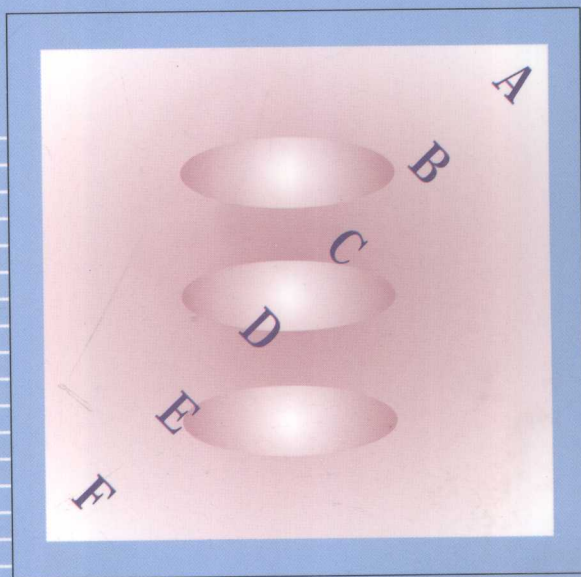


北京市高等教育学历文凭考试计算机专业教材

# 数据结构

(第二版)

唐发根 编著



科学出版社

[www.sciencep.com](http://www.sciencep.com)

北京市高等教育学历文凭考试计算机专业教材

# 数据结构

(第二版)

唐发根 编著

科学出版社

北京

北京

8837851

## 内 容 简 介

数据结构(第二版)是1998年出版的原书的修订版。修订版在保持了原书基本框架和特色的基础上对其中某些内容作了增删和修改。

书中讨论包括线性表、堆栈、队列、树、图等在内的各种数据结构和文件的基本概念,逻辑结构与存储结构,以及在这些结构的基础上实施的有关操作。算法用C语言书写,通俗易懂,具有较好的可读性与可移植性。全书共十一章,每一章都配有丰富的、各种类型的习题,并且提供了体现各章基本内容的上机实践题。

本书可作为高等教育自学考试计算机专业文凭考试课程的理想教材,也可作为普通高等院校计算机专业本科学生“数据结构”课程的教材与教学参考书。

### 图书在版编目(CIP)数据

数据结构/唐发根编著.—2版.—北京:科学出版社,2004  
北京市高等教育学历文凭考试计算机专业教材  
ISBN 7-03-012935-0

I.数… II.唐… III.数据结构-高等教育-自学考试-教材  
IV.TP311.12

中国版本图书馆CIP数据核字(2004)第011025号

责任编辑:赵卫江/责任校对:都 岚

责任印制:吕春珉/封面设计:黄华斌

科学出版社 出版

北京东黄城根北街16号

邮政编码:100717

<http://www.sciencep.com>

源海印刷有限责任公司印刷

新华书店北京发行所发行 各地新华书店经售

1998年8月第 一 版 \* 开本:787×1092 1/16

2004年3月第 二 版 印张:20

2005年8月第十二次印刷 字数:460 000

印数:38 001—39 000

定价:27.00元

(如有印装质量问题,我社负责调换<环伟>)

## 编委会名单

### 主任

怀进鹏 周 轩

### 委员

董存稳 刘 泓 沈旭昆 马殿富

李昭原 吴保国 刘 瑞 檀凤琴

何自强 唐发根 任爱华 熊桂喜

邵鸿余 郭俊美 于守谦

## 第二版前言

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。基于这样的原因,我们根据北京市高等教育自学考试委员会关于高等教育文凭考试“数据结构”课程考试大纲,在1998年编写了《数据结构》(科学出版社出版)。经过5年多的使用,我们感觉书中某些内容在描述,甚至在内容的选择和安排方面都有一些需要改进和完善的地方。于是,作者在本书第一版的基础上进行了许多必要的修订,形成了今天的《数据结构》(第二版)。

从课程性质上说,“数据结构”是高等学校计算机专业十分重要的专业基础课之一。为此,本书第二版仍然较详细地介绍线性结构、非线性结构和文件三大类数据结构,阐述在这些结构上实施的有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。

凡使用过《数据结构》(第一版)的读者不难看出,第二版与第一版相比,无论从内容本身还是内容的先后次序安排上都是相同的。全书仍然分为11章。第一章阐述数据结构的基本概念;第二章至第六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;第七章与第八章讨论非线性结构,重点讨论树和二叉树、图的基本概念及其应用;第九章至第十一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以及有关操作。

在本书的修订过程中,我们依旧本着着重基础、注意应用的原则,在第一版的基础上,每一章明显地增加了不少例题以及例题解析。全书总的习题与上机实践题的数量较第一版增加了近50%。凡是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本书所要求的基本概念、基本技术与基本方法。

数据结构是一门实践性很强的专业基础课程。通过该课程的学习,使学生能够运用“数据结构”提供的方法与技巧更好地进行算法设计和程序设计。所以,本书在讨论各种数据结构基本运算的同时都给出了相应的算法。全部算法采用C语言描述,并且都经过上机调试验证。需要说明的一点是,在函数引用时我们采用了C++语言的调用参数传递方式,目的是使得算法具有更好的可读性。

本书取材广泛、内容全面,作为高等教育自学考试计算机专业学生的教材,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,本书完全可以作为高等学校计算机专业本科学生的学习用书。同时,本书还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。

本书是作者在高等学校从事多年教学实践的基础上,对原有讲义和教材进行多次修改并参考了兄弟院校同类教材编著完成的。因此,在内容上努力遵循深入浅出的原则,在



## 序

高等教育学历文凭考试是我国高等教育事业发展过程中出现的一个新生事物。它是社会力量办学与国家考试相结合,以宽进严出、教考分离、全日制教学为特点的高等教育形式。这种形式从它产生之日起,就受到社会各界的重视和赞誉,认为它为我国社会主义经济建设对人才的大量需求又提供了一种培养手段,同时它也得到了国家有关领导同志的称赞,认为这是“穷国办大教育”的一条好的途径。北京在全国是最早进行这项试点工作的,目前有 24 所民办高校参加,开设的专业有 16 个,在校生 3 万余人,年均招生规模都在万人左右,其中计算机应用专业 1997 年成为招生规模最大的专业。经过五年的实践和总结,特别是结合国家为民办高等教育培养目标“应用性、职业性”定位的理解,我们感到,我市试点工作在发展过程中,基本建设做得还不够,其中一个表现就是抓教材建设做得不够,特别是目前市场上还缺乏与高等职业技术教育相匹配的有关教材,导致目前参加文凭考试的民办学校在教学上基本上是借用普通高校的教材,因而教材与培养要求上的矛盾就尤为突出。

我们非常感谢的是,北京航空航天大学计算机系非常积极和认真地提出了要组织编写一套适合这种考试的教材的建议,并做了很具体的安排,北京航空航天大学的许多专家、教授克服了学校教学、科研任务繁忙的困难,按时、保质地完成了撰稿工作。同时此项工作也得到了科学出版社有关同志的大力配合,没有他们耐心、细致地做组稿和出版等工作,这套教材能这样快地面世是不可想象的。

经过各方的通力协作,这套教材终于可以奉献给大家了。我们认为这套教材基本上体现了高等教育学历文凭考试计算机应用专业培养方向上的要求,在内容上也是科学、严谨的,我们同意把这套教材作为推荐使用的教材。同时,我们也希望社会各界将对这套教材的意见和建议及时反馈给我们,以便使它不断完善。

谢谢大家。

北京市高等教育自学考试委员会办公室

1998年6月12日

音 序

北京十月二十日

# 前 言

随着计算机科学的迅速发展,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。正是基于这样的原因,我们根据国务院发布的《高等教育自学考试暂行条例》精神,以及北京市高等教育自学考试委员会关于高等教育文凭考试“数据结构”课程考试大纲,编写了这本《数据结构》。

本书较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有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。为此,本书强调了数据结构本身的技术及其在程序设计中的应用。

全书共分十一章。第一章阐述了数据结构的一些基本概念;第二章至第六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;第七章与第八章讨论非线性结构,重点讨论树和二叉树、图的基本概念及其应用;第九章至第十一章重点讨论文件的基本概念,其中主要是各种数据文件的组织方法及有关的操作。

在本书的编写过程中,我们本着着重基础、注意应用的原则,每一章除了必要的例题之外,还配有适量的习题与上机题供读者练习。凡是阅读过本书并独立完成习题的读者,都能掌握本书所要求的基本概念、基本技术与基本方法。

数据结构是计算机专业一门实践性很强的专业基础课程。通过该课程的学习,应使学生能够运用数据结构提供的方法与技巧更好地进行算法设计和程序设计。所以,本书在讨论各种数据结构基本运算的同时,都给出了相应的算法。算法全部采用 SPARKS 语言描述,具有较好的可读性与可移植性。SPARKS 语言属于一种类 PASCAL 语言,它几乎包括了所有程序设计语言所共有的、基本的可执行语句,且不受具体程序设计语言在词法、语法以及语义上的严格限制,因而,用 SPARKS 语言描述的算法具有简洁、易懂、易学等特点,熟悉任何一种高级程序设计语言的读者,只要对书中的算法稍做修改或补充,便可以得到计算机所能接受的程序。

本书取材广泛、内容全面,作为高等教育自学考试计算机专业学生的教材,讲授时间为 50~70 学时,也可以根据具体情况做一些增减,有些内容只作为阅读参考,不作要求,如书中带 \* 号的章节。考虑到专科教育的针对性、实践性和应用性,本书也可以作为高等专科学校、职工大学、职业大学、夜大学以及函授大学等大专类计算机硬件、软件专业的教材或参考书;同时,本书还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。

本书是作者在高等学校从事多年教学实践的基础上,对原有讲义和教材进行多次修改并参考了兄弟院校同类教材编著完成的。因此,在内容上努力遵循深入浅出的原则,在基本概念的阐述方面力求通俗详尽,以便于读者自学。

由于数据结构本身是一门年轻的学科,内容还在不断变化与更新,加上作者水平有限以及时间仓促等原因,书中的错误在所难免,恳切希望读者批评指正。

作 者

1998 年 6 月于北京



# 目 录

( 02 )	.....	( 1 )
( 02 )	.....	( 1 )
( 03 )	.....	( 4 )
( 06 )	.....	( 5 )
( 07 )	.....	( 5 )
<b>第一章 绪论</b>	.....	<b>( 1 )</b>
( 10)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 )
( 15)1.3	算法.....	( 5 )
( 15 )	1.3.1 算法及其性质 .....	( 5 )
( 25 )	1.3.2 基本算法 .....	( 6 )
( 25 )	1.3.3 算法的描述 .....	( 7 )
( 35)1.4	算法分析.....	( 9 )
( 35 )	1.4.1 时间复杂度 .....	( 10 )
( 35 )	1.4.2 空间复杂度 .....	( 12 )
( 85 )	1.4.3 其他方面 .....	( 12 )
( *1.5	算法设计的基本步骤.....	( 12 )
( 85)习题	.....	( 13 )
<b>第二章 线性表</b>	.....	<b>( 16 )</b>
( 082)1	线性表的定义及其基本操作.....	( 16 )
( 18 )	2.1.1 线性表的定义 .....	( 16 )
( 18 )	2.1.2 线性表的基本操作 .....	( 17 )
( 182)2	线性表的顺序存储结构.....	( 18 )
( 282)3	线性链表及其操作.....	( 22 )
( 28 )	2.3.1 线性链表的构造 .....	( 23 )
( 08 )	2.3.2 线性链表的基本算法 .....	( 25 )
( 102)4	循环链表及其操作.....	( 38 )
( 202)5	双向链表及其操作.....	( 42 )
( 30 )	2.5.1 双向链表的构造 .....	( 42 )
( 30 )	2.5.2 双向链表的插入与删除算法 .....	( 43 )
( *2.6	链表的应用举例.....	( 45 )
( 80 )	2.6.1 链式存储结构下的一元多项式加法 .....	( 45 )
( 80 )	2.6.2 打印文本文件的最后 n 行 .....	( 48 )
( 00)习题	.....	( 50 )
<b>第三章 数组</b>	.....	<b>( 56 )</b>
( 103.1	数组的概念.....	( 56 )
( 101 )	3.1.1 一维数组 .....	( 56 )
( 101 )	3.1.2 多维数组 .....	( 56 )
( 203.2	数组的存储结构.....	( 57 )

3.3 矩阵的压缩存储	( 59 )
3.3.1 对称矩阵的压缩存储	( 59 )
3.3.2 对角矩阵的压缩存储	( 60 )
3.4 稀疏矩阵的三元组表表示	( 60 )
3.4.1 稀疏矩阵的三元组表存储方法	( 60 )
* 3.4.2 稀疏矩阵的转置算法	( 61 )
* 3.4.3 稀疏矩阵相乘算法	( 64 )
* 3.5 稀疏矩阵的十字链表表示	( 66 )
3.6 数组的应用举例	( 71 )
3.6.1 一元多项式的数组表示	( 71 )
3.6.2 n阶魔方	( 72 )
习题	( 73 )
<b>第四章 堆栈和队列</b>	<b>( 77 )</b>
4.1 堆栈的概念及其操作	( 77 )
4.1.1 堆栈的定义	( 77 )
4.1.2 堆栈的基本操作	( 78 )
4.2 堆栈的顺序存储结构	( 78 )
4.2.1 顺序堆栈的构造	( 78 )
4.2.2 顺序堆栈的基本算法	( 79 )
* 4.2.3 多个堆栈共享连续空间问题	( 80 )
4.3 堆栈的链式存储结构	( 83 )
4.3.1 链接堆栈的构造	( 83 )
4.3.2 链接堆栈的基本算法	( 84 )
4.4 堆栈的应用举例	( 85 )
4.4.1 进制转换	( 85 )
4.4.2 堆栈在递归中的应用	( 86 )
4.4.3 表达式的计算	( 91 )
4.4.4 又一个趣味游戏——迷宫	( 95 )
4.5 队列的概念及其操作	( 97 )
4.5.1 队列的定义	( 97 )
4.5.2 队列的有关操作	( 98 )
4.6 队列的顺序存储结构	( 98 )
4.6.1 顺序队列的构造	( 98 )
4.6.2 顺序队列的基本算法	( 100 )
4.6.3 循环队列	( 101 )
4.7 队列的链式存储结构	( 103 )
4.7.1 链接队列的构造	( 103 )
4.7.2 链接队列的基本算法	( 104 )
习题	( 106 )

第五章 广义表 .....	( 110 )
5.1 广义表的概念 .....	( 110 )
5.2 广义表的存储结构 .....	( 111 )
* 5.3 多元多项式的表示 .....	( 113 )
习题 .....	( 114 )
第六章 串 .....	( 116 )
6.1 串的基本概念 .....	( 116 )
6.1.1 串的定义 .....	( 116 )
6.1.2 串的几个概念 .....	( 117 )
6.2 串的基本操作 .....	( 117 )
6.3 串的存储结构 .....	( 118 )
6.3.1 串的顺序存储结构 .....	( 119 )
6.3.2 串的链式存储结构 .....	( 120 )
6.4 串的几个操作 .....	( 121 )
习题 .....	( 126 )
第七章 树与二叉树 .....	( 127 )
7.1 树的基本概念 .....	( 127 )
7.1.1 树的定义 .....	( 127 )
7.1.2 树的逻辑表示方法 .....	( 129 )
7.1.3 基本术语 .....	( 130 )
7.1.4 树的性质 .....	( 131 )
7.1.5 树的基本操作 .....	( 132 )
* 7.2 树的存储结构 .....	( 132 )
7.2.1 多重链表表示 .....	( 132 )
7.2.2 三重链表表示 .....	( 134 )
7.3 二叉树 .....	( 135 )
7.3.1 二叉树的定义 .....	( 135 )
7.3.2 二叉树的基本操作 .....	( 135 )
7.3.3 两种特殊形态的二叉树 .....	( 136 )
7.3.4 二叉树的性质 .....	( 137 )
* 7.3.5 二叉树与树、树林之间的转换 .....	( 138 )
7.4 二叉树的存储结构 .....	( 140 )
7.4.1 二叉树的顺序存储结构 .....	( 140 )
7.4.2 二叉树的链式存储结构 .....	( 142 )
7.5 树的遍历 .....	( 146 )
7.5.1 二叉树的遍历 .....	( 146 )
* 7.5.2 树和树林的遍历 .....	( 153 )
7.5.3 由遍历序列恢复二叉树 .....	( 154 )
7.6 线索二叉树 .....	( 156 )

( 011 )	7.6.1 线索二叉树的构造	( 156 )
( 011 )	7.6.2 线索二叉树的利用	( 158 )
( 111 )	* 7.6.3 二叉树的线索化算法	( 160 )
( 111 )	* 7.6.4 线索二叉树的更新	( 160 )
( 111 )	7.7 二叉排序树	( 161 )
( 011 )	7.7.1 二叉排序树的定义	( 162 )
( 011 )	7.7.2 二叉排序树的建立	( 162 )
( 011 )	* 7.7.3 在二叉排序树中删除结点	( 164 )
( 111 )	7.7.4 二叉排序树的查找	( 166 )
( * )	7.8 平衡二叉树	( 169 )
( 111 )	7.9 哈夫曼树及其应用	( 175 )
( 011 )	7.9.1 哈夫曼树的概念	( 175 )
( 031 )	* 7.9.2 哈夫曼编码	( 176 )
( 10 )	习题	( 180 )
	<b>第八章 图</b>	( 185 )
( 10 )	8.1 图的基本概念	( 185 )
( 121 )	8.1.1 图的定义和基本术语	( 185 )
( 121 )	8.1.2 图的基本操作	( 188 )
( 02 )	8.2 图的存储方法	( 189 )
( 061 )	8.2.1 邻接矩阵存储方法	( 189 )
( 161 )	8.2.2 邻接表存储方法	( 191 )
( 121 )	* 8.2.3 有向图的十字链表存储方法	( 194 )
( 132 )	* 8.2.4 无向图的多重邻接表存储方法	( 195 )
( 12 )	8.3 图的遍历	( 196 )
( 161 )	8.3.1 深度优先搜索	( 196 )
( 161 )	8.3.2 广度优先搜索	( 198 )
( 12 )	8.4 最小生成树	( 200 )
( 121 )	8.4.1 普里姆算法	( 200 )
( 061 )	8.4.2 克鲁斯卡尔算法	( 203 )
( 12 )	8.5 最短路径问题	( 204 )
( 12 )	8.6 AOV 网与拓扑排序	( 208 )
( 04 )	8.6.1 AOV 网	( 208 )
( 041 )	8.6.2 拓扑排序	( 209 )
( 121 )	8.6.3 拓扑排序算法	( 210 )
( 04 )	8.7 AOE 网与关键路径	( 215 )
( 041 )	8.7.1 AOE 网	( 215 )
( 121 )	8.7.2 关键路径	( 216 )
( 121 )	8.7.3 关键路径的确定	( 216 )
( 02 )	习题	( 220 )

<b>第九章 文件及查找</b> .....	( 224 )
<b>9.1 文件概述</b> .....	( 224 )
9.1.1 基本术语 .....	( 224 )
9.1.2 文件的存储介质 .....	( 225 )
9.1.3 文件的基本操作 .....	( 227 )
<b>9.2 顺序文件</b> .....	( 228 )
9.2.1 连续顺序文件及其查找 .....	( 228 )
9.2.2 链接顺序文件及其查找 .....	( 232 )
<b>9.3 索引文件</b> .....	( 232 )
9.3.1 稠密索引文件 .....	( 233 )
9.3.2 非稠密索引文件 .....	( 234 )
* 9.3.3 多级索引文件 .....	( 235 )
<b>9.4 B-树和 B+ 树</b> .....	( 236 )
9.4.1 B-树的基本概念 .....	( 236 )
* 9.4.2 B-树的基本操作 .....	( 237 )
9.4.3 B+ 树的基本概念 .....	( 242 )
* 9.4.4 B+ 树的基本操作 .....	( 243 )
<b>9.5 散列(Hash)文件</b> .....	( 244 )
9.5.1 概述 .....	( 244 )
9.5.2 散列函数的几种构造方法 .....	( 245 )
9.5.3 处理冲突的方法 .....	( 248 )
9.5.4 散列文件的操作 .....	( 250 )
* 9.5.5 散列法的平均查找长度 .....	( 253 )
习题 .....	( 253 )
<b>第十章 内排序</b> .....	( 259 )
<b>10.1 概述</b> .....	( 259 )
10.1.1 排序的基本概念 .....	( 259 )
10.1.2 排序的分类 .....	( 260 )
<b>10.2 插入排序</b> .....	( 261 )
<b>10.3 选择排序</b> .....	( 263 )
<b>10.4 泡排序</b> .....	( 265 )
<b>10.5 谢尔排序</b> .....	( 266 )
<b>10.6 快速排序</b> .....	( 268 )
<b>10.7 堆积排序</b> .....	( 271 )
10.7.1 堆积的定义 .....	( 271 )
10.7.2 堆积排序算法 .....	( 271 )
* <b>10.8 二路归并排序</b> .....	( 275 )
10.8.1 归并子算法 .....	( 276 )
10.8.2 一趟归并扫描子算法 .....	( 277 )

10.8.3	二路归并排序算法	(277)
* 10.9	基数排序	(278)
10.10	各种内排序方法的比较	(282)
10.10.1	稳定性比较	(282)
10.10.2	复杂性比较	(282)
习题		(283)
* 第十一章	外排序	(287)
11.1	概述	(287)
11.2	磁带排序	(288)
11.2.1	多路平衡归并排序法	(288)
11.2.2	多步归并排序	(290)
11.3	初始归并段的合理分布与产生	(291)
11.3.1	初始归并段的合理分布	(291)
11.3.2	一种产生初始归并段的方法—置换选择排序	(292)
11.4	磁盘排序	(294)
习题		(297)
上机实践题		(298)
附录	北京市高等教育学历文凭考试“数据结构”课程考试大纲	(300)
主要参考文献		(306)

# 第一章 绪 论

被人们称之为第二次工业革命标志的电子计算机自 20 世纪 40 年代问世以来,发展神速,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估计。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的问题,更多、更广泛的应用是在非数值计算的各个领域,例如企业管理、工业过程控制、经济系统工程、管理信息系统等等。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、图像、声音、信号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,必须深入研究这些数据自身的特性以及它们之间的相互联系。这也正是数据结构这门新兴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,它们之间有着一定的结构关系。如何表示这些结构关系,如何在计算机中存储数据和信息,采用什么样的方法和技巧去加工处理这些数据,这些都是数据结构这门课程要努力解决的问题。

究竟什么是数据结构呢?在回答这个问题以前,先来复习几个并不陌生的名词术语。

**数据(data)** 描述客观事物的数字、字符以及一切能够输入到计算机中,并能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”。数据的含义十分广泛,在不同场合有着不同的含义。例如,在数值计算的问题中,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,计算机处理的对象是一些符号串;而在某些控制过程问题中,可能又是某种信号。在很多场合,人们对数据和信息的引用没有加以区分。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。

**数据元素(data element)** 数据的基本单位,即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。如数学中的一个数列,我们可以称之为数据,其中的一个一个的元素称为数据元素。有时候,一个数据元素可以由若干个子数据项组成(可见,数据项是数据不可分割的最小单位)。例如,数据文件中每一个数据记录就是一个数据元素,而每个数据记录又由若干个描述客体某一方面特征的数据项组成。

**数据对象(data object)** 具有相同性质的数据元素的集合,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 $\{1, 2, 3, \dots\}$ ,而由 26 个大写英文字母字符组成的数据对象则是集合 $\{ 'A', 'B', 'C', \dots, 'Z' \}$ 。

数据结构的概念目前还没有一个被一致公认的定义,不同的人在使用这个概念时所表达的意思有所不同。尽管如此,我们还是从几个不同角度给数据结构下一个定义。

在客观世界中,任何事物及活动都不会孤立地存在,都在一定意义上相互影响,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,我们称这种联系为结构。显然,我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,数据结构就是具有结构的数据元素的集合。我们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

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

其中, $D$ 是数据元素的有限集合, $R$ 是 $D$ 上关系的集合。在上述定义中的“关系”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在计算机中实现对它的操作,因此还需要研究数据在计算机存储器中的表示。于是,与数据的逻辑结构相对应的还有数据的物理结构,即数据结构在计算机中的表示(或者称映像)。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。

根据数据元素之间具有的关系的不同,我们可以将数据的逻辑结构主要分为集合结构、线性结构与非线性结构几大类。其中,在集合结构中,数据元素仅存在“同属于一个集合”的关系。在线性结构中,数据元素之间的逻辑关系是“一对一”的关系,即除了第一个数据元素和最后那个数据元素外,其他每个数据元素有且仅有一个直接前驱元素,有且仅有一个直接后继元素。对于非线性结构,我们又将其分成层次结构和网状结构两种。层次结构又称为树型结构,在这种逻辑结构中,数据元素之间的逻辑关系一般不是“一对一”的关系,而是“一对多”或者“多对一”的关系,即每个元素若有直接前驱元素,只能有一个,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“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(其实,此时这种前后关系已经没有意义)。网状结构也称图型结构,简称图。

一种逻辑结构通过映像便可以得到它的存储结构,由于映像的方法不同,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,相应的存储结构称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等;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,任何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。

图 1.1 是一个反映某个班级 30 名学生情况的表格,若用计算机来管理这个表格,它可以对应着一个数据文件。该文件有 30 个记录,每一个记录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。

姓名	性别	民族	年龄	其他
刘晓光	男	汉	15	.....
王 敏	女	汉	15	.....
马广生	男	回	16	.....
⋮	⋮	⋮	⋮	.....
张玉华	女	汉	20	.....

图 1.1 学生情况表



对于这样一个数据文件,一个记录就是一个数据元素,它由若干个反映学生某一方面情况的数据项组成。我们称紧挨着一个数据元素的前面那一个数据元素为该数据元素的直接前驱元素,紧挨着一个数据元素的后面那一个数据元素为该数据元素的直接后继元素。这个文件中除了第一个数据元素与最后那个数据元素外,其余每个数据元素有且仅有一个直接前驱元素,有且仅有一个直接后继元素,数据元素之间存在着一对一的逻辑关系。也就是说,该数据文件的逻辑结构为线性结构,我们也称该文件为一个线性表。对于这个逻辑结构,在计算机内可以采用多种存储结构,除了索引存储结构和散列存储结构外,最经常采用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 1.2 给出了这个存储结构。图中  $l_i (i=1,2,\dots,30)$  分别代表数据元素的存储地址。顺序存储结构的优点是简单,易理解,并且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,并且存储空间要事先准备,另外,对于一些操作的时间效率低也是这种存储结构的主要缺陷之一。

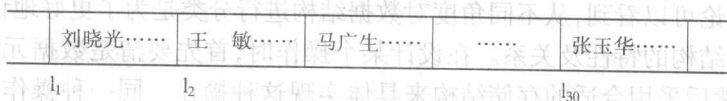


图 1.2 一个顺序存储结构的例子

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)存储单元依次存放这 30 个数据元素的信息,我们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。每个链结点不仅用来存放一个数据元素的数据信息,还要存放一个地址,该地址给出了这个数据元素在逻辑关系中的直接后继元素所在链结点的地址,我们称该地址为指针。也就是说,数据元素之间的逻辑关系通过指针间接地反映。由于要求的存储空间连续与否无所谓,因此,逻辑上相邻的数据元素在物理地址上不一定相邻。链式存储结构也称链表结构,简称链表。图 1.3 给出了图 1.1 所示的数据文件的链式存储结构的映像,图中的一个箭头表示一个指针。

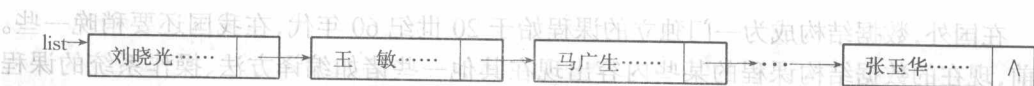


图 1.3 一个链式存储结构的例子

这种存储结构的优点是存储空间不必事先准备,需要时可以临时申请,这样不会造成存储空间的浪费。读者将来会看到,像插入和删除这样的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。但由于在这种存储结构中,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,仅从这一点来说,链式存储结构比顺序存储结构的开销要大。索引结构是利用数据元素的索引关系来确定数据元素的存储位置,这种结构由数据元素本身的数据信息以及索引表两个部分组成,而散列结构由事先准备好的散列函数关系与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