 章为进阶部分，介绍高级搜索树、词典和优先级队列等高级数据结构，这部分内容对更加注重计算效率的读者将很有帮助；最后两章分别以串匹配和高级排序算法为例，着重介绍算法性能优化以及针对不同应用需求的调校方法与技巧，这部分内容可以帮助读者深入理解各类数据结构与算法在不同实际环境中适用性的微妙差异。

■ 注重整体认识，着眼系统思维

全书体例参照现代数据结构普遍采用的分类规范进行编排，其间贯穿以具体而本质的线索，帮助读者在了解各种具体数据结构之后，通过概括与提升形成对数据结构家族的整体性认识。行文从多个侧面体现“转换-化简”的技巧，引导读者逐步形成和强化计算思维（computational thinking）的意识与习惯，从方法论的高度掌握利用计算机求解问题的一般性规律与方法。

比如从逻辑结构的角度，按照线性、半线性和非线性三个层次对数据结构进行分类，并以遍历算法为线索，点明不同层次之间相互转换的技巧。又如，通过介绍动态规划、减而治之、分而治之等算法策略，展示如何将人所擅长的概括化简思维方式与计算机强大的枚举迭代能力相结合，高效地求解实际问题。再如，从数据元素访问形式的角度，按照循序访问（call-by-rank）、循位置访问（call-by-position）、循关键码访问（call-by-key）、循值访问（call-by-value）、循优先级访问（call-by-priority）等方式对不同的数据结构做了归类，并指明它们之间的联系与区别。

通过引入代数判定树模型以及对应的下界等概念，并讲解如何针对具体计算模型确定特定问题的复杂度下界，破除了部分读者对计算机计算能力的盲目迷信。按照 CC 大纲综述报告的归纳结论，这也是对计算学科所有专业本科毕业生共同要求中的第三点——不仅需要了解计算机技术可以做什么（possibilities）以及如何做，更需要了解不能做什么（limitations）以及为什么。

■ 尊重认知规律，放眼拓展提升

在相关学科众多的专业基础课程中，数据结构与算法给学生留下的印象多是内容深、难度大，而如何让学生打消畏难情绪从而学有所乐、学有所获，则是摆在每位任课教师面前的课题。计算机教学有其独特的认知规律，整个过程大致可以分为记忆（remember）、理解（understand）、应用（apply）、分析（analyze）、评估（evaluate）和创造（create）等若干阶段，本书也按照这一脉络，在叙述方式上做了一些粗浅的尝试。

为加深记忆与理解，凡重要的知识点均配有插图。全书共计 230 余幅插图，借助视觉通道，从原理、过程、实例等角度使晦涩抽象的知识点得以具体化、形象化，也就是鲁迅先生“五到”读书法中的第一条“眼到”。

为加深对类似概念或系列概念的综合理解，完成认识上的提升，还普遍采用“对比”的手法。例如，优先级队列接口不同实现方式之间的性能对比、快速排序算法不同版本在适用范围上的对比，等等。又如，通过 **Dijkstra** 算法和 **Prim** 算法的横向对比，提炼和抽象出更具一般性的优先级搜索框架，并反过来基于这一认识实现统一的搜索算法模板。

为强化实践能力的培养，多从具体的应用问题入手，经逐步分析导出具体的解决方法。所列 230 余段代码，均根据讲述的侧重按模块划分，在力求简洁的同时也配有详实的备注解。读者可以下载代码，边阅读边编译执行，真正做到“手到”和“心到”。几乎所有实现的数据结构均符合对应的抽象数据类型接口标准，在强化接口规范的同时，从习惯与方式上为读者日后的团队协作做铺垫与准备。

在分析与评估方面，介绍了算法复杂度的典型层次及分析技巧，包括常规的最坏情况和平均情况分析，以及分摊分析。针对递归算法，还着重介绍了递归跟踪法与递推方程法。另外从实用的角度，还引入了稳定性、就地性等更为精细的性能评估尺度，并结合部分算法做了相关的分析对比。

数据结构与算法这二者之间相辅相成的关系，也是本书着重体现的一条重要线索。为此，本书的体例与多数同类教材不尽相同。以排序算法为例，除最后一章外，大部分排序算法都作为对应数据结构的应用实例，分散编入相应的章节：其中起泡排序、归并排序、插入排序、选择排序等算法以排序器的形式归入序列部分；桶排序和基数排序归入散列部分；而堆排序则归入优先级队列部分。再如，图算法及其基本实现均前置到第 6 章，待到后续章节引入高级数据结构时再介绍其优化方法，如此前后呼应。行文讲述中也着力突出数据结构对高效算法的支撑作用，以及源自应用的算法问题对数据结构发展的反向推动作用，优先级队列之于 **Huffman** 编码算法、完全二叉堆之于就地堆排序、伸展树之于基于局部性原理的缓存算法、散列表之于数值空间与样本空间规模差异的弥合算法等，均属于这方面的实例。

与许多课程的规律一样，习题对于数据结构与算法而言也是强化和提升学习效果的必由之途，否则无异于“入宝山而空返”。当然，好的习题不应仅限于对讲授内容的重复与堆砌，而应更多地侧重于拓展与反思。本书的拓展型习题既包括对书中数据结构接口的扩充、算法性能的改进，也包括通过查阅文献资料补充相关的知识点。另外，一些难度极大或者难度不大但过程繁琐的内容，在这里也以习题的形式留待课后进一步探讨。在求知求真的过程中，质疑与批判是难能可贵的精神，反诘与反思更是创造创新的起点。从吸收到反思，在某种意义上也就是学习（**learning**）与反学习（**unlearning**）反复迭代、不断上升的过程。为此，部分习题的答案并非简单地重复正文的结论，甚至并不具有固定的答案，以给读者日后灵活的运用与创新留下足够的空间。

说明

书中凡重要的专业词汇均注有原文，插图中的标注也多以英文给出，因为编者认为这都是进一步探究以及与国际同行交流的基础。公式多采用接近代码的风格，而非严格的数学格式，以利于理解算法。

书中涉及的所有代码以及大量尚未在书中列出的辅助代码，均按 **Visual Studio** 工程形式分成 50 多组，并统一到名为 **DSACPP** 的解决方案之下，完整的代码包可从本书主页下载后直接编译执行。

为精简篇幅、突出重点，在一定程度上牺牲了软件规范性甚至计算效率，读者不必全盘效仿。比如，为尽量利用页面宽度和便于投影式播放，全文源代码统一采用 **Java** 风格编排，但代码的

层次感却因此有所削弱，代码片段的切分也有过度之嫌。同样出于简化的考虑，代码中一些本可优化的细节也被忽略。以第 11 章的串匹配算法为例，总是直接调用计算成本较高的 `strlen()` 获取串长，在尚未对串结构做封装处理之前，这并非高效的做法。另外，对错误与意外的处理也采用了简化的处理方式。

限于编者的水平与经验，书中一定不乏纰漏与谬误之处，恳请读者及专家批评指正。

邓俊辉

2011 年夏末于清华园

关于教学计划编排的建议

采用本书作为教材时，视学生具体情况、教学目标及课时总量的不同，授课教师可从以下若干种典型的学时分配方案中选择其一。

教学方案与学时分配			方案 A	方案 B	方案 C	方案 D	方案 E	方案 F	
教学内容	部分	章	节	64	48	64	64	48	32
一 基础知识	第 1 章 绪论	1.1 + 1.2 + 1.3 + 1.5		2.5	2.5	3.5	4.5	3.5	2.5
		1.4*		1.5	1.5	2	2.5	2.5	
二 基本数据结构	第 2 章 向量	2.1 ~ 2.6		3	3	3	4	3	2.5
		2.7*		1		1.5	2	1	
		2.8		2	2	2	3	2	2
	第 3 章 列表	3.1 ~ 3.4		2	2	3	4	3	2
		3.5		2	2	3	4	3	2
	第 4 章 栈与队列	4.1 ~ 4.3		2	2	2	3	2	2
		4.4*		3	3	3	3	3	
		4.5 ~ 4.6		1	1	1	2	1	2
	第 5 章 二叉树	5.1 + 5.3		2	2	2	2	2	2
		5.2 + 5.4		2	2	3	3	3	
		5.5		2	2	3	3	3	2
	第 6 章 图	6.1 ~ 6.4		1.5	1.5	2	2	2	2
		6.5 ~ 6.8		2.5	2.5	2	4	3	3
		6.9*		1	1	2	2		
		6.10 ~ 6.12		2	2	2	4	3	2
	第 7 章 搜索树	7.1 ~ 7.4		4	4	6	12	8	6
三 高级数据结构	第 8 章 高级搜索树	8.1 ~ 8.2		2	2	3			
		8.3* ~ 8.4*		3		3			
	第 9 章 词典	9.1 + 9.3		2	2	2			
		9.2* + 9.4*		4		4			
	第 10 章 优先级队列	10.1 ~ 10.2		4	4	4			
		10.3*		2		2			
四 算法	第 11 章 串	11.1 ~ 11.3		2	2				
		11.4* ~ 11.5*		2					
	第 12 章 排序	12.1		2	2				
		12.2* ~ 12.3*		4					

本书所有相关教学资料均向公众开放，包括勘误表、插图、代码、部分习题解答、讲义等。
欢迎访问本书主页：<http://166.111.138.40/~deng/dsacpp/> 或出版社网站：
www.tup.com.cn。

致 谢

首先要感谢严蔚敏教授，廿年前，是她引领我进入了数据结构的殿堂；感谢吴文虎教授，在追随他参与信息学相关竞赛组织工作的过程中，我更加切实地感受到了算法之宏之美。感谢殷人昆、王宏、朱仲涛、徐明星等老师，在与他们的教学合作过程中我获益良多。感谢众多的同行，与他们的交流和探讨每每令我思路顿开。感谢数以千计的学生，他们是我写作的最终动机与不竭动力，无论是在课堂或是课后，与他们相处的时光都属于我在清华园最美好的记忆。

感谢龙启铭先生及出版社各位编辑，无论偏执于完美的我每次将“定稿”改得如何面目全非，他们始终报以宽容和耐心。正是依靠他们的鼎力支持，本书才得以顺利出版。

随着最后一个字符的敲出，编辑器的记录显示，本书的写作已逾 **1000** 小时，这还不包括纸面修订以及编写调试代码的用时。因此我要特别感谢我的父母、太太还有女儿，离开他们的关心和鼓励，很难想象自己如何能够坚持到底，并最终完成这一时间复杂度高达三年的艰巨工程。我欠他们的太多。

今年恰逢清华大学百年华诞，南昌二中（心远中学）也迎来 **110** 周年校庆，谨奉上此书，权作对母校培育之恩的点滴回报。

目录

第 1 章 绪论 1

§1.1 计算机与算法	2
1.1.1 古埃及人的绳索	2
1.1.2 欧几里德的尺规	3
1.1.3 起泡排序	4
1.1.4 算法	5
1.1.5 算法效率	8
§1.2 复杂度度量	8
1.2.1 问题规模、运行时间及时间复杂度	9
1.2.2 渐进复杂度	9
1.2.3 空间复杂度	11
§1.3 复杂度分析	12
1.3.1 常数复杂度 $O(1)$	12
1.3.2 对数复杂度 $O(\log n)$	13
1.3.3 线性复杂度 $O(n)$	14
1.3.4 多项式复杂度 $O(\text{polynomial}(n))$	15
1.3.5 指数复杂度 $O(2^n)$	15
1.3.6 复杂度层次	16
1.3.7 输入规模	16
§1.4 *递归	17
1.4.1 线性递归	17
1.4.2 递归分析	18
1.4.3 递归模式	20
1.4.4 递归消除	22
1.4.5 二分递归	24
§1.5 抽象数据类型	27
习题	28

第 2 章 向量 31

§2.1 从数组到向量	32
2.1.1 数组	32
2.1.2 向量	32
§2.2 接口	33
2.2.1 ADT 接口	33
2.2.2 操作实例	34
2.2.3 Vector 模板类	34
§2.3 构造与析构	36
2.3.1 默认构造方法	36
2.3.2 基于复制的构造方法	36
2.3.3 析构方法	37
§2.4 动态空间管理	37
2.4.1 静态空间管理	37
2.4.2 可扩充向量	38
2.4.3 扩容	38
2.4.4 分摊分析	39
2.4.5 缩容	40
§2.5 向量	41
2.5.1 直接引用元素	41
2.5.2 置乱器	41
2.5.3 判等器与比较器	42
2.5.4 无序查找	43
2.5.5 插入	44
2.5.6 删除	45
2.5.7 唯一化	46
2.5.8 遍历	47
§2.6 有序向量	49
2.6.1 比较器	49

2.6.2 有序性甄别	49
2.6.3 唯一化	49
2.6.4 查找	52
2.6.5 二分查找 (版本 A)	53
2.6.6 Fibonacci 查找	56
2.6.7 二分查找 (版本 B)	59
2.6.8 二分查找 (版本 C)	60
§2.7 *排序与下界	61
2.7.1 有序性	61
2.7.2 排序及其分类	62
2.7.3 下界	62
2.7.4 比较树	63
2.7.5 估计下界	64
§2.8 排序器	64
2.8.1 统一入口	64
2.8.2 起泡排序	65
2.8.3 归并排序	66
习题	69
第 3 章 列表	73
§3.1 从向量到列表	74
3.1.1 从静态存储到动态存储	74
3.1.2 由秩到位置	75
3.1.3 列表	75
§3.2 接口	75
3.2.1 列表节点	75
3.2.2 列表	76
§3.3 列表	79
3.3.1 头、尾节点	79
3.3.2 默认构造方法	79
3.3.3 由秩到位置的转换	80
3.3.4 查找	80
3.3.5 插入	81

3.3.6 基于复制的构造	83
3.3.7 删除	84
3.3.8 析构	85
3.3.9 唯一化	86
3.3.10 遍历	86
§3.4 有序列表	87
3.4.1 唯一化	87
3.4.2 查找	88
§3.5 排序器	88
3.5.1 统一入口	88
3.5.2 插入排序	89
3.5.3 选择排序	91
3.5.4 归并排序	92
习题	94

第 4 章 栈与队列 97

§4.1 栈	98
4.1.1 概述	98
4.1.2 ADT 接口	99
4.1.3 操作实例	99
4.1.4 Stack 模板类	99
§4.2 栈与递归	100
4.2.1 递归的实现	101
4.2.2 避免递归	102
§4.3 典型应用	103
4.3.1 逆序输出	103
4.3.2 递归嵌套	105
4.3.3 延迟缓冲	108
4.3.4 逆波兰表达式	113
§4.4 *试探回溯法	116
4.4.1 试探与回溯	116
4.4.2 八皇后	117
4.4.3 迷宫寻径	119

§4.5 队列	122	习题	172
4.5.1 概述	122	第6章 图	175
4.5.2 ADT 接口	122	§6.1 概述	176
4.5.3 操作实例	123	§6.2 抽象数据类型	179
4.5.4 Queue 模板类	124	6.2.1 操作接口	179
§4.6 队列应用	124	6.2.2 Graph 模板类	180
4.6.1 循环分配器	124	§6.3 邻接矩阵	181
4.6.2 银行服务模拟	125	6.3.1 原理	181
习题	126	6.3.2 实现	182
第5章 二叉树	129	6.3.3 时间性能	184
§5.1 二叉树及其表示	130	6.3.4 空间性能	184
5.1.1 树	130	§6.4 邻接表	184
5.1.2 二叉树	131	6.4.1 原理	184
5.1.3 多叉树	132	6.4.2 复杂度	184
§5.2 编码树	134	§6.5 图遍历算法概述	185
5.2.1 二进制编码	134	§6.6 广度优先搜索	186
5.2.2 二叉编码树	136	6.6.1 策略	186
§5.3 二叉树的实现	137	6.6.2 实现	186
5.3.1 二叉树节点	137	6.6.3 实例	187
5.3.2 二叉树节点操作接口	140	6.6.4 复杂度	187
5.3.3 二叉树	141	6.6.5 应用	188
§5.4 Huffman 编码	146	§6.7 深度优先搜索	188
5.4.1 PFC 编码	146	6.7.1 策略	188
5.4.2 最优编码树	149	6.7.2 实现	189
5.4.3 Huffman 编码树	152	6.7.3 实例	190
5.4.4 Huffman 编码算法	154	6.7.4 复杂度	191
§5.5 遍历	160	6.7.5 应用	192
5.5.1 递归式遍历	161	§6.8 拓扑排序	192
5.5.2 迭代版先序遍历	163	6.8.1 应用	192
5.5.3 迭代版中序遍历	165	6.8.2 有向无环图	193
5.5.4 迭代版后序遍历	169	6.8.3 算法	193
5.5.5 层次遍历	171	6.8.4 实现	193

6.8.5 实例	194
6.8.6 复杂度	194
§6.9 *双连通域分解	195
6.9.1 关节点与双连通域	195
6.9.2 蛮力算法	195
6.9.3 算法	196
6.9.4 实现	196
6.9.5 实例	198
6.9.6 复杂度	199
§6.10 优先级搜索	199
6.10.1 优先级与优先级数	199
6.10.2 基本框架	200
6.10.3 复杂度	201
§6.11 最小支撑树	201
6.11.1 支撑树	201
6.11.2 最小支撑树	201
6.11.3 歧义性	201
6.11.4 蛮力算法	202
6.11.5 Prim 算法	202
§6.12 最短路径	204
6.12.1 最短路径树	204
6.12.2 歧义性	205
6.12.3 Dijkstra 算法	205
习题	208

第 7 章 搜索树 211

§7.1 搜索	212
7.1.1 循关键码访问	212
7.1.2 词条	213
7.1.3 序与比较器	213
§7.2 二叉搜索树	213
7.2.1 顺序性	213
7.2.2 中序遍历序列	214

7.2.3 BST 模板类	214
7.2.4 查找算法及其实现	215
7.2.5 插入算法及其实现	217
7.2.6 删除算法及其实现	218
§7.3 平衡二叉搜索树	220
7.3.1 树高与性能	220
7.3.2 理想平衡与适度平衡	221
7.3.3 等价二叉搜索树	222
7.3.4 等价变换与局部调整	222
§7.4 AVL 树	223
7.4.1 AVL 树	223
7.4.2 节点插入	225
7.4.3 节点删除	228
7.4.4 统一重平衡算法	230
习题	232

第 8 章 高级搜索树 235

§8.1 伸展树	236
8.1.1 局部性	236
8.1.2 逐层伸展	237
8.1.3 双层伸展	238
8.1.4 分摊分析	240
8.1.5 伸展树的实现	243
§8.2 B-树	247
8.2.1 多路平衡搜索	247
8.2.2 ADT 接口及其实现	249
8.2.3 关键码查找	251
8.2.4 性能分析	252
8.2.5 关键码插入	253
8.2.6 上溢与分裂	254
8.2.7 关键码删除	257
8.2.8 下溢与合并	258
§8.3 *红黑树	263

8.3.1 概述.....	263	9.3.8 插入.....	310
8.3.2 红黑树接口定义.....	265	9.3.9 更多开放定址策略.....	312
8.3.3 节点插入算法.....	266	9.3.10 散列码转换.....	314
8.3.4 节点删除算法.....	269	§9.4 *散列应用.....	316
§8.4 *kd-树.....	275	9.4.1 桶排序.....	316
8.4.1 范围查询.....	275	9.4.2 最大间隙.....	317
8.4.2 kd-树.....	278	9.4.3 基数排序.....	318
8.4.3 基于 2d-树的范围查询算法.....	279	习题.....	319
习题.....	280	第 10 章 优先级队列.....	323
第 9 章 词典.....	283	§10.1 优先级队列.....	324
§9.1 词典.....	284	10.1.1 优先级与优先级队列.....	324
9.1.1 操作接口.....	284	10.1.2 关键码、比较器与偏序关系.....	325
9.1.2 操作实例.....	285	10.1.3 操作接口.....	325
9.1.3 接口定义.....	286	10.1.4 操作实例：选择排序.....	325
9.1.4 判等器.....	286	10.1.5 接口定义.....	326
9.1.5 实现方法.....	287	10.1.6 应用实例：Huffman 编码树.....	326
§9.2 *跳转表.....	287	§10.2 堆.....	328
9.2.1 Skiplist 模板类.....	287	10.2.1 完全二叉堆.....	329
9.2.2 总体逻辑结构.....	288	10.2.2 元素插入.....	331
9.2.3 四联表.....	288	10.2.3 元素删除.....	333
9.2.4 查找.....	290	10.2.4 建堆.....	335
9.2.5 空间复杂度.....	292	10.2.5 就地堆排序.....	338
9.2.6 时间复杂度.....	292	§10.3 *左式堆.....	341
9.2.7 插入.....	293	10.3.1 堆合并.....	341
9.2.8 删除.....	296	10.3.2 单侧倾斜.....	341
§9.3 散列表.....	298	10.3.3 PQ_LeftHeap 模板类.....	342
9.3.1 完美散列.....	298	10.3.4 空节点路径长度.....	342
9.3.2 装填因子与空间利用率.....	299	10.3.5 左倾性与左式堆.....	343
9.3.3 散列函数.....	300	10.3.6 右侧链.....	343
9.3.4 散列表.....	303	10.3.7 合并算法.....	344
9.3.5 冲突及其排解.....	305	10.3.8 合并操作 merge()的实现.....	344
9.3.6 开放定址策略.....	307	10.3.9 实例.....	345
9.3.7 查找与删除.....	309	10.3.10 复杂度.....	345

10.3.11 基于合并的插入和删除操作	346
习题	347
第 11 章 串	349
§11.1 串及串匹配	350
11.1.1 串	350
11.1.2 串匹配	351
11.1.3 测评标准与策略	352
§11.2 蛮力算法	353
11.2.1 算法描述	353
11.2.2 算法实现	353
11.2.3 时间复杂度	354
§11.3 KMP 算法	355
11.3.1 构思	355
11.3.2 next 表	356
11.3.3 KMP 算法	357
11.3.4 next[0] = -1	357
11.3.5 构造 next 表	358
11.3.6 性能分析	358
11.3.7 继续改进	359
§11.4 *BM 算法	361
11.4.1 思路与框架	361
11.4.2 坏字符策略	362
11.4.3 好后缀策略	364
11.4.4 综合性能	368
§11.5 *Karp-Rabin 算法	369
11.5.1 构思	369
11.5.2 算法与实现	370
习题	373

第 12 章 排序	375
§12.1 快速排序	376
12.1.1 分治策略	376
12.1.2 轴点	376
12.1.3 快速排序算法	377
12.1.4 快速划分算法	377
12.1.5 复杂度	379
12.1.6 应对退化	381
§12.2 *选取与中位数	382
12.2.1 概述	382
12.2.2 主流数	383
12.2.3 归并向量的中位数	385
12.2.4 基于优先级队列的选取	388
12.2.5 基于快速划分的选取	389
12.2.6 k-选取算法	390
§12.3 *希尔排序	392
12.3.1 递减增量策略	392
12.3.2 增量序列	394
习题	397
附录	399
插图索引	400
表格索引	406
算法索引	407
代码索引	408
关键词索引	414