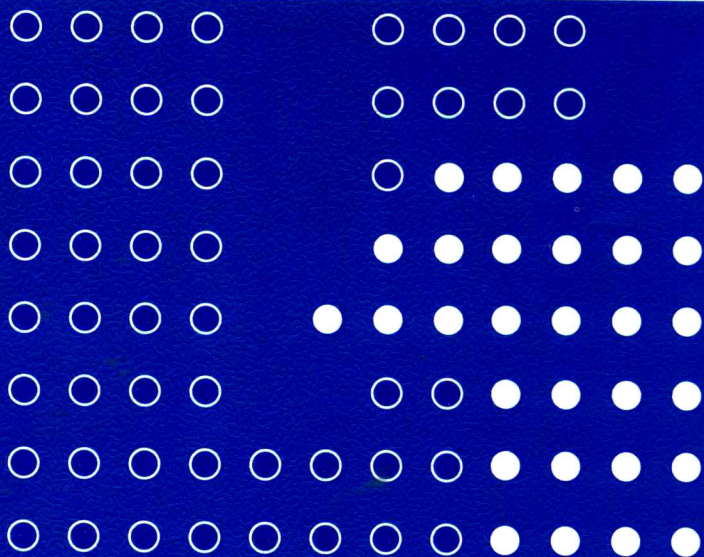




普通高等教育“十一五”国家级规划教材 计算机系列教材

数据结构与算法



熊岳山 祝恩 编著



清华大学出版社

内容简介

计算机系列教材

熊岳山 祝恩 编著

数据结构与算法

清华大学出版社

北京

内 容 简 介

数据结构与算法是计算机专业的重要基础课,是该专业的核心课程之一,是一门集技术性、理论性和实践性于一体的课程。本书重点介绍抽象数据类型、基本数据结构、C语言数据结构描述、数据结构的应用、算法设计与分析以及算法性能评价等内容,重点是让读者理解数据抽象与编程实现的关系,提高用计算机解决实际问题的能力。

本书内容包括基本数据类型、抽象数据类型、顺序表、链表、串、树和二叉树、图、递归与分治算法、贪心算法、分支限界和动态规划等内容。

本书结构合理,内容丰富,算法描述清晰,用C语言编写的算法代码都已调试通过,便于自学,可作为高等院校计算机专业、军事院校的基础合训专业和其他相关专业的教材和参考书,也可供从事计算机软件开发的科技工作者参考。

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

数据结构与算法/熊岳山,祝恩编著. —北京:清华大学出版社,2013.2
(计算机系列教材)

ISBN 978-7-302-30750-1

I. ①数… II. ①熊… ②祝… III. ①数据结构—高等学校—教材 ②算法分析—高等学校—教材 IV. ①TP311.12

中国版本图书馆CIP数据核字(2012)第284538号

责任编辑:白立军 徐跃进

封面设计:常雪影

责任校对:梁毅

责任印制:李红英

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦A座 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

课件下载: <http://www.tup.com.cn>, 010-62795954

印刷者:北京富博印刷有限公司

装订者:北京市密云县京文制本装订厂

经 销:全国新华书店

开 本:185mm×260mm 印 张:16.75

字 数:399千字

版 次:2013年2月第1版

印 次:2013年2月第1次印刷

印 数:1~3000

定 价:29.00元

产品编号:048134-01

《数据结构与算法》 前言

数据结构与算法是计算机科学与技术一级学科相关专业的重要基础课程之一,是软件开发和维护的基础。计算机的数据处理能力是计算机解决各种实际问题的关键。现实世界中的实际问题经过抽象,得出反映实际事物本质的数据表示后,才有可能被计算机处理。从实际问题抽象出数学模型,得出它的数据表示后,如何用计算机所能接受的形式来描述这些数据(包括数据本身与数据之间的关系)?如何将这些数据以及它们之间的关系存储在计算机中?如何用有效的方法去处理这些数据?如何在构建的数据结构上设计高效的算法?这些问题皆是数据结构与算法研究的主要问题。

本书是在深入研究国内外同类教材的基础上,结合多年“数据结构”、“算法设计与分析”课程教学经验编写而成的,书中将重点围绕抽象数据类型的C语言实现进行介绍。

第1章为数据结构概述,主要介绍数据结构概念,内容包括逻辑结构、存储方法、算法复杂度分析、基本数据类型、抽象数据类型与结构描述。第2章介绍向量、栈和队列及其应用,内容有向量、栈和队列的逻辑结构,抽象数据类型向量、栈和队列的描述;Josephus问题求解、栈与后缀表达式求值、栈与递归、递归效率分析、队列与离散事件模拟等应用实例。第3章介绍链表及其应用,内容有动态存储、单链表、循环表、双链表、栈和队列的链接存储。第4章介绍串,内容有串的定义、串的存储以及串的模式匹配算法(BMP算法)。第5章介绍各种排序方法,内容包括排序的基本概念,被排序文件的存储表示,折半插入排序、Shell排序、起泡排序、快速排序、归并排序和外排序等各种排序方法,各种排序方法的时空效率,算法的实现细节和算法的时空效率等。第6章介绍线性表的查找,内容包括有关查找的概念,顺序查找、折半查找、分块查找和散列查找。第7章介绍树和二叉树,内容包括树(树林)和二叉树的概念、树(树林)和二叉树的遍历、抽象数据类型 BinaryTree 与 BinaryTree 结构、二叉树的遍历算法。第8章介绍树状结构的应用,内容包括二叉排序树、平衡的二叉排序树、B-树和 B⁺-树、键树和 2-3 树、Huffman 最优树、堆排序、判定树、等价类和并查集、红黑树。第9章介绍图结构,内容包括图的基本概念、图的存储表示、Graph 结构的构造与实现、图的遍历、最小代价生成树、单源最短路径问题、每一对顶点间的最短路径问题、有向无回路图。第10章为算法设计与分析,内容包括递归与分治、回溯法、分支限界法、贪心法和动态规划法等。

本书是作为计算机专业本科生、军事院校的基础合训专业和其他相关专业的“数据结构与算法”课程教材编写的,也可供从事计算机软件开发和计算机应用的工程与科技人员

参考。具备了C语言基础的读者便可学习本书。

书中不加注*的章节可用60学时的时间讲授,全部内容的讲授可在70~80学时内完成。此外,为配合课堂教学,便于学生理解和掌握所学知识,提高程序设计编程能力,应另外配有20~30小时的上机时间。

本书的出版得到清华大学出版社,国防科技大学计算机学院、计算机系、603教研室的大力支持,在此深表谢意。特别感谢陈怀义教授和姚丹霖教授在《数据结构——C++描述》一书中的辛勤工作与许多富有创新的思想,感谢殷建平、肖晓强、刘越等老师为本书提出的宝贵意见,正是因为这些同志的热情帮助,才使得本书能顺利出版。由于时间仓促,加之作者水平有限,书中错误在所难免,敬请广大读者和专家批评指正。

作者

2012年9月

FOREWORD

第 1 章 数据结构概述 /1

1.1 基本概念 /1

1.1.1 数据、数据元素、数据对象 /1

1.1.2 数据结构 /2

1.2 数据结构的分类 /3

1.3 数据类型 /5

1.3.1 基本类型、组合类型 /5

1.3.2 抽象数据类型 /5

1.4 算法和算法分析 /8

1.4.1 算法概念 /8

1.4.2 算法分析 /9

习题 /11

第 2 章 向量、栈和队列 /13

2.1 线性表 /13

2.1.1 线性表的抽象数据类型 /13

2.1.2 线性表的结构表示 /15

2.2 向量 /18

2.2.1 向量的抽象数据类型 /18

2.2.2 向量的插入和删除 /20

2.2.3 向量的应用 /23

2.3 栈 /26

2.3.1 栈的抽象数据类型及其实现 /26

2.3.2 栈的应用 /29

2.4 递归效率分析 /36

2.4.1 递归方程求解 /36

2.4.2 生成函数求解递归方程 /37

2.4.3 特征方程求解递归方程 /38

2.4.4 递归树方法 /39

2.5 队列 /40

目录 《数据结构与算法》

参考。具备了C语言基础的同学使用学习本书。
书中不加注释的章节可用60学时的时间讲授,全部内容的讲授可在70~80学时内完成。本书与教材紧密结合,便于学生巩固课堂所学理论知识,提高程序设计编程能力,应另加2~3学时~30分钟的上课时间。

本书的出版得到清华大学出版社、清华大学计算机系、清华大学计算机系、清华大学计算机系的大力支持,在此深表谢意。特别感谢陈怀义教授和姚丹萍教授在《数据结构——C语言描述》一书中的辛勤工作与许多富有创新的思想,感谢殷建宇、白晓强、刘越等老师对本书提出的宝贵意见,正是因为这些同志的热情帮助,加之作者水平有限,书中错误在所难免。

作者
2012年9月

第2章	队列	1
2.1	队列的基本概念	1
2.2	队列的实现	1
2.3	队列的应用	1
2.4	队列的抽象数据类型及其实现	40
2.5	队列的应用——模拟银行活动	46
习题		54
第3章	链表	56
3.1	单链表	56
3.1.1	基本概念	56
3.1.2	单链表结点结构	57
3.1.3	单链表结构	59
3.1.4	栈的单链表实现	70
3.1.5	队列的单链表实现	71
3.1.6	单链表的应用举例	75
3.2	循环链表	80
3.3	双链表	82
习题		84
第4章	串	87
4.1	基本概念	87
4.2	串的存储	88
4.3	串结构和串的运算	89
4.4	模式匹配	91
4.4.1	朴素的模式匹配算法	91
4.4.2	KMP匹配算法	92
*4.4.3	BM匹配算法	95
习题		98
第5章	排序	99
5.1	基本概念	99
5.2	插入排序	100
5.2.1	直接插入排序	100
5.2.2	折半插入排序	102

5.2.3	Shell 排序	/104
5.3	选择排序	/105
5.3.1	直接选择排序	/105
5.3.2	树形选择排序	/107
5.4	交换排序	/108
5.4.1	起泡排序	/108
5.4.2	快速排序	/109
5.5	分配排序	/113
5.5.1	基本思想	/113
5.5.2	基数排序	/114
5.6	归并排序	/117
5.7	外部排序	/120
5.7.1	二路合并排序	/120
5.7.2	多路替代选择合并排序	/121
5.7.3	最佳合并排序	/122
	习题	/123
第 6 章 查找 /125		
6.1	基本概念	/125
6.2	顺序查找	/125
6.3	折半查找	/127
6.4	分块查找	/129
6.5	散列查找	/131
6.5.1	概述	/131
6.5.2	散列函数	/132
6.5.3	冲突的处理	/134
6.5.4	散列查找的效率	/137
	习题	/138
第 7 章 树和二叉树 /140		
7.1	树的概念	/140
7.2	二叉树	/141

7.2.1	二叉树的概念	/141
7.2.2	二叉树的性质	/141
7.2.3	二叉树的存储方式	/144
7.2.4	树(树林)与二叉树的相互转换	/146
7.3	树(树林)、二叉树的遍历	/147
7.3.1	树(树林)的遍历	/147
7.3.2	二叉树的遍历	/147
7.4	抽象数据类型 BinaryTree 以及 BinaryTree 结构	/148
7.4.1	抽象数据类型 BinaryTree	/148
7.4.2	一个完整的包含构建二叉树与遍历实现的例子	/150
7.5	二叉树的遍历算法	/151
7.5.1	非递归(使用栈)的遍历算法	/151
7.5.2	线索化二叉树的遍历	/153
	习题	/157
第 8 章 树状结构的应用 /159		
8.1	二叉排序树	/159
8.1.1	二叉排序树与 BinarySTree 结构	/159
8.1.2	二叉排序树的检索、插入、删除运算	/160
8.1.3	等概率查找对应的最佳二叉排序树	/164
8.2	平衡的二叉排序树	/166
8.2.1	平衡的二叉排序树	/166
* 8.2.2	平衡的二叉排序树的插入、删除	/167
* 8.2.3	AVL 树高度	/170
* 8.3	B-树、B ⁺ -树	/171
* 8.4	键树和 2-3 树	/175
8.4.1	键树	/175

1.1 基本概念	8.4.2 2-3 树 /176
1.1.1 数据、数据元素、数据对象	8.5 Huffman 最优树 /178
	8.5.1 Huffman 最优树 /178
	8.5.2 树编码 /181
	8.6 堆排序 /183
	*8.7 判定树 /189
	*8.8 等价类和并查集 /190
	8.8.1 等价类 /190
	8.8.2 并查集 /190
	*8.9 红黑树 /193
	习题 /197
	第9章 图 /199
	9.1 基本概念 /199
	9.2 图的存储表示 /201
	9.2.1 相邻矩阵表示图 /201
	9.2.2 图的邻接表表示 /202
	9.2.3 邻接多重表 /203
	9.3 基于邻接表表示的 Graph 结构 /205
	9.4 图的遍历 /206
	9.4.1 深度优先遍历 /206
	9.4.2 广度优先遍历 /208
	9.5 最小代价生成树 /209
	9.6 单源最短路径问题 /213
	9.7 每一对顶点间的最短路径问题 /216
	9.8 有向无回路图 /218
	9.8.1 DAG 图和 AOV、AOE 网 /218
	9.8.2 AOV 网的拓扑排序 /220
	9.8.3 AOE 网的关键路径 /222
	习题 /224

第 10 章	算法设计与分析	/226
10.1	递归与分治	/226
10.1.1	递归方法设计	/226
10.1.2	分治法	/227
10.2	回溯法	/229
10.3	分支限界法	/234
10.4	贪心算法	/241
10.5	动态规划法	/242
	习题	/245
	图目录	/247
	算法目录	/252
	关键字索引	/254
	参考文献	/257

第1章 数据结构概述

计算机的数据处理能力是计算机解决各种实际问题的基础,但是现实世界中的实际问题必须经过抽象,得出反映实际事物本质的数据表示后,才有可能被计算机处理。如何从实际问题抽象出它的数学模型,得出它的数据表示,这不属于本课程的主要内容;但如何用计算机所能接受的形式来描述这些数据(包括数据本身及数据与数据之间的关系),如何将这些数据以及它们之间的关系存储在计算机中,如何用有效的方法去处理这些数据,则是数据结构课程要研究的主要问题。

1.1 基本概念

1.1.1 数据、数据元素、数据对象

数据是客观事物的符号表示,是对现实世界的事物采用计算机能够识别、存储和处理的形式进行描述的符号的集合。计算机能处理多种形式的数。例如,科学计算软件处理的是数值数据;文字处理软件处理的是字符数据;多媒体软件处理的是图像、声音等多媒体数据。

数据元素是数据的基本单位。在计算机程序中,数据元素通常是作为一个整体处理的。一个数据元素又可以由若干个数据项组成。数据项包括两种:一种是初等项,是数据的不可分割的最小单位;一种是组合项,它又由若干个数据项组成。

例如,表 1-1 所示的学生情况表就是描述学生基本情况的数据。

表 1-1 学生情况表

学号	姓名	性别	年龄	籍贯	班别	成绩			
						数学	物理	化学	外语
6001	张三	女	19	北京	15	82	85	90	92
6002	李四	男	20	上海	15	90	92	91	95
6003	王五	男	18	湖南	15	93	95	91	94
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

每个学生的情况占一行,每一行则是一个数据元素。每一个数据元素由学号、姓名、性别、年龄、籍贯、班别、成绩等数据项组成。学号、姓名、性别、年龄、籍贯、班别为初等项。而成绩为组合项,它又分为数学成绩、物理成绩、化学成绩、外语成绩等初等项,而且学号这个数据项是一个关键字(或关键码),因为它的值能唯一标识一个数据元素。

数据对象是性质相同的数据元素的集合,是数据集合的一个子集。数据元素则是数

据对象集合中的数据成员。例如,学生情况表就是一个数据对象。整数集合、复数集合都是数据对象。

1.1.2 数据结构

在任何数据对象中,数据元素都不是孤立存在的,它们相互之间存在一种或多种特定的关系,这种关系称为结构。

数据的结构是指数据的组织形式,由数据对象及该对象中数据元素之间的关系组成。数据结构可以形式地描述为一个二元组:

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

其中: D 是数据对象,为数据元素的有限集;

R 是该数据对象中所有数据元素之间关系的有限集。

例 1-1 用符号 $\langle k_i, k_j \rangle$ 表示两数据元素 k_i, k_j 的先后次序关系,即 k_i 在 k_j 的前面,则下面两种二元组分别表示两个不同的逻辑结构。

结构 1 = (D, R)

$$D = \{k_0, k_1, \dots, k_{n-1}, k_n\}, \quad R = \{r\}$$

$$r = \{\langle k_0, k_1 \rangle, \langle k_1, k_2 \rangle, \dots, \langle k_{n-2},$$

$$k_{n-1} \rangle, \langle k_{n-1}, k_n \rangle\}.$$

结构 2 = (D, R)

$$D = \{k_0, k_1, k_2, k_3, k_4\}, \quad R = \{r\}$$

$$r = \{\langle k_0, k_1 \rangle, \langle k_0, k_2 \rangle, \langle k_1, k_3 \rangle,$$

$$\langle k_1, k_4 \rangle\}.$$

以上是数学意义的数据结构概念,是数据结构的逻辑描述,即逻辑结构。尽管在一般情况下,人们所说的数据结构即指数据的逻辑结构,但当这种数据结构要放在计算机中进行处理时,数据结构概念的含义就不仅如此了。涉及计算机的数据结构概念,至今尚未有一个公认的标准定义,但一般认为应包括以下三个方面:

(1) 数据元素及数据元素之间的逻辑关系,也称为数据的逻辑结构;

(2) 数据元素及数据元素之间的关系在计算机中的存储表示,也称为数据的存储结构或物理结构;

(3) 数据的运算,即对数据施加的操作。

数据的逻辑结构是从逻辑关系上描述数据,是根据问题所要实现的功能而建立的。数据的逻辑结构是面向问题的,是独立于计算机的。数据的存储结构是指数据在计算机中的物理表示方式,是根据问题所要求的响应速度、处理时间、存储空间和处理速度等建立的。数据的存储结构是依赖于计算机的。每种逻辑结构都有一个运算的集合,例如最常见的运算有检索、插入、排序等,这些运算在数据的逻辑结构上定义,只规定“做什么”,在数据的存储结构上考虑运算的具体实现,规定“如何做”。

例如表 1-1 就是一个数据结构,它由若干个数据元素组成,每个学生的情况数据就是一个数据元素。对任何一个数据元素,除第一个元素外,其他每个元素都有且仅有一个前驱,第一个元素没有前驱;除最后一个元素外,其他每个元素都有且仅有一个后继,最后一个元素没有后继。这就是数据的一种逻辑结构。

要利用这张表对学生的情况进行计算机管理,首先要将这张表存入计算机。可以把

表中的这些数据元素顺序邻接地存储在一片连续的存储单元中,也可以用指针把各自存储的数据元素按表中顺序链接在一起,这种表在计算机中的存储方式就是数据的存储结构。学生情况的管理,必然涉及学生成绩的登记和查询、学生的插班和退学等,这些管理行为就是数据的运算,这里涉及数据元素的查找、插入和删除等操作。这些操作在数据的逻辑结构上定义,在数据的存储结构上实现。

1.2 数据结构的分类

对于一个数据结构而言,每一个数据元素可称为一个结点,数据结构中所包含的数据元素之间的关系就是结点之间的关系。如果两个结点 k, k' 之间存在的关系可用有序对 $\langle k, k' \rangle$ 表示,则称 k' 是 k 的后继, k 是 k' 的前驱, k 和 k' 互为相邻结点。如果 k 没有后继,则称 k 为终端结点。如果 k 没有前驱,则称 k 为开始结点。如果 k 既不是终端结点,也不是开始结点,则称 k 为中间结点。

数据的逻辑结构可分为两大类:一类是线性结构;一类是非线性结构。

线性结构中有且仅有一个开始结点和一个终端结点,并且所有的结点最多只有一个前驱和一个后继。线性表是典型的线性结构。学生情况表就是一个线性表。

非线性结构中的一个结点可能有多个前驱和后继。如果一个结点最多只有一个前驱,而可以有多个后继,这种结构就是树。树是最重要的非线性结构之一。如果对结点的前驱和后继的个数不作限制,这种结构就是图。图是最一般的非线性结构。数据的这几种逻辑结构可用图 1.1 表示。

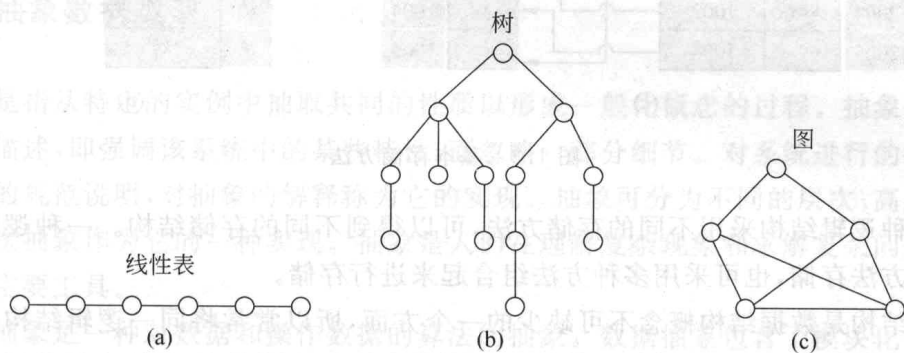


图 1.1 基本的逻辑结构

数据的存储结构取决于四种基本的存储方法:顺序存储、链接存储、索引存储、散列存储。顺序存储方法是把逻辑上相邻的结点存储在物理位置相邻的存储单元里。结点之间的逻辑关系用存储单元的邻接关系来体现。顺序存储主要用于线性结构,非线性结构也可以通过某种线性化的方法实现顺序存储。通常顺序存储是用程序语言的数组描述的。

链接存储方法对逻辑上相邻的结点不要求在存储的物理位置上亦相邻,结点之间的逻辑关系由附加的指针表示。非线性结构常用链接存储,线性结构也可以链接存储。通

常链接存储是用程序语言的指针描述的。

索引存储方法是在存储结点数据的同时,还建立附加的索引表。索引表的每一项称为索引项。一般情况下索引项由关键字(关键字是结点的一个字段或多个字段的组合,其值能唯一确定数据结构中的一个结点)和地址组成。一个索引项唯一对应于一个结点,其中的关键字是能唯一标识该结点的数据项,地址指示该结点的存储位置。

散列存储方法是根据结点的关键字计算出该结点的存储地址,然后按存储地址存放该关键字对应的数据元素。

这四种基本的存储方法形成四种不同的存储结构,如图 1.2 所示。

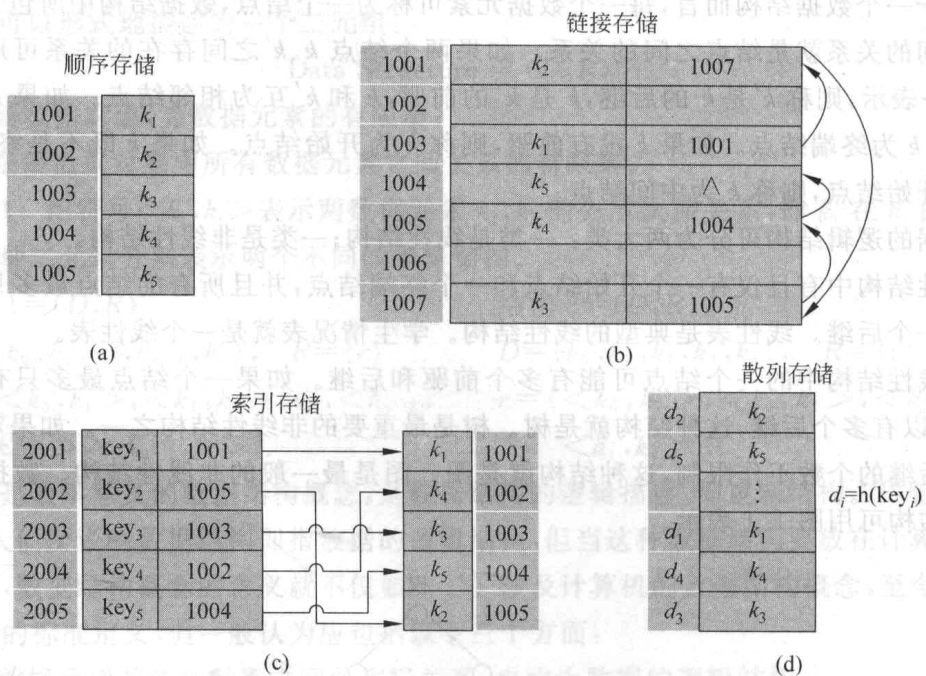


图 1.2 基本存储方法

同一种逻辑结构采用不同的存储方法,可以得到不同的存储结构。一种逻辑结构可采用一种方法存储,也可采用多种方法组合起来进行存储。

存储结构是数据结构概念不可缺少的一个方面,所以常常将同一逻辑结构的不同存储结构用不同的名称标识。例如,线性表的顺序存储称为顺序表,线性表的链接存储称为链表,线性表的散列存储称为散列表。

数据的运算也是数据结构概念不可缺少的一个方面。同一种逻辑结构采用同一种存储方式,如果定义的运算不同,也将以不同的名称标识。例如,若在线性表上插入、删除操作限制在表的一端进行,则称为栈;若插入限制在表的一端进行,删除限制在表的另一端进行,则称为队列。如果这种线性表顺序存储,则称为顺序栈或顺序队列;如果链接存储,则称为链式栈或链式队列。

在计算机环境下研究数据结构,应该将数据的逻辑结构、数据的存储结构和数据的运算看成一个整体,只有从这三个方面都了解清楚了,才能真正了解这个数据结构。

1.3 数据类型

1.3.1 基本类型、组合类型

在早先的高级程序语言中都有数据类型的概念。数据类型是一组性质相同的值的集合以及定义在这个集合上的一组操作的总称。

N. Wirth 曾指出：算法+数据结构=程序。在早先的高级程序语言中，数据结构是通过数据类型来描述的。数据类型用于刻画操作对象的特性。每个变量、常量和表达式都属于一个确定的数据类型，例如整型、实型、字符型等，这些数据类型规定了数据可能取值的范围以及允许进行的操作。例如，C 程序中的变量 k 定义为整型 int，则它可能取值的范围是 $\{0, +1, -1, +2, -2, \dots, \text{maxint}, \text{minint}\}$ ，其中 maxint 和 minint 是所用计算机上整型量所能表示的最大整数和最小整数。对它可施行的操作有算术一元运算+、-，算术二元运算+、-、*、/、%，关系运算<=、>=、==、!=，赋值运算=等。

在高级程序语言中，数据类型分为两种：一种是基本类型，如整型、实型、字符型等，其取值范围和允许的操作都是由系统预先规定的；另一种是组合类型，它由一些基本类型组合构造而成，如记录、数组、结构等。基本数据类型通常是由程序语言直接提供的，而组合类型则由用户借助程序语言提供的描述机制自己定义。这些数据类型都可以看成是程序设计语言已实现了的数据结构。

1.3.2 抽象数据类型

抽象是指从特定的实例中抽取共同的性质以形成一般化概念的过程。抽象是对某系统的简化描述，即强调该系统中的某些特性，而忽略一部分细节。对系统进行的抽象描述称为对它的规范说明，对抽象的解释称为它的实现。抽象可分为不同的层次，高层次抽象将其低层次抽象作为它的一种实现。抽象是人们在理解复杂现象和求解复杂问题中处理复杂性的主要工具。

数据抽象是一种对数据和操作数据的算法的抽象。数据抽象包含了模块化和信息隐蔽两种抽象。模块化是将一个复杂的系统分解成若干个模块，每个模块与系统某个特定模块有关的信息保持在该模块内。一个模块是对整个系统结构的某一部分的自包含和完整的描述。模块化的优点是便于修改或维护。系统发现问题后，容易定位问题出在哪个模块上。信息隐蔽是将一个模块的细节部分对用户隐藏起来，用户只能通过一个受保护的接口来访问某个模块，而不能直接访问一个模块的内部细节。这个接口一般由一些操作组成，这些操作定义了一个模块的行为。这样，错误的影响被限制在一个模块内，增强了系统的可靠性与可维护性。

数据类型已经体现了数据的抽象。例如，在计算机中使用二进制定点数和浮点数实现数据的存储和运算，在汇编语言中程序员可直接使用它们的自然表示，如 15.5、

1.35E10、10 等,不必考虑它们实现的细节,这是二进制数据的抽象。在高级语言中,出现了整型、实型、双精度实型等数据类型,给出了更高一级的数据抽象。面向对象程序语言出现后,可以用抽象数据类型定义更高层次的数据抽象,如各种表、树、图,甚至窗口、管理器等。这种数据抽象的层次为软件设计者提供了有利的手段,使设计者可从抽象的概念出发,从整体上进行考虑,然后自顶向下,逐步展开,最后得到所需的结果。

抽象数据类型 (Abstract Data Type, ADT) 是指抽象数据的组织和与之相关的操作。抽象数据类型通常是由用户定义,用以表示应用问题的数据模型,它可以看做是数据的逻辑结构及在逻辑结构上定义的操作。ADT 可用如下形式描述:

```
ADT ADT_Name
{
    Data:                                /* 数据说明 */
        数据元素之间逻辑关系的描述
    Operations:                            /* 操作说明 */
        Operation1
            Input:                          对输入数据的说明
            Preconditions:                  执行操作前系统应满足的状态
            Process:                        对数据执行的操作
            Output:                          对返回数据的说明
            Postconditions:                 执行操作后系统的状态
        Operation2
        :
} /* ADT */
```

抽象数据类型体现了数据抽象,它将数据和操作封装在一起,使得用户程序只能通过 ADT 里定义的某些操作来访问其中的数据,从而实现了信息隐蔽。ADT 既独立于它的具体实现,又与具体的应用无关,这可使软件设计者把注意力集中在数据及其操作的理想模型的选择上。

下面给出一个矩形抽象数据类型的例子。矩形由长和宽决定,不同的长和宽构成不同的矩形。与矩形相关的主要操作,一个是计算面积,一个是计算周长。因此,数据长和宽及求面积和周长的操作就构成了矩形的抽象数据类型。

封装的矩形数据和操作的抽象数据类型如下:

```
ADT Rectangle
{
    Data:
        非负实数,给出矩形的长和宽
    Operations:
        InitRectangle
            Input:                          长方形的长和宽
            Preconditions:                    无
            Process:                          建立长方形
```