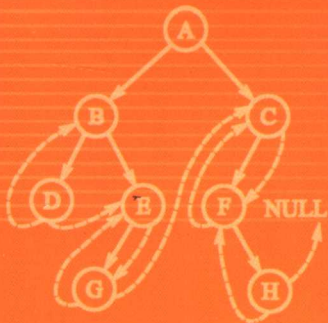


高等学校通用教材

# 数据结构教程 (第二版)

唐发根 编著



SHUJU JIEGOU  
JIAOCHENG



北京航空航天大学出版社

高等学校通用教材

# 数据结构教程

## (第二版)

唐发根 编著

北京航空航天大学出版社

## 内 容 简 介

《数据结构教程》(第二版)是1996年出版的第一版的修订版。修订版在保持第一版基本框架和特色的基础上,对其中的内容做了大量的增删和修改,书中所有算法采用C语言描述。

书中讨论了包括线性表、堆栈、队列、树和图在内的各种数据结构和数据文件的基本概念、逻辑结构与存储结构,以及在这些结构的基础上所实施的相关操作。全书仍分为11章。每一章在增加了大量例题解析的同时,还配有丰富的、各种类型的习题及相应解答,并且提供了体现各章基本内容的上机实践题。

本书可以作为高等院校计算机专业本科学生的教材,也可以作为报考高等学校计算机专业硕士研究生入学考试的复习用书,同时还可以作为从事计算机系统软件和应用软件设计与开发人员的参考资料。

### 图书在版编目(CIP)数据

数据结构教程/唐发根编著. —2版. —北京:北京  
航空航天大学出版社, 2005.5  
ISBN 7-81077-586-3

I. 数… II. 唐… III. 数据结构—高等学校—教材  
IV. TP311.12

中国版本图书馆CIP数据核字(2005)第022443号

## 数据结构教程(第二版)

唐发根 编著

责任编辑 宋淑娟

\*

北京航空航天大学出版社出版发行

北京市海淀区学院路37号(100083) 发行部电话:(010)82317024 传真:(010)82328026

<http://www.buaapress.com.cn>

E-mail: bhpress@263.net

涿州市新华印刷有限公司印装 各地书店经销

\*

开本:787×960 1/16 印张:26.25 字数:588千字

2005年5月第1版 2005年10月第2次印刷 印数:4 001~7 000册

ISBN 7-81077-586-3 定价:34.00元

# 前言

随着计算机科学与技术的迅速发展,“数据结构”作为一门新兴学科,已经越来越受到计算机界的重视,被认为是计算机领域的一门十分重要的基础学科。从课程性质上说,“数据结构”是高等学校计算机专业重要的专业基础课程之一。基于这样的原因,我们在1996年编写了《数据结构教程》。经过这些年的使用,感觉到书中某些内容的描述、选择和安排都有一些需要改进和进一步完善的地方。基于此,在本书第一版的基础上,对其做了许多必要的修订,形成了今天的《数据结构教程》(第二版)。

第二版仍然较为详细地介绍了线性结构、非线性结构和文件三大类数据结构,阐述了在这些结构上实施的相关算法的设计与实现。程序设计语言与数据结构之间存在着密切的联系:程序设计语言为数据结构的描述提供了很好的手段,数据结构为程序设计语言类型系统的发展与完善奠定了基础。同时,本书仍然强调了数据结构本身的技术及其在程序设计中的应用。

全书仍然分11章。第1章阐述数据结构的一些基本概念;第2章至第6章主要讨论线性结构,其中包括线性表、堆栈、队列以及字符串等;第7章与第8章讨论非线性结构,重点介绍树、二叉树和图的基本概念及其应用;第9章至第11章重点讨论文件的基本概念,其中包括各种数据文件的组织方法及有关的操作。

凡使用过《数据结构教程》的读者不难看出,第二版与第一版相比,虽然章节的先后次序安排基本相同,但内容还是有较大变化,去掉了一些作者认为有些过时的内容,增加了一些新的知识点,尤其是书中涉及到的所有算法都采用C语言描述。只是需要说明的一点是,在函数引用时采用了C++语言的调用参数传递方式,目的是使算法具有更好的可读性,除此之外,所有算法都完全遵循C语言标准。另外,为了与书中算法对应及保证全书体例上的统一,本书中的符号全部采用正体书写。

在本书修订过程中,依旧遵循着重基础和注意应用的原则,在第一版的基础上,每一章增加了许多例题及其解析。与第一版相比,第二版的习题与上机实践题的数量增加了近50%。可以说,阅读过本书并独立完成习题的读者,都能够比较容易掌握本书所要求的基本概念、基本技术与基本方法。

本书是作者在高等学校多年从事“算法与数据结构”课程教学实践的基础上,对原有讲义和教材进行修改,并参考了兄弟院校的同类教材编著完成的,因此,本书取材广泛、内容全面。课内讲授时间为50~70学时,也可以根据具体情况和不同要求,对内容做某些增减(如

书中带 \* 号的章节),以适合不同层次的读者。因此,本书不仅可以作为高等学校计算机专业本科学生的学习用书和报考计算机专业硕士研究生的入学考试的复习材料,还可以用做从事计算机系统软件和应用软件设计与开发人员的参考资料。

本书经过一段时间的使用,发现了若干错误或不妥之处,借此重印机会,对它们进行了修正,并应读者要求,对书中绝大部分习题进行了详细解答。由于“数据结构”本身依然是一门年轻的学科,内容还在不断变化与更新,加上作者水平有限及时间仓促等原因,书中的错误可能仍然存在,恳切希望读者给予批评指正。

作 者

2005 年 10 月于北京

# 目 录

## 第 1 章 绪 论

1.1 什么是数据结构 .....	1
1.2 数据结构的发展简史及其在计算机科学中的地位 .....	5
1.3 算 法 .....	5
1.3.1 算法及其性质 .....	5
1.3.2 基本算法 .....	7
1.3.3 算法的描述 .....	8
1.4 算法分析 .....	12
1.4.1 时间复杂度 .....	12
1.4.2 空间复杂度 .....	15
1.4.3 其他方面 .....	16
习 题 .....	16

## 第 2 章 线性表

2.1 线性表的定义及其基本操作 .....	21
2.1.1 线性表的定义 .....	21
2.1.2 线性表的基本操作 .....	22
2.2 线性表的顺序存储结构 .....	23
2.2.1 顺序存储结构的构造 .....	23
2.2.2 几种常见操作的实现 .....	25
2.2.3 顺序存储结构小结 .....	30
2.3 线性链表及其操作 .....	31
2.3.1 线性链表的构造 .....	31
2.3.2 线性链表的基本算法 .....	34
2.4 循环链表及其操作 .....	50
2.5 双向链表及其操作 .....	53
2.5.1 双向链表的构造 .....	53
2.5.2 双向链表的插入与删除算法 .....	54

* 2.6 链表的应用举例 .....	57
2.6.1 链式存储结构下的一元多项式相加 .....	57
2.6.2 打印文本文件的最后 n 行 .....	60
习 题 .....	63

### 第 3 章 数 组

3.1 数组的概念 .....	69
3.2 数组的存储结构 .....	69
3.3 矩阵的压缩存储 .....	71
3.3.1 对称矩阵的压缩存储 .....	72
3.3.2 对角矩阵的压缩存储 .....	73
3.4 稀疏矩阵的三元组表表示 .....	74
3.4.1 稀疏矩阵的三元组表存储方法 .....	74
* 3.4.2 稀疏矩阵的转置算法 .....	75
* 3.4.3 稀疏矩阵的相加算法 .....	78
* 3.4.4 稀疏矩阵的相乘算法 .....	79
* 3.5 稀疏矩阵的链表表示 .....	81
3.5.1 线性链表存储方法 .....	82
3.5.2 带行指针向量的链表存储方法 .....	83
3.5.3 十字链表存储方法 .....	83
3.6 数组的应用举例 .....	88
3.6.1 一元多项式的数组表示 .....	88
3.6.2 n 阶魔方 .....	89
习 题 .....	91

### 第 4 章 堆栈和队列

4.1 堆栈的概念及其操作 .....	95
4.1.1 堆栈的定义 .....	95
4.1.2 堆栈的基本操作 .....	96
4.2 堆栈的顺序存储结构 .....	96
4.2.1 顺序堆栈的构造 .....	97
4.2.2 顺序堆栈的基本算法 .....	97
* 4.2.3 多个堆栈共享连续空间 .....	99
4.3 堆栈的链式存储结构 .....	102

4.3.1	链接堆栈的构造 .....	103
4.3.2	链接堆栈的基本算法 .....	103
4.4	堆栈的应用举例 .....	106
4.4.1	符号匹配检查 .....	106
4.4.2	数制转换 .....	107
4.4.3	堆栈在递归中的应用 .....	108
4.4.4	表达式的计算 .....	113
4.4.5	又一个趣味游戏——迷宫 .....	117
4.5	队列的概念及其操作 .....	120
4.5.1	队列的定义 .....	120
4.5.2	队列的基本操作 .....	121
4.6	队列的顺序存储结构 .....	121
4.6.1	顺序队列的构造 .....	121
4.6.2	顺序队列的基本算法 .....	123
4.6.3	循环队列 .....	124
4.7	队列的链式存储结构 .....	127
4.7.1	链接队列的构造 .....	127
4.7.2	链接队列的基本算法 .....	128
	习 题 .....	131

## 第5章 广义表

5.1	广义表的基本概念 .....	136
5.2	广义表的存储结构 .....	137
*5.3	多元多项式的表示 .....	141
	习 题 .....	143

## 第6章 串

6.1	串的基本概念 .....	145
6.1.1	串的定义 .....	145
6.1.2	串的几个概念 .....	146
6.2	串的基本操作 .....	146
6.3	串的存储结构 .....	148
6.3.1	串的顺序存储结构 .....	149
6.3.2	串的链式存储结构 .....	150



6.4 串的几个操作 .....	151
习 题.....	157

## 第7章 树与二叉树

7.1 树的基本概念 .....	158
7.1.1 树的定义 .....	158
7.1.2 树的逻辑表示方法 .....	160
7.1.3 基本术语 .....	161
7.1.4 树的性质 .....	163
7.1.5 树的基本操作 .....	164
* 7.2 树的存储结构 .....	164
7.2.1 多重链表表示法 .....	165
7.2.2 三重链表表示法 .....	166
7.3 二叉树 .....	167
7.3.1 二叉树的定义 .....	167
7.3.2 二叉树的基本操作 .....	168
7.3.3 两种特殊形态的二叉树 .....	169
7.3.4 二叉树的性质 .....	169
* 7.3.5 二叉树与树、树林之间的转换 .....	171
7.4 二叉树的存储结构 .....	174
7.4.1 二叉树的顺序存储结构 .....	174
7.4.2 二叉树的链式存储结构 .....	176
7.5 二叉树与树的遍历 .....	180
7.5.1 二叉树的遍历 .....	181
7.5.2 由遍历序列恢复二叉树 .....	189
7.5.3 二叉树的等价性 .....	190
* 7.5.4 树和树林的遍历 .....	191
7.5.5 基于二叉树遍历操作的算法举例 .....	192
7.6 线索二叉树 .....	199
7.6.1 线索二叉树的构造 .....	200
7.6.2 线索二叉树的利用 .....	201
* 7.6.3 二叉树的线索化 .....	204
* 7.6.4 线索二叉树的更新 .....	205
7.7 二叉排序树 .....	206

7.7.1 二叉排序树的定义 .....	206
7.7.2 二叉排序树的建立 .....	206
* 7.7.3 在二叉排序树中删除结点 .....	209
7.7.4 二叉排序树的查找 .....	212
* 7.8 平衡二叉树 .....	215
7.9 哈夫曼树及其应用 .....	222
7.9.1 哈夫曼树的概念 .....	222
* 7.9.2 哈夫曼编码 .....	223
习 题 .....	227
<b>第 8 章 图</b>	
8.1 图的基本概念 .....	233
8.1.1 图的定义和基本术语 .....	233
8.1.2 图的基本操作 .....	237
8.2 图的存储方法 .....	238
8.2.1 邻接矩阵存储方法 .....	238
8.2.2 邻接表存储方法 .....	240
* 8.2.3 有向图的十字链表存储方法 .....	244
* 8.2.4 无向图的多重邻接表存储方法 .....	245
8.3 图的遍历 .....	246
8.3.1 深度优先搜索 .....	247
8.3.2 广度优先搜索 .....	250
8.3.3 连通分量 .....	251
8.4 最小生成树 .....	252
8.4.1 普里姆算法 .....	253
8.4.2 克鲁斯卡尔算法 .....	255
8.5 最短路径 .....	258
8.6 AOV 网与拓扑排序 .....	262
8.6.1 AOV 网 .....	262
8.6.2 拓扑排序 .....	263
8.6.3 拓扑排序算法 .....	264
8.7 AOE 网与关键路径 .....	269
8.7.1 AOE 网 .....	270
8.7.2 关键路径 .....	271

8.7.3 关键路径的确定 .....	271
习 题 .....	276

## 第9章 文件及查找

9.1 文件概述 .....	281
9.1.1 文件的基本概念 .....	281
9.1.2 文件的存储介质 .....	283
9.1.3 文件的基本操作 .....	285
9.2 顺序文件 .....	287
9.2.1 连续顺序文件及其查找 .....	287
9.2.2 链接顺序文件及其查找 .....	291
9.3 索引文件 .....	292
9.3.1 稠密索引文件 .....	292
9.3.2 非稠密索引分块文件 .....	293
*9.3.3 多级索引文件 .....	294
9.4 B-树和 B+树 .....	296
9.4.1 B-树的基本概念 .....	296
*9.4.2 B-树的基本操作 .....	297
9.4.3 B+树的基本概念 .....	303
*9.4.4 B+树的基本操作 .....	304
9.5 散列(hash)文件 .....	305
9.5.1 概 述 .....	305
9.5.2 散列函数的几种常见构造方法 .....	306
9.5.3 处理冲突的方法 .....	309
9.5.4 散列文件的操作 .....	313
*9.5.5 散列法的平均查找长度 .....	315
习 题 .....	316

## 第10章 内排序

10.1 概 述 .....	323
10.1.1 排序的基本概念 .....	323
10.1.2 排序的分类 .....	324
10.2 插入排序 .....	325
10.3 选择排序 .....	328

10.4	泡排序	330
10.5	谢尔排序	332
10.6	快速排序	334
10.7	堆积排序	338
10.7.1	堆积的定义	338
10.7.2	堆积排序算法	338
10.8	二路归并排序	343
10.8.1	归并子算法	343
10.8.2	一趟归并扫描子算法	344
10.8.3	二路归并排序算法	345
* 10.9	基数排序	346
10.10	各种内排序方法的比较	350
10.10.1	稳定性比较	350
10.10.2	复杂性比较	350
习 题		352
<b>* 第 11 章 外排序</b>		
11.1	概 述	358
11.2	磁带排序	359
11.2.1	多路平衡归并排序法	359
11.2.2	多步归并排序	361
11.3	初始归并段的合理分布与产生	362
11.3.1	初始归并段的合理分布	362
11.3.2	一种产生初始归并段的方法——置换选择排序	364
11.4	磁盘排序	366
习 题		369
<b>附录 上机实践题</b>		
<b>习题答案</b>		
<b>参考文献</b>		

# 第 1 章 绪 论

自 20 世纪 40 年代世界上第一台计算机问世以来,计算机产业飞速发展。就计算机系统本身而言,无论是在软件方面,还是在硬件方面,目前都已经远远超出了人们对它的预料。尤其随着计算机技术的高速发展以及微型计算机的日益普及,计算机已经广泛深入到人类社会的各个领域。现在,计算机已经不再局限于解决那些纯数值计算问题,而是更多、更广泛地应用在控制、管理以及数据处理等非数值计算的各个领域;与此相对应,计算机处理的对象也由纯粹的数值数据发展到诸如字符、表格、图像、声音、视频等各种各样具有一定结构的数据。为了有效地组织和管理好这些数据,设计出高质量的程序,高效率地使用计算机,就必须深入研究这些数据自身的特性以及它们之间存在的相互联系。这正是“数据结构”这门学科形成与发展的背景。

## 1.1 什么是数据结构

众所周知,计算机是一种信息处理装置。信息中的各个元素在客观世界中不是孤立存在的,它们之间具有一定的结构关系。如何表示这些结构关系,如何在计算机中存储数据,采用什么样的方法和技巧去加工处理这些数据,都成为数据结构这门课程所要努力解决的问题。

下面对书中常用到的几个名词术语赋以确定的含义。

### 1. 数据(data)

数据是描述客观世界的数字、字符以及一切能够输入到计算机中,并且能够被计算机程序处理的符号集合。简言之,数据就是计算机加工处理的“原料”,是信息的载体。

人们日常所涉及到的数据主要分为两类:一类是数值数据,包括整数、实数和复数等,它们主要用于工程和科学计算以及商业事务处理;另一类是非数值数据,主要包括字符和字符串以及文字、图形、语音等,它们多用于控制、管理和数据处理等领域。

数据的含义十分广泛,在不同场合可以有着不同的含义。例如,在数值计算问题中,计算机处理的对象大多数都是整数或者实数;在文字处理程序中,计算机处理的数据大多是一些字符串;而在一些控制过程问题中,数据可能又是某种信号。然而,在许多场合,人们对数据和信息的引用没有加以严格区分。信息是指数据这个集合中元素的含义;而数据则是信息的某种特定的符号表示形式,可以从中提取信息。

## 2. 数据元素(data element)

数据元素是数据整体中相对独立的基本单位,它是数据这个集合中的一个一个的客体。数据元素也称为数据结点,甚至简称结点。如数学中的一个数列称为数据,其中的一个一个元素称为数据元素。字符串就是数据,串中的单个字符就是该字符串的一个数据元素。有的时候,一个数据元素由若干个数据项组成(可见,数据项是数据不可分割的最小单位)。例如,数据文件中一个记录就是它所属文件的一个数据元素,而每个记录又是由若干个描述客体某一方面特征的数据项组成的。当然,一个记录相对于所包含的数据项而言又可以看做是数据。可以说,数据元素是本书中出现频率最高的名词。

## 3. 数据对象(data object)

一个数据对象被定义为具有相同性质的数据元素的集合。它是数据这个集合的一个子集。例如,自然数的数据对象是集合 $\{1, 2, 3, \dots\}$ ,该集合中的每一个数据元素都是一个自然数,而由 26 个大写英文字母组成的数据对象则是集合 $\{‘A’, ‘B’, ‘C’, \dots, ‘Z’\}$ 。

## 4. 结 构

在客观世界中,任何事物及活动都不会孤立地存在,都在一定程度上相互影响,相互联系,甚至相互制约。同样,数据元素之间也必然存在着某种联系,这种联系称为结构。人们不仅要考虑到数据的这种结构,而且还要考虑将要施加于数据上的各种操作及其种类。从这个意义上说,数据结构是具有结构的数据元素的集合。

这里也可以给数据结构一个形式化的描述。数据结构是一个二元组

$$\text{Data-Structure} = (D, R)$$

其中,  $D$  是数据元素的有限集合,  $R$  是  $D$  上关系的集合。

上述定义中的“关系”通常是指数据元素之间存在的逻辑关系,也称为数据的逻辑结构。由于讨论数据结构的目的在于实现计算机中对它的操作,因此还需要研究数据在计算机存储器中的表示。通常把数据结构在计算机中的表示(或者称映像)称为数据的物理结构。物理结构又称存储结构,它包括数据元素的表示以及关系的表示两个方面。

根据数据元素之间具有的不同关系,可以将数据的逻辑结构主要分为集合、线性结构与非线性结构几大类。在集合结构中,数据元素仅存在“同属于一个集合”的关系;而在线性结构中,数据元素之间的逻辑关系是“一对一”的关系,即除了第 1 个数据元素和最后那个数据元素之外,其他每一个数据元素有且仅有一个直接前驱元素,及有且仅有一个直接后继元素,这就是说,结构中的各个数据元素依次排列在一个线性序列中。对于非线性结构,一般情况下,各个数据元素不再保持在一个线性序列中,每个数据元素可能与零个或多个其他数据元素发生联系。非线性结构又分成层次结构与网状结构两种,其中层次结构称为树型结构,在这种逻辑

结构中,数据元素之间的逻辑关系一般都是“一对多”或者“多对一”的,即每个数据元素有且仅有一个直接前驱元素,但可以有多个直接后继元素。在网状结构中,数据元素之间存在的一般是“多对多”的关系,即每个数据元素可以有多个直接前驱元素,也可以有多个直接后继元素(此时这种前后关系已经没有实际意义)。网状结构也称图形结构,简称图。

一种逻辑结构通过映像便可以得到它的存储结构。由于映像的方式不同,同一种逻辑结构可以映像成不同的存储结构,如顺序映像与非顺序映像,相应的存储结构为顺序存储结构与非顺序存储结构,后者包括链式存储结构(简称链表)、索引结构和散列结构;反之,在很大程度上,数据的存储结构要能够正确反映数据元素之间具有的逻辑结构。数据的逻辑结构与存储结构是密不可分的两个方面,以后读者将会看到,一个算法的设计取决于选定的逻辑结构,而算法的实现则依赖于采用的存储结构。

为了说明这些,下面看一个例子。

表 1.1 是一个反映某个班级 30 名学生情况的表格,若利用计算机来管理这个表格,它就对应着一个数据文件。该数据文件有 30 个记录,每一个记录由若干个数据项组成,一个记录就是一个数据元素,它反映了该班一个学生的情况,30 个记录在文件中的先后顺序是按学生年龄从小到大排列的。

表 1.1 学生情况表

姓 名	性 别	民 族	年 龄	其 他
刘晓光	男	汉	15	...
王 敏	女	汉	17	...
马广生	男	回	18	...
⋮	⋮	⋮	⋮	⋮
张玉华	女	汉	20	...

对于这样一个数据文件,由于数据元素之间的逻辑关系为线性关系,因此,该数据文件的逻辑结构为线性结构,在第 2 章的讨论中将会看到,该文件被称为一个线性表。对于这个线性表,在计算机内可以有多种存储结构,除了索引存储结构和散列存储结构外,采用较多的是顺序存储结构和链式存储结构。前者在计算机存储器中用一片地址连续的存储单元(即存储单元之间不能间隔)依次存放数据元素的信息,数据元素之间的逻辑关系通过数据元素的存储地址来直接反映。在这种存储结构中,逻辑上相邻的数据元素在物理地址上也必然相邻。图 1.1 给出了这个存储结构。顺序存储结构的优点是简单,易理解,并且实际占用最少的存储空间;缺点是需要占用一片地址连续的整块空间,并且存储分配要事先进行;另外,对于一些操作的时间效率较低也是这种存储结构的主要缺陷之一。

刘晓光...	王 敏...	马广生...	...	张玉华...
--------	--------	--------	-----	--------

图 1.1 一个顺序存储结构的例子

链式存储结构是指在计算机存储器中用一片地址任意的(连续的或者不连续的)存储单元依次存放数据元素的信息,一般称每个数据元素占用的若干存储单元的组合为一个链结点。每个链结点中不仅要存放一个数据元素的数据信息,还要存放一个指出这个元素在逻辑关系中的直接后继元素所在链结点的地址,该地址被称为指针。这就是说,数据元素之间的逻辑关系通过指针间接地反映。由于不要求存储空间地址连续,因此,逻辑上相邻的数据元素在物理上不一定相邻。链式存储结构也称链表结构,简称链表。图 1.2 给出了表 1.1 所示的数据文件的链式存储结构的映像,图中用一个箭头来图形化地表示一个指针。

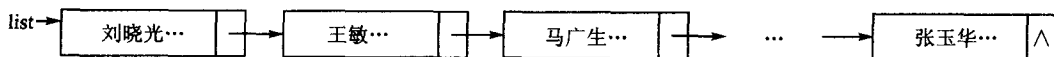


图 1.2 一个链式存储结构的例子

这种存储结构的优点是存储空间不必事先分配,在需要存储空间时可以临时申请,不会造成存储空间的浪费。在以后的讨论中将会看到,像插入和删除这样操作的时间效率采用链式存储结构远比采用顺序存储结构要高。但在这种存储结构中,不仅数据元素本身的数据信息需要占用存储空间,而且指针也有存储空间的开销,因此,从这一点来说,链式存储结构要比顺序存储结构的空间开销大。

索引结构是利用数据元素的索引关系来确定数据元素存储位置的一种存储结构,它由数据元素本身的数据信息以及索引表两个部分组成。散列结构是由事先构造的散列函数关系及处理冲突的方法来确定数据元素在散列表中的存储位置。有关这两种存储结构的具体内容,将留到后面相应的章节中讨论。

对于表 1.1 所示的数据文件,对它可以进行的操作有:若某个学生因故离开该班,则相应的记录要从文件中去掉,对应的操作就是从该线性表中删除一个数据元素;若从班外新来一个同学,则相应的操作是在该线性表中插入一个数据元素;而新年伊始,线性表中的每个数据元素中的“年龄”这个数据项都要增加一岁,对应的操作是数据元素的修改;新来的班主任想从表中了解某个同学的情况,对应操作就是在表中查找一个数据元素;上述文件中记录是按学生年龄大小排列的,若要使记录按姓名的字典顺序排列,对应的操作是对线性表进行排序,等等。

综上所述,可以看到,从不同角度对数据结构进行分类是为了更好地认识它们,更深入地了解各种结构的特性及关系。在设计某个操作时,首先要清楚数据元素之间具有什么逻辑结构,然后采用合适的存储结构来具体实现这种操作。同一种操作在不同存储结构中的实现方法可以不同,有的则完全依赖于所采用的存储结构。

因此,数据结构课程所要研究的主要内容可以简要地归纳为以下 3 个方面。

- ① 研究数据元素之间固有的客观联系(逻辑结构)。
- ② 研究数据在计算机内部的存储方法(存储结构)。
- ③ 研究在数据的各种结构(逻辑的和物理的)的基础上如何对数据实施有效的操作或处



理(算法)。

为此,应该说数据结构是一门抽象的、研究数据之间结构关系的学科。

## \* 1.2 数据结构的发展简史及其 在计算机科学中的地位

在国外,数据结构成为一门独立的课程始于20世纪60年代,在我国稍晚一些。此前,现在的“数据结构”课程的某些内容出现在其他一些诸如“编译方法”和“操作系统”的课程中。尽管后来美国一些大学的计算机系在教学计划中将“数据结构”列为一门独立的课程,但对课程的范围仍然没有做出明确的规定。当时的“数据结构”几乎同图论,特别是表处理和树的理论,是一回事。随后,数据结构的概念被扩充到包括网和集合代数论等方面,从而变成了称为“离散结构”的内容。20世纪60年代中期出现了类似于现在的“数据结构”课程,叫做“表处理语言”,它主要介绍处理表结构和树结构的语言。1968年,美国出版了一本大型著作《计算机程序设计技巧》第一卷《基本算法》(作者D·E·克努特),较为系统地阐述了数据的逻辑结构与物理结构以及运算的理论方法与技巧。20世纪60年代末到70年代初,出现了大型程序,软件也相对独立,结构程序设计逐步成为程序设计方法学的主要内容。人们越来越重视数据结构,已经认识到程序设计的实质就是对所确定的问题选择一种好的结构,从而设计一种好的算法。从那以后,各种版本的《数据结构》著作相继问世。

目前我国,随着计算机基础教育的普及以及计算机越来越广泛地应用于非数值计算问题的处理,“数据结构”不仅作为计算机专业教学计划中的重点核心课程之一,而且也开始成为其他许多非计算机专业的主要选修课。它已作为计算机专业在“程序设计语言”课程之后及各专业课程之前的一门最重要的专业基础课之一。“数据结构”不仅为学习“编译原理”、“操作系统”和“数据库原理”等一系列后继专业课程提供必要的基础,也直接为从事各类系统软件和应用软件的设计开发提供了必备的知识与方法。

“数据结构”在计算机科学领域有着十分重要的地位。它有着自己的理论、研究对象和应用范围,而且其研究内容还在不断扩充和深化。为此,作为一门课程或者一本教材,因受到特定对象和时期的限制,故难以全面反映整个数据结构的全貌。数据结构作为一门方兴未艾的新兴学科,目前仍然处在一个蓬勃发展的阶段。

## 1.3 算 法

### 1.3.1 算法及其性质

数据结构与算法之间存在着密切的联系。可以说,不了解施加于数据上的算法需求就无