



高等学校通用教材

# 数据结构教程

## (第3版)

唐发根 编著



北京航空航天大学出版社  
BEIHANG UNIVERSITY PRESS

高等学校通

# 数据结构教程

(第3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3版)》是第2版的修订版。修订版继续保持了第2版的基本框架和表达风格,对其中部分内容做了增删与补充,尤其是增加了大量的习题和解答。

书中按照“数据结构”课程教学大纲系统地讨论了数据的各种逻辑结构、存储结构以及在这些结构的基础上对数据所实施的操作。全书仍然分为11章。

本书不仅可以作为高等学校计算机专业和其他相关专业本科学生的学习用书,也可以作为计算机软件开发人员的参考资料,更是报考高等院校计算机专业硕士研究生的考生考前重要的复习资料。

### 图书在版编目(CIP)数据

数据结构教程 / 唐发根编著. -- 3版. -- 北京 :

北京航空航天大学出版社, 2017.6

ISBN 978-7-5124-2432-6

I. ①数… II. ①唐… III. ①数据结构—高等学校—  
教材 IV. ①TP311.12

中国版本图书馆 CIP 数据核字(2017)第 120061 号

版权所有,侵权必究。

### 数据结构教程(第3版)

唐发根 编著

责任编辑 陈守平

\*

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(邮编100191) <http://www.buaapress.com.cn>

发行部电话:(010)82317024 传真:(010)82328026

读者信箱:goodtextbook@126.com 邮购电话:(010)82316936

北京兴华昌盛印刷有限公司印装 各地书店经销

\*

开本:787×1092 1/16 印张:25.25 字数:646千字

2017年8月第3版 2017年8月第1次印刷 印数:4000册

ISBN 978-7-5124-2432-6 定价:56.00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。它作为计算机程序设计的灵魂,为计算机系统软件和应用软件的设计与开发提供必要的方法性的理论指导。

《数据结构教程(第2版)》一书自出版以来,得到了广大读者的认可与好评。由于作者水平所限,加上计算机技术的发展日新月异,第2版图书在内容的选取与充实方面存在进一步改进和完善的需要。本着适用与实用的原则,并结合读者的使用情况,在征询和听取了部分读者的意见和建议之后,作者对第2版进行了修正和补充,形成了今天的《数据结构教程(第3版)》。

第3版秉承第2版的思路,仍然将全书分为11章,系统地讨论了各种数据结构的基本概念和相关操作。其中,第1章绪论简要介绍数据结构与算法的基本概念、算法描述与分析的基本方法;第2章线性表,主要讨论线性表的基本概念、线性表的顺序存储结构与链式存储结构的构造原理,以及在这两种存储结构的基础上对线性表实施的基本操作;第3章重点讨论数组的基本概念,以及几种特殊矩阵的压缩存储方法;第4章讨论堆栈和队列这两种操作受限制的线性表,包括堆栈和队列的基本概念、存储结构,以及基本操作的实现;第5章简要讨论广义表的概念和存储方法;第6章讨论字符串这一非数值数据结构;第7章讨论树形结构的基本概念,包括名词术语、存储结构,以及在二叉链表存储结构基础上对二叉树实施的基本操作;第8章讨论图结构的基本概念,重点讨论图的存储方法以及几种常用算法;第9章讨论查找的基本概念和几种常用的查找方法及其时空效率分析;第10章内排序,详细讨论几种常用的内排序方法及其性能分析;第11章外排序,主要讨论如何在计算机内存与外存之间进行数据组织与数据调动来实现排序。

本书取材广泛,内容丰富,表达清晰,对“数据结构”课程中的重点和难点内容进行了较为深入细致的分析,对于一些经典算法、重点算法及其应用进行了详细的讨论,目的是使读者能够更好地掌握各种数据结构的应用。书中涉及的算法采用C语言函数描述,除个别算法外,大多数算法无须经过修改即可被其他函数调用。

“数据结构”课程是一门实践性较强的课程。本着这一认识,第3版依旧遵循着重基础与注重应用的原则,在第2版的基础上,在讨论具体内容过程中适时增加了一些算法实例,旨在进一步增强读者对课程中相关概念和内容的理解。值得说明的是,第3版对第2版的习题作了较大幅度的更改,替换了第2版中的大部分习题,新习题量超过60%;大量新习题的出现增加了全书的习题量,并且在书的后部对绝大部分习题都给出了相应的分析与参考答案。可以说,阅读过本书并独立完成习题的读者,都能够比较容易地掌握课程所要求的基本概念、基本技术与基本方法。

本书不仅可以作为高等学校计算机专业和其他相关专业本科学生的学习用书,也可以用

作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。本书的课内讲授时间建议为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。由于书中习题参考了近年来国内众多高等院校计算机专业硕士研究生入学考试以及硕士研究生入学考试计算机专业基础综合全国联考试题,因此,本书也可以作为报考高等院校计算机专业硕士研究生的考生考前复习的重要资料。

“数据结构”课程属于一门年轻的学科。随着计算机技术的飞速发展,课程的内容仍然在不断变化与更新,加上作者水平有限,书中某些地方可能考虑不周,一些内容的描述和表达尚待改进,疏漏与错误一定存在,恳请读者批评指正,并给予宝贵意见和建议。

作者

2017年2月于北京

# 目 录

第 1 章 绪 论	1
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	4
1.3 算 法	5
1.3.1 算法及其性质	5
1.3.2 基本算法	6
1.3.3 算法的描述	7
1.4 算法分析	10
1.4.1 时间复杂度	11
1.4.2 空间复杂度	13
1.4.3 其他方面	14
习 题	14
第 2 章 线性表	19
2.1 线性表的定义及其基本操作	19
2.1.1 线性表的定义	19
2.1.2 线性表的基本操作	20
2.2 线性表的顺序存储结构	21
2.2.1 顺序存储结构的构造	21
2.2.2 几种常见操作的实现	22
2.2.3 顺序存储结构小结	28
2.3 线性链表及其操作	28
2.3.1 线性链表的构造	29
2.3.2 线性链表的基本算法	31
2.4 循环链表及其操作	45
2.5 双向链表及其操作	47
2.5.1 双向链表的构造	48
2.5.2 双向链表的插入与删除算法	49
* 2.6 链表的应用举例	51
2.6.1 链式存储结构下的一元多项式相加	51
2.6.2 打印文本文件的最后 n 行	54
习 题	56

<b>第3章 数组</b> .....	63
3.1 数组的概念 .....	63
3.2 数组的存储结构 .....	63
3.3 矩阵的压缩存储 .....	65
3.3.1 对称矩阵的压缩存储 .....	65
3.3.2 对角矩阵的压缩存储 .....	66
3.4 稀疏矩阵的三元组表表示 .....	67
3.4.1 稀疏矩阵的三元组表存储方法 .....	67
* 3.4.2 稀疏矩阵的转置算法 .....	68
* 3.4.3 稀疏矩阵的相加算法 .....	70
* 3.4.4 稀疏矩阵的相乘算法 .....	72
* 3.5 稀疏矩阵的链表表示 .....	74
3.5.1 线性链表存储方法 .....	74
3.5.2 带行指针向量的链表存储方法 .....	75
3.5.3 十字链表存储方法 .....	75
3.6 数组的应用举例 .....	79
3.6.1 一元多项式的数组表示 .....	79
3.6.2 n阶魔方 .....	80
习 题 .....	82
<b>第4章 堆栈和队列</b> .....	85
4.1 堆栈的概念及其操作 .....	85
4.1.1 堆栈的定义 .....	85
4.1.2 堆栈的基本操作 .....	86
4.2 堆栈的顺序存储结构 .....	86
4.2.1 顺序堆栈的构造 .....	87
4.2.2 顺序堆栈的基本算法 .....	87
* 4.2.3 多个堆栈共享连续空间 .....	89
4.3 堆栈的链式存储结构 .....	92
4.3.1 链接堆栈的构造 .....	92
4.3.2 链接堆栈的基本算法 .....	93
4.4 堆栈的应用举例 .....	95
4.4.1 符号匹配检查 .....	95
4.4.2 数制转换 .....	95
4.4.3 堆栈在递归中的应用 .....	96
4.4.4 表达式的计算 .....	101
4.4.5 趣味游戏——迷宫 .....	104
4.5 队列的概念及其操作 .....	107

4.5.1 队列的定义 .....	107
4.5.2 队列的基本操作 .....	108
4.6 队列的顺序存储结构 .....	108
4.6.1 顺序队列的构造 .....	108
4.6.2 顺序队列的基本算法 .....	109
4.6.3 循环队列 .....	111
4.7 队列的链式存储结构 .....	113
4.7.1 链接队列的构造 .....	113
4.7.2 链接队列的基本算法 .....	114
习 题 .....	117
<b>第 5 章 广义表</b> .....	<b>122</b>
5.1 广义表的基本概念 .....	122
5.2 广义表的存储结构 .....	123
* 5.3 多元多项式的表示 .....	126
习 题 .....	128
<b>第 6 章 串</b> .....	<b>130</b>
6.1 串的基本概念 .....	130
6.1.1 串的定义 .....	130
6.1.2 串的几个概念 .....	131
6.2 串的基本操作 .....	131
6.3 串的存储结构 .....	133
6.3.1 串的顺序存储结构 .....	133
6.3.2 串的链式存储结构 .....	134
6.4 串的几个操作 .....	135
习 题 .....	140
<b>第 7 章 树与二叉树</b> .....	<b>142</b>
7.1 树的基本概念 .....	142
7.1.1 树的定义 .....	142
7.1.2 树的逻辑表示方法 .....	144
7.1.3 基本术语 .....	145
7.1.4 树的性质 .....	146
7.1.5 树的基本操作 .....	147
* 7.2 树的存储结构 .....	148
7.2.1 多重链表表示法 .....	148
7.2.2 三重链表表示法 .....	149
7.3 二叉树 .....	150



7.3.1	二叉树的定义 .....	150
7.3.2	二叉树的基本操作 .....	151
7.3.3	两种特殊形态的二叉树 .....	151
7.3.4	二叉树的性质 .....	152
* 7.3.5	二叉树与树、树林之间的转换 .....	154
7.4	二叉树的存储结构 .....	156
7.4.1	二叉树的顺序存储结构 .....	156
7.4.2	二叉树的链式存储结构 .....	158
7.5	二叉树与树的遍历 .....	163
7.5.1	二叉树的遍历 .....	163
7.5.2	由遍历序列恢复二叉树 .....	170
7.5.3	二叉树的等价性 .....	172
* 7.5.4	树和树林的遍历 .....	173
7.5.5	基于二叉树遍历操作的算法举例 .....	174
7.6	线索二叉树 .....	179
7.6.1	线索二叉树的构造 .....	180
7.6.2	线索二叉树的利用 .....	181
* 7.6.3	二叉树的线索化 .....	183
* 7.6.4	线索二叉树的更新 .....	184
7.7	二叉排序树 .....	185
7.7.1	二叉排序树的定义 .....	185
7.7.2	二叉排序树的建立(插入) .....	186
* 7.7.3	在二叉排序树中删除结点 .....	188
7.7.4	二叉排序树的查找 .....	191
* 7.8	平衡二叉树 .....	193
7.9	哈夫曼树及其应用 .....	199
7.9.1	哈夫曼树(Huffman)的概念 .....	199
* 7.9.2	哈夫曼编码 .....	200
习 题	.....	203
<b>第8章</b>	<b>图</b> .....	<b>209</b>
8.1	图的基本概念 .....	209
8.1.1	图的定义和基本术语 .....	209
8.1.2	图的基本操作 .....	213
8.2	图的存储方法 .....	213
8.2.1	邻接矩阵存储方法 .....	214
8.2.2	邻接表存储方法 .....	215
* 8.2.3	有向图的十字链表存储方法 .....	219
* 8.2.4	无向图的多重邻接表存储方法 .....	220

8.3 图的遍历 .....	221
8.3.1 深度优先搜索 .....	221
8.3.2 广度优先搜索 .....	224
8.3.3 连通分量 .....	225
8.4 最小生成树 .....	226
8.4.1 普里姆算法 .....	227
8.4.2 克鲁斯卡尔算法 .....	230
8.5 最短路径 .....	231
8.6 AOV 网与拓扑排序 .....	235
8.6.1 AOV 网 .....	235
8.6.2 拓扑排序 .....	236
8.6.3 拓扑排序算法 .....	237
8.7 AOE 网与关键路径 .....	241
8.7.1 AOE 网 .....	241
8.7.2 关键路径 .....	242
8.7.3 关键路径的确定 .....	243
习 题 .....	247
<b>第 9 章 文件及查找</b> .....	<b>253</b>
9.1 文件概述 .....	253
9.1.1 文件的基本概念 .....	253
9.1.2 文件的存储介质 .....	255
9.1.3 文件的基本操作 .....	256
9.2 顺序文件 .....	258
9.2.1 连续顺序文件及其查找 .....	258
9.2.2 链接顺序文件及其查找 .....	262
9.3 索引文件 .....	262
9.3.1 稠密索引文件 .....	262
9.3.2 非稠密索引分块文件 .....	263
* 9.3.3 多级索引文件 .....	264
9.4 B-树和 B+树 .....	266
9.4.1 B-树的基本概念 .....	266
* 9.4.2 B-树的基本操作 .....	267
9.4.3 B+树的基本概念 .....	272
* 9.4.4 B+树的基本操作 .....	273
9.5 散列(hash)文件 .....	273
9.5.1 概 述 .....	273
9.5.2 散列函数的几种常见构造方法 .....	275
9.5.3 处理冲突的方法 .....	278

9.5.4 散列文件的操作 .....	281
*9.5.5 散列法的平均查找长度 .....	284
习 题 .....	285
<b>第 10 章 内排序</b> .....	<b>290</b>
10.1 概 述 .....	290
10.1.1 排序的基本概念 .....	290
10.1.2 排序的分类 .....	291
10.2 插入排序 .....	292
10.3 选择排序 .....	295
10.4 泡排序 .....	296
10.5 谢尔排序 .....	298
10.6 快速排序 .....	300
10.7 堆积排序 .....	303
10.7.1 堆积的定义 .....	303
10.7.2 堆积排序算法 .....	304
10.8 二路归并排序 .....	308
10.8.1 归并子算法 .....	308
10.8.2 一趟归并扫描子算法 .....	309
10.8.3 二路归并排序算法 .....	310
*10.9 基数排序 .....	311
10.10 各种内排序方法的比较 .....	314
10.10.1 稳定性比较 .....	315
10.10.2 复杂性比较 .....	315
习 题 .....	316
<b>* 第 11 章 外排序</b> .....	<b>321</b>
11.1 概 述 .....	321
11.2 磁带排序 .....	322
11.2.1 多路平衡归并排序法 .....	322
11.2.2 多步归并排序 .....	324
11.3 初始归并段的合理分布与产生 .....	325
11.3.1 初始归并段的合理分布 .....	325
11.3.2 一种产生初始归并段的方法——置换选择排序 .....	326
11.4 磁盘排序 .....	328
习 题 .....	330
<b>习题答案</b> .....	<b>332</b>
<b>参考文献</b> .....	<b>393</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符号串;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)

数据元素是能够独立、完整地描述问题世界中的实体的最小数据单位,它是数据这个集合中的一个一个的元素。数据元素也称为数据结点,或者简称结点。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作是数据。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。数据结构是一个二重组

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

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**,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(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。

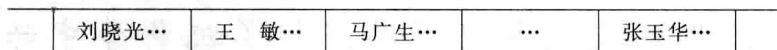


图 1.1 一个顺序存储结构的例子

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。

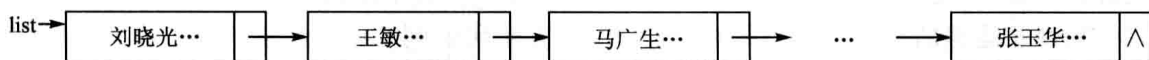


图 1.2 一个链式存储结构的例子

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,首先要搞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,叫作“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。

在我国,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”不仅成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

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,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法决定数据结构;反之,算法的结构设计和选择在很大程度上又依赖于作为其基础的数据结构,即数据结构为算法提供了工具,而算法则是运用这些工具来实施解决问题的最优方案。凡是从事过程序设计的人都会有这样一个体会:程序设计过程中相当多的时间花费在考虑如何解决问题上,即通常所说的构思算法,一旦有了合适的算法,用某种具体的程序设计语言来实现(编写程序)并不是一件很困难的事情。难怪有人说,算法+数据结构=程序。从这个角度上说,要设计出一个好的程序,很大程度上取决于设计出一个好的算法。

什么是算法?

算法是用来解决某个特定课题的一些指令的集合;或者说,是由人们组织起来准备加以实施的一系列有限的基本步骤。简单地说,算法就是解决问题的办法。如果需要解决的问题比较简单,可以采用自然语言来描述算法。可以说,程序就是用计算机语言表述的算法,而流程图是图形化的算法,甚至一个公式也可以称为算法。

待解决的特定课题一般可以分成数值的和非数值的两类。解决数值问题的算法称为数值算法。科学与工程计算方面的算法大都属于这一类算法,如求解数值积分、代数方程或线性方程组及微分方程等。解决非数值问题的算法称为非数值算法。数据处理方面的算法大多属于非数值算法,如各种各样的查找算法、排序算法、插入或删除算法以及遍历算法。非数值算法主要进行的操作是比较和逻辑运算。另外,由于特定的课题可能是递归的,也可能是非递归的,因而解决它们的算法就有递归算法与非递归算法之分。从理论上说,任何递归算法都可以通过循环的方法,或者利用堆栈或队列等机制转化成非递归算法。

在计算机领域,一个算法实质上是根据所处理问题的需要,在数据的逻辑结构和存储结构的基础上施加的一种运算。由于数据的逻辑结构与存储结构不是唯一的,在很大程度上可以由用户自行选择和设计,因而解决同一个问题的算法也不一定唯一;另外,即使具有相同的逻辑结构与存储结构,但如果算法的设计思想和技巧不同,设计出来的算法也可能大不相同。显然,根据数据处理问题的需要,为待处理的数据选择合适的逻辑结构与存储结构,进而设计出比较满意的算法,是学习“数据结构”这门课程的重要目的之一。

作为一个完整的算法应该满足下面5个标准,通常称之为算法的基本特性。

**输入** 由算法的外部提供  $n(n \geq 0)$  个有限量作为算法的输入。这里的“0个”指在算法内部确定初始条件。也就是说,在某些特殊情况下,一个算法可以没有输入。

**输出** 无论什么情况,至少有一个量作为算法的输出,没有输出的算法毫无意义。这些输



出与输入有着特定的联系。

**有穷性** 算法必须在有限的步骤内结束。这并不意味着在人们可以容忍的时间内结束,因为算法的性质不应该与具体的机器相联系,只是说明算法要有结束。

**确定性** 组成算法的每一条指令必须有着清晰明确的含义,不能让读者理解它时产生二义性或者多义性。就是说,算法的每一个步骤都必须准确定义。虽然采用自然语言传递的信息具有许多语义不清之处,但人们在理解它们时可以根据周围情景和上下文信息以及推理准确地接受它们,而机器却不能。

**有效性** 算法的每一条指令必须具有可执行性。也就是说,算法所实现的每一个动作都应该是基本的和可以付诸实施的。例如,一个数除以 0,这个动作是不能执行的。同时,实施了算法应该能够达到预期的目的。

### 1.3.2 基本算法

作为算法,还有一个重要的侧面,就是构成这个算法所依据的办法(公式、方案和原则),即通常所说的解题思路。有许多问题,只要对处理的对象进行仔细分析,处理的方法就有了,有的则不然。但是,作为寻找思路的基本思想方法对任何算法设计都是有用的。

#### 1. 枚举法(enumeration)

枚举原理是计算机领域十分重要的原理之一。所谓枚举,即从集合中一一列举各个元素。如果不知道一个命题是否为真,利用一一列举每一个元素的方法,如果都为真,找不到反例,则此命题为真。这就是枚举证明。

枚举的思想作为一个算法能够解决许多问题。一个简单的例子就是对不定方程求解的百鸡问题——公鸡每只 5 元,母鸡每只 3 元,小鸡 3 只 1 元,问 100 元钱买 100 只鸡能有多少种买法? 设  $x, y, z$  分别为 3 种鸡的只数,可得代数方程

$$x + y + z = 100$$

$$5x + 3y + z/3 = 100$$

再也找不到其他关系了。对于这个不定方程组的求解采用枚举法比较方便。若假设可以买到的公鸡、母鸡和小鸡的数目分别用  $x, y$  和  $z$  表示,则算法可以描述如下。

```
void BUYCHICKS{
    int x,y,z;
    for(x=1;x<=20;x++){
        for(y=1;y<=33;y++){
            z=100-x-y;
            if(5*x+3*y+z/3==100)
                printf("x=%d,y=%d,z=%d",x,y,z); /* 输出一组解 */
        }
    }
}
```

上述算法的基本思想是在  $x, y, z$  可能的范围内,把  $x, y, z$  可能的只数一一进行列举,如果有解,则解必在其中,而且可能不止一个。枚举法的实质是枚举所有可能的解,用检验条件来判断哪些是有用的,哪些是无用的。枚举法的特点是算法比较简单,但有时运算量较大,效率