

21 世纪高等学校计算机教育实用规划教材

数据结构 (C语言版)

唐国民 王国钧 主编



清华大学出版社

数据结构 (C语言版)

唐国民 王国钧 主 编
蒋云良 李树东 邵 斌 副主编

清华大学出版社
北 京

内 容 简 介

本书是为“数据结构”课程编写的教材,也可以作为学习数据结构及其算法的C语言程序设计的参考书。

书中系统地介绍了各种常用的数据结构与算法方面的基本知识。全书共分为9章。第1章为概论,引入了数据结构与算法的一些基本概念,是全书的综述;第2~7章分别介绍了线性表、栈、队列、串、多维数组、广义表、树、二叉树和图等几种基本的数据结构;第8章和第9章分别介绍了查找和排序,它们都是数据处理时广泛使用的技术。

本书的特色是深入浅出,既注重理论又重视实践。全书配有大量的例题和详尽的注释,各章都有小结和不同类型的习题。书中自始至终使用C语言来描述算法和数据结构,全部程序都在C-Free3.5或Visual C++6.0中调试通过。

本书可作为普通高等学校计算机及相关专业本科生的教材,也可以作为专科和成人教育的教材,还可供从事计算机应用的科技人员参考。与本书配套的《数据结构实验教程(C语言版)》也由清华大学出版社正式出版。

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/唐国民,王国钧主编. —北京:清华大学出版社,2009.9

(21世纪高等学校计算机教育实用规划教材)

ISBN 978-7-302-20237-0

I. 数… II. ①唐… ②王… 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. TP311.12 TP312

中国版本图书馆CIP数据核字(2009)第139511号

责任编辑:魏江江 李玮琪

责任校对:梁毅

责任印制:孟凡玉

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:010-62776969, c-service@tup.tsinghua.edu.cn

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

印 刷 者:北京密云胶印厂

装 订 者:北京鑫海金澳胶印有限公司

经 销:全国新华书店

开 本:185×260 印 张:16 字 数:398千字

版 次:2009年9月第1版 印 次:2009年9月第1次印刷

印 数:1~3000

定 价:26.00元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。联系电话:(010)62770177 转 3103 产品编号:033974-01

出版说明

随着我国高等教育规模的扩大以及产业结构调整的不断深入,社会对高层次应用型人才的需求将更加迫切。各地高校紧密结合地方经济建设发展需要,科学运用市场调节机制,合理调整和配置教育资源,在改革和改造传统学科专业的基础上,加强工程型和应用型学科专业建设,积极设置主要面向地方支柱产业、高新技术产业、服务业的工程型和应用型学科专业,积极为地方经济建设输送各类应用型人才。各高校加大了使用信息科学等现代科学技术提升、改造传统学科专业的力度,从而实现传统学科专业向工程型和应用型学科专业的发展与转变。在发挥传统学科专业师资力量强、办学经验丰富、教学资源充裕等优势的同时,不断更新教学内容、改革课程体系,使工程型和应用型学科专业教育与经济建设相适应。计算机课程教学在从传统学科向工程型和应用型学科转变中起着至关重要的作用,工程型和应用型学科专业中的计算机课程设置、内容体系和教学手段及方法等也具有不同于传统学科的鲜明特点。

为了配合高校工程型和应用型学科专业的建设和发展,急需出版一批内容新、体系新、方法新、手段新的高水平计算机课程教材。目前,工程型和应用型学科专业计算机课程教材的建设工作仍滞后于教学改革的实践,如现有的计算机教材中有不少内容陈旧(依然用传统专业计算机教材代替工程型和应用型学科专业教材),重理论、轻实践,不能满足新的教学计划、课程设置的需要;一些课程的教材可供选择的品种太少;一些基础课的教材虽然品种较多,但低水平重复严重;有些教材内容庞杂,书越编越厚;专业课教材、教学辅助教材及教学参考书短缺,等等,都不利于学生能力的提高和素质的培养。为此,在教育部相关教学指导委员会专家的指导和建议下,清华大学出版社组织出版本系列教材,以满足工程型和应用型学科专业计算机课程教学的需要。本系列教材在规划过程中体现了如下一些基本原则和特点。

(1) 面向工程型与应用型学科专业,强调计算机在各专业中的应用。教材内容坚持基本理论适度,反映基本理论和原理的综合应用,强调实践和应用环节。

(2) 反映教学需要,促进教学发展。教材规划以新的工程型和应用型专业目录为依据。教材要适应多样化的教学需要,正确把握教学内容和课程体系的改革方向,在选择教材内容和编写体系时注意体现素质教育、创新能力与实践能力的培养,为学生知识、能力、素质协调发展创造条件。

(3) 实施精品战略,突出重点,保证质量。规划教材建设仍然把重点放在公共基础课和

专业基础课的教材建设上;特别注意选择并安排一部分原来基础比较好的优秀教材或讲义修订再版,逐步形成精品教材;提倡并鼓励编写体现工程型和应用型专业教学内容和课程体系改革成果的教材。

(4) 主张一纲多本,合理配套。基础课和专业基础课教材要配套,同一门课程可以有多种具有不同内容特点的教材。处理好教材统一性与多样化,基本教材与辅助教材,教学参考书,文字教材与软件教材的关系,实现教材系列资源配套。

(5) 依靠专家,择优选用。在制订教材规划时要依靠各课程专家在调查研究本课程教材建设现状的基础上提出规划选题。在落实主编人选时,要引入竞争机制,通过申报、评审确定主编。书稿完成后要认真实行审稿程序,确保出书质量。

繁荣教材出版事业,提高教材质量的关键是教师。建立一支高水平的以老带新的教材编写队伍才能保证教材的编写质量和建设力度,希望有志于教材建设的教师能够加入到我们的编写队伍中来。

21 世纪高等学校计算机教育实用规划教材编委会

联系人: 丁岭 dingl@tup.tsinghua.edu.cn

前 言

在信息化社会的今天,对信息技术型人才的需求很大,而信息技术型人才的培养又是高等学校人才培养的重要组成部分,本教材就是基于培养信息化人才的需要而编写的。

数据结构是计算机科学的算法理论基础和软件设计的技术基础,主要研究信息的逻辑结构及其基本操作在计算机中的表示和实现。因此,数据结构不仅是计算机专业的一门核心课程,而且也是其他理工科专业的热门选修课。学会分析研究计算机加工的数据对象的特性,能够选择合适的数据结构、存储结构和相应的算法并加以实现,是计算机工作者和其他科技工作者不可缺少的知识和能力。

数据结构课程内容抽象,知识丰富,隐藏在各章节内容中的方法和技术比较多。编者长期从事数据结构课程的教学,对该课程的教学特点和知识难点有比较深切的体会,本教材中,作者对多年来形成的数据结构课程的教学内容进行了合理的剪裁和重组,既强调数据结构的原理和方法,又特别注重其实践性与实用性。

书中介绍了各种常用的数据结构和它们在计算机中的存储表示,讨论了基于这些数据结构的基本运算(操作)和实际的执行算法,简要介绍了算法的时间分析和空间分析的技巧,并阐述了各种常用数据结构内涵的逻辑关系。

本书共分为9章。第1章为概论;第2~4章分别介绍了线性表、栈、队列和串等几种基本的数据结构,它们都属于线性结构;第5~7章分别介绍了多维数组、广义表、树和图等非线性结构;第8章和第9章分别介绍了查找和排序,它们都是数据处理时需要广泛使用的技术。

本书的特色是深入浅出,注重基本理论、基本知识和基本技能,思想性、科学性、启发性贯穿于所有章节。每一章的开头都给出了本章要点和本章学习目标,内容介绍中配有大量的例题和详尽的注释,章末有本章小结,并配置了大量的不同类型的习题。书中自始至终使用C语言来描述算法和数据结构,各章的程序都在C-Free 3.5或Visual C++6.0中调试通过,以方便读者在计算机上实践,有助于理解算法的实质和基本思想。

本书可作为计算机专业本科学生的教材,其内容可以讲授一个学期。将本书用作其他相关专业本科学生的教材,或用作计算机专业专科学生的教材,或用作成人教育学员的教材时,建议讲授教师根据实际情况适当删减教材内容(带*的部分)。在整个教学过程中,除了理论教学以外,上机实践也是一个不可缺少的环节,与本书配套的《数据结构实验教程(C语言版)》也将由清华大学出版社出版。

另外,本书也可供从事计算机应用等工作的工程技术人员参考,只需掌握C语言编程的基本技术就可以学习本书。

本书是对2005年科学出版社出版的《数据结构——C语言描述》的修订,参加本书编写及修订工作的有唐国民、王国钧、蒋云良、李树东、邵斌等。

由于编著者水平有限、时间仓促,书中难免存在一些不足之处,殷切希望广大读者批评指正。

编 者
2009年5月

目 录

第 1 章 概论	1
1.1 什么是数据结构	1
1.1.1 数据和数据元素	1
1.1.2 数据类型与数据对象	1
1.1.3 数据结构	2
1.2 为什么要学习数据结构	4
1.2.1 学习数据结构的重要性	4
1.2.2 数据结构的应用举例	5
1.3 算法和算法分析	6
1.3.1 算法的概念	6
1.3.2 算法的描述和设计	7
1.3.3 算法分析	7
本章小结	9
习题	10
第 2 章 线性表	12
2.1 线性表的基本概念	12
2.1.1 线性表的定义	12
2.1.2 线性表的基本操作	13
2.2 线性表的顺序存储	13
2.2.1 顺序表	13
2.2.2 顺序表的基本操作	14
2.2.3 一个完整的例子(1)	17
2.3 线性表的链式存储	19
2.3.1 单链表的基本概念	20
2.3.2 单链表的基本操作	21
2.3.3 一个完整的例子(2)	25
2.3.4 循环链表	27
2.3.5 双向链表	30
2.3.6 双向循环链表	32

2.3.7 静态链表	33
2.4 线性表顺序存储与链式存储的比较	34
2.5 线性表的应用	35
2.5.1 约瑟夫问题	36
2.5.2 多项式加法	38
2.5.3 电文加密	40
本章小结	42
习题	42
第3章 栈和队列	44
3.1 栈	44
3.1.1 栈的定义与基本操作	44
3.1.2 顺序栈的存储结构和操作的实现	45
3.1.3 链栈的存储结构和操作的实现	49
3.2 栈的应用	50
3.2.1 数制转换	50
3.2.2 括号匹配问题	52
3.2.3 子程序的调用	53
3.2.4 利用一个顺序栈逆置一个带头结点的单链表	54
3.3 队列	57
3.3.1 队列的定义与基本操作	57
3.3.2 链队列的存储结构和操作的实现	58
3.3.3 顺序队列的存储结构和操作的实现	60
3.4 队列的应用	64
3.4.1 打印杨辉三角形	64
3.4.2 迷宫问题:寻找一条从迷宫入口到出口的最短路径	67
3.5 递归	69
3.5.1 递归的定义与实现	70
3.5.2 递归消除	73
本章小结	75
习题	76
第4章 串	79
4.1 串的定义和基本操作	79
4.1.1 串的定义	79
4.1.2 串的基本操作	81
4.2 串的实现	82
4.2.1 串的定长顺序存储	82
4.2.2 串的堆存储结构	85

4.2.3 串的块链存储结构	86
4.3 串的模式匹配算法	91
4.3.1 基本的模式匹配算法	91
4.3.2 模式匹配的改进算法——KMP 算法	93
本章小结	95
习题	95
第 5 章 多维数组和广义表	97
5.1 多维数组	97
5.1.1 多维数组的定义	97
5.1.2 数组的存储结构	98
5.2 矩阵的压缩存储	99
5.2.1 特殊矩阵	99
5.2.2 稀疏矩阵	101
5.3 广义表	107
本章小结	109
习题	109
第 6 章 树和二叉树	111
6.1 树的概念与基本操作	111
6.1.1 树的定义	111
6.1.2 树的一些基本概念	112
6.1.3 树的基本操作	112
6.2 二叉树	113
6.2.1 二叉树的定义和基本操作	113
6.2.2 二叉树的性质	114
6.2.3 二叉树的存储结构	115
6.3 二叉树的遍历与线索化	116
6.3.1 二叉树的遍历	116
6.3.2 线索二叉树	119
6.3.3 基于遍历的应用与线索二叉树的应用	121
6.4 树和森林	126
6.4.1 树的存储结构	126
6.4.2 树、森林和二叉树之间的转换	128
6.4.3 树和森林的遍历	132
6.5 哈夫曼树及其应用	133
6.5.1 与哈夫曼树相关的基本概念	133
6.5.2 哈夫曼树的应用	134
6.5.3 哈夫曼编码算法的实现	137

* 6.6 树的计数	138
本章小结	141
习题	141
第 7 章 图	144
7.1 基本概念	144
7.1.1 图的定义	144
7.1.2 图的相关术语	145
7.2 图的存储结构	147
7.2.1 邻接矩阵表示法	147
7.2.2 邻接表表示法	149
7.3 图的遍历	153
7.3.1 深度优先搜索法	153
7.3.2 广度优先搜索法	155
7.3.3 非连通图的遍历	156
7.4 生成树与最小生成树	157
7.4.1 生成树的概念	157
7.4.2 构造最小生成树的普里姆(Prim)算法	158
7.4.3 构造最小生成树的克鲁斯卡尔(Kruskal)算法	160
7.5 最短路径	163
7.5.1 从某个源点到其余各顶点的最短路径	163
7.5.2 每一对顶点之间的最短路径	167
7.6 拓扑排序	170
7.7 关键路径	173
本章小结	178
习题	179
第 8 章 查找	183
8.1 查找的基本概念	183
8.2 线性表的查找	184
8.2.1 顺序查找	184
8.2.2 二分查找	186
8.2.3 分块查找	189
8.3 树表的查找	191
8.3.1 二叉排序树	191
* 8.3.2 B-树	196
* 8.3.3 B-树上的基本运算	198
8.4 散列表的查找	203
8.4.1 散列表的概念	203

8.4.2	散列函数的构造方法	204
8.4.3	处理冲突的方法	205
8.4.4	散列表上的运算	208
	本章小结	212
	习题	212
第9章	排序	214
9.1	排序的基本概念	214
9.1.1	关键字与排序	214
9.1.2	排序的稳定性	215
9.1.3	排序方法的分类	215
9.1.4	排序算法性能评价	215
9.1.5	不同存储方式的排序过程	215
9.2	插入排序	216
9.2.1	直接插入排序	216
9.2.2	希尔排序	219
9.3	交换排序	221
9.3.1	冒泡排序	221
9.3.2	快速排序	222
9.4	选择排序	226
9.4.1	直接选择排序	226
9.4.2	堆排序	227
9.5	归并排序	231
9.6	基数排序	234
9.6.1	多关键字的排序	234
9.6.2	链式基数排序	235
9.7	内部排序算法比较	237
9.8	外部排序简介	239
	本章小结	240
	习题	240
	主要参考文献	243

本章要点

- ◇ 什么是数据结构
- ◇ 为什么要学习数据结构
- ◇ 数据对象和数据类型
- ◇ 算法和算法分析

本章学习目标

- ◇ 了解数据结构的基本概念,理解常用术语
- ◇ 掌握数据元素间的4类结构关系
- ◇ 掌握算法的定义及特性,算法设计的要求
- ◇ 掌握分析算法的时间复杂度和空间复杂度的方法

1.1 什么是数据结构

计算机科学是一门研究信息表示和处理的科学,而信息的表示和组织又直接关系到处理信息程序的效率。由于许多系统程序和应用程序的规模很大,结构又相当复杂,因此有必要对程序设计方法进行系统的研究,这不仅涉及程序的结构和算法,同时也涉及程序的加工对象(数据)的结构,因为数据的结构直接影响算法的选择和效率。

1.1.1 数据和数据元素

数据(data)是信息的载体,是对客观事物的符号表示,能够被计算机识别、存储和加工处理。可以说,数据是计算机程序加工的“原料”。例如,一个求解代数方程的程序所处理的对象是整数、实数或复数;一个编译程序或文本编辑程序所处理的对象是字符串。随着计算机科学和技术的发展,以及计算机应用领域的扩大,数据的含义也越来越广。目前,图像、声音、视频等都可以通过编码而由计算机处理,因此它们也属于数据的范畴。

数据元素(data element)是数据中具有独立意义的个体,是数据的基本单位,通常在计算机程序中作为一个整体进行考虑和处理,如成绩表中的学生成绩信息、通信录中的个人或组织的通信信息等。数据元素也称为元素、结点或记录。有时,一个数据元素可以划分为若干个数据项(也称字段、域),数据项是数据不可分割的最小单位。

1.1.2 数据类型与数据对象

在数据结构中往往涉及数据类型与数据对象的概念。

数据对象(data object)是性质相同的数据元素的集合,是数据的一个子集。例如,整数数据对象是集合 $N=\{0,\pm 1,\pm 2,\pm 3,\dots\}$; 大写字母字符数据对象是集合 $C=\{'A','B',\dots,'Z'\}$ 。要注意的是,计算机中的整数数据对象集合 N_1 应该是上述集合 N 的一个子集, $N_1=\{0,\pm 1,\pm 2,\dots,\pm \text{maxint}\}$, 其中 maxint 是依赖于所使用的计算机和语言的最大整数。

数据类型(data type)是计算机程序中的数据对象以及定义在这个数据对象集合上的一组操作的总称。例如,C语言中的整数类型是区间 $[-\text{maxint}, \text{maxint}]$ 上的整数,在这个集合上可以进行加、减、乘、整除、求余等操作。

数据类型可以分为原子数据类型和结构数据类型,原子数据类型是由计算机语言所提供的,如C语言中的整型、实型、字符型;结构数据类型是利用计算机语言提供的一种描述数据元素之间逻辑关系的机制,是由用户自己定义而成的,如C语言中的数组类型、结构类型等。

1.1.3 数据结构

数据结构不同于数据类型,也不同于数据对象,它不仅要描述数据类型的数据对象,而且还要描述数据对象各元素之间的相互关系。例如,需要描述数据对象元素之间的运算,并使这些运算能合法地用于数据对象各元素上。

数据结构(data structure)是指数据对象以及该数据对象集合中的数据元素之间的相互关系(数据元素的组织形式)。数据结构的研究范围主要包括:研究数据的逻辑结构和物理结构(数据结构在计算机中的表示),并对每种结构定义相适应的运算;使用某种高级程序设计语言给出各种运算的算法并分析算法的效率;研究各种数据结构在计算机科学和软件工程中的某些应用;讨论数据分类、检索等方面的技术。

数据元素的组织形式一般包含下列内容:

(1) 数据元素之间的逻辑关系,也称为数据的**逻辑结构**。数据的逻辑结构通常有下列4类(如图1.1所示):

- ① **集合**: 其中的数据元素之间除了“属于同一个集合”的关系以外,别无其他关系。
- ② **线性结构**: 其中的数据元素之间存在一对一的关系。
- ③ **树型结构**: 其中的数据元素之间存在一对多的关系。
- ④ **图状结构**(或称**网状结构**): 其中的数据元素之间存在多对多的关系。

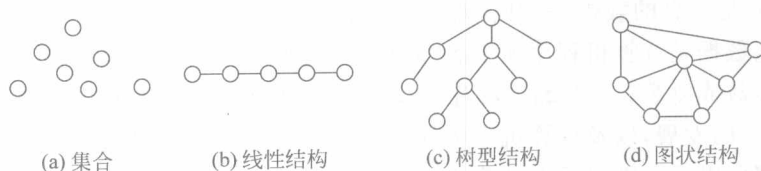


图 1.1 4类基本逻辑结构

(2) 数据元素以及它们之间的相互关系在计算机存储器内的表示(映像),称为数据的**存储结构**,也称数据的**物理结构**。

(3) 数据元素之间的**运算**,亦即对数据元素施加的操作,有时也直接称为数据的运算或操作。

由于数据的**逻辑结构**是从逻辑关系上描述数据,独立于计算机,与数据的存储无关,因

此,数据的逻辑结构可以看做是从具体问题抽象出来的数学模型。而数据的存储结构是逻辑结构用计算机语言实现的,依赖于计算机语言,因此对机器语言来说,存储结构是具体的,但需要注意的是,我们是在高级语言的层次上讨论存储结构。数据的运算是定义在数据的逻辑结构上的,每一种逻辑结构都有一个运算的集合。常用的运算有插入、删除、查找、排序等,这些运算实际上是对数据所施加的一系列抽象的操作。所谓抽象的操作,是指我们只知道这些操作是“做什么”,而不必考虑“怎么做”。只有在确定了数据的存储结构以后,才能考虑如何具体地实现这些运算。本书中所讨论的数据运算,均以 C 语言描述的算法来实现。

例 1.1 数据结构示例。

表 1.1 被称为一个数据结构,表中的每一行是一个结点(记录),由学号、姓名、各科成绩和平均成绩等数据项(字段)组成。

表 1.1 学生成绩表

学号	姓名	计算机导论	高等数学	普通物理	平均成绩
08051101	陈小洁	90	99	81	
08051102	马丽丽	85	68	78	
08051103	林春英	92	68	66	
08051104	卢华娟	70	79	93	
⋮	⋮	⋮	⋮	⋮	⋮
08051138	张晓祥	89	88	75	

首先,此表中数据元素之间的逻辑关系是:与表中任一结点(第一个结点除外)相邻且在它前面的结点(又称直接前趋)有且只有一个;与表中任一个结点(最后一个结点除外)相邻且在其后的结点(又称直接后继)也有且只有一个。表中只有第一个结点没有直接前趋,故称之为开始结点;也只有最后一个结点没有直接后继,故称之为终端结点。例如,表中“马丽丽”所在结点的直接前趋是“陈小洁”结点,其直接后继是“林春英”结点。上述结点之间的关系就构成了这张学生成绩表的逻辑结构。

其次,该表的存储结构是指用计算机语言如何表示各结点之间的这种关系,也就是说,表中的结点是按顺序邻接地存储在一些连续的单元之中,还是用指针链接在一起。

最后,在该表中,经常要查看一些学生的成绩;转入新学生时需要增加结点;学生退学时需要删除相应的结点。至于如何进行查找、插入、删除,这就是数据的运算问题。

搞清楚上述三个问题,也就弄清了学生成绩表这一数据结构。

数据结构可以理解为:按某种逻辑关系组织起来的一批数据,应用计算机语言,按一定的存储表示方式把它们存储在计算机的存储器中,并在这些数据上定义了一个运算的集合。

在不会产生混淆的前提下,可以将数据的逻辑结构简称为数据结构。本书的第 2~4 章介绍的都是线性结构;第 5~7 章介绍的都是非线性结构。

数据的存储结构可采用以下 4 种基本的存储方法得到。

(1) 顺序存储方法

该方法是将逻辑上相邻的结点存储在物理位置相邻的存储单元中,结点之间的逻辑关系由存储单元的邻接关系来体现。由此得到的存储结构称为顺序存储结构。通常顺序存储

结构是用计算机高级语言中的数组来描述的。

该方法主要应用于线性的数据结构,而非线性的数据结构可以通过某种线性化的方法来实现顺序存储。

(2) 链接存储方法

该方法不要求逻辑上相邻的结点在物理位置上也相邻,结点之间的逻辑关系是由附加的指针来表示的。由此得到的存储结构称为**链式存储结构**。通常链式存储结构是用计算机高级语言中的指针来描述的。

(3) 索引存储方法

该方法通常是在存储结点信息的同时,建立附加的**索引表**。索引表中的每一项称为**索引项**。索引项的一般形式是:(关键字,地址),所谓**关键字(key)**是指能够唯一地标识一个结点的那些数据项。若每个结点在索引表中都有一个索引项,则该索引称为**稠密索引**;若一组结点在索引表只对应一个索引项,则该索引称为**稀疏索引**。稠密索引中索引项地址指出了结点所在的存储位置,而稀疏索引中索引项地址则指出了一组结点的起始存储位置。

(4) 散列存储方法

该方法的基本思想是根据结点的关键字直接计算出该结点的存储地址。

上述4种基本的存储方法,既可以单独使用,也可以组合起来对数据结构进行存储映像。同一种逻辑结构,若采用不同的存储方法,则可以得到不同的存储结构。选择何种存储结构来表示相应的逻辑结构,应该根据具体要求而定,主要是考虑运算便捷和算法的时间、空间需求。

需要指出的是,不管怎样定义数据结构,都应该将数据的逻辑结构、数据的存储结构和数据的运算(操作)这三方面看成一个整体。读者学习时,不要孤立地去理解其中的某一个方面,而应该注意它们之间的联系。

由于存储结构是数据结构不可或缺的一个方面,因此常常将同一逻辑结构的不同存储结构,分别冠以不同的数据结构名称来标识。例如,线性表是一种逻辑结构,若采用顺序存储方法表示,则称为顺序表;若采用链式存储方法表示,则称为链表;若采用散列存储方法表示,则称为散列表。

同理,由于数据的运算也是数据结构不可分割的一个方面,因此,在给定了数据的逻辑结构和存储结构之后,按定义的运算集合及其运算性质的不同,也可以导出完全不同的数据结构。例如,若将线性表上的插入、删除运算限制在表的固定一端进行,则该线性表就称为栈;若将插入限制在表的一端进行,而将删除限制在表的另一端进行,则该线性表就称为队列。进一步而言,若线性表采用顺序表或链表作为存储结构,则对插入、删除运算做了上述限制以后,可以分别得到顺序栈或链栈、顺序队列或链队列。

1.2 为什么要学习数据结构

1.2.1 学习数据结构的重要性

数据结构是一门研究非数值计算的程序设计问题中计算机的操作对象以及它们之间的关系和操作等的学科。由于数据结构的研究不仅涉及计算机硬件,而且与计算机软件的研究

究有着密切的关系,因此它不仅是一般程序设计的基础,也是设计和实现编译程序、操作系统、数据库系统及其他系统程序和大型应用程序的重要基础。

目前,“数据结构”不仅是计算机科学与技术专业的核心课程之一,而且是其他非计算机专业的主要选修课程之一。由于在计算机系统软件和应用软件中都要用到各种数据结构,因此,仅仅掌握几种计算机语言是难以应付众多复杂问题的,要想更有效地使用计算机,就必须学习数据结构的有关知识。

在计算机发展初期,人们主要利用计算机处理数值计算问题。例如,在进行建筑设计时计算梁架结构的应力要求解线性方程组;预计人口增长情况要求解微分方程等。因为当时所涉及的运算对象比较简单(多为整数、实数或布尔型数据),所以程序设计主要考虑的是设计技巧,并不重视数据结构。随着软件、硬件的不断发展,计算机的应用领域也日益扩大,解决“非数值计算”的问题显得越来越重要。据统计,目前处理非数值计算问题大约占用了90%以上的计算机时间。由于非数值计算问题所涉及的数据结构更为复杂,数据元素之间的相互关系一般无法用数学方程式来描述,因此,解决此类问题的关键是设计出合适的数据结构。

瑞士著名的计算机科学家沃思(N. Wirth)曾指出:算法+数据结构=程序。这里的数据结构是指数据的逻辑结构和存储结构,而算法则是指对数据运算的描述。由此可见,程序设计的实质是针对所提出的问题选择一种好的数据结构,并且设计一个好的算法,而好的算法在很大程度上取决于描述该问题的数据结构。

1.2.2 数据结构的应用举例

例 1.2 电话号码查询问题。

要求编写一个电话号码的查询程序。对于任意给出的一个姓名,如果此人留有电话号码,那么就找出他的电话号码;否则就指出此人没有电话号码。要解决这个问题,首先应构造一张电话号码登记表,表中的每个结点存放姓名和电话号码两个数据项。设计的查找算法,取决于该表的结构及存储方式。第一种算法是将表中结点顺序地存储在计算机中,查找时从头开始依次核对姓名,若找到需要查找的姓名则可获得其电话号码,若找遍整张表均无所找的姓名,则表示此人无电话号码。此算法对于一个人数不多的单位是可行的,但对一个大单位或城市来说是不实用的。第二种算法是将电话号码登记表按姓氏排序,另外构造一张姓氏索引表,存储结构如图1.2所示。查找时首先在索引表中核对姓氏,然后根据索引表中的地址到电话号码登记表中查找姓名。

注意:这时已经不需要查找其他姓氏的名字了。相比之下,在新的结构上产生的第二种查找算法比第一种算法更为有效。在第8章中将进一步讨论查找策略。

例 1.3 n 个城市之间铺设光缆的问题。

假设需要在 n 个城市之间铺设光缆,并且任意两个城市之间都可以铺设。显然,在 n 个城市之间只要铺设 $n-1$ 条光缆,就能将这 n 个城市连成网络,但是由于地理位置的不同,所需经费也不同,问题是采用什么样的设计方案能使总投资最省。这个问题的数学模型如图1.3(a)所示,图中“顶点”表示城市,顶点之间的连线及其上面的数值表示可以铺设的光缆及所需经费。求解该问题的算法为:在可以铺设的 m 条光缆中选取 $n-1$ 条,既能连通 n 个城市,又使总投资为最少。实际上,这是一个“求图的最小生成树”的问题,如图1.3(b)所示。关于这个问题将在第7章中进一步讨论。