

清华大学

计算机系列教材

邓俊辉 编著

数据结构(C++语言版)

(第3版)



清华大学出版社

清华大学985名优教材立项资助

数据结构（C++语言版）

第 3 版

邓俊辉

清华大学出版社

2013年9月·北京

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 and 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,给出了具体的教学计划方案。

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,使学生在学数据结构的众多知识点的同时,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

第3版说明

在第2版的基础上，本书第3版推出了配套的《习题解析》，故在体例上也做了相应的调整，主要包括以下方面：

- 原各章所附习题，均统一摘出并汇编为《习题解析》；除了部分实践型和研究型习题，大部分习题均提供了详尽的分析和解答。
- 删除了少量习题，同时也补充了若干。大题的总数，已增至292道；因多数习题都是逐层递进式的，小题的总数已超过500道。
- 关于伸展树性能分摊分析的原8.1.4小节，作为习题转入《习题解析》。
- 图灵机模型、RAM模型等基本概念，以及（线性）归约、封底估算及基本技巧，也结合对应的习题予以介绍。

同时，结合读者反馈以及新一轮教学实践效果，也在以下方面做了相应修订：

- 1.4节补充了对记忆策略与动态规划策略的介绍，并通过实例展示二者的联系与区别。
- 鉴于前四章已经充分地展示了相关技巧，后续Bintree和Dictionary等结构不再过于严格地封装，使读者更好地将注意力集中于这些结构的机理本身。
- 通过多重继承，统一了ComplHeap、LeftHeap、ListHeap等结构的实现方式，使之封装更紧凑、代码更简洁。
- 精简了Vector<T>::mergesort()、GraphMatrix::insert()、Splay::splay()、RedBlack::solveDoubleRed()、trivialMedian()等算法的实现。
- 关于函数调用栈、栈与递归、Huffman编码算法等各节的叙述与讲解，也尽可能地做了精简。
- 统一了“环路”、“众数”、“最左/右侧通路”、“波峰集”、“输入/输出敏感”等概念。
- 严格了“完全二叉树”等概念以及“黑高度”等指标的定义。
- 参照BFS和DFS的实现方式改进PFS框架，使之支持多个连通域（或可达域）。
- 借助几何分布等概率模型，简化对跳转表、散列表的平均性能分析。
- 插图、表格、代码等均有大幅增加，关键词索引项进一步细化。
- 增加了若干重要的参考文献。
- 修正了原书及代码中的若干错误，详细对比请见勘误表。

最后，鉴于第3版采用双色印刷方式，故在版面及样式等方面也做了相应的调整。

第2版说明

本书的初稿完成于2009年冬季，随后在清华大学经过了三个学期共四个课堂的试用，根据各方的反馈意见做过调整补充之后，第1版于2011年夏季由清华大学出版社正式出版发行。此后，又在清华大学经过两个学期共三个课堂的教学实践，并汇总读者的反馈进一步修订完善之后，第2版终于2012年夏季出版发行，也就是目前读者所看到的这个版本。

第2版继承并强化了此前版本的叙述风格，基本保留了总体的体例结构，同时在针对性、简洁性、实用性和拓展性等方面，也做了大量的修改、删节与扩充。与此前的版本相比较，主要的变化包括以下几个方面：

- 针对多种数据结构的算法实现及其性能分析，精简了行文叙述与代码实现，比如有序向量的查找、树和图的遍历、Huffman编码、平衡二叉搜索树的重平衡、二叉堆的调整等。
- 更换并补充了大量的实例和插图，比如向量、词典、关联数组、高级平衡二叉搜索树和优先级队列等数据结构，以及表达式求值、KMP、BM、平衡二叉搜索树的重平衡、字符串散列、快速排序、中位数及众数等算法的原理及过程等等，插图增至260多组。
- 重写了多个章节的总结部分，比如针对各类查找算法、串匹配算法，就其性能特点均做了统一的归纳与梳理，指明其中的关键因素以及不同的适用范围。
- 进一步规范 and 统一了几个基本概念的定义及其表述方式，使得各章节之间的相互引述更趋一致，比如栈混洗、真二叉树、完全二叉树、满树、闭散列策略等概念的定义，以及遍历序列、红黑树不同类型节点等概念的图解示意方式。
- 细化了针对一些关键知识点的讲解，比如第1章的渐进复杂度层次和伪复杂度、第8章中B-树及kd-树的引入动机、第11章中BM算法好后缀策略中的`gs[]`表构造算法等。
- 添加了大量的习题，总量已超过280道。在帮助读者梳理主要知识点、加深对讲解内容理解的同时，还从以下方面为他们的进一步拓展，提供了必要的线索：插入排序算法性能与逆序对的关系、选择排序算法性能与循环节的关系、插值查找、指数查找、马鞍查找、CBA式排序算法平均性能的下界、栈混洗甄别、栈堆、队堆、算术表达式的组合搜索、键树、关联矩阵、Prim算法与Krusal算法的正确性、欧氏最小支撑树、并查集、计数排序、四叉树、八叉树、范围树、优先级搜索树、树堆、AVL树节点删除算法的平均性能、AVL树的合并与分裂、堆节点插入算法的平均性能、支持重复元素的二叉搜索树、双向平方试探、轴点构造算法版本C、希尔排序算法的正确性，等等。
- 提供了一批相关的参考文献，包括经典的教材专著20余册、拓展的学术论文30余篇。
- 修正了多处排版问题及若干实质错误。请此前版本的读者下载勘误表并做相应更正，同时感谢我的读者、学生和同行，他们的意见与建议是本教材不断完善的保证。

第1版前言

背景

伴随着计算学科 (Computing Discipline) 近年来的迅猛发展, 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, 以分卷的形式制订相应的教学大纲计划, 同时以综述报告的形式概括统领; 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。作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, 在为此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 and 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，本教材的出版也是作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

在读者群体定位、体例结构编排以及环节内容取舍等方面，全书尽力贯彻以下原则。

■ 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至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至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、循位置访问或循链接访问、循关键码访问、循值访问、循优先级访问等方式，对各种数据结构做了归类，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）以及为什么不能做。

■ 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多组300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说。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。

与许多课程的规律类似，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”。本书各章均针对性地附有大量习题，累计逾270道。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。其中，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

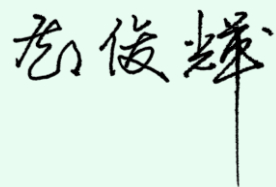
书中凡重要的专业词汇均注有原文，插图中的标注也多以英文给出，因为作者认为这都是进一步钻研以及与国际同行交流的基础。公式多采用接近代码的风格，而非严格的数学格式，以利

于按照代码注释的方式描述和理解算法。

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，读者不必盲目效仿。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但可能影响总体思路的细节也被忽略。另外，对错误与意外的处理也采用了简化的处理方式。

限于本人的水平与经验，书中一定不乏纰漏与谬误之处，恳请读者及专家批评指正。

A handwritten signature in black ink, reading '陈俊辉' (Chen Junhui). The signature is written in a cursive style with a long vertical line extending downwards from the bottom character.

2011年夏末于清华园

教学计划编排方案建议

采用本书作为教材时，视学生基础、专业方向、教学目标及允许课时总量的不同，授课教师可参照以下典型方案分配课内学时，通常还需另外设置约50%的课外编程实验学时。

教学方案与课内学时分配			方案A	方案B	方案C	方案D	方案E	方案F	方案G	
教学内容	部分	章	节 (视学时可省带*小节)	64	48	64	64	48	32	48
一 基础知识	第1章 绪论	1.1~1.3 + 1.5		2.5	2.5	3.5	4.5	3.5	2.5	3
		1.4 *		1.5	1.5	2	2.5	2.5	0	2
二 基本数据结构	第2章 向量	2.1 ~ 2.6		3	3	3	4	3	2.5	3
		2.7 *		1	0	1.5	2	1	0	0
		2.8		2	2	2	3	2	2	2
	第3章 列表	3.1 ~ 3.4		2	2	3	4	3	2	3
		3.5		2	2	3	4	3	2	3
	第4章 栈与队列	4.1 ~ 4.3		2	2	2	3	2	2	2
		4.4 *		3	3	3	3	3	0	3
		4.5 ~ 4.6		1	1	1	2	1	2	1
	第5章 二叉树	5.1 + 5.3		2	2	2	2	2	2	2
		5.4		2	2	3	3	3	2	2
		5.2 + 5.5		2	2	3	3	3	0	3
	第6章 图	6.1 ~ 6.4		1.5	1.5	2	2	2	2	2
		6.5 ~ 6.8		2.5	2.5	2	4	3	3	3
		6.9 *		1	1	2	2	0	0	0
		6.10 ~ 6.12		2	2	2	4	3	2	2
	第7章 搜索树	7.1 ~ 7.2		2	2	3	6	4	3	3
		7.3 ~ 7.4		2	2	3	6	4	3	3
三 高级数据结构	第8章 高级搜索树	8.1 ~ 8.2		2	2	3	0	0	0	0
		8.3 * ~ 8.4 *		3	0	3	0	0	0	0
	第9章 词典	9.1 + 9.3		2	2	2	0	0	0	0
		9.2 * + 9.4 *		4	0	4	0	0	0	0
	第10章 优先级队列	10.1 ~ 10.2		4	4	4	0	0	0	0
		10.3 *		2	0	2	0	0	0	0
四 算法	第11章 串	11.1 ~ 11.3		2	2	0	0	0	0	2
		11.4 * ~ 11.5 *		2	0	0	0	0	0	2
	第12章 排序	12.1		2	2	0	0	0	0	2
		12.2 * ~ 12.3 *		4	0	0	0	0	0	0

本书所有相关教学资料均向公众开放，包括勘误表、插图、代码以及配套讲义等。欢迎访问教材主页：<http://dsa.cs.tsinghua.edu.cn/~deng/dsacpp/>

致 谢

感谢严蔚敏教授，廿多年前是她引领我进入数据结构的殿堂；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、尹霞等老师，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。

历年的助教研究生不仅出色地完成了繁重的课外辅导与资源建设工作，他们丰富的想象力和创造力更是我重要的灵感来源，在此我要感谢他们对我的帮助！

截至**2012**年秋季，按担任助教时间先后，他们分别是：

王智、李云翀、赵乐、肖晶、刘汝佳、高岳、沈超慧、李锐喆
于泽、白彦冰、夏龙、向阳、姚姜源、刘雨辰、姜禹、方宇剑

在本书各版次的使用过程中，收到了学生及读者的大量反馈，其中杨凯峪、李雨田、许婷婷、郑斯陶、石梦凯、陈逸翀、王紫、朱剑男、李仁杰、许建林、吴育昕、刘苏齐、陈键飞、唐骞璘、徐霜晴等同学，分别指出了多处纰漏及错误。王笑尘同学作为第**3**版的试验读者，杨凯峪、李雨田同学作为配套《习题解析》的试验读者，均提出了许多很好的修改建议。清华美院的郭清华同学，对第**3**版的版式和色调设计提出了极具价值的建议。在此也谨向他们表示感谢！

感谢清华大学出版社的各位编辑，正是依靠他们的鼎力支持，本书才得以顺利出版。特别感谢龙启铭先生，他出色而高效的协调工作，使我得以将更多精力集中于书稿本身；在体例编排及编写风格等方面，他的许多建议都极具价值。

本教材第**3**版的撰写工作，得到了“清华大学**985**名优教材”立项资助，在此谨向清华大学，特别是校、系教务部门的大力支持表示感谢！

简要目录

第1章 绪论 1

§1.1 计算机与算法	2
§1.2 复杂度度量	8
§1.3 复杂度分析	11
§1.4 *递归	16
§1.5 抽象数据类型	26

第2章 向量 27

§2.1 从数组到向量	28
§2.2 接口	29
§2.3 构造与析构	32
§2.4 动态空间管理	33
§2.5 常规向量	37
§2.6 有序向量	44
§2.7 *排序与下界	57
§2.8 排序器	59

第3章 列表 65

§3.1 从向量到列表	66
§3.2 接口	67
§3.3 列表	71
§3.4 有序列表	77
§3.5 排序器	78

第4章 栈与队列 85

§4.1 栈	86
§4.2 栈与递归	88
§4.3 栈的典型应用	90

§4.4 *试探回溯法	99
§4.5 队列	104
§4.6 队列应用	107

第5章 二叉树 109

§5.1 二叉树及其表示	110
§5.2 编码树	114
§5.3 二叉树的实现	117
§5.4 遍历	123
§5.5 Huffman编码	136

第6章 图 149

§6.1 概述	150
§6.2 抽象数据类型	153
§6.3 邻接矩阵	155
§6.4 邻接表	158
§6.5 图遍历算法概述	159
§6.6 广度优先搜索	159
§6.7 深度优先搜索	162
§6.8 拓扑排序	165
§6.9 *双连通域分解	168
§6.10 优先级搜索	172
§6.11 最小支撑树	174
§6.12 最短路径	178

第7章 搜索树 181

§7.1 查找	183
§7.2 二叉搜索树	184

§7.3 平衡二叉搜索树	191
§7.4 AVL树	194
第8章 高级搜索树	203
§8.1 伸展树	204
§8.2 B-树	212
§8.3 *红黑树	227
§8.4 *kd-树	239
第9章 词典	245
§9.1 词典ADT	247
§9.2 *跳转表	249
§9.3 散列表	259
§9.4 *散列应用	277
第10章 优先级队列	281
§10.1 优先级队列ADT	282
§10.2 堆	286
§10.3 *左式堆	297

第11章 串	305
§11.1 串及串匹配	306
§11.2 蛮力算法	309
§11.3 KMP算法	311
§11.4 *BM算法	317
§11.5 *Karp-Rabin算法	327
第12章 排序	333
§12.1 快速排序	334
§12.2 *选取与中位数	341
§12.3 *希尔排序	350
附录	357
参考文献	358
插图索引	362
表格索引	369
算法索引	370
代码索引	371
关键词索引	377

详细目录

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 欧几里得的尺规3
- 1.1.3 起泡排序4
- 1.1.4 算法5
- 1.1.5 算法效率7

§1.2 复杂度度量 8

- 1.2.1 时间复杂度8
- 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 11

- 1.3.1 常数 $O(1)$ 12
- 1.3.2 对数 $O(\log n)$ 12
- 1.3.3 线性 $O(n)$ 13
- 1.3.4 多项式 $O(\text{polynomial}(n))$
.....14
- 1.3.5 指数 $O(2^n)$ 14
- 1.3.6 复杂度层次15
- 1.3.7 输入规模16

§1.4 *递归 16

- 1.4.1 线性递归17
- 1.4.2 递归分析17
- 1.4.3 递归模式19
- 1.4.4 递归消除21
- 1.4.5 二分递归22

§1.5 抽象数据类型 26

第2章 向量 27

§2.1 从数组到向量 28

- 2.1.1 数组28

- 2.1.2 向量29

§2.2 接口 29

- 2.2.1 ADT接口29
- 2.2.2 操作实例30
- 2.2.3 Vector模板类30

§2.3 构造与析构 32

- 2.3.1 默认构造方法32
- 2.3.2 基于复制的构造方法32
- 2.3.3 析构方法33

§2.4 动态空间管理 33

- 2.4.1 静态空间管理33
- 2.4.2 可扩充向量34
- 2.4.3 扩容34
- 2.4.4 分摊分析35
- 2.4.5 缩容36

§2.5 常规向量 37

- 2.5.1 直接引用元素37
- 2.5.2 置乱器37
- 2.5.3 判等器与比较器38
- 2.5.4 无序查找39
- 2.5.5 插入40
- 2.5.6 删除40
- 2.5.7 唯一化42
- 2.5.8 遍历43

§2.6 有序向量 44

- 2.6.1 比较器44
- 2.6.2 有序性甄别44
- 2.6.3 唯一化45
- 2.6.4 查找47
- 2.6.5 二分查找 (版本A)48
- 2.6.6 Fibonacci查找51
- 2.6.7 二分查找 (版本B)54
- 2.6.8 二分查找 (版本C)55

§2.7 *排序与下界 _____ 57

2.7.1 有序性 57

2.7.2 排序及其分类 57

2.7.3 下界 57

2.7.4 比较树 58

2.7.5 估计下界 59

§2.8 排序器 _____ 59

2.8.1 统一入口 59

2.8.2 起泡排序 60

2.8.3 归并排序 61

第3章 列表 65

§3.1 从向量到列表 _____ 66

3.1.1 从静态到动态 66

3.1.2 由秩到位置 67

3.1.3 列表 67

§3.2 接口 _____ 67

3.2.1 列表节点 67

3.2.2 列表 68

§3.3 列表 _____ 71

3.3.1 头、尾节点 71

3.3.2 默认构造方法 71

3.3.3 由秩到位置的转换 72

3.3.4 查找 72

3.3.5 插入 72

3.3.6 基于复制的构造 74

3.3.7 删除 75

3.3.8 析构 76

3.3.9 唯一化 76

3.3.10 遍历 77

§3.4 有序列表 _____ 77

3.4.1 唯一化 77

3.4.2 查找 78

§3.5 排序器 _____ 78

3.5.1 统一入口 78

3.5.2 插入排序 79

3.5.3 选择排序 80

3.5.4 归并排序 82

第4章 栈与队列 85

§4.1 栈 _____ 86

4.1.1 ADT接口 86

4.1.2 操作实例 87

4.1.3 Stack模板类 88

§4.2 栈与递归 _____ 88

4.2.1 函数调用栈 88

4.2.2 避免递归 89

§4.3 栈的典型应用 _____ 90

4.3.1 逆序输出 90

4.3.2 递归嵌套 91

4.3.3 延迟缓冲 94

4.3.4 逆波兰表达式 96

§4.4 *试探回溯法 _____ 99

4.4.1 试探与回溯 99

4.4.2 八皇后 100

4.4.3 迷宫寻径 102

§4.5 队列 _____ 104

4.5.1 概述 104

4.5.2 ADT接口 105

4.5.3 操作实例 105

4.5.4 Queue模板类 106

§4.6 队列应用 _____ 107

4.6.1 循环分配器 107

4.6.2 银行服务模拟 107

第5章 二叉树 109

§5.1 二叉树及其表示 _____ 110

5.1.1 树 110

5.1.2 二叉树 111

5.1.3 多叉树 112

\$5.2 编码树 114

5.2.1 二进制编码114

5.2.2 二叉编码树116

\$5.3 二叉树的实现 117

5.3.1 二叉树节点117

5.3.2 二叉树节点操作接口...119

5.3.3 二叉树120

\$5.4 遍历 123

5.4.1 递归式遍历124

5.4.2 *迭代版先序遍历126

5.4.3 *迭代版中序遍历128

5.4.4 *迭代版后序遍历131

5.4.5 层次遍历133

\$5.5 Huffman编码 136

5.5.1 PFC编码及解码136

5.5.2 最优编码树139

5.5.3 Huffman编码树141

5.5.4 Huffman编码算法142

第6章 图 149

\$6.1 概述 150

\$6.2 抽象数据类型 153

6.2.1 操作接口153

6.2.2 Graph模板类153

\$6.3 邻接矩阵 155

6.3.1 原理155

6.3.2 实现155

6.3.3 时间性能157

6.3.4 空间性能157

\$6.4 邻接表 158

6.4.1 原理158

6.4.2 复杂度158

\$6.5 图遍历算法概述 159

\$6.6 广度优先搜索 159

6.6.1 策略159

6.6.2 实现160

6.6.3 实例161

6.6.4 复杂度161

6.6.5 应用161

\$6.7 深度优先搜索 162

6.7.1 策略162

6.7.2 实现162

6.7.3 实例163

6.7.4 复杂度164

6.7.5 应用165

\$6.8 拓扑排序 165

6.8.1 应用165

6.8.2 有向无环图166

6.8.3 算法166

6.8.4 实现167

6.8.5 实例168

6.8.6 复杂度168

\$6.9 *双连通域分解 168

6.9.1 关节点与双连通域.....168

6.9.2 蛮力算法169

6.9.3 可行算法169

6.9.4 实现170

6.9.5 实例171

6.9.6 复杂度172

\$6.10 优先级搜索 172

6.10.1 优先级与优先级数....172

6.10.2 基本框架173

6.10.3 复杂度174

\$6.11 最小支撑树 174

6.11.1 支撑树174

6.11.2 最小支撑树174

6.11.3 歧义性175

6.11.4 蛮力算法175

6.11.5 Prim算法175

\$6.12 最短路径 178

6.12.1 最短路径树178

6.12.2 Dijkstra算法	178
-------------------------	-----

第7章 搜索树	181
---------	-----

§7.1 查找	183
---------------	-----

7.1.1 循关键码访问	183
7.1.2 词条	183
7.1.3 序与比较器	183

§7.2 二叉搜索树	184
------------------	-----

7.2.1 顺序性	184
7.2.2 中序遍历序列	184
7.2.3 BST模板类	185
7.2.4 查找算法及其实现	185
7.2.5 插入算法及其实现	188
7.2.6 删除算法及其实现	189

§7.3 平衡二叉搜索树	191
--------------------	-----

7.3.1 树高与性能	191
7.3.2 理想平衡与适度平衡	192
7.3.3 等价变换	192
7.3.4 旋转调整	193

§7.4 AVL树	194
-----------------	-----

7.4.1 定义及性质	194
7.4.2 节点插入	196
7.4.3 节点删除	198
7.4.4 统一重平衡算法	200

第8章 高级搜索树	203
-----------	-----

§8.1 伸展树	204
----------------	-----

8.1.1 局部性	204
8.1.2 逐层伸展	205
8.1.3 双层伸展	206
8.1.4 伸展树的实现	208

§8.2 B-树	212
----------------	-----

8.2.1 多路平衡查找	212
8.2.2 ADT接口及其实现	215
8.2.3 关键码查找	216

8.2.4 性能分析	218
8.2.5 关键码插入	219
8.2.6 上溢与分裂	219
8.2.7 关键码删除	222
8.2.8 下溢与合并	223

§8.3 *红黑树	227
-----------------	-----

8.3.1 概述	228
8.3.2 红黑树接口定义	230
8.3.3 节点插入算法	231
8.3.4 节点删除算法	234

§8.4 *kd-树	239
------------------	-----

8.4.1 范围查询	239
8.4.2 kd-树	242
8.4.3 基于2d-树的范围查询	243

第9章 词典	245
--------	-----

§9.1 词典ADT	247
------------------	-----

9.1.1 操作接口	247
9.1.2 操作实例	247
9.1.3 接口定义	248
9.1.4 实现方法	248

§9.2 *跳转表	249
-----------------	-----

9.2.1 Skiplist模板类	249
9.2.2 总体逻辑结构	250
9.2.3 四联表	250
9.2.4 查找	252
9.2.5 空间复杂度	253
9.2.6 时间复杂度	254
9.2.7 插入	255
9.2.8 删除	258

§9.3 散列表	259
----------------	-----

9.3.1 完美散列	259
9.3.2 装填因子与空间利用率	260
9.3.3 散列函数	261
9.3.4 散列表	264
9.3.5 冲突及其排解	266

9.3.6 闭散列策略	268
9.3.7 查找与删除	271
9.3.8 插入	272
9.3.9 更多闭散列策略	273
9.3.10 散列码转换	275

§9.4 *散列应用 277

9.4.1 桶排序	277
9.4.2 最大间隙	278
9.4.3 基数排序	279

第10章 优先级队列 281

§10.1 优先级队列ADT 282

10.1.1 优先级与优先级队列 ..	282
10.1.2 关键码、比较器与偏序关系	283
10.1.3 操作接口	283
10.1.4 操作实例：选择排序 ..	283
10.1.5 接口定义	284
10.1.6 应用实例：Huffman编码树	284

§10.2 堆 286

10.2.1 完全二叉堆	286
10.2.2 元素插入	289
10.2.3 元素删除	291
10.2.4 建堆	292
10.2.5 就地堆排序	295

§10.3 *左式堆 297

10.3.1 堆合并	297
10.3.2 单侧倾斜	298
10.3.3 PQ_LeftHeap模板类 ..	298
10.3.4 空节点路径长度	299
10.3.5 左倾性与左式堆	299
10.3.6 最右侧通路	300
10.3.7 合并算法	300
10.3.8 实例	301
10.3.9 合并操作的实现	302

10.3.10 复杂度	302
10.3.11 基于合并的插入和删除	302

第11章 串 305

§11.1 串及串匹配 306

11.1.1 串	306
11.1.2 串匹配	307
11.1.3 测评标准与策略	308

§11.2 蛮力算法 309

11.2.1 算法描述	309
11.2.2 算法实现	309
11.2.3 时间复杂度	310

§11.3 KMP算法 311

11.3.1 构思	311
11.3.2 next表	312
11.3.3 KMP算法	312
11.3.4 $\text{next}[0] = -1$	313
11.3.5 $\text{next}[j + 1]$	313
11.3.6 构造next表	314
11.3.7 性能分析	315
11.3.8 继续改进	315

§11.4 *BM算法 317

11.4.1 思路与框架	317
11.4.2 坏字符策略	318
11.4.3 好后缀策略	321
11.4.4 $\text{gs}[]$ 表构造算法	323
11.4.5 算法纵览	326

§11.5 *Karp-Rabin算法 327

11.5.1 构思	327
11.5.2 算法与实现	328

第12章 排序 333

§12.1 快速排序 334

12.1.1 分治策略	334
12.1.2 轴点	334

12.1.3 快速排序算法	335
12.1.4 快速划分算法	335
12.1.5 复杂度	338
12.1.6 应对退化	339
§12.2 *选取与中位数	341
12.2.1 概述	341
12.2.2 众数	342
12.2.3 归并向量的中位数 ...	343
12.2.4 基于优先级队列的选取	346
12.2.5 基于快速划分的选取 ..	347
12.2.6 k-选取算法	348
§12.3 *希尔排序	350

12.3.1 递减增量策略	350
12.3.2 增量序列	353

附录	357
参考文献	358
插图索引	362
表格索引	369
算法索引	370
代码索引	371
关键词索引	377