

高等学校21世纪教材

GAODENG XUEXIAO 21 SHIJI JIAOCAI

数据结构

——使用C++语言描述

◎ 陈慧南 主编



人民邮电出版社
POSTS & TELECOM PRESS

高等学校 21 世纪教材

数 据 结 构

——使用 C++ 语言描述

陈慧南 主编

人 民 邮 电 出 版 社

图书在版编目 (CIP) 数据

数据结构: 使用 C++语言描述 / 陈慧南主编. —北京: 人民邮电出版社, 2006.10
高等学校 21 世纪教材

ISBN 7-115-15122-9

I. 数... II. 陈... III. ①数据结构—高等学校—教材 ②C 语言—程序设计—高等学校—教材 IV. ①TP311.12 ②TP312

中国版本图书馆 CIP 数据核字 (2006) 第 091638 号

内 容 简 介

本书不仅系统介绍各种传统的数据结构以及搜索和内、外排序算法, 还引入了一些新数据结构, 如伸展树和跳表。本书在重视理论的同时, 强调应用性和实践性, 对上机实习赋予足够重视。每章包括程序设计题, 并将实习指导和实习题作为专门的章节编写。

全书条理清晰, 内容详实。书中算法都有完整的 C++程序。程序结构清晰, 构思精巧。所有程序都已在 VC++环境下编译通过并能正确运行。本书深入浅出, 配有大量的实例和图示, 并有丰富的习题和实习题, 适于自学。

本书可作为电气信息类、电子信息科学类以及计算机、管理信息系统、电子商务, 教育技术等其他相关专业学生数据结构课程的教材, 并可供其他计算机应用工程技术人员参考。

高等学校 21 世纪教材

数据结构——使用 C++语言描述

◆ 主 编 陈慧南

责任编辑 滑 玉

执行编辑 蒋 亮

◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号

邮编 100061 电子函件 315@ptpress.com.cn

网址 <http://www.ptpress.com.cn>

北京艺辉印刷有限公司印刷

新华书店总店北京发行所经销

◆ 开本: 787×1092 1/16

印张: 17.5

字数: 417 千字

2006 年 10 月第 1 版

印数: 1—3 000 册

2006 年 10 月北京第 1 次印刷

ISBN 7-115-15122-9/TP • 5620

定价: 23.00 元

读者服务热线: (010)67170985 印装质量热线: (010)67129223

前 言

本书是作为江苏省教育厅重点建设精品教材立项编写的。作者多年在南京邮电大学讲授《数据结构》课程，已编写《数据结构》、《数据结构与算法》以及《算法设计与分析》等教材。本书是在前者基础上，参考了近年来国内外多种数据结构和算法的优秀教材编写而成。

本书内容符合 CC2001 对数据结构知识领域的要求，可作为电气信息类、电子信息科学类，以及管理信息系统、电子商务、教育技术等其他相关专业学生数据结构课程的教材。也可供其他计算机应用工程技术人员参考。

近年来，《数据结构》课程的内容和讲授体系有了很大的变革。在介绍数据结构知识的同时，普遍重视软件设计原理和程序设计方面的训练。本书采用抽象数据类型的观点讨论数据结构，使用 C++ 语言描述。书中不仅系统介绍了各种传统的数据结构以及搜索和内、外排序算法，还引入了一些新数据结构，如伸展树和跳表。本书在重视理论的同时，对上机实习给予足够重视。每章包括程序设计题，并将实习指导和实习题作为专门的章节编写。

本书条理清晰、内容详实、深入浅出。书中对算法都做了详细的解释，尽可能做到可读易懂，并配有大量的实例和图示，有利于读者深入理解算法的设计思想。每章引言和结尾处的小结帮助读者掌握各章要点。丰富的习题和实习题便于练习和自测。本书适于自学。

本书共分 12 章和附录，由陈慧南主编。其中，第 1 章、第 5~9 章和第 11 章由陈慧南编写，第 2~4 章和第 10 章由陈春玲编写，第 12 章由朱立华和陈慧南共同完成，附录由朱立华编写。全书由陈慧南统稿、审校。

第 1 章是基础知识，讨论有关数据结构、算法、数据抽象和抽象数据类型的概念，给出数据结构和算法的描述方法，讨论算法分析的基本方法。

第 2~4 章介绍几种线性数据结构：线性表、堆栈和队列、数组和字符串。第 3 章还讨论递归和递归算法，以及编译程序实现函数嵌套调用的机制。

第 5 章讨论树结构。第 6~8 章讨论集合的表示和运算。其中，第 6 章为顺序搜索和二分搜索，第 7 章为二叉搜索树、二叉平衡树、B 树和伸展树，第 8 章为跳表和散列表。第 9 章讨论图。

第 10 章介绍多种内排序算法。第 11 章讨论外排序过程和算法。文件作为一种外存的数据结构，也在第 11 章介绍。

数据结构上机实习是课程教学不可或缺的重要环节，第 12 章和附录是专门为数据结构上机实习而编写的。第 12 章为实习指导和实习题。附录介绍调试技术和 VC++ 调试器。

本书包含“数据结构”课程 72 学时的教学内容。对于学时数少于 72 学时的授课计划，可根据实际学时加以剪裁。作者已在目录中对难度较大，或非基本的章节标上*号，供读者选取时参考。除*号部分内容外的基本部分适合 48~56 学时授课计划。

本书的编写得到了南京邮电大学和计算机学院领导的推荐和关心，在此表示衷心感谢。

书中若有不当之处，敬请读者批评指正。

作 者

2006 年 5 月 28 日于南京

目 录

第 1 章 基础知识	1
1.1 算法与数据结构	1
1.2 什么是数据结构	2
1.2.1 基本概念	2
1.2.2 数据的逻辑结构	3
1.2.3 数据的存储表示	4
1.2.4 数据结构的运算	5
1.3 数据抽象和抽象数据类型	6
1.3.1 抽象、数据抽象和过程抽象	6
1.3.2 封装与信息隐蔽	6
1.3.3 数据类型和抽象数据类型	7
1.3.4 数据结构与抽象数据类型	7
1.4 描述数据结构和算法	8
1.4.1 数据结构的规范	8
1.4.2 实现数据结构	9
1.5 算法分析的基本方法	10
1.5.1 算法及其性能标准	10
1.5.2 算法的时间复杂度	11
1.5.3 渐近时间复杂度	12
1.5.4 最坏、最好和平均情况时间复杂度	13
1.5.5 算法的空间复杂度	14
本章小结	14
习题	14
第 2 章 线性表	17
2.1 线性表 ADT	17
2.2 线性表的顺序表示	18
2.3 线性表的链接表示	23
2.3.1 单链表	23
2.3.2 带表头结点的单链表	29
2.3.3 单循环链表	30
2.3.4 双向链表	31
2.4 多项式的算术运算	32

2.4.1 项结点的 C++ 类	32
2.4.2 多项式的 C++ 类	35
2.4.3 多项式类的实现	35
本章小结	38
习题	38
第 3 章 堆栈和队列	40
3.1 堆栈	40
3.1.1 堆栈 ADT	40
3.1.2 堆栈的顺序表示	41
3.1.3 堆栈的链接表示	42
3.2 队列	43
3.2.1 队列 ADT	43
3.2.2 队列的顺序表示	44
3.2.3 队列的链接表示	46
3.3* 表达式计算	46
3.3.1 表达式	46
3.3.2 计算后缀表达式的值	47
3.3.3 中缀表达式转换为后缀表达式	50
3.4 递归	53
3.4.1 递归的概念	53
3.4.2 递归的实现	54
本章小结	55
习题	55
第 4 章 数组和字符串	57
4.1 数组	57
4.1.1 数组 ADT	57
4.1.2 数组的顺序表示	58
4.1.3 一维数组的 C++ 类	58
4.2 特殊矩阵	61
4.2.1 对称矩阵	61
4.2.2* 带状矩阵	61
4.3 稀疏矩阵	63
4.3.1 稀疏矩阵 ADT	63
4.3.2 稀疏矩阵的顺序表示	64
4.3.3 稀疏矩阵转置	65
4.4 字符串	67
4.4.1 字符串 ADT	68
4.4.2 字符串的存储表示	68
4.4.3 简单模式匹配算法	69

4.4.4* 模式匹配的 KMP 算法	71
本章小结	75
习题	75
第 5 章 树	77
5.1 树的基本概念	77
5.1.1 树的定义	77
5.1.2 基本术语	78
5.2 二叉树	79
5.2.1 二叉树的定义	80
5.2.2 二叉树的性质	80
5.2.3 二叉树 ADT	82
5.2.4 二叉树的存储表示	82
5.2.5 二叉树类	83
5.2.6 实现二叉树基本运算	84
5.3 二叉树的遍历	86
5.3.1 二叉树遍历算法	86
5.3.2 二叉树遍历的递归算法	88
5.3.3 二叉树遍历的应用实例	89
5.4* 二叉树遍历的非递归算法	90
5.4.1 遍历器类	90
5.4.2 中序遍历器类	91
5.5 树和森林	93
5.5.1 森林与二叉树的转换	93
5.5.2 树和森林的存储表示	95
5.5.3 树和森林的遍历	97
5.6 堆和优先权队列	98
5.6.1 堆	98
5.6.2 优先权队列 ADT	100
5.6.3 优先权队列类	101
5.6.4 实现优先权队列	102
5.7 哈夫曼树和哈夫曼编码	104
5.7.1 树的路径长度	104
5.7.2 哈夫曼树和哈夫曼算法	105
5.7.3 哈夫曼树类	106
5.7.4 构造哈夫曼树	106
5.7.5 哈夫曼编码	107
5.8* 并查集和等价关系	109
5.8.1 并查集 ADT	109
5.8.2 并查集的存储表示	109

5.8.3	并查集类	110
5.8.4	函数 Union 和 Find	111
5.8.5	改进的函数 Union 和 Find	111
5.8.6	按等价关系分组	112
本章小结		113
习题		113
第 6 章	集合与搜索	116
6.1	基本概念	116
6.1.1	集合与搜索	116
6.1.2	动态集 ADT	118
6.1.3	集合的表示	119
6.2	顺序搜索	119
6.2.1	无序表的顺序搜索	119
6.2.2	有序表的顺序搜索	120
6.2.3	平均搜索长度	121
6.3	二分搜索	121
6.3.1	二分搜索算法	121
6.3.2	对半搜索	122
6.3.3	二叉判定树	123
本章小结		125
习题		125
第 7 章	搜索树	126
7.1	二叉搜索树	126
7.1.1	二叉搜索树的定义	126
7.1.2	二叉搜索树的搜索	127
7.1.3	二叉搜索树的插入	128
7.1.4	二叉搜索树的删除	129
7.1.5	平均情况时间分析	131
7.2*	二叉平衡树	132
7.2.1	二叉平衡树的定义	132
7.2.2	二叉平衡树类	132
7.2.3	二叉平衡树的平衡旋转	133
7.2.4	二叉平衡树的插入	139
7.2.5	二叉平衡树的删除	141
7.2.6	二叉平衡树的高度	144
7.3	B-树	145
7.3.1	m 叉搜索树	145
7.3.2	B-树的定义	147
7.3.3	B-树的高度	147

7.3.4 B-树的搜索	148
7.3.5 B-树的插入	148
7.3.6 B-树的删除	150
7.4* 伸展树	152
本章小结	154
习题	155
第8章 跳表和散列表	156
8.1 字典	156
8.2* 跳表	156
8.2.1 什么是跳表	157
8.2.2 跳表类	159
8.2.3 跳表的搜索	160
8.2.4 跳表的插入	161
8.2.5 跳表的删除	162
8.3 散列表	163
8.3.1 散列技术	163
8.3.2 散列函数	164
8.3.3 拉链法	166
8.3.4 开地址法	167
8.3.5 线性探查法	167
8.3.6 其他开地址法	171
8.3.7 性能分析	172
本章小结	173
习题	173
第9章 图	174
9.1 图的基本概念	174
9.1.1 图的定义与术语	174
9.1.2 图 ADT	176
9.2 图的存储结构	178
9.2.1 图的矩阵表示法	178
9.2.2 图的邻接矩阵实现	179
9.2.3 图的邻接表表示法	182
9.2.4 图的邻接表实现	182
9.3 图的遍历	185
9.3.1 扩充的图类	185
9.3.2 深度优先遍历	186
9.3.3 宽度优先遍历	188
9.4 拓扑排序	189
9.4.1 用顶点代表活动的 AOV 网	189

9.4.2	什么是拓扑排序	191
9.4.3	拓扑排序算法	191
9.5*	关键路径	194
9.5.1	用边代表活动的 AOE 网	194
9.5.2	什么是关键路径	195
9.5.3	关键路径算法	197
9.6	最小代价生成树	198
9.6.1	基本概念	198
9.6.2	普里姆算法	198
9.6.3*	克鲁斯卡尔算法	200
9.7	单源最短路径	202
9.7.1	最短路径问题	202
9.7.2	迪杰斯特拉算法	203
9.7.3	选择数据结构	204
9.7.4	迪杰斯特拉算法	204
9.8	所有顶点之间的最短路径	207
9.8.1	选择数据结构	207
9.8.2	弗洛伊德算法	207
	本章小结	209
	习题	209
第 10 章	内排序	212
10.1	基本概念	212
10.2	简单排序算法	213
10.2.1	简单选择排序	213
10.2.2	直接插入排序	214
10.2.3	冒泡排序	215
10.3	快速排序	217
10.4	两路合并排序	219
10.5	堆排序	221
10.6*	基数排序	222
	本章小结	226
	习题	227
第 11 章*	文件和外排序	228
11.1	辅助存储器简介	228
11.1.1	主存储器和辅助存储器	228
11.1.2	磁盘存储器	228
11.2	文件	229
11.2.1	文件的基本概念	229
11.2.2	文件的组织方式	230

11.3 文件的索引结构	233
11.3.1 静态索引结构	233
11.3.2 动态索引结构	233
11.4 外排序	234
11.4.1 外排序的基本过程	235
11.4.2 初始游程的生成	235
11.4.3 多路合并	237
11.4.4 最佳合并树	239
本章小结	240
习题	241
第 12 章 实习指导和实习题	242
12.1 实习目的和要求	242
12.1.1 实习目的	242
12.1.2 实习要求	242
12.2 实习步骤	243
12.3 面向对象方法及其表示法	243
12.3.1 面向对象方法	243
12.3.2 表示法	245
12.4 实习报告和样例	247
12.4.1 实习报告	247
12.4.2 实习样题	247
12.4.3 实习报告样例	248
12.5 实习题	253
12.5.1 实习一	253
12.5.2 实习二	254
12.5.3 实习三	255
12.5.4 实习四	256
12.5.5 实习五	256
12.5.6 实习六	257
12.5.7 实习七	258
12.5.8 实习八	259
12.5.9 实习九	259
12.5.10 实习十	260
12.5.11 实习十一	261
附录 程序调试	262
附录 1 调试步骤	262
附录 2 VC++调试器	263
参考文献	267

第1章 基础知识

本章首先介绍什么是数据结构，算法和数据结构在计算机学科中的地位如何；然后介绍数据抽象和抽象数据类型概念，阐明其与数据结构的关系，给出数据结构和算法的描述方法；最后讨论算法分析的基本方法。

1.1 算法与数据结构

计算机由硬件和软件组成，硬件通过软件发挥效用。硬件是躯体，软件是灵魂，软件的核心是程序。学习程序设计需要掌握一门程序设计语言，它是学习计算机后续课程所必须的技能。但程序设计不等于编码，为了充分利用计算机资源，开发高效的程序，计算机人员还必须掌握计算机学科多方面知识，如数据的组织、算法的设计和分析、软件工程技术等。

随着计算机科学与技术的发展，计算机应用已远远超出了单纯进行科学计算的范围。从传统的应用领域，如工业控制、情报检索、企业管理、商务处理、图形图像、人工智能等诸多的数据处理领域，发展到电子政务、电子商务、办公自动化、企业资源管理、电子图书馆、远程教育、远程医疗等更广泛的领域。计算机技术已渗透到国民经济的各行各业和人们日常生活的方方面面。今天，信息技术作为现代技术的标志，已成为世界各国经济增长的主要动力。

现实世界各领域中的大量信息都必须转换成数据才能在计算机中存储、处理。数据是信息的载体，应用程序处理各种各样的数据。笼统地说，所谓数据，就是计算机加工处理的对象。数据一般分为两类：数值数据和非数值数据。数值数据是一些整数、实数或复数，主要用于工程计算、科学计算、商务处理等。非数值数据包括字符、文字、图形、图像、语音、表格等。这类数据的特点是量大，而且往往有着复杂的内在联系。如果单纯依靠改进程序设计技巧，已无法编制出高效可靠的程序，而必须对数据本身的结构加以研究。数据的组织和表示方法直接影响使用计算机求解问题的效率。算法设计通常建立在所处理数据的一定组织形式之上。在许多应用中，对于相同数据的同样处理要求，如果选择不同的数据结构，会有不同的处理效率：运算时间和存储空间。数据结构和算法两者是紧密结合的。

对计算机学科来说，数据结构与算法的概念是至关重要的，它们是计算机学科的基础之一，更是软件技术的基础。数据结构与算法之间有着本质的联系。当谈论一种算法时，

自然要涉及算法所处理的数据问题；而讨论数据的组织或结构，离开了对处理此类数据的运算及其算法的研究也是无意义的。有人概括过一个公式，即

程序=算法+数据结构

在计算学科教学计划 1991 中，数据结构和算法是计算机学科的 9 个研究领域之一。计算学科教学计划 2001 调整成 14 个领域，数据结构和算法的基本内容主要涵盖在算法与复杂性和程序设计基础 2 个领域中。CC2001 强调了算法和程序设计，它建议使用 90 个核心学时讲授这方面知识，占整个核心学时数的 32.2%。

数据结构主要是为研究和解决如何使用计算机组织和处理非数值问题而产生的理论、技术和方法。它已成为计算机学科研究的基本课题之一。本书讨论数据的结构技术，以及在数据结构上的算法。

1.2 什么是数据结构

1.2.1 基本概念

前文提到，数据是计算机加工处理的对象。一个数据可以由成分数据构成的具有某种结构的数据。在这里，我们称组成数据的成分数据为数据元素。通常，数据元素可以是简单类型的，如整数、实数、字符等，也可以是结构类型，如记录。如果把每个学生的记录看作一个数据元素，它包括学号、姓名、性别等数据项——数据项是不可再分割的，一个班学生的记录组成了表 1-1 所示的学生情况表。

表 1-1 学生情况表

学 号	姓 名	性 别	其 他 信 息
B02040101	王小红	女	...
B02040102	林悦	女	...
B02040103	陈菁菁	女	...
B02040104	张可可	男	...
⋮	⋮	⋮	⋮

那么，什么是数据结构？数据结构是计算机科学与技术领域中广泛使用的术语，但究竟什么是数据结构，在计算机科学界至今没有标准的定义。

Sartaj Sahni 在他的《数据结构、算法与应用》一书中称“数据结构是数据对象，以及存在于该对象的实例和组成实例的数据元素之间的各种联系。这些联系可以通过定义相关的函数来给出”。该书中，Sartaj Sahni 将数据对象定义为“一个数据对象是实例或值的集合。”

Clifford A. Shaffer 在《数据结构与算法分析》一书中的定义是：“数据结构是 ADT^①的物理实现。”

① ADT (abstract data type, 抽象数据类型)

Robert L.Kruse 在他的《数据结构与程序设计》一书中, 将一个数据结构的设计过程分成抽象层、数据结构层和实现层。其中, 抽象层是指抽象数据类型层, 它讨论数据的逻辑结构及其运算, 数据结构层和实现层讨论一个数据结构的表示和在计算机内的存储细节以及运算的实现。

我们认为, 一个数据结构是由数据元素依据某种逻辑联系组织起来的, 对数据元素间逻辑关系的描述称为数据的逻辑结构; 数据必须在计算机内存储, 数据的存储结构是数据结构的实现形式, 是其在计算机内的表示; 此外讨论一个数据结构必须同时讨论在该类数据上执行的运算才有意义。

本书中, 我们将一个数据结构的讨论分为两个层次, 即抽象层和实现层。抽象层讨论数据的逻辑结构及其运算定义, 实现层讨论数据的存储表示以及运算的算法实现。

1.2.2 数据的逻辑结构

从概念上讲, 一个数据结构是由数据元素依据某种逻辑联系组织起来的。对数据元素间逻辑关系的描述被称为数据的逻辑结构, 它可以用一个二元组表示, 即

$$DS = (D, R)$$

其中, D 是数据元素的有限集合, R 是 D 中元素序偶的集合。

例如, $DS = \{D, R\}$, $D = \{a, b, c, d\}$, $R = \{\langle a, b \rangle, \langle b, c \rangle, \langle c, d \rangle\}$, 其中, 序偶 $\langle a, b \rangle$ 表示元素 a 和 b 之间的关系, 我们称 a 是 b 的直接前驱, b 是 a 的直接后继。在不引起混淆的情况下, 常简称直接前驱为“前驱”, 直接后继为“后继”。数据结构 DS 是一个线性数据结构。

我们用小圆圈表示数据元素, 用带箭头的线表示元素间的关系。两个不同元素的序偶称为边, 则 DS 可由图 1-1 表示。

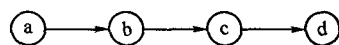


图 1-1 DS 示意图

根据数据结构中数据元素之间关系的不同特征, 可划分为以下四种基本逻辑结构。

(1) 集合结构

集合结构中, 元素间的次序是随意的。元素之间除了“属于同一个集合”的联系之外没有其他关系。由于集合结构的元素间没有固有的关系, 因此往往需要借助其他结构才能在计算机中实际表示此结构。

(2) 线性结构

线性结构是数据元素的有序序列, 其中, 第一个元素没有前驱只有后继, 最后一个元素只有前驱没有后继, 其余元素有且仅有一个前驱和一个后继。

(3) 树形结构

树中, 除一个特殊元素称为根, 它没有前驱只有后继外, 其余元素都有且仅有一个前驱, 但后继的数目不限。对于非根元素, 都存在着一从根到该元素的路径。树是层次数据结构。

(4) 图状结构

图是最一般的数据结构, 图中每个元素的前驱和后继的数目都不限。

这四种基本结构还可进一步分成两类: 线性结构和非线性结构。把除了线性结构以外的几种结构, 即树、图和集合都归入非线性结构一类。图 1-2 所示为四种基本结构的示意图。

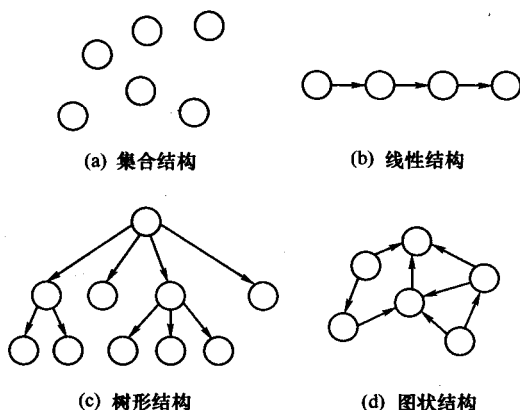


图 1-2 四种基本的结构关系

1.2.3 数据的存储表示

数据的逻辑结构是面向应用问题的，是从用户角度看到的数据结构。数据必须在计算机内存储，数据的存储结构是数据在计算机内的表示形式，是逻辑数据的存储映象，它是面向计算机的。

我们知道，计算机内存是由有限个存储单元组成的一个连续存储空间，这些存储单元是字节编址或者是字编址的。从存储器角度看，内存中是一堆二进制数据，它们可以被机器指令解释为指令、整数、字符、布尔数等，也可以被数据结构的算法解释为具有某种结构的数据。

对一个数据结构，找到一种有效的存储表示方法，使它适于计算机表示是十分重要的。顺序和链接是两种最基本的存储表示方法。

顺序（或称连续）表示方法需要一块连续的存储空间，并把逻辑上相关的数据元素依次存储在该连续的存储区中。例如，由 4 个元素组成的线性数据结构 (a_0, a_1, a_2, a_3) ，存储在某个连续的存储区内，设存储区的起始地址是 102，假定每个元素占 2 个存储单元，则其顺序存储表示如图 1-3(a)所示。

在顺序表示下，可以容易地用一个数学公式来确定每个元素的存储地址。对于图 1-3(a)的顺序存储结构，可使用式 (1-1) 计算元素 a_k 的存储位置 $\text{Loc}(a_k)$ ，即

$$\text{Loc}(a_k) = 102 + 2 \times k \quad (1-1)$$

顺序表示方法并不仅限于存储线性数据结构。对于非线性数据结构，如树结构，有时也可采用顺序存储的方法表示之。这将在以后讨论。

另一种基本的存储表示方法是链接存储表示。在链接存储表示下，为了在计算机内存储一个元素，除了需要存放该元素本身的信息外，还需要存放与该元素相关的其他元素的地址信息。这两部分信息组成存放一个数据元素的结点。图 1-3(b)所示为线性结构 (a_0, a_1, a_2, a_3) 的链接存储表示。其中，每个结点的存储块分成两部分，一部分存放元素自身，另一部分包含该元素逻辑上的后继结点的存储地址。这种关于相关结点的地址信息被称为链。图 1-3(b)中，符号“^”表示空链（即不代表任何具体结点的存储地址）。注意，一个结点的存储地址通常是指存放该结点的存储块的起始存储单元地址。

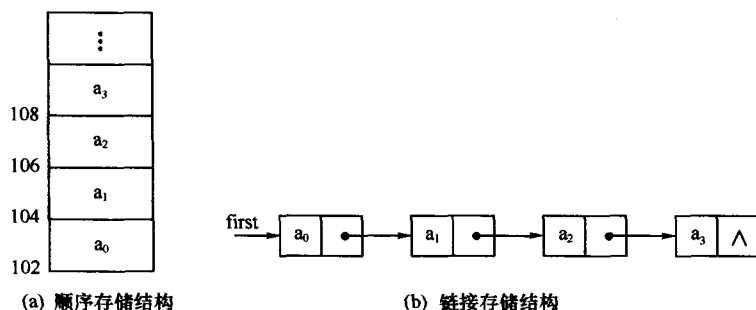


图 1-3 两种基本的存储表示方法

以后的讨论中,在不会引起混淆的场合,将不明确区分结点和元素这两个术语。但必要时,我们将包括地址信息在内的存储块整体称为结点,而将其中的元素信息部分称为该结点的元素。

除顺序和链接表示外,还可以有其他存储数据的方法,如索引方法和散列方法。关于这两种方法,请参看以后相关章节。

这些基本的存储表示方法,以及它们的组合,可衍生出各种数据的存储结构。

1.2.4 数据结构的运算

如果说数据的逻辑结构描述了数据的静态特性,那么在数据的逻辑结构上定义的一组运算给出了数据被使用的方式,即数据的动态特性。使用数据结构上定义的运算,用户可对数据结构的实例或组成实例的数据元素实施相应的操作。运算的结果可使数据改变状态。

请注意,在下面的讨论中,我们一般不明确区分数据结构和它的实例。例如,我们称创建一个数据结构,很显然是指创建某种数据结构的一个实例。

数据结构的最常见的运算有:

- (1) 创建运算——创建一个数据结构;
- (2) 清除运算——删除数据结构中的全部元素;
- (3) 插入运算——在数据结构中插入一个新元素;
- (4) 删除运算——将数据结构中的某个指定元素删除;
- (5) 搜索运算——在数据结构中搜索满足一定条件的元素;
- (6) 更新运算——修改数据结构中某个指定元素的值;
- (7) 访问运算——访问数据结构中某个元素;
- (8) 遍历运算——按照某种次序,系统地访问数据结构的各元素,使得每个元素恰好被访问一次。

如果一个数据结构一旦创建,其结构不发生改变,则称为静态数据结构,否则称为动态数据结构。

堆栈是一种线性数据结构,可以向栈中加入元素,但只允许访问和删除最后入栈的元素。例 1.1 给出了在栈上定义的若干运算。

例 1.1 栈数据结构

- (1) 向栈中加入一个元素 (Push 运算);