

丛书主编 刘平

高等院校信息技术应用型规划教材

数据结构

李莹 孙承福 主编
李文超 王艳 梁爽 时巍 副主编



清华大学出版社



高等院校信息技术应用型规划教材

数据结构

李莹 孙承福 主编
李文超 王艳 梁爽 时巍 副主编

清华大学出版社
北京

内 容 简 介

本书根据应用型人才的培养目标和“应用为本、学以致用”的办学理念,精选必需的内容,对常用的数据结构作了系统介绍,以通俗易懂的讲解配以图示,注重实际应用。全书共分 10 章,每一章的主要算法以调试通过的程序演示讲解,并附有小结与典型例题,有助于学生理解和应用。

本书既可作为应用型本科计算机及相关专业的专业教材,也可作为非计算机专业学生及计算机爱好者学习数据结构的入门书籍。

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/李莹,孙承福主编. --北京:清华大学出版社,2013

高等院校信息技术应用型规划教材

ISBN 978-7-302-32791-2

I. ①数… II. ①李… ②孙… 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(2013)第 136195 号

责任编辑:孟毅新

封面设计:傅瑞学

责任校对:袁芳

责任印制:王静怡

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>, 010-62795764

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:15.75 字 数:360千字

版 次:2013年8月第1版 印 次:2013年8月第1次印刷

印 数:1~3000

定 价:32.00元

前言

PREFACE

数据结构是计算机及相关专业的一门重要的专业基础课程和专业核心课程。它不仅是计算机程序设计的重要理论基础,还是计算机及相关专业课程的先导,也逐渐为其他工科类专业所重视。

数据结构课程的学习难度较大。主要是数据结构的内容和应用数据结构解决实际问题之间存在着很大的距离,逐步缩短进而消除这个距离是解决本课程学习难度大的关键所在。目前,数据结构的教材很多,各有特点,但适合应用型本科的数据结构教材却凤毛麟角。本书的原则是“以应用为目的”,避免大而全,从实际应用的需求出发,坚持科学性、应用性与先进性的统一,注重实用性。

本书对数据结构的概念和原理的阐述通俗易懂,习题丰富,对于书中重要的算法均给出了完整的 C 语言源程序,并全部在 Visual Studio 2010 环境中运行通过。

全书共分为 10 章。第 1 章介绍数据、数据结构和算法的基本概念;第 2~5 章介绍线性表、栈和队列、串、数组等线性结构的常用算法的实现及基本应用;第 6 章和第 7 章介绍树和图两种非线性数据结构的常用算法的实现及基本应用;第 8 章和第 9 章介绍查找和内部排序操作的各种实现方法及其特点。第 10 章为数据结构课程设计,是学生根据综合实验的例子和自己的基础,进一步加强对学生实际应用能力及动手能力的培养,并建立起系统设计的初步概念。

各章的基本体例结构如下。

- (1) 内容提要:概括本章讲解的主要内容;
- (2) 学习目标与重点:说明本章的学习重点及学习收获;
- (3) 关键术语:本章需要重点理解的关键术语;
- (4) 本章正文;
- (5) 例题:穿插于正文中,通过例题加深对重点问题和难点问题的理解和掌握;
- (6) 本章小结;
- (7) 综合训练。

学习本课程的学生应具备 C 语言的初步编程能力。使用本教材的总学时为 40~70 学时,实验时数在 15~30 学时,课程设计可在课程结束后集中安排 2~3 周。

本书既可以作为应用型本科计算机及相关专业的专业教材,也可以作为非计算机专业学生及计算机爱好者学习数据结构的入门书籍。

本书由沈阳理工大学应用技术学院李莹、孙承福担任主编,辽宁石油化工大学计算机与通信工程学院李文超、沈阳师范大学软件学院王艳、沈阳理工大学应用技术学院梁爽、沈阳大学职业技术学院时巍担任副主编。具体分工如下:李莹、孙承福编写第1~3章和第6~9章;李文超编写第4章;王艳编写第5章;梁爽编写各章的综合训练;时巍编写第10章;石元博(辽宁石油化工大学)参与了第3~5章的编写;史秀男(沈阳理工大学应用技术学院)参与了第8~10章的编写和部分文字校对工作。

由于编者水平有限,书中难免有不足之处,恳请广大读者批评指正。

编 者

2013年5月

目录

CONTENTS

第 1 章 绪论	1
1.1 数据结构课程的地位	1
1.2 数据结构的定义和基本术语	2
1.2.1 数据结构研究的内容及定义	2
1.2.2 基本术语	4
1.3 数据的逻辑结构和存储结构	5
1.3.1 逻辑结构	5
1.3.2 存储结构	5
1.4 算法和算法分析	7
1.4.1 算法	7
1.4.2 算法分析	8
本章小结	9
综合训练	10
第 2 章 线性表	17
2.1 线性表的定义及逻辑结构	17
2.2 顺序存储结构	19
2.2.1 顺序表	19
2.2.2 顺序表中基本操作的实现	20
2.3 链式存储结构	26
2.3.1 单链表	27
2.3.2 单链表中基本操作的实现	28
2.3.3 循环链表	34
2.3.4 双向链表	35
2.4 应用：一元多项式的表示和相加	37
实验一 线性表的顺序存储	39
实验二 单链表的基本操作	43

实验三 链表综合操作	51
本章小结	54
综合训练	55
第3章 栈和队列	65
3.1 栈	65
3.1.1 栈的定义	65
3.1.2 顺序栈的存储结构和操作的实现	66
3.1.3 链栈的存储结构和操作的实现	68
3.2 栈的应用	69
3.3 队列	73
3.3.1 队列的定义	73
3.3.2 链队列的存储结构和操作的实现	74
3.3.3 顺序队列的存储结构和操作的实现	76
3.4 队列的应用	79
实验四 栈和队列的表示与实现	79
实验五 栈和队列的应用	82
本章小结	84
综合训练	84
第4章 串	91
4.1 串的定义	91
4.1.1 基本术语	91
4.1.2 基本运算	92
4.2 串的存储结构	93
4.2.1 定长顺序存储	93
4.2.2 堆分配存储	94
4.2.3 块链存储	94
4.3 串的基本操作的实现	95
本章小结	96
综合训练	96
第5章 数组	98
5.1 数组的定义	98
5.2 数组的顺序存储结构	99
5.3 矩阵的压缩存储	100
5.3.1 对称矩阵	100
5.3.2 三角矩阵	101

5.3.3 对角矩阵	102
5.3.4 稀疏矩阵	102
本章小结	105
综合训练	106
第6章 树和二叉树	110
6.1 树的定义和基本术语	110
6.2 二叉树	111
6.2.1 二叉树的定义和相关概念	111
6.2.2 二叉树的性质	112
6.2.3 二叉树的存储结构	113
6.3 遍历二叉树和线索二叉树	115
6.3.1 遍历二叉树	115
6.3.2 线索二叉树	118
6.4 树和森林	120
6.4.1 树的存储结构	120
6.4.2 森林与二叉树的转换	122
6.4.3 树和森林的遍历	123
6.5 哈夫曼树及其应用	124
6.5.1 最优二叉树	124
6.5.2 哈夫曼编码	125
实验六 树与二叉树	127
本章小结	128
综合训练	129
第7章 图	133
7.1 图的定义和基本术语	133
7.2 图的存储结构	135
7.2.1 邻接矩阵表示法(数组表示法)	135
7.2.2 邻接表表示法	137
7.3 图的遍历	139
7.3.1 深度优先搜索	139
7.3.2 广度优先搜索	139
7.4 图的应用	140
7.4.1 最小生成树	140
7.4.2 拓扑排序	142
7.4.3 关键路径	143
7.4.4 最短路径	146

实验七 图的操作	148
本章小结	148
综合训练	149
第 8 章 查找	153
8.1 查找的基本概念	153
8.2 基于线性表的查找	155
8.2.1 顺序查找	155
8.2.2 折半查找	157
8.2.3 索引查找	161
8.3 基于树的查找	162
8.3.1 二叉排序树	162
8.3.2 平衡二叉树	168
8.3.3 B-树	171
实验八 查找	172
8.4 哈希表	174
8.4.1 哈希表的定义	174
8.4.2 哈希函数的构造方法	175
8.4.3 处理冲突的方法	176
8.4.4 哈希表的查找分析	177
实验九 哈希表的查找	179
本章小结	180
综合训练	180
第 9 章 内部排序	184
9.1 概述	184
9.1.1 排序的定义	184
9.1.2 内部排序和外部排序	184
9.2 插入排序	185
9.2.1 直接插入排序	185
9.2.2 希尔排序	187
9.3 交换排序	188
9.3.1 冒泡排序	189
9.3.2 快速排序	191
9.4 选择排序	193
9.4.1 简单选择排序	194
9.4.2 堆排序	196
9.5 归并排序	200

9.6 基数排序	202
9.6.1 多关键字排序	202
9.6.2 链式基数排序	202
9.7 各种排序方法的比较	205
实验十 内部排序	206
本章小结	208
综合训练	209
第 10 章 课程设计	214
10.1 目的和题目	214
10.2 考核方式及报告要求	214
附录 A 课程设计报告模板	216
附录 B 模拟试题	237
参考文献	240

Chapter 1

第 1 章 绪 论

内容提要

本章首先阐述数据结构课程的地位，然后介绍数据结构的定义、基本术语、数据的逻辑结构和存储结构，最后介绍算法和算法分析。

学习目标与重点

- ◆ 理解数据、数据元素、数据结构和数据类型的概念；
- ◆ 了解算法的概念和算法设计的基本要求；
- ◆ 了解算法的时间复杂度和空间复杂度。

关键术语

数据；数据元素；数据结构；数据类型；算法；时间复杂度；空间复杂度

1.1 数据结构课程的地位

早在 1968 年，数据结构已经作为一门独立的课程开设。我国从 20 世纪 80 年代初才开始正式开设数据结构这门课程。

数据结构是计算机及相关专业的一门主要的专业基础课，是介于数学、计算机硬件和计算机软件之间的一门核心课程，是高级程序设计语言、编译原理、操作系统、数据库、人工智能等课程的基础。同时，数据结构技术也广泛应用于信息科学、系统工程、应用数学以及各种工程技术领域。

数据结构与其他课程之间的关系如图 1.1 所示。

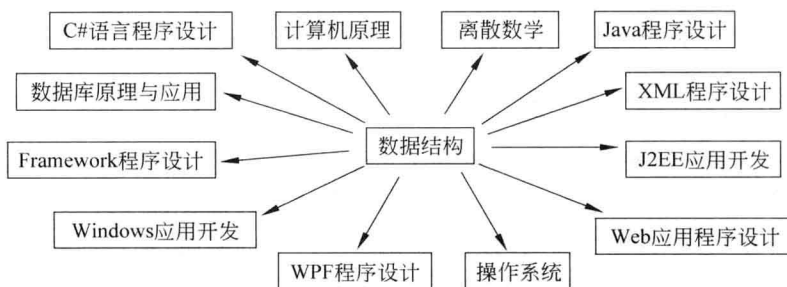


图 1.1 数据结构与其他课程的关系

数据结构课程较系统地介绍了软件设计中经常使用的数据结构及相应的存储结构和算法、常用的查找和排序技术,并对各种结构和技术进行分析和比较,适用于什么环境,内容丰富。

从某种意义上说,数据结构是程序设计的后继课程。因此,学习数据结构时,仅从书本上学习是不够的,必须经过大量的实践,在实践中体会构造性思维方法,掌握数据组织与程序设计的技术。本书中配有大量的例题、习题、实验和课程设计,希望通过循序渐进的理论学习和实践环节,能够掌握学习数据结构的方法和技巧。

1.2 数据结构的定义和基本术语

1.2.1 数据结构研究的内容及定义

随着计算机应用领域的扩大和软、硬件的发展,非数值计算问题越来越显得重要。据统计,当今处理非数值计算性问题占用了 90% 以上的机器时间。这类问题涉及的数据结构更为复杂,数据元素之间的相互关系一般无法用数学方程式加以描述。因此,解决这类问题的关键是要设计出合适的数据结构,才能有效地解决问题。下面所列举的就是属于这一类的具体问题。

【例 1.1】 学生信息管理系统分析。

要查询某个学生的有关情况,或者想查询某个专业或年级的学生的有关情况时,只要建立了相关的数据结构,按照某种算法编写了相关程序,就可以实现计算机自动检索。由此,可以在学生信息查询系统中建立一张按学号顺序排列的学生信息表,如表 1.1 所示。由这张表构成的文件便是学生信息查询系统的数学模型,计算机的主要操作便是按照某个特定要求(如给定姓名)对学生信息文件进行查询。

表 1.1 学生信息表

学号	姓名	性别	专业
10301106	班金玉	女	计算机科学与技术
10301208	宋天成	男	计算机科学与技术
10301303	徐粤玲	女	计算机科学与技术
10306110	魏兴	男	软件工程
10306217	孙国杰	男	软件工程
10306305	王小红	女	软件工程
10307105	王驰文	男	网络工程
10307120	王慕鑫	男	网络工程

表 1.1 中的每一行是一条记录,第一条记录称为开始结点,最后一条记录称为终端结点。除了第一条记录和最后一条记录以外,其余的记录都有且仅有一条直接前驱记录和一条直接后继记录,这些结点之间的关系是“一对一”的关系,是一种简单的线性关系,即“线性表”。这是学生信息表的逻辑结构。

学生信息表在计算机中又是如何存储的呢?表中各个元素在存储器中是连续存放

的,还是不连续存放的呢?这是学生信息表的存储结构。

在学生信息表中查询、插入或删除记录等操作,即是数据的运算。

数据结构就是研究数据的逻辑结构、存储结构和运算方法(算法)的学科。

诸如此类的还有图书信息查询系统、考试查分系统、仓库库存管理系统等。

【例 1.2】 企业组织结构分析。

企业组织结构是企业组织内部各个机构相互作用的联系方式或形式,以求有效、合理地把组织成员组织起来,为实现共同目标而协同努力。组织结构是企业资源和权力分配的载体,它在人的能动行为下,通过信息传递,承载着企业的业务流动,推动或者阻碍企业使命的进程。某企业组织结构图如图 1.2 所示。

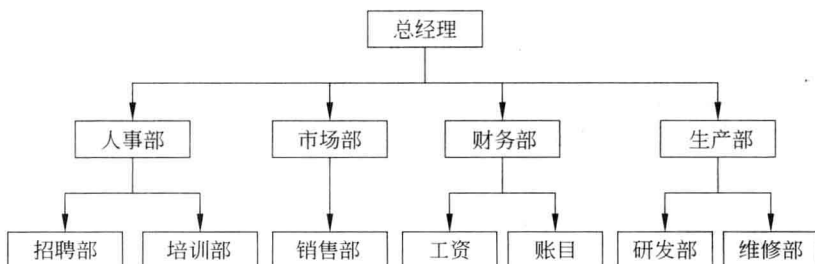


图 1.2 某企业组织结构图

从图 1.2 可以看出,企业组织结构是具有层次结构的数据模型,是一种树状结构。“树根”称为开始结点,没有直接前驱,而所有的“叶子”则是具体的某个部门,结点之间存在着“一对多”的关系,这是企业组织结构的逻辑结构。

【例 1.3】 教学计划编排。

一个教学计划包含许多课程,在教学计划包含的许多课程之间,有些必须按规定的先后次序进行,有些则没有次序要求。即有些课程之间有先修和后修的关系,有些课程可以任意安排次序,如表 1.2 所示。这种各个课程之间的次序关系可用一个称作“图”的数据结构来表示,如图 1.3 所示。

表 1.2 某专业的课程设置

课程编号	课程名称	先修课程
C ₁	C 程序设计语言	无
C ₂	C# 程序设计	C ₁
C ₃	数据库原理	C ₄ , C ₈
C ₄	数据结构	C ₁
C ₅	数据访问技术	C ₂ , C ₃
C ₆	XML 程序设计	C ₃
C ₇	计算机网络	C ₈
C ₈	操作系统	C ₄
C ₉	软件工程	C ₂ , C ₄ , C ₅
C ₁₀	软件测试	C ₉

图 1.3 中每个顶点表示一门课程,如果从顶点 v_i 到 v_j 之间存在有向边 $\langle v_i, v_j \rangle$,则表示课程 i 必须先于课程 j 进行。如何在规定的年限(如 4 年)内完成所有课程,哪个学期安排哪门课程,就是要进行的一种操作。

由以上几个例子可知,描述非数值计算问题的数学模型不再是数学方程,而是诸如“线性表”、“树”、“图”之类的数据结构。因此,简单说来,数据结构是一门研究非数值计算的程序设计问题中计算机的操作对象以及它们之间的关系和操作等的学科。

著名的计算机科学家 Niklaus Wirth 提出“Algorithm+Data Structures=Programs”的思想,即“算法+数据结构=程序”。可以看出,数据结构是程序的重要组成部分。概括地说:数据结构是一门讨论“描述现实世界实体的数学模型(非数值计算)及其上的操作在计算机中如何表示和实现”的学科。

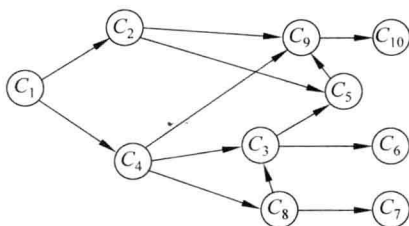


图 1.3 课程之间优先关系图

1.2.2 基本术语

(1) 数据: 所有能被输入计算机,且能被计算机处理的符号的集合;是计算机操作对象的总称;是计算机处理的信息的某种特定的符号表示形式。

(2) 数据元素: 是数据(集合)中的一个“个体”,是数据结构中讨论的基本单位。

(3) 数据项: 数据元素可以是数据项的集合,是数据结构中讨论的最小单位,如学生记录中的学号、姓名等。

(4) 数据结构: 带结构的数据元素的集合,或者说,数据结构是相互之间存在着某种逻辑关系的数据元素的集合。

在任何问题中,数据元素之间都不会是孤立的,在它们之间都存在着这样或那样的关系,这种数据元素之间的关系称为结构。

根据数据元素间关系的不同特性,通常有下列 4 类基本结构。

(1) 线性结构: 该结构的数据元素之间存在着一对一的关系。

(2) 树状结构: 该结构的数据元素之间存在着一对多的关系。

(3) 图状结构: 该结构的数据元素之间存在着多对多的关系,该结构也称作网状结构。

(4) 集合结构: 在集合结构中,数据元素间的关系是“属于同一个集合”。集合是元素关系极为松散的一种结构。

图 1.4 为表示上述 4 类基本结构的示意图。

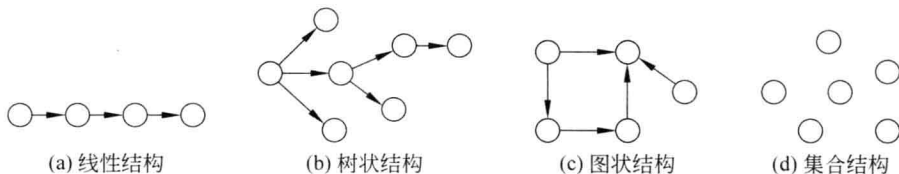


图 1.4 数据的 4 类基本结构

1.3 数据的逻辑结构和存储结构

数据结构包括数据的逻辑结构和数据的存储结构。数据的逻辑结构可以看做是从具体问题抽象出来的数学模型,与数据的存储无关。研究数据结构的目的是在计算机中实现对它的操作,为此还需要研究如何在计算机中表示一个数据结构。数据结构在计算机中的表示(又称映像)称为数据的物理结构,或称存储结构。它所研究的是数据结构在计算机中的实现方法,包括数据结构中元素的表示及元素间关系的表示。

1.3.1 逻辑结构

从 1.2.2 小节介绍的数据结构的术语中可以知道,一个数据结构有两个要素:一个是数据元素的集合;另一个是关系的集合。在形式上,数据结构通常可以采用一个二元组来表示。

数据结构的定义如下。

数据结构是一个二元组:

$$\text{Data_Structure} = (D, R)$$

其中, D 是数据元素的有限集; R 是 D 上关系的有限集。

【例 1.4】 一种数据结构 $\text{Line} = (D, R)$, 其中:

$$D = \{1, 2, 3, 4, 5, 6, 7\},$$

$$R = \{\langle 3, 7 \rangle, \langle 7, 1 \rangle, \langle 1, 5 \rangle, \langle 5, 2 \rangle, \langle 2, 4 \rangle, \langle 4, 6 \rangle\}$$

其中,尖括号表示关系集合,如 $\langle 3, 7 \rangle$ 表示是有向的,即表示从 3 指向 7。以此类推,这种数据结构的表示如图 1.5 所示。其特点是除了头结点 3 和尾结点 6 以外,其余结点都只有一个直接前驱和一个直接后继,即结构的元素之间存在着一对一的关系。把具有这种特点的数据结构称为线性结构。



图 1.5 线性结构

把数据元素之间存在一对多关系的树形结构和数据元素之间存在多对多关系的图形结构统称为非线性结构。

1.3.2 存储结构

数据的存储结构是数据的逻辑结构在存储器中的映像。数据元素在计算机中主要有以下 4 种不同的存储结构。

(1) 顺序存储:以相对的存储位置表示后继关系。例如,令 y 的存储位置和 x 的存储位置之间差一个常量 C ,而 C 是一个隐含值,整个存储结构中只含数据元素本身的信息。

(2) 链式存储:以附加信息(指针)表示前驱或后继关系。需要用一个或几个和 x 在一起的附加信息指示 y 的存储位置。

(3) 索引存储: 在原有存储数据结构的基础上, 附加建立一个索引表, 索引表中的每一项都由关键字和地址组成。索引存储的主要作用是提高数据的检索速度。具体方法将在第 8 章中介绍。

(4) 散列存储: 通过构造散列函数来确定数据存储地址或查找地址。具体方法将在第 8 章中介绍。

在不同的编程环境中, 存储结构可有不同的描述方法。当用程序设计语言进行编程时, 通常可用编程语言中提供的数据类型描述之。例如, 以三个带有次序关系的整数表示一个长整数时, 可利用 C 语言中提供的整数数组类型, 定义长整数的格式如下:

```
typedef int Long_int [3]
```

在用高级程序语言编写的程序中, 必须对程序中出现的每个变量、常量或表达式, 明确说明它们所属的数据类型。不同类型的变量, 其所能取的值的范围不同, 所能进行的操作不同。

数据类型是一个值的集合和定义在此集合上的一组操作的总称。

抽象数据类型 (Abstract Data Type, ADT) 是指一个数学模型以及定义在此数学模型上的一组操作。

例如, 抽象数据类型复数的定义如下:

```
ADT Complex {
    数据对象:
        D = { e1, e2 | e1, e2 ∈ RealSet }
    数据关系:
        R1 = { < e1, e2 > | e1 是复数的实数部分 | e2 是复数的虚数部分 }
    基本操作:
        AssignComplex (&Z, v1, v2)
        操作结果: 构造复数 Z, 其实部和虚部分别被赋以参数 v1 和 v2 的值
        DestroyComplex (&Z)
        操作结果: 复数 Z 被销毁
        GetReal (Z, &realPart)
        初始条件: 复数已存在
        操作结果: 用 realPart 返回复数 Z 的实部值
        GetImag (Z, &ImagPart)
        初始条件: 复数已存在
        操作结果: 用 ImagPart 返回复数 Z 的虚部值
        Add (z1, z2, &sum)
        初始条件: z1, z2 是复数
        操作结果: 用 sum 返回两个复数 z1, z2 的和值
} ADT Complex
```

假设 z_1 和 z_2 是上述定义的复数, 则 $\text{Add}(z_1, z_2, z_3)$ 操作的结果即为用户所求的结果 $z_3 = z_1 + z_2$ 。

抽象数据类型可用 (D, S, P) 三元组表示。其中: D 是数据对象; S 是 D 上的关系集; P 是对 D 的基本操作集。

```
ADT 抽象数据类型名 {
```


数据对象 D: <数据对象的定义>
数据关系 S: <数据关系的定义>
基本操作 P: <基本操作的定义>
} ADT 抽象数据类型名

其中,基本操作的定义格式如下:

基本操作名(参数表)

初始条件: <初始条件描述>描述了操作执行之前数据结构和参数应满足的条件,若不满足,则操作失败,并返回相应出错信息

操作结果: <操作结果描述>说明了操作正常完成之后,数据结构的变化状况和应返回的结果。若初始条件为空,则省略之

赋值参数: 只为操作提供输入值

引用参数: 以 & 打头,除可提供输入值外,还将返回操作结果

抽象数据类型需要通过固有数据类型(高级编程语言中已实现的数据类型)来实现。

本书在讨论各种数据结构时,就是针对其逻辑结构和具体的存储结构给出相应的数据类型,并在确定的数据类型上通过各种算法实现其各种操作。

1.4 算法和算法分析

1.4.1 算法

算法(Algorithm)是为了解决某类问题而规定的一个有限长的操作序列。一个算法必须满足以下五个重要特性。

(1) 有穷性。对于任意一组合法输入值,在执行有穷步骤之后一定能结束,即算法中的每个步骤都能在有限时间内完成。

(2) 确定性。对于每种情况下所应执行的操作,在算法中都有确切的规定,使算法的执行者或阅读者都能明确其含义及如何执行。并且在任何条件下,算法都只有一条执行路径。

(3) 可行性。算法中的所有操作都必须足够基本,都可以通过已经实现的基本操作运算有限次实现之。

(4) 有输入。作为算法加工对象的量值,通常体现为算法中的一组变量。有些输入量需要在算法执行过程中输入,而有的算法表面上可以没有输入,实际上已被嵌入算法之中。

(5) 有输出。它是一组与“输入”有确定关系的量值,是算法进行信息加工后得到的结果,这种确定关系即为算法的功能。

设计算法时,通常应考虑达到以下目标。

(1) 正确性。算法应当满足具体问题的要求。目前多数是用自然语言描述需求,它至少应包括对于输入、输出和处理等的明确描述。

(2) 可读性。算法主要是为了人的阅读与交流,其次才是为计算机执行,因此算法应该易于人理解;另一方面,晦涩难读的程序易于隐藏较多错误而难以调试。

(3) 健壮性。当输入的数据非法时,算法应当恰当地作出反应或进行相应处理,而不