

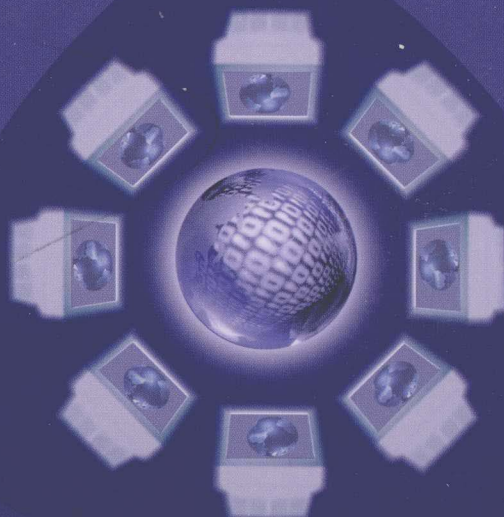


高等学校计算机精品课程系列教材

数据结构与算法

(第二版)

王昆仑 李 红 主编



中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE



高等学校计算机精品课程系列教材

数据结构与算法

（第二版）

王昆仑 李 红 主编

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

内 容 简 介

本书为高等院校本科计算机类专业、信息技术类专业和相关专业“应用型”人才培养而编写。以学习软件设计开发中涉及的各种数据结构、常用算法和解决基本应用问题的实际应用需求为基本点,深入介绍了各种数据结构的定义(逻辑结构、存储结构和基本算法)和基本应用等方面的知识。本书以“数据结构”的逻辑结构作为引线,突出以实例和应用为特色,把数据结构与算法同应用结合起来,缩短了理论知识与应用之间的距离。

全书共分11章,在本书编写过程中,编者以利于读者学习为目的,对书中的有关数据结构与算法的基本知识重新用通俗的语言进行了描述,更换了部分例题;对所有基本算法和源程序进行了调试;为方便教师教学和学生自学,重新编写了每章的教学目标和教学提示;对每章后配备的多种类型的习题,重新进行了设计和编排,同时给出了所有习题的参考答案。

本书适合作为高等学校理工科计算机工程类、软件工程类和信息技术类等相关专业的教材,也可供从事相关工作的科技工作者参考。

图书在版编目(CIP)数据

数据结构与算法 / 王昆仑, 李红主编. -- 2版. --
北京: 中国铁道出版社, 2012.9
高等学校计算机精品课程系列教材

ISBN 978-7-113-15256-7

I. ①数… II. ①王… ②李… III. ①数据结构—高等学校—教材②算法分析—高等学校—教材 IV.
①TP311.12

中国版本图书馆CIP数据核字(2012)第202370号

书 名: 数据结构与算法(第二版)
作 者: 王昆仑 李 红 主编

策 划: 吴宏伟 孟 欣

读者热线: 400-668-0820

责任编辑: 孟 欣

编辑助理: 刘丽丽

封面设计: 付 巍

封面制作: 刘 颖

责任印制: 李 佳

出版发行: 中国铁道出版社(100054, 北京市西城区右安门西街8号)

网 址: <http://www.51eds.com>

印 刷: 北京鑫正大印刷有限公司

版 次: 2007年6月第1版 2012年9月第2版 2012年9月第3次印刷

开 本: 787mm×1092mm 1/16 印张: 21 字数: 509千

印 数: 4 001~7 000册

书 号: ISBN 978-7-113-15256-7

定 价: 39.80元

版权所有 侵权必究

凡购买铁道版图书, 如有印制质量问题, 请与本社教材图书营销部联系调换。电话: (010) 63550836

打击盗版举报电话: (010) 63549504

“数据结构与算法”是高等学校计算机专业的一门核心基础课程。本书自 2007 年 6 月第一版问世后，受到兄弟院校同仁和学生的厚爱，已被数十所高校采用作为授课教材或教学参考书。作为省级精品课程“数据结构与算法”的配套教材，本书已被审定省级“十一五”规划教材。

为达到高等院校“应用型”人才的培养目标，本书在吸收了国内外教材知识体系结构的基础上，参考了众多资料并结合编者多年在高校讲授“数据结构”课程的体会与经验编写而成。本书的知识结构以“数据结构”的逻辑结构作为引线展开，突出强化数据结构的基础知识学习和算法应用能力培养，介绍了大量的应用例题，把数据结构与算法学习同解决应用实践结合起来，适合工院校相关专业学生参考使用。

本书编者以利于读者学习为目的，对本书第一版中的有关数据结构与算法中的基本知识重新用通俗的语言进行了描述，更换了部分例题，使之更适合读者自主学习；对所有基本算法和源程序进行了调试；为方便教师教学和学生自学，重新编写了每章的教学目标和教学提示；对每章后配备的填空题、判断题、选择题、简答题和算法设计题等类型的习题，重新进行了设计和编排，同时给出了所有习题的参考答案。在第二版中，由于篇幅有限，删除了第一版中有关算法实现时与 C 语言有关的内容和个别应用实例，本书习题参考答案和有关算法实例的源程序以素材的形式放在了资源网站 www.51eds.com 中，需要的读者可自行下载。

全书共分 11 章（作者分工见第 1 版前言），适合作为计算机专业或相关专业数据结构课程的教材，也可供有关科技人员自学或参考。其中基础部分教学内容，也可以作为专科教学使用。同时本书也是相关专业的读者自学数据结构与算法的一本入门教材，欢迎读者使用。

书中带“*”的章节为选学内容，可根据教学培养目标要求以及学生实际情况进行安排。

本书的编写工作得到了省级教学质量工程建设项目基金的资助，以及计算机教育界同行的关心和帮助，在此一并致谢！

由于数据结构与算法的应用发展迅速，加之编者水平有限，书中疏漏和不足之处在所难免，敬请广大读者批评指正。

主 编

2012 年 7 月

第一版前言

FOREWORD

如何合理地组织数据、高效率地处理数据是扩大计算机应用领域、提高软件效率的关键。在软件开发过程中要求“高效地”组织数据和设计出“好的”算法，并使算法用程序来实现，通过调试而成为软件，必须具备数据结构领域和算法设计领域的专门知识。

“数据结构与算法”课程主要学习在软件开发中涉及的各种常用数据结构及其常用算法，在此基础上，学习如何利用数据结构和算法解决一些基本的应用问题。通过学习，使读者基本掌握相关领域的基础知识和基本应用。

本教材为达到高等工科院校“应用型”人才的培养目标，在吸收了国、内外教材的知识体系结构的基础上，参考了众多的应用资料并根据主编多年在高校讲授《数据结构》课程的体会而编写。教材有以下几个特色：

(1) 学习一种数据结构必须掌握该数据结构的定义（逻辑结构、存储结构和基本算法）和基本应用两个方面的知识。所以，本教材以“结构”为特色，每一个数据结构的学习都围绕着该数据结构的定义，通过数据结构的逻辑结构、存储结构、基本算法和相关应用问题来介绍其基本知识和应用知识。数据的存储结构是学习和掌握算法的基础，本书在“算法的数据类型”部分中加以突出介绍。学生掌握的是数据结构的完整知识并能学为所用。

(2) 本书的知识结构以“数据结构”的逻辑结构作为引线。第1章作为本书的学习基础和预备知识。其后内容分为4个部分，从数据的逻辑结构来看，第一部分学习的是逻辑结构为“线性”的数据结构，包括第2章~第6章；第二部分学习的是逻辑结构为“树形”的数据结构，包括第7章~第8章；第三部分学习的是逻辑结构为“集合型”的数据元素，在散列存储方法下的数据结构，包括第9章；第四部分学习的是逻辑结构为“图形”的数据结构，包括第10章。第11章重点讨论了算法和程序性能分析以及算法设计方法的基本问题。这样划分教材章节的原因是：使采用本书作为教材的读者能够掌握什么是数据结构，如何设计算法以及能解决什么问题。以突出“数据结构”主题。

(3) 教材中注重工科“应用型”人才培养的需求和学习方法。吸收理工科教材的特色，在介绍新的知识点时，没有大段的文字描述，而是尽可能地采用具体的例题来加强其学习效果。

(4) 本教材中介绍了大量的应用问题。把数据结构与算法问题同应用问题结合起来，缩短了理论知识与应用问题之间的距离，在附录A中，给出了第2章~第10章中重要的基本算法和应用问题的源程序，适合工科院校相关专业的学生参考使用。对课程设计环节有很好的辅助、指导作用。

(5) 教材中注意了算法设计能力的培养。学习和培养算法设计能力是本课程的主要教学目的之一，本教材中注重介绍算法的设计过程和算法分析；在上机实验环节中，将引领学生编写十余个从简单到有一定难度的算法；在习题中，安排了一定量的基础题和适量的算法设计题，供教师和学生在学习中参考使用。

本书共分11章。第1章“数据结构和算法”作为全书的导引，主要包括有关数据、数据类型、

数据结构、算法、算法实现（算法描述工具——C 语言）、C 语言使用中的相关问题和算法分析等基本概念和相关知识。其中重点是数据、数据类型、数据结构和算法等概念；对于本教材使用的算法描述工具——C 语言，则介绍了指针、结构变量、函数、递归、动态存储分配、文件操作、程序测试与调试问题等内容，以方便本课程与 C 语言课程的衔接，便于教学。

第 2 章~第 6 章是逻辑结构为“线性”的数据结构及其应用知识内容。

第 2 章“顺序表及其应用”主要介绍的是线性逻辑结构的数据在顺序存储方法下的数据结构顺序表（包括顺序串）的概念、数据类型、数据结构、基本运算及相关应用问题。其中重点一是顺序表的定义、数据类型、数据结构、基本算法和性能分析等概念和相关知识；二是顺序表的应用，包括查找问题（简单顺序查找、二分查找、分块查找）、排序问题（直接插入排序、希尔排序、冒泡排序、快速排序、直接选择排序、归并排序）、字符处理问题（模式匹配）等内容。第 3 章“链表及其应用”主要介绍的是线性逻辑结构的数据在链接存储方法下的数据结构链表的相关知识。主要是单链表、循环链表的数据类型描述、数据结构、基本运算及其实现以及链表的相关应用问题，在此基础上介绍了链串的相关知识。在应用方面有多项式的相加问题、归并问题、箱子排序问题和链表在字符处理方面的应用问题等。第 4 章“堆栈及其应用”介绍在两种不同的存储结构下设计的堆栈，即顺序栈和链栈的相关知识，了解堆栈的相关应用，掌握应用堆栈来解决实际问题的思想及方法。第 5 章“队列及其应用”主要介绍顺序存储和链接存储方法下的两种队列、顺序（循环）队列和链队列的数据结构、基本运算及其性能分析以及应用。第 6 章“特殊矩阵、广义表及其应用”将学习数组、稀疏矩阵和广义表的基本概念，几种特殊矩阵的存储结构及基本运算，在此基础上学习特殊矩阵的计算算法与广义表应用等相关问题。本章的重点是相关数据结构的存储结构及基本运算算法。

第 7 章和第 8 章是逻辑结构为“树形”的数据结构及其应用知识内容。

第 7 章“二叉树及其应用”的知识结构主要是：非线性数据结构二叉树的定义、性质、逻辑结构、存储结构及其各种基本运算算法，包括二叉树的建立、遍历、线索化和表达式求值等算法。在此基础上，介绍二叉树的一些应用问题，包括哈夫曼编码问题、（平衡）二叉排序树问题和堆排序问题等。第 8 章“树和森林及其应用”介绍树和森林的数据结构、基本算法及其性能分析，树和森林与二叉树之间的转换算法等，在此基础上介绍树的应用——B 树，应用 B 树来实现数据元素的动态查找。

第 9 章“散列结构及其应用”是逻辑结构为“集合型”的数据元素在散列存储方法下的数据结构及其应用知识内容。主要介绍散列结构的概念、散列存储结构——散列表、散列函数和散列表中解决冲突的处理方法——开放定址法、链地址法以及散列表的基本算法及其性能分析，在散列结构的应用方面介绍散列结构的查找问题、LZW 压缩/解压缩问题和直接存取文件问题。

第 10 章“图及其应用”是逻辑结构为“图形”的数据结构及其应用知识内容，主要介绍图的定义和基础知识，图的 4 种存储结构，图的基本算法以及图的典型应用问题（最小生成树、最短路径、拓扑排序和关键路径等）。

第 11 章“算法性能分析和算法设计方法简介”主要对算法和程序性能分析中的目的、时间复杂性和空间复杂性、复杂性要素和分析方法、时间复杂性上（下）限值、算法性能测量等问题进行讨论，并结合货箱装船、0/1 背包和迷宫老鼠等问题介绍优化问题、分而治之、贪婪算法和回溯算法等基本的算法设计方法的基本知识，介绍 NP-复杂问题和 NP-完全问题。通过对本章的学

习,使读者初步了解算法设计的常用方法,知道什么是“优质”算法和程序以及如何测量、评价算法的知识。

每章都有教学目标和教学提示,每章后面都配备有一定量的填空题、判断题、选择题、简答题和算法设计题,供读者选用。习题的参考答案或者提示在教学网站中给出,主页地址:
<http://www.51eds.com>。

本教材适合理工科高等院校本科计算机工程类专业、信息技术类专业和软件工程类相关专业使用。减少部分教学内容,也可以作为专科教学使用。同时也是相关专业的读者了解和学习数据结构与算法的一本很好的入门教材。

需要特别说明的是,本书中较细致地介绍了较多的应用问题及其算法,教学过程中由于受到教学课时的限制或者根据本校培养目标要求以及学生的实际情况,可以选讲本书中打“*”号的章节内容,也可安排有兴趣的学生选修。

本教材由王昆仑、李红主编,编写工作由王昆仑、李红和许强完成,其中王昆仑编写了第1、5、6、9、11章,李红编写了第2、3、4、7章,许强编写了第8、10章。教材中的全部算法由项响琴在 Microsoft Visual C++ 6.0 环境中进行了调试,董靖完成了部分章节的绘图工作,屠菁、黄小杰、林晓燕、彭晓舟等为本书的编写工作给予了很大的支持。另外,本书的编写工作得到了省、校两级精品课程建设项目基金的资助,还得到了计算机教育界同行的关心和帮助,在此一并致谢!

由于数据结构与算法的应用发展迅速,加之编者水平有限,书中疏漏和不妥之处恳请读者批评指正。

编者
2007年2月

目 录

CONTENTS

第 1 章 数据结构与算法概述..... 1	2.5.4 插入排序——希尔排序算法.....39
1.1 数据和数据类型..... 1	2.5.5 交换排序——冒泡排序算法.....41
1.1.1 数据和数据元素..... 1	2.5.6 交换排序——快速排序算法.....43
1.1.2 数据类型..... 2	2.5.7 选择排序——直接选择 排序算法.....46
1.1.3 抽象数据类型..... 4	2.5.8 归并排序算法.....47
1.1.4 抽象数据类型程序应用实例..... 5	2.5.9 排序算法的性能分析与比较.....51
1.1.5 数据对象..... 6	2.6 顺序表的应用 3——字符处理问题...51
1.2 数据结构..... 6	2.6.1 串和顺序串的定义及相关概念...51
1.2.1 数据的逻辑结构..... 6	2.6.2 顺序串的数据结构分析.....52
1.2.2 数据元素的存储结构..... 8	2.6.3 顺序串的基本运算.....52
1.2.3 常用的数据运算..... 10	2.6.4 顺序串的数据类型定义.....54
1.3 算法和算法评价..... 11	2.6.5 顺序串的基本运算算法.....54
1.3.1 算法的概念..... 11	2.6.6 串的模式匹配算法.....56
1.3.2 算法描述工具..... 12	本章小结.....57
1.3.3 算法的性质..... 13	本章习题.....58
1.3.4 算法的评价标准..... 14	第 3 章 链表及其应用.....63
1.4 算法性能分析..... 15	3.1 链表的基本概念.....63
1.4.1 算法的时间性能分析..... 15	3.1.1 链表的定义.....63
1.4.2 算法的空间性能分析..... 18	3.1.2 链表的逻辑结构.....63
本章小结..... 19	3.1.3 链表的存储结构.....64
本章习题..... 19	3.1.4 静态链表和动态链表.....64
第 2 章 顺序表及其应用..... 22	3.1.5 链表基本运算.....65
2.1 顺序表的基本概念..... 22	3.2 单链表的数据结构.....66
2.1.1 顺序表的定义..... 22	3.2.1 单链表的逻辑结构.....66
2.1.2 顺序表的数据结构分析..... 22	3.2.2 单链表的存储结构.....66
2.1.3 顺序表的数据类型描述..... 23	3.3 单链表基本算法及性能分析.....67
2.2 顺序表基本算法..... 24	3.3.1 单链表的基本算法.....67
2.3 顺序表基本算法性能分析..... 27	3.3.2 单链表基本算法性能分析.....71
2.3.1 时间性能分析..... 27	3.4 单链表基本运算应用实例.....71
2.3.2 空间性能分析..... 28	3.5 循环链表.....73
2.4 顺序表的应用 1——查找问题..... 28	3.5.1 单向循环链表.....74
2.4.1 查找的概念..... 28	3.5.2 双向循环链表.....75
2.4.2 简单顺序查找算法..... 29	3.5.3 双向循环链表的结点插入算法...76
2.4.3 有序表的二分查找算法..... 31	3.5.4 双向循环链表的结点删除算法...77
2.4.4 分块查找算法..... 33	3.6 链表的应用.....78
2.4.5 3 种查找算法的性能比较..... 34	3.6.1 多项式相加问题.....78
2.5 顺序表的应用 2——排序问题..... 35	3.6.2 两个链表的归并问题.....81
2.5.1 排序的概念..... 35	3.6.3 箱子排序问题.....83
2.5.2 顺序表的数据类型..... 36	3.6.4 链表在字符处理方面的 应用——链串.....85
2.5.3 插入排序——直接插入排序 算法..... 37	

本章小结	86	5.4.3 链式基数排序算法实现的 技术要点	135
本章习题	86	5.4.4 链表基数排序算法及其 性能分析	136
第4章 堆栈及其应用	91	*5.5 火车车厢重排问题	137
4.1 堆栈的基本概念	91	5.5.1 问题分析及算法思想	137
4.1.1 堆栈的定义	91	5.5.2 火车车厢重排算法设计	138
4.1.2 堆栈的逻辑结构	91	5.5.3 火车车厢重排算法及其 性能分析	139
4.1.3 堆栈的基本算法	92	*5.6 电路布线问题	140
4.2 顺序栈及其基本算法	92	5.6.1 电路布线问题分析	140
4.2.1 顺序栈的概念及其数据类型	92	5.6.2 电路布线问题算法思想	140
4.2.2 顺序栈的基本算法	93	5.6.3 电路布线问题算法设计	141
4.2.3 顺序栈基本算法性能分析	94	5.6.4 电路布线问题算法及其 性能分析	141
4.3 链栈及其基本算法	94	*5.7 识别图元问题	143
4.3.1 链栈的概念及数据类型	94	5.7.1 识别图元问题分析	143
4.3.2 链栈的基本算法	95	5.7.2 识别图元算法设计	143
4.3.3 链栈基本算法性能分析	96	5.7.3 识别图元算法及其性能分析	144
4.4 堆栈的应用	97	*5.8 工厂仿真问题	145
4.4.1 数制转换问题	97	5.8.1 工厂仿真问题分析	145
4.4.2 简单的文字编辑器问题	98	5.8.2 工厂仿真问题算法设计	146
4.4.3 表达式计算问题	99	5.8.3 工厂仿真问题算法及其 性能分析	147
4.4.4 括号匹配问题	104	本章小结	149
4.4.5 汉诺塔问题	105	本章习题	149
*4.4.6 火车车厢重排问题	109	第6章 特殊矩阵、广义表及其应用	152
4.4.7 开关盒布线问题	113	6.1 数组与矩阵	152
*4.4.8 离线等价类问题	116	6.1.1 数组与矩阵的概念及其 相互关系	152
*4.4.9 迷宫老鼠问题	119	6.1.2 数组的存储结构	153
本章小结	122	6.2 特殊矩阵的压缩存储	154
本章习题	122	6.2.1 对称矩阵及其存储结构	155
第5章 队列及其应用	125	6.2.2 三角矩阵及其存储结构	155
5.1 队列的基本概念	125	6.2.3 对角矩阵及其存储结构	156
5.1.1 队列的定义	125	6.2.4 稀疏矩阵及其存储结构	157
5.1.2 队列的逻辑结构	126	6.3 矩阵的应用实例	161
5.1.3 队列的基本运算	126	6.3.1 稀疏矩阵的转置问题	161
5.2 顺序队列及其基本运算	126	6.3.2 稀疏矩阵的加法运算	163
5.2.1 顺序队列的概念及数据类型	126	6.4 广义表	167
5.2.2 循环队列	127	6.4.1 广义表的概念	167
5.2.3 循环队列的基本运算实现	129	6.4.2 广义表的存储结构	168
5.2.4 循环队列基本算法性能分析	130	6.4.3 广义表的应用—— m 元多项式 的表示	170
5.3 链队列及其基本算法	130		
5.3.1 链队列的概念及数据类型	130		
5.3.2 链队列的基本运算实现	131		
5.3.3 链队列基本算法性能分析	133		
5.4 基数排序问题	133		
5.4.1 多关键字排序问题	133		
5.4.2 链式基数排序算法思想	134		

本章小结	172	7.8.1 堆和堆排序的概念	212
本章习题	172	7.8.2 堆的调整算法	213
第 7 章 二叉树及其应用	176	7.8.3 建堆	214
7.1 二叉树的基本概念	176	7.8.4 堆排序算法及性能分析	214
7.1.1 二叉树的定义	176	本章小结	216
7.1.2 二叉树的基本术语	177	本章习题	216
7.1.3 两种特殊的二叉树	177	第 8 章 树和森林及其应用	224
7.1.4 二叉树的性质	178	8.1 树和森林的基本概念	224
7.2 二叉树的存储结构	179	8.1.1 树和森林的定义	224
7.2.1 二叉树的顺序存储结构	179	8.1.2 树的性质	225
7.2.2 二叉树的链接存储结构	180	8.1.3 树和森林的遍历	226
7.2.3 建立二叉树的算法	181	8.1.4 树、森林与二叉树的关系	227
7.3 二叉树的遍历算法	182	8.2 树的存储结构	228
7.3.1 二叉树遍历的概念	182	8.2.1 树的顺序存储结构和 建树算法	228
7.3.2 二叉树遍历递归算法	184	8.2.2 树的链接存储结构和 建树算法	229
7.3.3 二叉树先序遍历非递归算法	185	8.3 树的基本算法及性能分析	231
7.3.4 二叉树中序遍历非递归算法	187	8.3.1 树转换为二叉树算法及 性能分析	231
7.3.5 二叉树后序遍历非递归算法	187	8.3.2 森林转换为二叉树算法及 性能分析	232
7.4 线索二叉树	188	8.3.3 二叉树转换为树算法及 性能分析	233
7.4.1 线索二叉树的概念	188	8.3.4 二叉树转换为森林算法及 性能分析	234
7.4.2 二叉树的线索化	189	8.3.5 树的遍历算法	235
7.4.3 线索二叉树的查找算法	190	8.3.6 森林的遍历算法	235
7.5 二叉树的应用 1——二叉树 遍历的应用实例	191	8.4 树的应用——B 树	236
7.6 二叉树的应用 2——哈夫曼树	193	8.4.1 B 树的概念	236
7.6.1 基本概念	193	8.4.2 B 树的数据存储类型	237
7.6.2 哈夫曼树定义	194	8.4.3 B 树的查找算法	237
7.6.3 哈夫曼树的构造过程	195	8.4.4 B 树的插入算法	239
7.6.4 哈夫曼树的存储结构及哈夫曼 算法思想	195	8.4.5 B 树的删除算法及实例	243
7.6.5 哈夫曼算法	196	本章小结	244
7.6.6 哈夫曼树的应用——哈夫曼 编码	197	本章习题	245
7.7 二叉树的应用 3——二叉排序树	198	第 9 章 散列结构及其应用	248
7.7.1 二叉排序树定义	198	9.1 散列结构的基本概念	248
7.7.2 二叉排序树的存储结构	199	9.1.1 散列结构的概念	248
7.7.3 二叉排序树的结点查找算法	199	9.1.2 散列存储结构——散列表	249
7.7.4 二叉排序树的结点插入算法	200	9.1.3 散列函数	250
7.7.5 二叉排序树的生成算法	201	9.1.4 冲突处理方法——开放定址法	253
7.7.6 二叉排序树的结点删除算法	202	9.1.5 冲突处理方法——链地址法	254
7.7.7 二叉排序树的结点查找算法 性能分析	204	9.2 线性探测散列算法	255
7.7.8 平衡二叉排序树	205		
7.8 二叉树的应用 4——堆和堆排序	212		

9.2.1 线性探测散列算法的数据类型	255	*10.5.3 最小生成树算法——克鲁斯卡尔算法	290
9.2.2 线性探测散列基本算法	255	10.6 非连通图的生成森林算法	291
9.3 链地址法散列算法	257	10.7 最短路径	293
9.3.1 链地址散列算法的数据类型	257	10.7.1 最短路径问题	293
9.3.2 链地址散列基本算法	258	*10.7.2 从某源点到其余各顶点之间的最短路径问题——迪杰斯特拉算法	294
9.4 散列结构的查找性能分析	260	*10.7.3 每一对顶点之间的最短路径问题——弗洛伊德算法	296
*9.5 散列结构应用——LZW 压缩问题	261	10.8 有向无环图及其应用	298
9.5.1 LZW 压缩问题	261	10.8.1 有向无环图及其应用问题实例	298
9.5.2 LZW 压缩算法及其算法性能分析	262	10.8.2 AOV 网及其特性	300
9.5.3 LZW 解压缩问题	262	10.8.3 拓扑排序及其算法	300
9.5.4 LZW 解压缩算法及其算法性能分析	263	10.8.4 AOE 网	302
本章小结	264	10.8.5 关键路径及有关概念	303
本章习题	264	10.8.6 求解关键路径算法思想	303
第 10 章 图及其应用	267	10.8.7 AOE 网关键路径求解算法的数据类型	304
10.1 图的概念	267	10.8.8 AOE 网中求解关键路径的算法	305
10.1.1 图的定义	267	10.8.9 关键路径求解算法分析	306
10.1.2 图的基础知识	268	本章小结	307
10.1.3 图的基本运算	271	本章习题	308
10.2 图的存储结构及其基本算法	271	*第 11 章 算法性能分析和算法设计方法	312
10.2.1 邻接矩阵及其数据类型	272	11.1 算法和程序	312
10.2.2 基于邻接矩阵的基本算法	273	11.1.1 算法与程序分析	312
10.2.3 邻接表及其数据类型	274	11.1.2 空间复杂性分析	313
10.2.4 基于邻接表的基本算法	276	11.1.3 时间复杂度分析	314
10.2.5 逆邻接表	277	11.2 再谈算法性能问题	315
10.2.6 十字链表及其数据类型	277	11.2.1 时间复杂度的上限值	315
10.2.7 基于十字链表的基本算法	279	11.2.2 时间复杂度的下限值	316
10.2.8 邻接多重表及其数据类型	280	11.2.3 有关渐进符号的计算	317
10.2.9 基于邻接多重表的基本算法	281	11.3 算法性能测量问题	318
10.3 图的遍历及算法	283	11.3.1 算法性能测量问题分析	318
10.3.1 深度优先搜索遍历	283	11.3.2 算法性能测量实例	318
10.3.2 深度优先搜索遍历算法	284	11.4 算法设计方法简介	319
10.3.3 广度优先搜索遍历	284	11.4.1 分而治之的算法设计方法	319
10.3.4 广度优先搜索遍历算法	285	11.4.2 最优化问题简介	320
10.4 有向图的连通性和最小生成树	286	11.4.3 贪婪算法设计方法	321
10.4.1 有向图的连通性分析	286	11.4.4 回溯算法设计方法	322
10.4.2 连通网的最小生成树问题——通信网络问题	286	11.5 NP 问题简介	323
10.5 图的(最小)生成树问题	287	11.6 散列表查找性能	324
10.5.1 图的生成树	287	11.7 讨论题	326
*10.5.2 最小生成树算法——普利姆算法	288		

第1章 数据结构与算法概述

教学目标：本书主要学习如何组织数据和设计算法，学习软件开发中所涉及的各种常用数据结构。作为全书导引，本章要求掌握数据、数据类型、数据结构、算法及算法分析等基本概念和基础知识。另外，本章还结合课程学习要求，复习和掌握算法描述工具——C语言中的指针类型与指针变量、结构类型与结构变量、函数与参数、递归定义和递归函数、动态存储分配、文件操作、程序测试与测试集、测试数据的设计和程序调试等问题。

教学提示：如何合理地组织数据、高效率地处理数据是扩大计算机应用领域、提高软件效率的关键。因此，必须完整地讲解和理解数据结构（逻辑结构、存储结构和相关算法）的定义及其实现的方法，算法数据类型的定义本质上是存储结构的实现。算法的时间性能分析是难点，算法的空间性能分析不可忽视。程序调试问题将影响算法的实现，在实践中要注意。

1.1 数据和数据类型

通常，我们将计算机的处理对象称为“数据”。数据的类型可分为数值型和非数值型两大类。本节介绍数据、数据元素、数据项、关键项、关键字、数据类型和抽象数据类型等有关知识。

1.1.1 数据和数据元素

1. 数据

定义 1.1 在计算机科学中，数据是指描述客观事物的数值、字符、相关符号等所有能够输入到计算机中并能被计算机程序处理的符号的总称。

在计算机系统的表示层次，数据以各种数据类型来表示，在计算机系统的物理层次，数据都是以二进制形式表示的。

【例 1-1】在计算机系统中，除数值型数据之外，字符、声音、图像、图形等信息是数据吗？

答：在计算机高级语言程序设计课程中，描述的客观事物通常以数值数据来表示。例如，从客观事物抽象而来的数值，通常用整型、实型、布尔型等基本数据类型数值来表示。

字符、声音、图像、图形等以及数据之间带有更复杂的结构关系的数据就是非数值型的数据，但能够通过编码后以二进制码输入到计算机中存储、处理和输出。所以，数据包括数值型和非数值型两大类。利用非数值型数据处理的问题很多，可以举出很多例子。

在计算机科学与技术专业中，数值型数据的处理方法通常在“计算方法”（或者叫“数值计算”）课程中学习，非数值型数据的处理方法通常在“数据结构与算法”课程中学习。

2. 数据元素

定义 1.2 数据中具有独立意义的个体称为数据元素。

数据元素是数据的基本单位，在程序设计时通常作为一个整体进行考虑和处理。在有些场合，数据元素又称元素或者记录、结点、顶点等。有时，一个数据元素可由一个数据项组成（简单型数据元素），也可由若干个数据项组成（复杂型数据元素）。

定义 1.3 数据项是数据不可分割的最小单位。

定义 1.4 关键项是可以唯一标识一条数据元素的数据项。关键项可以是一个数据项，也可以由多个数据项组合生成。

定义 1.5 关键项中的每一个值称为所在数据元素的关键字（Key Word 或 Key）。

【例 1-2】为实现图书馆书目的自动检索，将与图书相关的数据做成如表 1-1 所示的表，试分析表中的数据元素（记录）、数据项、关键项、关键字。

表 1-1 图书目录关系表

书 号	书 名	作 者	价 格	...
8420001	计算机原理	张明	17.00	...
8420002	数据结构	陈英	23.00	...
8420003	C 语言	王范	17.60	...
8420004	大学英语	解东红	21.00	...
8420005	大学物理	洪亮	23.50	...
...	...	...	...	...

表 1-1 中某一本书的相关数据（表中每一行）都是一个数据元素，每一个数据元素都具有独立意义。每一个数据元素由 4 个简单数据项（书号、书名、作者、价格）组成。书号是关键项，8420001 是第一条记录的关键字。用关键项代替所有记录，用关键字代替所在记录。也可以由“书号”和“作者”这两个数据项组合而生成一个关键项。满足“书号=8420003，作者=王范”的关键项的记录是表 1-1 中的第 3 个记录。

表中数据元素之间的关系是自上而下的线性顺序关系，故表 1-1 又称为线性表。书目的自动检索即计算机按照某个特定的要求（如给定书名），对某张表按某种查询方法（如按自上而下的顺序）进行查询。依此类推，学生成绩表、工资表、员工信息表甚至人口普查表等都可线性表。

一般来讲，数据元素是相对于所讨论的问题而言的，如对二维表来说，每个记录就是它的数据元素；对字符串来说，每个字符就是它的数据元素；对数组来说，每个成分就是它的数据元素等。

1.1.2 数据类型

1. 数据类型的概念和定义

数据类型是和数据结构密切相关的一个概念，在高级程序语言中，用以表示程序的操作对象的特性。根据计算机所处理数据的方式和结果的不同，高级程序语言中定义了几种数据类型，例如整型、实型、字符型、指针、枚举、数组、结构、共同体等。

用高级程序语言定义一种数据类型后，程序编译时计算机语言编译系统就知道以下信息：

- （1）一组性质相同的值集合。
- （2）一个预定的存储体系。
- （3）定义在这个值集合上的一组操作。

定义 1.6 数据类型是一个同类值的集合和定义在这个值集上的一组操作的总称。

数据类型可分为两类：简单数据类型和结构数据类型。

2. 简单数据类型

简单数据类型的数据是不可分解的整体，如整数、实数、字符、指针、枚举量等。

【例 1-3】解释整型数据类型。

整型数据类型通常有 short (2 字节)、int (2 字节)、long (4 字节) 等形式，其值集为某个区间上的整数，如果整型是由两个字节表示的，则其值集范围是 -32 768~32 767，定义在整型数据上的操作有单目正 (+) 操作、负 (-) 操作，双目加 (+) 操作、减 (-) 操作、乘 (*) 操作、除 (/) 操作和取模 (mod) 操作等算术运算，双目关系 (>、<、>=、<=、<>) 操作运算以及赋值 (=) 操作等。

【例 1-4】解释字符型数据类型。

字符型数据类型通常用一个字节表示，其无符号值集范围是 0~255，定义在其上的操作为赋值运算和各种关系运算。汉字通常是用两个字节表示的无符号值集。

【例 1-5】解释浮点型数据类型。

浮点型数据类型通常为 float (4 字节)、double (4 字节)、long double (8 字节) 形式，其值集为某个区间上的浮点数，如果整型是 4 个字节表示的，其值集范围是 -32 768~32 767，定义在其上的操作有单目正、负，双目加、减、乘、除等算术运算，双目关系 (比较) 运算以及赋值运算。

【例 1-6】解释指针型数据类型。

指针型数据类型通常为 2 字节的形式，其值集为某个区间上的整数，定义在其上的操作为有限的加、减和赋值等运算。

对于这些基本数据类型的数据，一般用户不需要了解它们在计算机内是如何表示、运算细节是如何实现的，只需了解其外部特性，就可以运用高级语言进行程序设计对其进行操作。

3. 结构数据类型

实际上，仅有基本数据类型的数据不足以解决某些现实问题，许多程序设计语言都允许程序员利用基本数据类型作为基础来定义更多的数据类型。例如 C 语言中为用户提供了利用基本数据类型为方便某个应用问题的程序设计而定义的新的数据类型，称为用户自定义数据类型。

结构数据类型由简单数据类型按一定规则构造而成。结构数据类型中还可包含结构数据类型，所以结构数据类型的数据可分解成若干个简单数据类型的数据或数据结构，又称复合数据类型。

【例 1-7】数组数据类型分析。

数组是结构数据类型，例如，char name[20]、int a[10][10]、float b[5][10][15] 等。一维数组由若干个同种简单数据类型顺序排列而成，数组中每个值的数据类型相同；二维数组看成是一个以“一行”为一个元素的一维数组；而“一行”中简单元素有序。三维数组看成是一个以“一个面 (行 × 列)”为一个元素的一维数组；“面”为二维数组。

【例 1-8】记录数据类型分析。

例如记录 worker 定义为：

```
struct worker{
    int id;
    char name[20];
    float wage;
};
struct worker Warray[50];
```

记录是结构数据类型, 记录数据类型由若干个不同种数据类型顺序排列而成, 每一个记录值包含不同类型的数据。

【例 1-9】定义表 1-1 表示的数据类型。

已知表 1-1 中的每一行是线性表中的一个数据元素, 每一个数据元素的数据项可以由长整型的书号、字符型的书名和作者名以及实型的价格等基本类型数据表示, 可以采用如下的 C 语言语句来定义一个称为 EmployeeType 的、新的 (用户自定义) 数据类型:

```
typedef struct{
    long mun;
    char name[10], book[100];
    float price;
}EmployeeType;
```

然后把这个新的类型名为 EmployeeType 的数据类型当作一个基本数据类型来使用。使用该数据类型定义一个变量 x 如下:

```
EmployeeType x;
```

则它表达的是: 变量 x 将在后面的程序中用到, 它指向一个大约 118 个字节的主存储器区域, 用于以二进制依次存储 4 个值: 一个整数、两个字符串和一个实数。

【例 1-10】解释字符串数据类型。

字符串数据类型为字符数据类型的顺序排列结构, 定义在其上的操作主要有求串的长度、串复制、两串连接、两串比较等。

可以看出, 在数据类型的存储体系上, 用户自定义数据类型与基本数据类型是相同的。

1.1.3 抽象数据类型

定义 1.7 抽象数据类型 (Abstract Data Type, ADT) 是由用户自定义, 用以表示应用问题的数据模型。它由基本的数据类型构成, 并包括一组相关的服务 (或称操作)。

【例 1-11】如何理解“抽象数据类型”定义中的抽象意义?

例如, 各个计算机都拥有的整型数据类型就是一个抽象数据类型, 尽管它们在不同处理器上实现的方法可以不同, 但由于其定义的数学特性相同, 在用户看来都是相同的。因此, “抽象”的意义在于数据类型的数学抽象特性。

【例 1-12】在计算机高级语言中, 抽象数据类型是如何定义的?

例如, 在 C++ 语言中是通过“类”类型来描述抽象数据类型的。在“类”类型的数据部分要求只定义到数据的逻辑结构和操作说明, 不考虑数据的存储结构和操作的具体实现。数据部分被定义为类的私有数据成员, 只能给该类或派生类直接使用; 在“类”类型的操作部分定义为类的公共 (Public) 的成员函数, 提供给该类或派生类使用, 也可提供给外部操作使用。操作部分在头文件中只给出函数声明, 操作的具体实现在单独的文件中给出, 与类的定义相分离, 这样实现信息的隐藏、封装、重用和继承。

【例 1-13】抽象数据类型定义形式是怎样的?

在 C 语言中, 抽象数据类型定义形式如下:

```
ADT<抽象数据类型名> is
Data:
    <数据描述>
```

Operations:

<操作声明>

End <抽象数据类型名>

抽象数据类型可以通过固有的数据类型（如整型、实型、字符型等）来表示和实现。

【例 1-14】抽象数据类型 RECTangle 定义如下：

ADT RECTangle is

Data:

float length,width;

Operations:

Rectangle *InitRectangle(float len,float wid);

float Circumference(Rectangle &r);

float Area(Rectangle &r);

End RECTangle

其中，结构数据类型 Rectangle 定义如下：

```
typedef struct{
```

```
    float length,width;
```

```
} Rectangle;
```

```
Rectangle *InitRectangle(float len,float wid){
```

```
    Rectangle r;
```

```
    r.length=len;
```

```
    r.width=wid;
```

```
    return &r;
```

```
}
```

```
float Circumference(Rectangle *R){
```

```
    return 2*(R->length+R->width);
```

```
}
```

```
float Area(Rectangle *r){
```

```
    return R->length*R->width;
```

```
}
```

1.1.4 抽象数据类型程序应用实例

【例 1-15】C 语言程序例子。

```
typedef struct{
```

```
    float length,width;
```

```
} Rectangle;
```

```
Rectangle r;
```

```
Rectangle *InitRectangle(float len,float wid);
```

```
float Circumference(Rectangle &r);
```

```
float Area(Rectangle &r);
```

```
void main(void){
```

```
    float x,y;
```

```
    float p,s;
```

```
    printf("Input the Length and Width for a Rectangle!\n");
```

```
    scanf("%f%f",&x,&y);
```

```
    InitRectangle(x,y);
```

```
    p=Circumference(&r);
```

```
    s=Area(&r);
```

```
    printf("\nThe Circumference is:%f",p);
```

//先声明后使用

```

    printf("\nThe Area is:%f ",s);
}
Rectangle *InitRectangle(float len,float wid){
    r.length=len;
    r.width=wid;
    return &r;
}
float Circumference(Rectangle *R){
    return 2*(R->length+R->width);
}
float Area(Rectangle *R ){
    return R->length*R->width;
}

```

该程序执行情况如下:

```

Input the Length and Width for a Rectangle!
The Circumference is: 14.000000
The Area is: 12.000000

```

抽象数据类型的定义仅取决于它的一组逻辑特性,而与其在计算机内部如何表示和实现无关,即不论其内部结构如何变化,只要它的数学特性不变,都不影响其外部的使用。

1.1.5 数据对象

定义 1.8 数据对象 (Data Object) 是数据类型的实例,简称对象。

【例 1-16】数据对象举例。

例如,25 是整型数据对象;'A'是字符数据对象;char **p*,定义 *p* 为一个字符指针对象;int *a*[10],定义 *a* 为一个含有 10 个整型数的整型数组对象;Rectangle *r*,定义 *r* 为一个 Rectangle 类型的对象;RECTangle *rec*,定义 *rec* 为一个 RECTangle 抽象数据类型的对象。

1.2 数据 结 构

数据结构主要讨论数据元素之间的关系。本节介绍数据结构的有关知识,包括数据元素的逻辑结构、存储结构和常用的数据运算。

定义 1.9 在计算机科学中,数据结构是指数据元素之间的关系,它包括 3 方面的内容:

- (1) 数据元素间的逻辑关系,即数据的逻辑结构。
- (2) 数据元素以一定的存储方式存放在计算机的存储器中,形成数据元素的存储结构。
- (3) 在这些数据元素上定义的一组运算集合。

1.2.1 数据的逻辑结构

任何事物及其活动相互之间都是有联系的,所以,数据也不是孤立存在的,它们之间存在着某种关系。

定义 1.10 数据元素之间的相互联系称为数据的逻辑结构。数据元素的逻辑结构的形式定义为一个二元组 $B=(K,R)$ 。其中, B 是一种数据结构, K 是数据元素的有限集合, $K=\{k_i|1\leq i\leq n, n$