



“十二五”普通高等教育本科国家级规划教材

“高等学校本科计算机类专业应用型人才培养研究”项目规划教材

数据结构与算法

(第2版)

Data Structures and Algorithms
(Second Edition)

陈卫卫 王庆瑞 编著

高等教育出版社



“十二五”普通高等教育本科国家级规划教材

“高等学校本科计算机类专业应用型人才培养研究”项目规划教材



数据结构与算法

(第2版)

Shuju Jiegou yu Suanfa

陈卫卫 王庆瑞 编著

高等教育出版社·北京

内容简介

本书是对 2010 年第 1 版教材的内容进行优化重组、修订而成。全书共 6 章,分别为概述、表结构、树结构、图结构、排序和问题的固有难度和算法设计的一般主法简介。主要内容包括数据结构和算法的基本概念;顺序表、链表、栈、队、矩阵、字符串、散列表、广义表、树、二叉树、检索树、最优检索树、AVL 树、红黑树、B 树、B+树、2-3 树、Trie 树、哈夫曼树、判定树、union-find 树、图等基本结构及各结构的特点和存储方法;实现查找、插入、删除、遍历、搜索算法的设计方法和时空效率分析,实现图的最小生成树和最短路径求解算法、DAG 图的拓扑排序和关键路径求解算法,以及实现各种内排序算法、文件结构和外排序算法;讲解问题的固有难度、算法设计的一般方法,并给出表、树、图等典型基本结构的 C++类实现示例。全书配有 400 多道各种题型的习题。

本书语言通俗流畅,叙述简洁,可读性强,配有丰富的教学资源,可作为普通高等学校本科计算类专业教材和教学参考书,也可作为程序设计爱好者的理论指导书。

图书在版编目 (C I P) 数据

数据结构与算法 / 陈卫卫,王庆瑞编著.--2 版
.--北京:高等教育出版社,2015.7
ISBN 978-7-04-043307-4

I. ①数… II. ①陈… ②王… III. ①数据结构-高等学校-习题集②算法分析-高等学校-习题集 IV.
①TP311.12-44

中国版本图书馆 CIP 数据核字(2015)第 155760 号

策划编辑 倪文慧 责任编辑 韩 飞 封面设计 张 志
插图绘制 杜晓丹 责任印制 赵义民

出版发行	高等教育出版社	网 址	http://www.hep.edu.cn
社 址	北京市西城区德外大街 4 号		http://www.hep.com.cn
邮政编码	100120	网上订购	http://www.landaco.com
印 刷	北京市白帆印务有限公司		http://www.landaco.com.cn
开 本	787mm×1092mm 1/16	版 次	2010 年 11 月第 1 版
印 张	26.5		2015 年 7 月第 2 版
字 数	570 千字	印 次	2015 年 7 月第 1 次印刷
购书热线	010-58581118	定 价	41.00 元
咨询电话	400-810-0598		

本书如有缺页、倒页、脱页等质量问题,请到所购图书销售部门联系调换
版权所有 侵权必究
物 料 号 43307-00

数字课程资源使用说明

与本书配套的数字课程资源发布在高等教育出版社易课程网站,请登录网站后开始课程学习。

1. 访问 <http://abook.hep.com.cn/1871816>, 点击“注册”。在注册页面输入用户名、密码及常用的邮箱进行注册。已注册的用户直接输入用户名和密码登录即可进入“我的课程”界面。

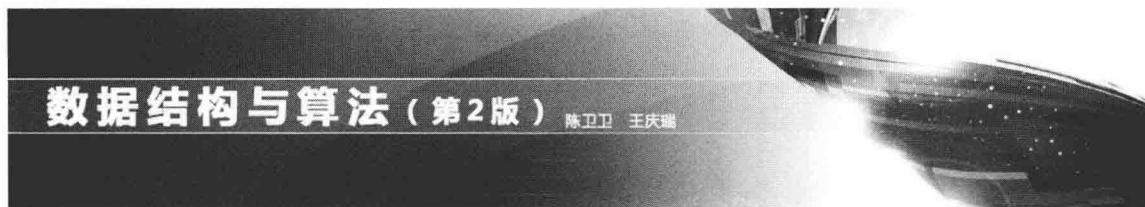
2. 课程充值: 登录后点击右上方的“充值”图标, 正确输入教材封底标签上的明码和密码, 点击“确定”完成课程充值。

3. 在“我的课程”列表中选择已充值的数字课程, 点击“进入课程”即可开始课程学习。

4. 与本书配套的易课程数字课程资源按照章、节知识树的形式构成, 包括电子教案、微视频、示例代码、习题解答、专业名词和术语的中英文对照索引等, 以便读者学习使用。

账号自登录之日起一年内有效, 过期作废。

使用本账号如有任何问题, 请发邮件至: ecourse@pub.hep.cn。



用户名 密码 验证码 2682

数字课程介绍

纸质教材

版权信息

联系方式

重要通知



书名: 《数据结构与算法（第2版）》
作者: 陈卫卫 王庆瑞
ISBN:
出版商: 高等教育出版社
出版日期: 2015年7月
定价:

因系统升级, 所有用户都需要先注册
(不能用书后的明码暗码直接登录),
注册后的用户登录后, 请先点击页面右
上方“充值”, 正确输入教材封底标签上
的明码和密码完成课程选择。

出版说明

信息化社会需要大量的计算机类专业人才。据统计,目前我国计算机类专业布点总数已逾2 800个,这些专业点为国家的现代化建设培养了大批计算机类专业人才,其中绝大多数是应用型人才。如何按照社会需求,确定合理的人才培养目标,并在其制导下培养特色突出的应用型人才,是提高教育质量和水平的重要任务。

为了更好地引导高校计算机类各专业点构建有特色的培养方案,例如,能够体现行业特色、区域需求,同时建设体现这些特色的学科基础课和专业课,促进本科计算机类专业应用型人才培养,出版一批体现应用型人才培养特色的新形态教材,教育部高等学校计算机类专业教学指导委员会、全国高等学校计算机教育研究会与高等教育出版社联合组建了“高等学校本科计算机类专业应用型人才培养研究”课题组,基于《计算机类专业教学质量国家标准》,围绕软件工程、网络工程、物联网工程等专业应用型人才培养的研究展开相关工作。

在研究的基础上,课题组汇聚80多所高校的教学经验,协同创新,开展了核心课程教学资源建设以及教材建设,这套教材作为课题研究的重要成果之一,具有以下几个显著特点。

- 以课题研制的《高等学校本科计算机类专业应用型人才培养指导意见》为指导,委托有丰富教学实践经验的教师编写,内容覆盖了不同专业的学科基础课、专业核心课及专业方向课。
- 教材内容基于理论适用,突出理论与实践相结合,强调“做中学”,引入丰富的实验案例,摒弃大而全、重理论轻实践的做法,结构新颖、努力突出专业特色。
- 采用纸质教材与数字资源相结合的形式,将教学内容与课程建设充分展示出来,使教师和学生借助网络实现全方位的个性化教学。

相信这套教材的出版能够起到推动各高校计算机类专业建设、提高教学水平和人才培养质量的作用。希望广大教师在教学过程中对教材提出宝贵的意见和建议,使其在使用过程中不断完善。

教育部高等学校计算机类专业教学指导委员会
全国高等学校计算机教育研究会
高等教育出版社
2015年3月

第2版前言

本书1版是依据《高等学校计算机科学与技术专业核心课程教学实施方案》，面向计算机类专业应用型人才培养的要求编写的，也是教育部高等理工教育教学改革与实践项目的研究成果，在使用过程中得到了广大读者的肯定和厚爱。本书之所以能够入选“十二五”普通高等教育本科国家级规划教材，与各方领导、同事、朋友以及关注、使用本教材的同仁们的大力支持和帮助是密不可分的，特表致谢！

2013年以来，中国逐步兴起MOOC，在教育教学中不断地融入各种新媒体元素，学习的渠道和途径越来越多。如何与时代的发展相适应，发挥纸质教材的特点和优势，形成线上线下的有机补充，是本版教材修订的着眼点和立足点。

如果说，本书第1版更着重于解决教学内容的选择和易懂易学的问题，那么第2版则在继承前一版特点的基础上，更着重于教学内容的编排组织，从学习者的角度出发，更有力地展现“读、仿、改、究”的数据结构学习和训练模式，突出“以学生为主体”的教育理念，并辅以适当的MOOC资源。这既基于作者亲力亲为的教学实践，也得益于广大读者的意见和建议，同时也渗透着作者日积月累的教学经验和与时俱进的教学理念。

本书每一章前有内容提要、后有内容小结是本版的一个特色。设在一章开篇处的内容提要使读者对本章主要内容有一个宏观的认识，知道本章共有哪些内容，重点、难点是什么，而这一章的正文部分就围绕这一知识架构，按照“读、仿、改、究”的模式组织教学内容、配备相关习题。处于一章结尾处的内容小结，其价值和意义不仅是对本章内容的归纳和总结，同时也是对这些知识的深入理解和拓展。

“读、仿、改、究”是作者针对数据结构课程提出的一套行之有效的学习方法，具有很强的可操作性，对学生如何学好数据结构课程具有现实的指导意义。该方法针对学生学习和掌握数据结构与算法的知识提出了4个不同等级要求（或称阶段）：简单地说，就是让学生能够读懂算法（初级要求）、模仿算法（中级要求）、改进算法（高级要求）和自行设计算法（最高要求）。按照这一方法，学生能够自我把握学习难度，由浅入深地掌握各项基本知识，并自测掌握程度，逐步建立自信心，由初级一步步达到中级、高级，乃至最高级要求。

具体地说，“读”就是研读书中的经典算法及其实现代码，研读的同时细心领会这些算法的设计思路、描述方式和其实现代码的程序结构，这是学习算法设计的第一步，即启蒙阶段。在熟读众多经典算法的基础上，能将已经掌握的知识进行简单搭建，用来求解与原有问题相近的新问题，即编写出模仿算法，也就是达到了“仿”阶段。而“改”阶段则要求当应用场景或用户需求发生改变时，学生能够对现有算法进行一番改进，写出改进算法，使算法渐达最佳。这一阶段至关

重要,不仅起到了深化理解知识的作用,而且对学生的算法设计和分析能力、改进点的定位、改进措施的选定都是极大的考验和锻炼。“究”指的是研究和探索,要求学生能够围绕给定问题,探究该问题的性质、特征,以确定求解策略,设计出性能良好的求解算法,是培养学生各种能力,提高素质的最高境界。

为了配合“读、仿、改、究”的训练模式,各章均配有“节习题”和“章习题”两部分习题。有基本概念题、普通填空题、程序填空题、选择题、算法设计题、基础实验题、综合实验题等多种题型。“节习题”放在每一节的尾部,主要用于考查学生对本节相关知识的掌握程度,强调基础性和针对性,一般说来这部分习题的求解难度不大。“章习题”(即综合习题)放在每一章的尾部,主要用于考查学生对跨章节的多个知识的掌握程度,更加强调综合性和应用性,其求解难度要大于节习题,属于中等难度或较高难度的一类习题,其中的综合实验题难度最高,对培养学生的算法设计能力和程序设计能力起到至关重要的作用。

第2版教材对全书的章节内容作了较大的调整。调整后,全书共分6章(本版的章节内容与第1版的章节没有简单的对应关系)。第1章概括地介绍数据结构和算法的概念以及算法的描述方法和评价方法。第2~4章分别介绍基本数据结构——表、树、图。第5章介绍基本排序方法(内排序和外排序)。第6章用通俗的语言介绍问题的固有难度和NP-完全问题的概念,以及常用的算法设计方法。

建议本书使用者根据实际教学要求和学时分配,按照下述方式进行教学内容的取舍。

第一层次(最高要求):全面完成本书内容。但可以将第二层次所忽略的内容列为课外阅读(不占学时)。

第二层次(较高要求):可以忽略2.2.5小节中有关单链域的双向链表,2.4.4小节中有关BM算法和KR算法,2.5.3小节散列表的性能分析,3.5.3小节Trie树,5.7.3小节初始顺串的生成,5.7.4小节最佳合并树,5.7.5小节磁带排序,6.1.3小节不确定性算法,6.1.4小节三大重要的问题类,6.2.5小节搜索-回溯法等内容。

第三层次(一般要求):可以再忽略1.2.3小节计算时间复杂性的一般方法,2.2.5小节中的静态链表应用,2.4.4小节中的KMP算法,3.3.4小节最优检索树,3.5.1小节B树,3.5.2小节B+树,4.1.2小节中的邻接表变种,4.2.3小节无向图的关节点,4.4.3小节关键路径,5.5.1小节递归的合并排序,5.7节外部排序,6.1节问题的固有难度,6.2.4小节动态规划法,以及表、树、图的一类实现等内容。

第四层次(较低要求):可以在第三层次的基础上再忽略2.2.5小节静态链表,2.4.2小节中的矩阵相乘算法,2.6节广义表,3.2.2小节遍历序列的前驱和后继,3.2.5小节非递归的遍历算法,3.4.1小节AVL树,3.6.2小节判定树,3.6.3小节union-find树,4.3.4小节Floyd算法,6.2节算法设计的一般方法等内容。

本书配有全套PPT课件、重点难点讲解教学视频、名词和术语的“中英文对照索引表”(正文中用黑色字体标注的词语)、“部分习题参考答案”和示例代码,访问与本书配套的数字课程网站

可获取相应资源。

本书在编写过程中得到了高等教育出版社和解放军理工大学的大力支持,在此一并致谢。
对于书中不当甚至错误之处,恳请读者批评、指正。

作 者

2015年7月

第 1 版前言

为了更好地贯彻教育部高等学校计算机科学与技术教学指导委员会发布的《高等学校计算机科学与技术专业发展战略研究报告暨专业规范(试行)》和《高等学校计算机科学与技术专业公共核心知识体系与课程》的指导精神,加速建设优质教学资源,落实分类培养的战略目标,解决相关课程建设中存在的问题,教育部高等学校计算机科学与技术专业教学指导分委员会、中国计算机学会教育专业委员会、高等教育出版社相联合,从全国 30 多所高校中遴选富有教学经验的教师和专家,就“高等学校计算机科学与技术专业核心课程教学实施方案”展开研讨,并于 2009 年制定出包括“数据结构与算法”在内的 8 门核心课程的教学实施方案。方案按课程和培养类型分别列出详细的教学内容大纲,知识矩阵,学时分配,各知识点的讲授提示,重点和难点,实验内容和要求,考核内容、形式和要求,以及对教材写作的要求和建议。

作者参与制定了“数据结构与算法”(应用型)课程教学实施方案,并按照要求,精心为广大读者打造出这本内容丰富、适用面广、易学易用、具有特色的教材。“注重基础,强调实践,开发智力,培养能力,提高素质”是本书的写作宗旨。

首先,本书紧扣方案的主导思想,涵盖方案要求的全部知识点。考虑到不同学校教学要求上的差异和培养计算机专业应用型人才的实际需求,本着“有用、够用、实用”的原则,对某些知识点作了适当的扩充和提高。这些内容一部分可用于选讲,一部分可作为学生课外阅读,以进一步改善学生的知识结构,扩大知识面。

其二,在突出重点、有效化解难点方面作了认真的考虑和合理的安排。对于那些既是重点又是难点的内容(比如,如何建立链表中“链”的概念以及遍历二叉树的递归概念),不惜加重笔墨;而对于那些非重点的难点,有的只介绍大致“轮廓”(如文件的组织结构),有的只作“科普式”的介绍(比如问题固有难度)。对于基本运算的算法,都给出 C/C++ 实现程序(这些程序全部在 Visual C++6.0 环境下上机通过);而对于那些实现程序过于冗长的非基本运算(如 AVL 树、红黑树,以及 B 树、B+ 树的处理),只给出算法的描述形式,而将算法的实现程序作为提高性的课后练习。将数据结构的类实现方法放在全书的最后也是出于这一考虑。因为用类去封装数据结构,如同给数据结构穿上一层厚厚的“铠甲”,不仅涉及的语法概念过于繁杂,初学者很难理解,而且需要透过“铠甲”才能看到算法的核心处理步骤,与其让学生带着“铠甲”训练,不如先轻装学透数据结构的处理算法,之后再学习如何“穿戴铠甲”。这样,既有效地化解了教学难点,也为学习面向对象的程序设计技术作了铺垫。

第三,以查找、插入、删除运算为主线介绍表、树、图等基本数据结构的特点、存储方法、时空效率。因为,查找、插入、删除不仅是最基本的、最常用的,而且往往也是不可分的(通常联合使

用,极少单独使用)。将这三种运算构成的运算集作为一个整体,可以得出结构的整体时空效率。正因为如此,本书未将“查找”作为一个独立问题单用一章集中介绍,而是将其作为基本结构的常用运算之一,与结构的其他运算一起讨论。

第四,作者力图通过本书为读者构建多层面全方位的知识体系。具体地说,在问题层面上,揭示问题的固有难度和 NP 难题的存在,使读者不至于为试图谋求优于固有难度的算法(尤其是为 NP-完全问题谋求有效算法)而做无用功(或说外行话);在算法层面上,通过算法复杂性和评价方法,使读者了解什么是“有效算法”,什么是“好”算法,什么是“最优”算法,以及(通过学习算法设计的一般方法)如何设计出“好”算法;在数据结构层面,介绍各种数据结构的特点以及如何合理地_为算法配置适当的数据结构,以获取预期的处理效率;在程序设计层面上,通过大量的示例算法的实现程序,向读者展示程序设计技术,从而将算法、数据结构、程序设计有机地融合在一起。

第五,精心设计的题量大、题型多、考查面宽、具有一定难度层次分布的 400 多道练习题,足以构建一个中等规模的习题库。题型包括基本概念题、普通填空题、程序填空题、选择题、算法设计题、基础实验题、综合实验题等。习题的考查面覆盖书中所有主要知识点。对重点内容配备的习题不但题量大、题型多,而且更偏重于编程练习(包括算法设计题、基本实验题和综合实验题),以帮助学生充分消化吸收所学内容。这 400 多道习题中,一般难度的约占 30%,中等难度的约占 40%,稍高难度的约占 20%,难度较大的约占 10%,教师可以针对不同的教学对象和不同的教学要求,选配不同难度的习题。通过多题型、多难度层次习题的综合训练(尤其是上机练习),可以有效地加深学生对基本概念的理解,强化学生的程序设计能力。

此外,附录 B 给出一部分习题的参考答案(给出答案的题都是一些具有启发性的典型题),既便于教师备课授课,又便于学生自学自测和复习。正文中带波浪线的词语在附录 A 给出中英文对照。

总之,在整体结构、内容取舍、主次分布、重点和难点、习题配备等诸方面都作了周满地考虑。既考虑到由易逐步过渡到难,也照顾到知识的系统性和完整性。

全书共分为 8 章。第 1 章概括介绍数据结构和算法的概念以及算法的描述方法和评价方法。第 2~4 章分别介绍基本数据结构——表、树、图。第 5 章介绍常用集合运算的算法设计和效率分析。第 6 章介绍基本排序方法(内排序和外排序)。第 7 章用极通俗的语言介绍问题的固有难度和 NP-完全问题的概念,以及常用的算法设计方法。第 8 章介绍基本数据结构的类实现方法。其中,第 1~4 章、第 5 章和第 6 章中的散列表是基本内容,而第 2~4 章和第 6 章是重点内容。标题带“*”的章节是可选讲内容(偏难的和非重点的)。

各校根据实际教学要求和学时分配,可以适当地对本书进行节选。建议按照下述方式进行教学内容的取舍。

第一层次(最高要求):全面完成本书内容。但可以将第二层次忽略的内容列为课外阅读(不占学时)。

第二层次(较高要求):可以忽略 2.5.3, 2.7.4, 5.2.3, 5.4.5, 6.7.3, 6.7.4, 6.7.5, 7.2, 7.3.4 各节内容。

第三层次(普通要求):可以再忽略 1.2.3, 2.5.2, 2.7.3, 2.9, 4.2.4, 4.4, 4.7.3, 5.3, 5.4.3, 5.4.4, 5.5.2, 6.5.1, 6.7, 7.1, 7.3.3 各节和第 8 章的内容。

第四层次(最低要求):可以在第三层次的基础上再忽略 2.5, 2.6.2(其中的矩阵乘法内容), 2.8, 3.2.2, 3.2.5, 3.5, 4.6.2, 5.4.1, 7.3 各节的内容。

陈卫卫制定了全书的章节结构和内容编选原则,并编写了第 1、2、3、8 章;第 4~7 章由王庆瑞编写。

要特别感谢蒋宗礼教授认真详尽地审阅了本书,并提出宝贵的修改意见。

本书在编写过程中,一直得到高等教育出版社、解放军理工大学的关注和支持,在此一并致谢。

本书有配套的 PPT 课件,可以从中国高校计算机课程网(<http://computer.cncourse.com>)下载)。

对于书中不当甚至错误之处,恳请读者批评、指正。

作 者

2010 年 9 月

目录

第 1 章 概述	1	2.3.1 基本概念	72
1.1 基本概念	1	2.3.2 进栈和退栈算法	75
1.1.1 数据结构的概念	1	2.3.3 进队和出队算法	78
1.1.2 抽象数据类型	3	2.3.4 应用举例	82
1.1.3 算法的概念	5	习题 2.3	85
习题 1.1	6	2.4 矩阵和字符串	90
1.2 算法的描述和评价	6	2.4.1 矩阵的基本概念和存储方法	90
1.2.1 算法的描述	6	2.4.2 稀疏矩阵运算示例	94
1.2.2 算法的评价标准和评价方法	11	2.4.3 字符串的基本概念和简单匹配算法	99
1.2.3 计算时间复杂性的一般方法	15	2.4.4* 其他匹配算法	102
习题 1.2	17	习题 2.4	111
内容小结	19	2.5 散列表	113
综合习题	20	2.5.1 散列函数	113
第 2 章 表结构	21	2.5.2 散列表的处理算法	117
2.1 基本概念和顺序表	21	2.5.3* 散列表的性能分析	120
2.1.1 基本概念	21	习题 2.5	122
2.1.2 顺序表的插入和删除	28	2.6* 广义表	124
2.1.3 顺序表的查找	30	习题 2.6	126
习题 2.1	34	2.7 表结构的类实现示例	126
2.2 链表	38	习题 2.7	135
2.2.1 基本概念和链表种类	38	内容小结	137
2.2.2 链表的构造	45	综合习题	139
2.2.3 链表的遍历	48	第 3 章 树结构	144
2.2.4 链表的插入和删除	50	3.1 基本概念和存储方法	144
2.2.5 静态链表	56	3.1.1 普通树的基本概念	144
习题 2.2	63	3.1.2 二叉树的基本概念	148
2.3 栈和队	72	3.1.3 普通树与二叉树的相互	

转换	152	综合习题	241
3.1.4 树的存储方法	154	第4章 图结构	246
习题 3.1	157	4.1 基本概念和存储方法	246
3.2 二叉树的遍历和构造	160	4.1.1 图的定义和有关术语	246
3.2.1 二叉树的遍历	160	4.1.2 图的存储方法	250
3.2.2* 遍历序列的前驱和 后继	164	习题 4.1	256
3.2.3 遍历的应用示例	165	4.2 图的遍历和应用示例	259
3.2.4 二叉树的构造	169	4.2.1 先深搜索	259
3.2.5* 非递归的遍历算法	172	4.2.2 先广搜索	266
习题 3.2	175	4.2.3* 无向图的关节点	268
3.3 检索树	183	习题 4.2	273
3.3.1 检索树的查找	183	4.3 最小生成树和最短路径	276
3.3.2 检索树的插入和构造	184	4.3.1 Kruskal 算法	276
3.3.3 检索树的删除	186	4.3.2 Prim 算法	281
3.3.4* 最优检索树	190	4.3.3 Dijkstra 算法	284
习题 3.3	195	4.3.4* Floyd 算法	288
3.4 平衡树	196	习题 4.3	290
3.4.1 AVL 树	196	4.4 有向无回路图	292
3.4.2 红黑树	203	4.4.1 基本概念	292
习题 3.4	212	4.4.2 拓扑排序	294
3.5* B 树和 Trie 树	213	4.4.3* 关键路径	297
3.5.1 B 树	213	习题 4.4	301
3.5.2 B+树	216	4.5 图结构的类实现示例	302
3.5.3 Trie 树	221	习题 4.5	305
习题 3.5	222	内容小结	305
3.6 几个实用树结构	222	综合习题	306
3.6.1 哈夫曼树	222	第5章 排序	308
3.6.2* 判定树	226	5.1 基本概念	309
3.6.3* union-find 树	229	习题 5.1	310
习题 3.6	233	5.2 插入排序	310
3.7 树结构的类实现示例	234	5.2.1 直接插入排序	310
习题 3.7	239	5.2.2 二分插入排序	313
内容小结	239	5.2.3 希尔排序	314
		习题 5.2	318

5.3 交换排序	319
5.3.1 冒泡排序	319
5.3.2 快速排序	322
习题 5.3	327
5.4 选择排序	330
5.4.1 一般原理和效率分析	330
5.4.2 树选排序	331
5.4.3 堆排序	332
习题 5.4	338
5.5 合并排序	340
5.5.1 递归的合并排序	340
5.5.2* 非递归的合并排序	342
习题 5.5	345
5.6 基数排序	346
5.6.1 基本原理和示例	346
5.6.2 算法的实现和分析	349
习题 5.6	352
5.7 外部排序	354
5.7.1 文件的组织结构	354
5.7.2 顺串的合并	358
5.7.3* 初始顺串的生成	367
5.7.4* 最佳合并树	369
5.7.5* 磁带排序	371

习题 5.7	373
内容小结	374
综合习题	375
第 6 章* 问题的固有难度和算法	
设计的一般方法简介	377
6.1 问题的固有难度和分类	377
6.1.1 算法的重要地位	377
6.1.2 问题的固有难度	379
6.1.3 不确定性算法	381
6.1.4 三大重要的问题类	383
习题 6.1	385
6.2 算法设计的一般方法	386
6.2.1 集合运算的数据结构	
选取	386
6.2.2 递归、分治和平衡	388
6.2.3 贪心法	394
6.2.4 动态规划法	396
6.2.5 搜索-回溯法	399
习题 6.2	402
内容小结	403
综合习题	404
参考文献	405

内容提要

本章的目的有两个:其一是介绍数据结构和算法的基本概念;其二是介绍算法的表现形式和评价方法。基本概念和算法表现形式等内容不难理解,只要了解即可。算法评价涉及的评价标准、正确性、时间复杂性以及最坏情况和平均情况等概念却很重要。

“抽象数据类型”和“算法时间复杂性的计算方法”两部分内容是本章的两个难点。抽象数据类型的确有些“抽象”,可结合后续内容(表、树、图的类化示例)逐步理解。算法时间复杂性的计算方法虽然对于学习算法的人是至关重要的,但是对于刚刚开始学习本书的读者来说,一时难以理解是在所难免的,需要通过后续各章对众多算法的分析、评价、时间空间复杂性计算,逐步加以理解。

1.1 基本概念

1.1.1 数据结构的概念

1. 数据和数据结点

在计算机领域中,数据是对客观事物的描述形式和编码形式的统称,是计算机算法和程序的处理对象和计算结果。

在研究算法时,着重考虑由众多相同类型数据组成的数据集合,以及集合中数据之间的内在联系,需要对数据和数据集合进行哪些运算,以及如何提高运算效率,设计出高质量的程序。

数据元素也称为**数据结点**,简称**结点**。通常,一个数据结点由用来描述一个独立事物的名称、数量、特征、性质的一组相关信息组成。多数情况下,一个结点含有多个**数据项**,每个数据项是结点的一个**域**,能够用来唯一标识结点的域称为**关键字域**,人们往往用关键字指明结点。当然,某些情况下,一个结点只含一个数据项,即单值结点。

例如,在设计处理学生成绩问题的程序时,一名学生的相关信息,包括学生的姓名、学号、各科考试成绩等,构成一个数据结点,学号可以作为关键字。在处理库存商品问题时,一个数据结点对应一种商品的相关数据项,包括商品编号、名称、规格、数量、生产厂家、单价、入库日期等,商品编号可以作为关键字。在后面的示例中,多选用单值结点,也可理解成用关键字值代替结点值。



视频
数据结构的
基本概念

2. 数据结构的定义

一个有穷的结点集合 D , 以及该集合中各结点之间的关系 R , 组成一个**数据结构**, 表示成 $B = (D, R)$ ^①。

这里, D 和 R 是对客观事物的抽象描述, R 表示结点之间的逻辑关系, 所以, (D, R) 指的是数据的**逻辑结构**。

例如, $D = \{a, b, c, d, e\}$, $R_1 = \{ \langle a, b \rangle, \langle d, c \rangle, \langle b, d \rangle, \langle c, e \rangle \}$, $R_2 = \{ \langle a, b \rangle, \langle b, c \rangle, \langle b, d \rangle, \langle b, e \rangle, \langle d, e \rangle \}$ 。那么, (D, R_1) 组成一个数据结构; 而 (D, R_2) 则组成另一个数据结构。

数据结构在计算机内的存储形式称为**存储结构**, 也称为**物理结构**。在存储数据结点值的同时, 还必须存储(或结点的存储地址能够体现出)结点之间的关系。用来存储一个数据结点, 并在必要时存储该结点与其他结点之间关系的一组存储单元称为一个**存储结点**。因为一个数据结点对应一个存储结点, 所以, 在不会混淆的情况下, 也把存储结点简称为**结点**。当前尚未存储数据结点的存储结点叫**空白结点**, 或称为**空结点**、**自由结点**。

例如, n 个数据结点的集合 $D = \{a_0, a_1, a_2, \dots, a_{n-1}\}$, 关系 $R = \{ \langle a_i, a_{i+1} \rangle \mid i = 0, 1, \dots, n-2 \}$ 。可以将 D 存储于数组 $b[m]$ ($n < m$) 中, 那么每个 $b[i]$ ($i = 0, 1, \dots, m-1$) 是一个存储结点, 除 n 个数据元素占用的 n 个单元外, 尚有 $m-n$ 个空白结点。如果将 n 个数据元素依次存储于 $b[0]$ 至 $b[n-1]$, 那么结点的存储地址(即数组元素下标)能够体现出结点之间的关系, 关系 R 用不着单独存储。如果将 n 个数据元素随机地存储于 n 个单元, 除存储数据元素 a_i 外, 还要存储能够体现关系 $\langle a_i, a_{i+1} \rangle$ 的信息。通常的做法是为存储结点增加一个域, 若 a_i 存储于 $b[j]$, 那么对于 $b[j]$ 来说, 增加的这个域则用于存储元素 a_{i+1} 的地址。

3. 数据结构的种类

基本数据结构大体上可分成表结构、树结构、图结构和散结构 4 大类, 如图 1-1 所示。

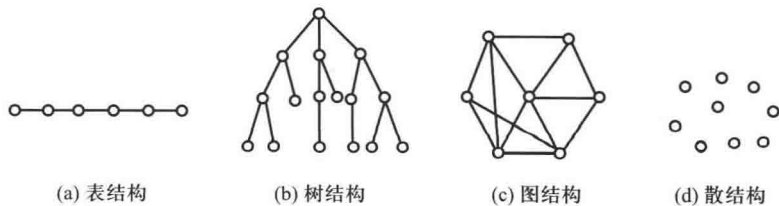


图 1-1 数据结构分类示意图

表结构用来表示结点之间某种先后次序关系(属于线性关系), 一个结点只有一个前驱和一个后继, 但首结点没有前驱, 尾结点没有后继, 如学生成绩单。

树结构用来表示结点之间的层次关系、分支关系和嵌套关系, 一个结点只有一个前驱(根结

^① 数据结构另一种定义形如: $B = (D, R, O)$, 其中“ O ”表示“运算集”。

点没有前驱),但可以有多个后继,如某部门的组织机构、行政区划图、文件目录结构。

图结构用来表示结点之间最一般的关系,任何两个结点之间都可能(或没有)关系,如城市交通网。

通常认为,表结构描述的是“一对一”关系,树结构描述的是“一对多”关系,而图结构描述的是“多对多”关系。

如果集合中任何两个结点之间都没有关系,或者说存在一种特殊的关系——无关关系,则可表示成散结构。

从数学角度来讲,树结构和散结构都是图的特例,表结构又是树的特例。例如,无回路的连通图就是树,散结构对应的是只含顶点没有边的“空图”。线性表也可以看作是“一元树”。

1.1.2 抽象数据类型

1. 抽象的意义

从一般意义上讲,抽象是指“抽取事物的共性,忽略个性;体现外部特征,掩盖具体细节”。

在计算机程序设计领域中,“抽象”起到十分关键的作用。在系统分析和设计阶段,通过对客观事物的抽象,能够让人们更好地把握全局,从更高层的宏观上考虑问题,着眼于制作更具有生命力的软件模块,以增强软件的复用性。同时,在编程实现阶段,通过抽象数据类型实现“信息隐蔽”,从而使软件具有易移植和易维护的特点。

抽象程度越高,对信息及其处理细节的隐藏就越深,对使用这些信息的人们的计算机专业知识要求也就越低,使用起来就越加方便。

最初,汇编语言的出现,使人们能够像数学那样直接使用形如 23、1.54、3.2E05 等数据,而不必关心数据在计算机中是如何表示的,如何进行计算的,其中的处理细节可能涉及定点数、浮点数的二进制存储形式以及运算方式,这些细节都由汇编程序自动完成,对使用者来说是隐蔽的,或者说是透明的。

其后,高级程序设计语言提供了整型、单/双精度实型、字符型等基本数据类型,以及这些类型的相应运算、函数库、输入输出语句。用户在使用这些类型的数据进行计算时,可以直接将它们写成表达式,系统便按照运算符的优先级,自动地对表达式进行加工处理,并给出计算结果。这中间究竟隐藏了多少细节,对于那些不懂得“编译”过程的人们是很难理解,也很难说清楚的。但是这种隐藏给程序设计者带来的方便却是实实在在的。

伴随着抽象不断地向高层扩展,也引发了程序设计方法的变革。从面向过程的程序设计,发展到面向对象的程序设计,再逐步发展到面向组件、面向服务的程序设计。这对软件复用技术的发展起到十分关键的作用。

2. 抽象数据类型的含义

抽象数据类型简称 ADT,是将“数据”连同对其的“处理操作”(即运算)封装在一起而形成的复合体。ADT 是对一个确定的数学模型,以及定义在该模型上的一组操作的抽象表示,不涉