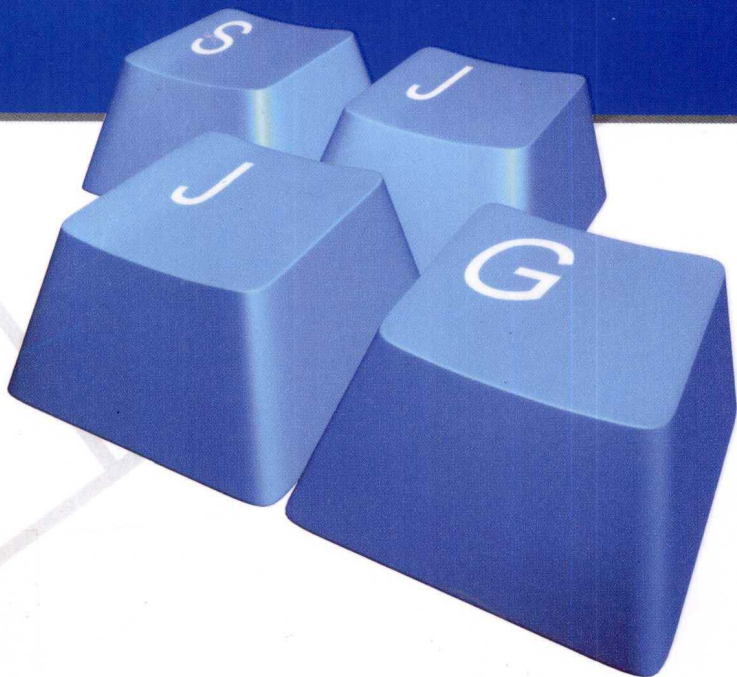


21 世纪面向工程应用型  
计算机人才培养规划教材

杨剑 主编

# 数据结构



清华大学出版社

21 世纪面向工程应用型计算机人才培养规划教材

# 数 据 结 构

杨 剑 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,其内容选取符合教学大纲要求,并兼顾学科的广度和深度,适用面广。

本书共分为9章,全面系统地介绍线性表、队列、堆栈、树、图等基本数据结构,以及这些数据结构在计算机中的存储及算法实现,介绍各种查找及排序算法的实现和效率分析,最后给出了数据结构综合应用实例。书中各种算法采用C语言描述。书中每一章除介绍相关知识点外,还给出了每章教学的建议课时、总体要求、学习重点、习题和上机实训题目,在最后一章的实习实训指导中,给出了实训题目、设计思想和设计过程,这非常有助于教师的教学安排以及学生对重点的掌握,提高其应用能力。

本书配有相关素材,包括各章的程序源代码、PPT电子教案、习题答案与解析,可从清华大学出版社网站([www.tup.com.cn](http://www.tup.com.cn))下载。

本书可以作为高等学校计算机类专业学生的教材和参考书,也可作为其他理工类专业的数据结构课程的教学用书,还可以作为计算机相关人员的自学参考书。

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/杨剑主编.--北京:清华大学出版社,2011.1

(21世纪面向工程应用型计算机人才培养规划教材)

ISBN 978-7-302-23359-6

I. ①数… II. ①杨… III. ①数据结构 IV. ①TP311.12

中国版本图书馆CIP数据核字(2010)第152194号

责任编辑:魏江江 王冰飞

责任校对:白蕾

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](mailto:jsjic@tup.tsinghua.edu.cn)

质 量 反 馈:010-62772015, [zhiliang@tup.tsinghua.edu.cn](mailto:zhiliang@tup.tsinghua.edu.cn)

印 刷 者:北京市世界知识印刷厂

装 订 者:三河市新茂装订有限公司

经 销:全国新华书店

开 本:185×260 印 张:17.5 字 数:416千字

版 次:2011年1月第1版 印 次:2011年1月第1次印刷

印 数:1~3000

定 价:28.00元

产品编号:035980-01


数据结构是计算机程序设计的重要理论技术基础,它不仅是计算机学科的核心课程,而且已成为其他理工专业的热门选修课。通过数据结构的學習,读者可以运用相关知识和技能更好地进行算法和程序的设计,也为后继课程的学习打下良好的基础。本书是为“数据结构”课程编写的教材,其内容选取符合教学大纲要求,并兼顾学科的广度和深度,适用面广。

数据结构课程主要培养以下几个方面的知识和能力:(1)掌握并能根据实际问题灵活应用基本数据结构的抽象数据类型、存储方法和主要算法。(2)掌握基本的算法设计和分析技术。(3)掌握并能应用常用的排序、查找方法。(4)具备一定的调试算法和程序、项目测试的能力。显然,合理地组织数据,有效地表示数据和正确地处理数据,这三者是提高程序设计质量的关键因素。

本书注意基本概念的引入和阐述,通过实例引入基本概念,然后对主要数据结构及其相关算法分析进行深入比较。本书是作者多年教学经验的总结,在编写本书时注意继承已有教材的优点,同时又结合实际教学和社会需求进行了大范围的内容更新。数据结构是一门理论和实践结合比较紧密的课程,本书每一章都设计有比较新颖的综合上机实习题(即实训),因此可以说,这是一本具有崭新内容的教学用书,它可以作为高等学校计算机类专业学生的教材和参考书,也可作为其他理工类专业的数据结构课程的教学用书,还可以作为计算机相关人员的自学参考书。

本书的优势是:(1)注意基本概念的引入和阐述,注重算法设计的分析方法,强调实践环节的重要性。(2)语言简明流畅,结构清晰合理,内容深入浅出。(3)每章都设有算法设计举例,每章后附有习题,供读者进行练习。(4)配套素材包括各章电子教案和习题答案,可从清华大学出版社网站([www.tup.com.cn](http://www.tup.com.cn))下载,方便老师授课和答疑。

全书共分为9章。第1章介绍数据结构的基本概念和常用术语;第2~6章介绍基本的数据结构,如线性表、栈、队列、串、数组、广义表、树和图结构,分别讨论了这几种数据类型的逻辑结构和存储结构,以及相应的算法;第7~8章介绍几种常用的查找和排序算法;第9章是实习实训指导内容,并给出一个实习实训报告示例及模板,使读者在学完基础知识后,能够利用所学知识完成一些综合实训题目的分析和设计。书中所有程序都在 Visual C++ 6.0 环境中编译通过,可以直接上机运行。另外,每章后面还配有习题和上机实训内容,方便教学和自学。



## II 数据结构

本书由电子科技大学成都学院杨剑、杨雪、郭建新、杨菊英编写。杨剑编写第1、6、7、8、9章；杨雪编写第2章；郭建新编写第3、4章；杨菊英编写第5章。最后由杨剑统稿。在本书编写过程中，电子科技大学成都学院计算机系的领导和同行给予了大力支持并提出了宝贵的意见，在此表示衷心的感谢。

由于编者水平有限，书中难免有一些不足之处，请读者批评指正。

编 者

2010年8月

<b>第 1 章 绪论</b> .....	1
1.1 数据结构的作用和意义 .....	2
1.1.1 数据结构的作用 .....	2
1.1.2 数据结构的意义 .....	2
1.2 数据结构的基本概念 .....	5
1.2.1 基本概念和术语 .....	5
1.2.2 数据结构的逻辑结构与物理结构 .....	7
1.3 数据结构的表示 .....	9
1.4 算法和算法分析 .....	11
1.4.1 算法的基本概念 .....	11
1.4.2 算法效率的度量 .....	12
1.4.3 算法效率分析 .....	13
1.5 习题 .....	16
<b>第 2 章 线性表</b> .....	19
2.1 线性表的逻辑结构 .....	20
2.1.1 线性表的概念 .....	20
2.1.2 线性表的基本操作 .....	21
2.1.3 线性表的抽象数据类型描述 .....	21
2.2 线性表的顺序表示和实现 .....	22
2.2.1 线性表的顺序表示 .....	22
2.2.2 顺序表的实现 .....	23
2.2.3 顺序表的应用 .....	29
2.3 线性表的链式表示和实现 .....	30
2.3.1 线性表的链式表示 .....	30
2.3.2 单链表的实现 .....	31
2.3.3 循环链表 .....	36
2.3.4 双向链表 .....	37

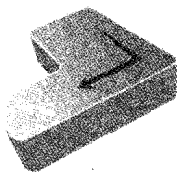
2.3.5	链表的应用 .....	39
2.4	小结 .....	41
2.4.1	线性表小结 .....	41
2.4.2	顺序表和链表的比较 .....	42
2.5	习题 .....	42
2.6	实训 .....	46
<b>第3章</b>	<b>栈和队列 .....</b>	<b>48</b>
3.1	栈 .....	48
3.1.1	栈的定义及基本运算 .....	48
3.1.2	顺序栈 .....	50
3.1.3	链栈 .....	57
3.2	队列 .....	58
3.2.1	队列的定义及基本运算 .....	58
3.2.2	顺序队列 .....	59
3.2.3	链队列 .....	68
3.3	栈和队列的应用举例 .....	72
3.3.1	栈应用之一：数制转换 .....	73
3.3.2	栈应用之二：括号匹配 .....	74
3.3.3	栈应用之三：表达式求值 .....	75
3.3.4	队列应用之一：模拟服务台前的排队现象问题 .....	79
3.3.5	队列应用之二：模拟打印机缓冲区 .....	80
3.4	习题 .....	80
3.5	实训 .....	83
<b>第4章</b>	<b>串、数组和广义表 .....</b>	<b>85</b>
4.1	串及其运算 .....	86
4.1.1	串的基本概念 .....	86
4.1.2	串的抽象数据类型定义 .....	86
4.2	串的存储结构 .....	87
4.2.1	顺序存储结构 .....	87
4.2.2	链式存储结构 .....	88
4.3	串运算的实现 .....	89
4.3.1	串的基本运算 .....	89
4.3.2	模式匹配 .....	91
4.4	多维数组 .....	93

4.5	数组的顺序表示和实现 .....	93
4.6	矩阵的压缩存储 .....	94
4.6.1	特殊矩阵 .....	95
4.6.2	稀疏矩阵 .....	96
4.7	广义表 .....	97
4.7.1	广义表的逻辑结构 .....	97
4.7.2	广义表的存储结构及实现 .....	99
4.8	习题 .....	100
<b>第5章</b>	<b>树和二叉树 .....</b>	<b>103</b>
5.1	树的定义和基本术语 .....	104
5.1.1	树的定义 .....	104
5.1.2	树的表示方法 .....	105
5.1.3	树的术语 .....	106
5.2	二叉树 .....	107
5.2.1	二叉树基本概念 .....	107
5.2.2	二叉树的性质 .....	108
5.2.3	二叉树的存储结构 .....	109
5.2.4	二叉树的基本操作及实现 .....	112
5.3	二叉树遍历 .....	115
5.3.1	二叉树遍历简介 .....	115
5.3.2	线索二叉树 .....	119
5.3.3	线索二叉树的基本操作实现 .....	121
5.4	树和森林 .....	124
5.4.1	树的存储结构 .....	124
5.4.2	树和森林与二叉树之间的转换 .....	126
5.4.3	树和森林遍历 .....	129
5.5	哈夫曼树及其应用 .....	131
5.5.1	最优二叉树(哈夫曼树) .....	131
5.5.2	哈夫曼树的构造算法 .....	133
5.5.3	哈夫曼树在编码问题中的应用 .....	134
5.6	习题 .....	136
5.7	实训 .....	142
<b>第6章</b>	<b>图 .....</b>	<b>143</b>
6.1	概述 .....	144



6.1.1	图的定义	144
6.1.2	图的常用术语及含义	144
6.2	图的存储结构	147
6.2.1	邻接矩阵	147
6.2.2	邻接表	152
6.3	图的遍历	156
6.3.1	深度优先搜索	157
6.3.2	广度优先搜索	160
6.4	生成树和最小生成树	162
6.4.1	生成树	162
6.4.2	最小生成树	163
6.5	图的应用	169
6.5.1	最短路径	169
6.5.2	拓扑排序	174
6.5.3	关键路径	175
6.6	习题	179
6.7	实训	184
<b>第7章</b>	<b>查找</b>	<b>185</b>
7.1	基本概念	186
7.2	静态查找表	187
7.2.1	顺序查找	187
7.2.2	折半查找	189
7.3	动态查找表	193
7.3.1	二叉排序树	193
7.3.2	平衡二叉树	198
7.3.3	B-树	201
7.4	哈希表	205
7.4.1	哈希表的概念	205
7.4.2	哈希函数的构建	206
7.4.3	处理冲突	208
7.4.4	哈希表的查找及其分析	211
7.5	习题	212
7.6	实训	215
<b>第8章</b>	<b>排序</b>	<b>216</b>
8.1	基本概念	216

8.2	插入排序 .....	218
8.2.1	直接插入排序 .....	218
8.2.2	希尔排序 .....	220
8.3	交换排序 .....	222
8.3.1	冒泡排序 .....	222
8.3.2	快速排序 .....	224
8.4	选择排序 .....	227
8.4.1	简单选择排序 .....	227
8.4.2	堆排序 .....	228
8.5	归并排序(二路归并排序) .....	232
8.6	各种排序方法的比较 .....	234
8.7	习题 .....	235
8.8	实训 .....	238
<b>第9章</b>	<b>实习实训 .....</b>	<b>239</b>
9.1	实习实训题目及设计要求 .....	239
9.1.1	评分参考标准 .....	239
9.1.2	停车场管理 .....	240
9.1.3	运动会分数统计 .....	240
9.1.4	学生成绩管理系统 .....	241
9.1.5	飞机售票系统 .....	242
9.1.6	仓库货品管理系统 .....	242
9.1.7	校园导游系统 .....	243
9.2	实习实训报告模板 .....	243
9.3	校园导游系统设计 .....	245



# 绪 论

**建议学时：**2 学时

## 总体要求

- 了解数据结构的意义,数据结构在计算机领域的地位和作用;
- 掌握数据结构各名词、术语的含义和有关的基本概念;数据的逻辑结构和存储结构之间的关系;
- 了解使用 C 语言对数据结构进行抽象数据类型的表示和实现的方法;
- 了解算法的五要素;
- 掌握计算语句频度估算算法时间复杂度的方法。

## 相关知识点

- 相关术语:数据、数据元素、数据项、数据对象、数据结构;
- 数据逻辑结构:集合、线性结构、树和图;
- 数据的物理结构:顺序和非顺序结构;
- 算法的五要素和时间复杂度及空间复杂度。

## 学习重点

- 数据的逻辑结构和存储结构及其之间的关系;
- 算法时间复杂度、空间复杂度及其计算。

计算机科学是一门研究信息表示、组织和处理的科学,而信息的表示和组织直接关系到处理信息的效率。随着计算机产业的迅速发展和计算机应用领域的不断扩大,计算机应用已不仅仅限于早期的科学计算,而是更多地用于控制、管理和数据处理等方面,随之而来的是处理的数据量越来越大,数据类型越来越多,数据结构越来越复杂。因此,如要编制一个高效的处理程序,就需要解决如何合理地组织数据,建立合适的数据结构,设计好的算法,来提高程序执行的效率等问题。“数据结构”这门学科就是在这样的背景下逐步形成和发展起来的。

## 1.1 数据结构的作用和意义

数据是外部世界信息的计算机化,是计算机加工处理的对象。运用计算机处理数据时,必须解决四个方面的问题:一是如何在计算机中方便、高效地表示和组织数据;二是如何在计算机存储器(内存和外存)中存储数据;三是如何对存储在计算机中的数据进行操作,可以有哪些操作,如何实现这些操作以及如何对同一问题的不同操作方法进行评价;四是必须理解每种数据结构的性能特征,以便选择一个适合于某个特定问题的数据结构。这些问题就是数据结构这门课程所要研究的主要问题。

### 1.1.1 数据结构的作用

我们知道,虽然每个人都懂得英语的语法与基本类型,但是对于同样的题目,每个人写出的作文,水平却高低不一。程序设计也和写英语作文一样,虽然程序员都懂得语言的语法与语义,但是对于同样的问题,不同程序员写出来的程序不一样。有的人写出来的程序效率很高,有的人却用复杂的方法来解决一个简单的问题。

当然,程序设计水平的提高仅仅靠看几本程序设计书是不行的。只有多思索、多练习,才能提高自己的程序设计水平;否则,书看得再多,提高也不大。程序设计水平要想提高,要多看别人写的程序,多去思考问题。从别人写的程序中,可以发现效率更高的解决方法;从思考问题的过程中,可以了解解决问题的方法常常不止一个。运用先前解决问题的经验,来解决更复杂更深入的问题,是提高程序设计水平的最有效途径。

数据结构正是前人在思索问题的过程中所想出的解决方法。一般而言,在学习程序设计一段时间后,学习“数据结构”便能让你的程序设计水平上一个台阶。如果只学会了程序设计的语法和语义,那么你能只能解决程序设计三分之一的问题,而且运用的方法并不是最有效的。但如果学会了数据结构的概念,就能在程序设计上,运用最有效的方法来解决绝大多数的问题。

“数据结构”这门课程的目的有三个。第一个是讲授常用的数据结构,这些数据结构形成了程序员基本数据结构工具。对于许多常见的问题,这些数据结构是理想的选择。程序员可以直接拿来或经过少许的修改就使用,非常方便。第二个是讲授常用的算法,这和数据结构一样,是人们在长期实践过程中的总结,程序员也可以直接拿来或经过少许的修改就使用,并且可以通过算法训练来提高程序设计水平。第三个目的是通过程序设计的技能训练促进程序员综合能力的提高。

### 1.1.2 数据结构的意义

当我们用计算机解决一个问题时,必须告诉计算机如何去做,这需要先分析问题,确

定一个适合的数据模型,然后,需要设计一个求解这个数据模型的算法,最后编写程序,经过反复调试直至得到正确结果,这就像我们求解一个数学的应用题时,需要通过问题的描述列出一个方程或方程组,然后求解该方程(组)一样。但是,需要计算机求解的多数问题比数学方程要复杂得多。下面给出几个简单的例子加以说明。

**【例 1-1】** 一个班上有 30 名学生,现在需要设计一个管理系统,完成对学生信息的查找、修改、插入或删除。

首先需考虑如何表示这 30 名学生的信息。学生信息之间的关系可以看成是一个接一个排列的一对一关系,这是一种线性结构。可以建立一个线性表,线性表中的每一个元素表示一个学生信息,如表 1-1 所示,对学生信息的查找、修改、插入或删除都应该基于该线性表进行操作。

表 1-1 学生基本信息表

学号	姓名	性别	班级
2940710801	王实	男	软工 8 班
2940710802	张斌	男	软工 8 班
2940710803	徐玲玉	女	软工 8 班
2940710804	周安	男	软工 8 班
2940710805	马小勇	男	软工 8 班
2940710806	黄莉	女	软工 8 班
⋮	⋮	⋮	⋮

学生信息是按学号一个接一个存放的。要查找某个学生信息,可以从第一个学生开始,依次向后一一比较,找到后,就可以修改了;如果要插入一个学生信息,可以先找到插入位置,把插入位置到最后的所有学生信息依次向后移动,空出位置后插入;如果要删除一个学生信息,可以先找到删除位置,然后把后面的学生信息依次前移即可。

**【例 1-2】** 计算机和人对弈问题。由于对弈的过程是在一定的规则下进行的。为使计算机能灵活对弈就必须将对弈过程中所有可能发生的情况以及相应的对策都考虑周全。同时,作为一名“好”的棋手,还应能预测棋局的发展趋势,所以,为使计算机能够和人进行对弈,必须事先将对弈的策略存入计算机,图 1-1(a)所示为一个九宫格的棋盘格局。

黑白双方交替落子,如将对弈开始到结束过程中可能出现的棋盘格局画出来,则可得到一棵倒长的“树”,图 1-1(b)是其中的一部分。“树根”是对弈开始前的棋盘格局,而“叶子”是可能出现的结局。可以看到与线性结构不同,一个格局可以派生出几个格局,也就是说,一个格局只能有一个前驱,而可以有多个后继。如果计算机对弈开始前就计算出这样一棵树,就可以知道在对弈过程哪一种走法获胜的概率大一些,就像一位高手能预测棋局发展的趋势一样,从而选择一种较好的走法。

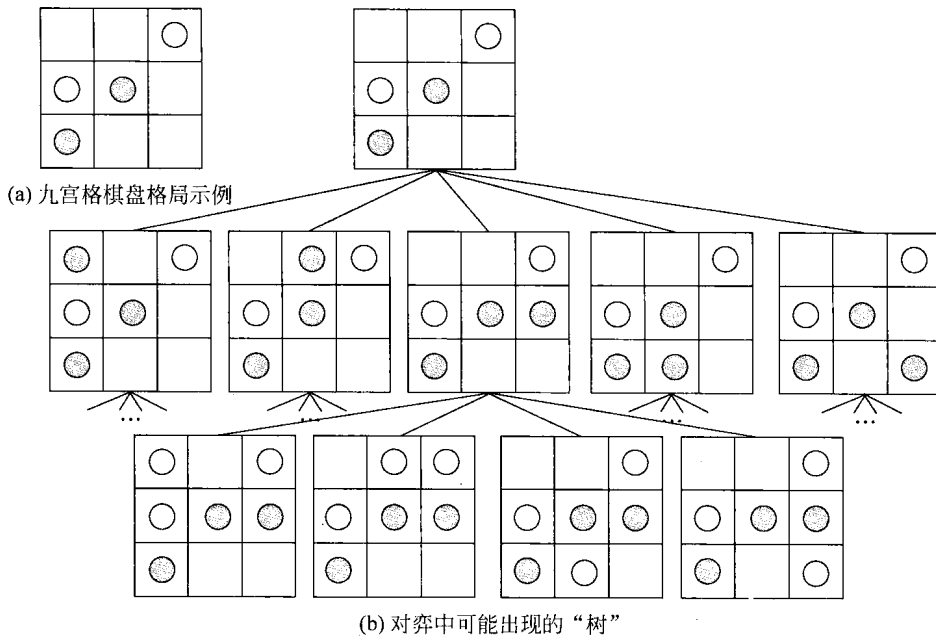


图 1-1 九宫格棋盘对弈“树”

**【例 1-3】** 田径比赛赛程安排问题。在一名选手参加多个项目的情况下,这些项目不能同时开始,否则会产生冲突,假设有一个比赛的参赛项目表如表 1-2 如示,那么 A、B、E 就不能同时开始。那么应如何安排赛程呢?

表 1-2 选手参赛项目表

姓名	参 赛 项		
ZHAO	A	B	E
QIAN	C	D	
SUN	C	E	F
LI	D	F	A
ZHOU	B	F	

在此例中,可以把一个参赛项目表示为图中的一个顶点,而当两个项目不能同时举行时,以两个顶点之间的连线表示互相矛盾的关系。图 1-2(a)所示的每个圆圈表示一个比赛项目,两个圆圈之间的连线表示这两个项目不能在同一时间安排。所以当安排项目 A 时,只能同时安排没有和 A 连线的项目,在此例中,应为 C,可以按此方法将没有冲突(互相没有连线)的项目用同一种颜色涂色,图 1-2(b)为一种涂色结果。该结果表示的安排方法为(A、C),(B、D),(E),(F)。

通常,这种模型中,每个项目表示为一个顶点。每个顶点可以和其他任意一个顶点联系。这类问题的数学模型是一种称为“图”的数据结构。

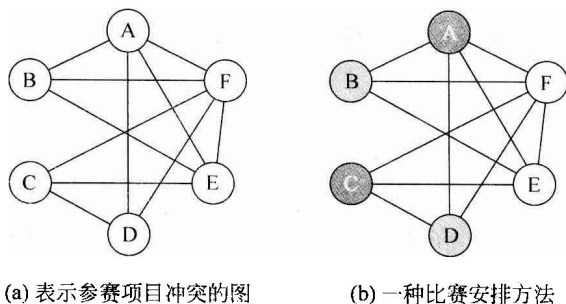


图 1-2 赛程安排示意图

综合前面 3 个例子,这类非数值计算问题的数学模型是诸如表、树、图之类的数据结构,因此,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的学科。

数据结构是一门介于数学、计算机硬件和计算机软件三者之间的计算机专业核心课程。

在计算机科学中,数据结构不仅是一般程序设计的基础,而且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。

学习“数据结构”既为进一步学习其他计算机专业课程提供必要的准备知识,也有助于提高软件设计和程序编制水平。1968 年,美国 D. E. Kunth 教授开创了数据结构的最初体系,他的名著《计算机程序设计技巧》较为系统地阐述了数据的逻辑结构和存储结构及其操作。随着计算机科学的飞速发展和应用领域的不断扩大,到 20 世纪 80 年代初期,数据结构的基础研究日臻成熟,已成为一门完整的学科。

## 1.2 数据结构的基本概念

### 1.2.1 基本概念和术语

在本小节中,将对一些常用的概念和术语进行介绍,这些概念和术语在以后的章节中会多次出现。

#### 1. 数据(Data)

数据是外部世界信息的载体,它能够被计算机识别、存储和加工处理,是计算机程序加工的原料。计算机程序处理各种各样的数据,可以是数值数据,如整数、实数或复数;也可以是非数值数据,如字符、文字、图形、图像、声音等。

## 2. 数据元素(Data Element)和数据项(Data Item)

数据元素是数据的基本单位,在计算机程序中通常被作为一个整体进行考虑和处理。数据元素有时也被称为元素、结点、顶点、记录等。一个数据元素可由若干个数据项组成。数据项是不可分割的、含有独立意义的最小数据单位,数据项有时也称为字段(Field)或域(Domain)。例如,在数据库信息处理系统中,学生数据表中的一条记录就是一个数据元素。这条记录中的学生学号、姓名、性别、籍贯、出生年月、成绩等字段就是数据项。数据项分为两种,一种叫做初等项,如学生的性别、籍贯等,在处理时不能再进行分割;另一种叫做组合项,如学生的成绩,它可以再分为数学、物理、化学成绩等更小的项。

## 3. 数据对象(Data Object)

数据对象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 $\{0, \pm 1, \pm 2, \pm 3, \dots\}$ ,字符数据对象是 $\{a, b, c, \dots\}$ 。

## 4. 数据类型(Data Type)

数据类型是高级程序设计语言中的概念,是数据的取值范围和对数据进行操作的总和。数据类型规定了程序中对象的特性。程序中的每个变量、常量或表达式的结果都应该属于某种确定的数据类型。例如,C语言中的整型(int),它可以设定的值的取值范围是 $[-32768, 32767]$ ,主要运算有 $+$ 、 $-$ 、 $*$ 、 $/$ 、 $\%$ (取模运算)等。

数据类型可分为两类:一类是非结构的原子类型,如C语言中的基本类型(整型、实型、字符型等);另一类是结构类型,它的成分可以由多个结构类型组成,并可以分解。结构类型的成分可以是非结构的,也可以是结构的。例如,C语言中数组的成分可以是整型等基本类型,也可以是数组等结构类型。

## 5. 数据结构(Data Structure)

数据结构是相互之间存在一种或多种特定关系的数据元素的集合。在任何问题中,数据元素之间都不是孤立的,而是存在着一定的关系,这种关系称为结构(Structure)。根据数据元素之间关系的不同特性,通常有4类基本数据结构:

(1) 集合(Set):如图1-3(a)所示,该结构中的数据元素除了存在“同属于一个集合”的关系外,不存在任何其他关系。

(2) 线性结构(Linear Structure):如图1-3(b)所示,该结构中的数据元素存在着一对一的关系。

(3) 树形结构(Tree Structure):如图1-3(c)所示,该结构中的数据元素存在着一对多的关系。

(4) 图状结构(Graphic Structure):如图1-3(d)所示,该结构中的数据元素存在着多对多的关系。



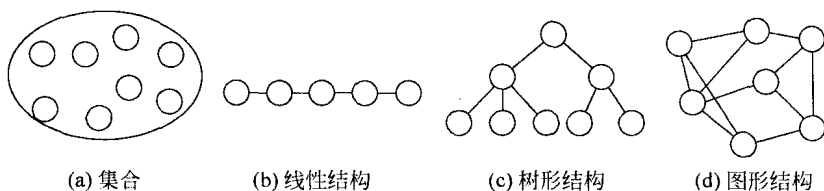


图 1-3 四类基本数据结构关系图

由于集合中元素的关系极为松散,可用其他数据结构来表示,所以本书不做专门介绍。

从数据类型和数据结构的概念可知,二者的关系非常密切。数据类型可以看作是简单的数据结构。数据的取值范围可以看作是数据元素的有限集合,而对数据进行操作的集合可以看作是数据元素之间关系的集合。

### 1.2.2 数据结构的逻辑结构与物理结构

数据结构包括数据的逻辑结构和物理结构。数据的逻辑结构(Logic Structure)是从具体问题抽象出来的数学模型,与数据在计算机中的具体存储没有关系。数据的逻辑结构独立于计算机,是数据本身所固有的特性。从逻辑上可以把数据结构分为线性结构和非线性结构,上述数据结构的定义就是数据的逻辑结构(Logic Structure),主要包括集合、线性、树和图形结构,其中集合、树和图形结构都属于非线性结构。

#### 【例 1-4】长整数表示。

以三个 4 位的十进制数表示一个含 12 位十进制数的“长整数”,则可用如下描述的数学模型表示:它是一个含三个数据元素 $\{a_1, a_2, a_3\}$ 的集合,且在集合上存在下列次序关系: $\{\langle a_1, a_2 \rangle, \langle a_2, a_3 \rangle\}$ 。则长整数“321465879345”可用 $a_1 = 3214, a_2 = 4658$ 和 $a_3 = 9345$ 的集合表示。

而  $3214, 6587, 9345 \neq 6587, 3214, 9345$

$a_1 \quad a_2 \quad a_3 \quad a_2 \quad a_1 \quad a_3$

**注意:**  $\langle x, y \rangle$  意为  $x$  和  $y$  之间存在“ $x$  领先于  $y$ ”的次序关系。而  $(x, y)$  表示  $x$  和  $y$  之间没有次序上的关系。

**【例 1-5】** 某单位的管理关系如图 1-4 所示。假设单位工作由以下人员参与:总经理、部门经理、小组长、普通职工。人员的关系为:一个总经理管理若干部门经理,每个部门经理管理若干小组长,每个小组长管理若干普通职工。

各个数据元素之间的逻辑关系可以表示为 $\{\langle \text{总经理}, \text{部门经理 } 1 \rangle, \dots, \langle \text{总经理}, \text{部门经理 } n \rangle; \langle \text{部门经理 } 1, \text{组长 } 1 \rangle, \dots, \langle \text{部门经理 } n, \text{组长 } m \rangle; \langle \text{组长 } 1, \text{职工 } 1 \rangle, \dots, \langle \text{组长 } m, \text{职工 } q \rangle\}$ 。

上述数据结构的定义仅是对操作对象的一种数学描述,是从操作对象抽象出来的数学模型,结构定义的“关系”描述的是数据元素之间的逻辑关系,所以称为逻辑结构。