

重点大学计算机专业系列教材

数据结构 (C++描述)

熊岳山 编著

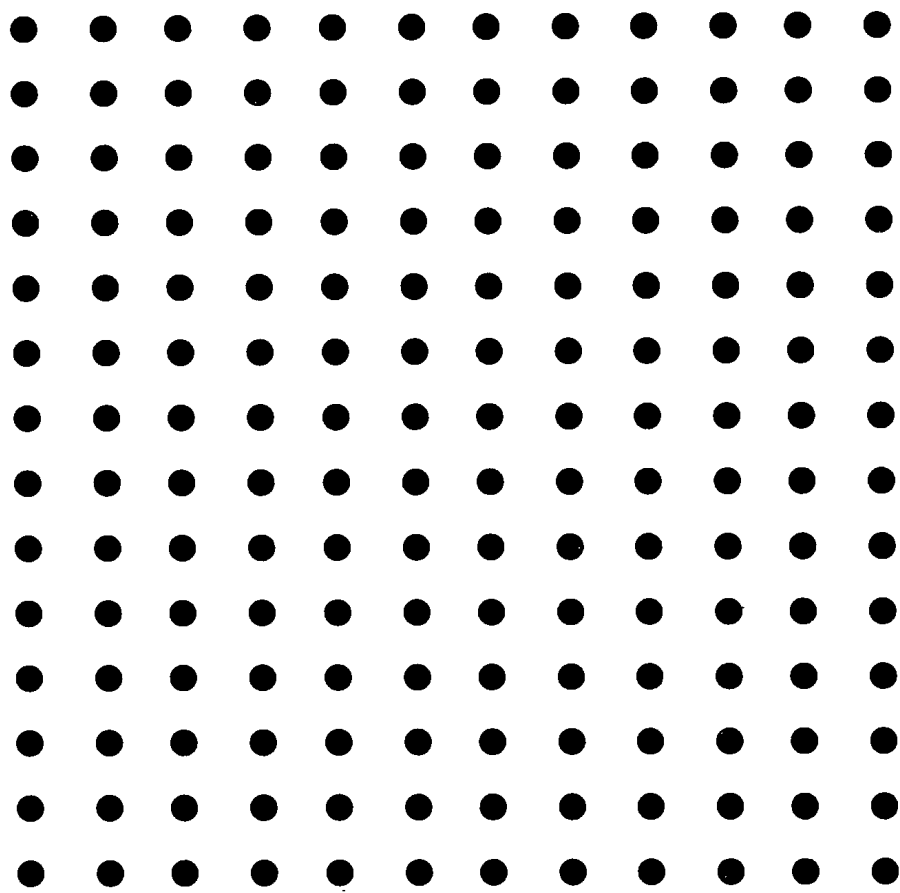


清华大学出版社

重点大学计算机专业系列教材

数据结构 (C++描述)

熊岳山 编著



清华大学出版社
北京

内 容 简 介

数据结构是计算机科学与技术、网络工程、软件工程、信息安全等专业的重要基础课,是这些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、算法性能评价、C++语言描述数据结构、数据结构的应用等内容,进一步使读者理解数据抽象与面向对象编程实现的关系,提高使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、算法效率分析、顺序表、链表、树和二叉树、图、多维数组等内容。本书结构合理,内容丰富,算法理论分析详细,数据结构的算法描述丰富,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科学与技术、网络工程、软件工程、信息安全等专业、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(C++描述)/熊岳山编著. —北京:清华大学出版社,2012.1

(重点大学计算机专业系列教材)

ISBN 978-7-302-27001-0

I. ①数… II. ①熊… III. ①数据结构 ②C语言—程序设计 IV. ①TP311.12
②TP312

中国版本图书馆 CIP 数据核字(2011)第 200262 号

责任编辑:魏江江 李玮琪

责任校对:梁 毅

责任印制:何 芊

出版发行:清华大学出版社

地 址:北京清华大学学研大厦 A 座

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

邮 编:100084

社 总 机:010-62770175

邮 购:010-62786544

投稿与读者服务:010-62795954, jsjic@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

印 刷 者:北京鑫海金澳胶印有限公司

经 销:全国新华书店

开 本:185×260 印 张:15.25 字 数:378千字

版 次:2012年1月第1版 印 次:2012年1月第1次印刷

印 数:1~3000

定 价:25.00元

产品编号:040148-01

出版说明

随着国家信息化步伐的加快和高等教育规模的扩大,社会对计算机专业人才的需求不仅体现在数量的增加上,而且体现在质量要求的提高上,培养具有研究和实践能力的高层次的计算机专业人才已成为许多重点大学计算机专业教育的主要目标。目前,我国共有 16 个国家重点学科、20 个博士点一级学科、28 个博士点二级学科集中在教育部部属重点大学,这些高校在计算机教学和科研方面具有一定优势,并且大多以国际著名大学计算机教育为参照系,具有系统完善的教学课程体系、教学实验体系、教学质量保证体系和人才培养评估体系等综合体系,形成了培养一流人才的教学和科研环境。

重点大学计算机学科的教学与科研氛围是培养一流计算机人才的基础,其中专业教材的使用和建设则是这种氛围的重要组成部分,一批具有学科方向特色优势的计算机专业教材作为各重点大学的重点建设项目成果得到肯定。为了展示和发扬各重点大学在计算机专业教育上的优势,特别是专业教材建设上的优势,同时配合各重点大学的计算机学科建设和专业课程教学需要,在教育部相关教学指导委员会专家的建议和各重点大学的大力支持下,清华大学出版社规划并出版本系列教材。本系列教材的建设旨在“汇聚学科精英、引领学科建设、培育专业英才”,同时以教材示范各重点大学的优秀教学理念、教学方法、教学手段和教学内容等。

本系列教材在规划过程中体现了如下一些基本组织原则和特点。

(1) 面向学科发展的前沿,适应当前社会对计算机专业高级人才的培养需求。教材内容以基本理论为基础,反映基本理论和原理的综合应用,重视实践和应用环节。

(2) 反映教学需要,促进教学发展。教材要能适应多样化的教学需要,正确把握教学内容和课程体系的改革方向。在选择教材内容和编写体系时注意体现素质教育、创新能力与实践能力的培养,为学生知识、能力、素质协调发展创造条件。

(3) 实施精品战略,突出重点,保证质量。规划教材建设的重点依然是专业基础课和专业主干课;特别注意选择并安排了一部分原来基础比较好的优秀教材或讲义修订再版,逐步形成精品教材;提倡并鼓励编写体现重点大学

计算机专业教学内容和课程体系改革成果的教材。

(4) 主张一纲多本,合理配套。专业基础课和专业主干课教材要配套,同一门课程可以有多本具有不同内容特点的教材。处理好教材统一性与多样化的关系;基本教材与辅助教材以及教学参考书的关系;文字教材与软件教材的关系,实现教材系列资源配套。

(5) 依靠专家,择优落实。在制订教材规划时要依靠各课程专家在调查研究本课程教材建设现状的基础上提出规划选题。在落实主编人选时,要引入竞争机制,通过申报、评审确定主编。书稿完成后要认真实行审稿程序,确保出书质量。

繁荣教材出版事业,提高教材质量的关键是教师。建立一支高水平的以老带新的教材编写队伍才能保证教材的编写质量,希望有志于教材建设的教师能够加入到我们的编写队伍中来。

教材编委会

前言

数据结构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。现实生活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,是数据结构研究的主要问题。

面向对象的程序设计(OOP)是当今流行的一种程序设计方法。它是人们以一种与传统的面向过程的程序设计方法完全不同的思维方式来认识软件设计的各个方面,并用于指导软件生产的整个过程。面向对象的程序设计有以下几个优点:首先,由于对象包含属性和行为,因此它支持模块化的程序设计,而模块化程序设计又便于程序的开发和维护;其次,在面向对象程序设计过程中可以充分地利用已有类库,在此基础上所做的修改与扩充不会损害原有类库,因此应用程序的新增代码明显减少,应用程序的可靠性得以提高。C++是一种支持面向对象的程序设计的语言。C++是C语言的超集,它不但实现了面向对象的程序设计的要求,而且还继承了C语言的所有优点,是开发系统软件和应用软件的有力工具,这就是C++备受青睐的主要原因。

用面向对象的程序设计语言C++描述数据结构,与用面向过程的程序设计语言(C或PASCAL)描述数据结构相比,有很大的变化。用面向对象的C++语言的类来描述抽象数据类型,可利用封装、继承和多态等特性。

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”课程教学经验编写而成的,选材充分考虑了教育部高等学校计算机科学与技术教学指导委员会对“数据结构”课程的内容要求,算法理论分析详细,数据结构的算法描述丰富。全书共分9章,第1章为数据结构概述,主要介绍数据结构概念:逻辑结构、存储方法、算法复杂度分析、基本数据类型、抽象数据类型与类。第2章介绍数组、栈和队列及其应用,内容有向量、栈和队列的逻辑结构,抽象数据类型数组、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、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章介绍各种排序方法,内容有排序的基本概念,被排序文件的存储表示,折半插入排序、Shell排序、冒泡排序、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5章介绍了线性表的查找,内容有有关查找的概念,顺序查找、折半查找、分块查找、字符串查找的KMP算法和散列查找。第6章介绍树和二叉树,内容有树(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 与类 BinaryTree、二叉树的遍历算法。第7章介绍树形结构的应用,内容有二叉排序树、平衡的二叉排序树、B-树和 B⁺-树、2-3 树、Huffman 最优树、堆排序和判定树、红黑树、并查集和键树。第8章介绍图结构,内容有图的基本概念、图的存储表示、Graph 类的构造与实现、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、有向无回路图的拓扑排序和关键路径。第9章为多维数组,内容有多维数组的存储方法、稀疏矩阵的存储和抽象数据类型——稀疏矩阵。

本书是作为计算机专业本科生“数据结构”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++ 的读者可学习本教材。书中没有标“*”的章节可用 50 学时的时间讲授。全部内容可用 60 学时完成。

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。

本书的出版得到国防科技大学计算机学院、计算机系、603 教研室以及清华大学出版社的大力支持,在此深表谢意。特别感谢陈怀义教授、姚丹霖教授在合作撰写数据结构教材中的辛勤工作。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。

编 者

2011 年 7 月

目录

第 1 章 数据结构概述	1
1.1 基本概念	1
1.1.1 数据、数据元素和数据对象	1
1.1.2 数据结构.....	2
1.2 数据结构的分类	3
1.3 抽象数据类型	5
1.3.1 两种软件设计方法.....	5
1.3.2 数据类型.....	5
1.3.3 抽象数据类型.....	5
1.4 算法和算法分析	8
1.4.1 算法的概念.....	8
1.4.2 算法分析.....	9
习题	11
第 2 章 顺序表	13
2.1 线性表.....	13
2.1.1 线性表的抽象数据类型表示	13
2.1.2 线性表的类表示	15
2.2 数组.....	18
2.2.1 数组的抽象数据类型	18
2.2.2 数组元素的插入和删除	20
2.2.3 数组的应用	22
2.3 栈.....	24
2.3.1 栈的抽象数据类型及其实现	24
2.3.2 栈的应用	26
2.4 队列.....	33
2.4.1 队列的抽象数据类型及其实现	33

2.4.2 优先级队列	36
2.4.3 队列的应用——离散事件驱动模拟	38
习题	45

第3章 链表

3.1 动态数据结构	47
3.2 单链表	49
3.2.1 基本概念	49
3.2.2 单链表结点类	50
3.2.3 单链表类	51
3.2.4 栈的单链表实现	61
3.2.5 链式队列	63
3.2.6 链表的应用举例	65
3.3 循环链表	69
3.4 双链表	71
习题	74

第4章 排序

4.1 基本概念	77
4.2 插入排序	78
4.2.1 直接插入排序	78
4.2.2 折半插入排序	79
4.2.3 Shell 排序	81
4.3 选择排序	83
4.3.1 直接选择排序	83
4.3.2 树形选择排序	84
4.4 交换排序	85
4.4.1 冒泡排序	85
4.4.2 快速排序	86
4.5 分配排序	90
4.5.1 基本思想	90
4.5.2 基数排序	91
4.6 归并排序	93
*4.7 外部排序	96
4.7.1 二路合并排序	96
4.7.2 多路替代选择合并排序	97
4.7.3 最佳合并排序	98
*4.8 排序算法的时间下界	99
习题	100

第 5 章 查找	102
5.1 基本概念	102
5.2 顺序查找	103
5.3 折半查找	104
5.4 分块查找	106
5.5 字符串的模式匹配	107
5.5.1 朴素的模式匹配算法	107
5.5.2 KMP 匹配算法	108
5.5.3 算法效率分析	111
5.6 散列查找	112
5.6.1 概述	112
5.6.2 散列函数	113
5.6.3 冲突的处理	115
5.6.4 散列查找的效率	118
习题	119
第 6 章 树和二叉树	121
6.1 树的概念	121
6.2 二叉树	122
6.2.1 二叉树的概念	122
6.2.2 二叉树的性质	122
6.2.3 二叉树的存储方式	125
6.2.4 树(树林)与二叉树的相互转换	126
6.3 树(树林)、二叉树的遍历	127
6.3.1 树(树林)的遍历	127
6.3.2 二叉树的遍历	128
6.4 抽象数据类型 BinaryTree 以及类 BinaryTree	128
6.4.1 抽象数据类型 BinaryTree	128
6.4.2 一个完整包含类 BinaryTreeNode 和类 BinaryTree 实现的例子	129
6.5 二叉树的遍历算法	132
6.5.1 非递归(使用栈)的遍历算法	133
6.5.2 线索化二叉树的遍历	134
习题	138
第 7 章 树形结构的应用	140
7.1 二叉排序树	140
7.1.1 二叉排序树与类 BinarySTree	140
7.1.2 二叉排序树的检索、插入和删除运算	141

7.1.3	等概率查找对应的最佳二叉排序树	144
7.2	平衡的二叉排序树	147
7.2.1	平衡的二叉排序树与类 AVLTree	147
7.2.2	平衡二叉排序树的插入和删除	148
7.2.3	类 AVLTree 与 AVL 树高度	154
7.3	B-树、B ⁺ -树	155
*7.4	2-3 树	159
*7.5	红黑树	161
7.6	Huffman 最优二叉树	164
7.6.1	Huffman 最优二叉树概述	164
7.6.2	树编码	167
7.7	堆排序	169
*7.8	判定树	175
*7.9	等价类和并查集	176
7.9.1	等价类	176
7.9.2	并查集	177
*7.10	键树	179
	习题	181
第8章	图	183
8.1	基本概念	183
8.2	图的存储表示	185
8.2.1	相邻矩阵表示图	185
8.2.2	图的邻接表表示	186
8.2.3	邻接多重表	187
8.3	构造 Graph 类	189
8.3.1	基于邻接表表示的 Graph 类	189
8.3.2	Graph 类的实现	191
8.4	图的遍历	194
8.4.1	深度优先遍历	194
8.4.2	广度优先遍历	196
8.5	最小代价生成树	196
8.6	单源最短路径问题——Dijkstra 算法	200
8.7	每一对顶点间的最短路径问题	203
8.8	有向无回路图	204
8.8.1	DAG 图和 AOV、AOE 网	204
8.8.2	AOV 网的拓扑排序	206
8.8.3	AOE 网的关键路径	208
	习题	210

第9章 多维数组	212
9.1 多维数组的顺序存储	212
9.2 特殊矩阵的顺序存储	213
9.3 稀疏矩阵的存储	214
9.4 抽象数据类型稀疏矩阵与 class SparseMatrix	217
习题	222
附录 Nodelib. h	223
参考文献	228

数据结构概述

第 1 章

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、数据元素和数据对象

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。数据。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来处理的。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;另一种是组合项,它由若干个数据项组成。

例 1.1 表 1.1 所示的学生情况表就是描述学生基本情况的数据。

表中,每个学生的情况占一行,每一行则是一个数据元素(或称为结点)。每一个数据元素由学号、姓名、年龄、籍贯、性别、专业、成绩等数据项组成。姓名、年龄、籍贯、性别、专业为初等项,而成绩为组合项,它又分为高等数学、线性代数、大学英语成绩等初等项。因为学号的值能唯一标识一个数据元素,学号称为关键字(或称为主关键字),非关键字的数据项又称为属性项(或次关键字)。

表 1.1 学生情况表

学号	姓名	年龄	籍贯	性别	专业	成 绩		
						高等数学	线性代数	大学英语
20100601	张三	18	湖南	男	信息安全	90	86	82
20100602	李四	19	北京	女	软件工程	89	84	88
20100603	王五	18	上海	男	信息安全	88	87	90
...	...	...	...	...	...	...	...	...

数据对象是性质相同数据元素的集合,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。整数集合和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 1.2 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。

$$\text{结构 1} = (D, R)$$

$$\text{结构 2} = (D, R)$$

$$D = \{k_0, k_1, \dots, k_{n-1}, k_n\}, R = \{r\} \quad D = \{k_0, k_1, k_2, k_3, k_4\}, R = \{r\}$$

$$r = \{\langle k_0, k_1 \rangle, \langle k_1, k_2 \rangle, \dots, \quad r = \{\langle k_0, k_1 \rangle, \langle k_0, k_2 \rangle,$$

$$\langle k_{n-2}, k_{n-1} \rangle, \langle k_{n-1}, k_n \rangle\} \quad \langle k_1, k_3 \rangle, \langle k_1, k_4 \rangle\}$$

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。尽管在一般情况下所说的数据结构即数据的逻辑结构,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连接在一起,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 为内部结点。

数据的逻辑结构可分为两大类:一类是线性结构;另一类是非线性结构。

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。

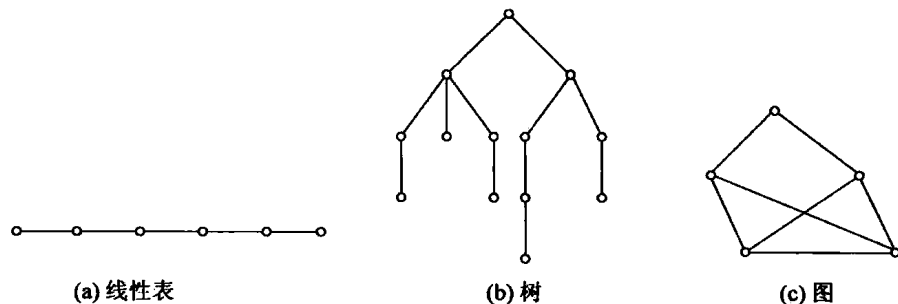


图 1.1 基本的逻辑结构

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、索引存储和散列存储。

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来实现顺序存储。通常顺序存储是用程序语言的数组来描述的。

链接存储方法对逻辑上相邻的结点不要求在存储空间的物理位置上亦相邻,结点之间的逻辑关系由附加的指针来表示。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来描述的。

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。

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。

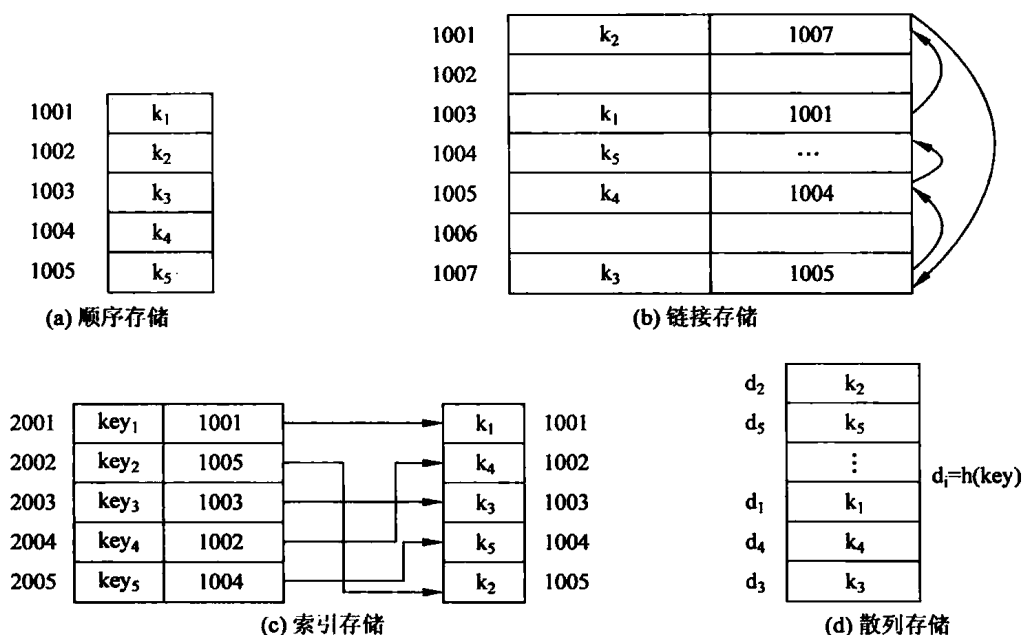


图 1.2 基本的逻辑结构

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,只有对这三个方面都清楚了,才能真正了解这个数据结构。

1.3 抽象数据类型

1.3.1 两种软件设计方法

在软件设计中,为了降低问题的复杂度,一种基本的策略就是“分解”。把复杂的问题分解成若干个较为简单的子问题,然后分别对这些子问题进行处理。传统的软件设计技术采用的是功能分解。它把软件看成一个处理过程,将软件分解为若干个表示过程步骤的模块,然后由编程语言的构件(子程序和函数)来实现。

面向对象的软件设计方法将软件看成由数据对象组成的集合,这些对象是应用问题所涉及的物理实体的数据模型,它们之间的相互作用构成了一个软件系统。首先,软件被分解成若干个数据对象,用抽象数据类型加以描述;然后,通过面向对象的程序设计语言 C++ 的类加以表示和实现;最后,通过各个类的实例之间的消息连接,实现软件的功能。

因为数据对象自然地对应于应用问题中的实体,所以与传统的软件设计方法相比,面向对象的软件设计方法分解更为直观,分解的难度降低。用面向对象设计方法生成的软件,测试和调试效率高,易于修改,质量可靠,而且便于重用。

之所以面向对象的软件设计方法优于传统的软件设计方法,主要原因是面向对象的软件设计方法采用了抽象数据类型的描述方式,实现了数据抽象和信息隐蔽。

1.3.2 数据类型

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。

1.3.3 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