



工业和信息化部职业教育教学指导委员会“十二五”规划教材
全国高等职业教育计算机系列规划教材

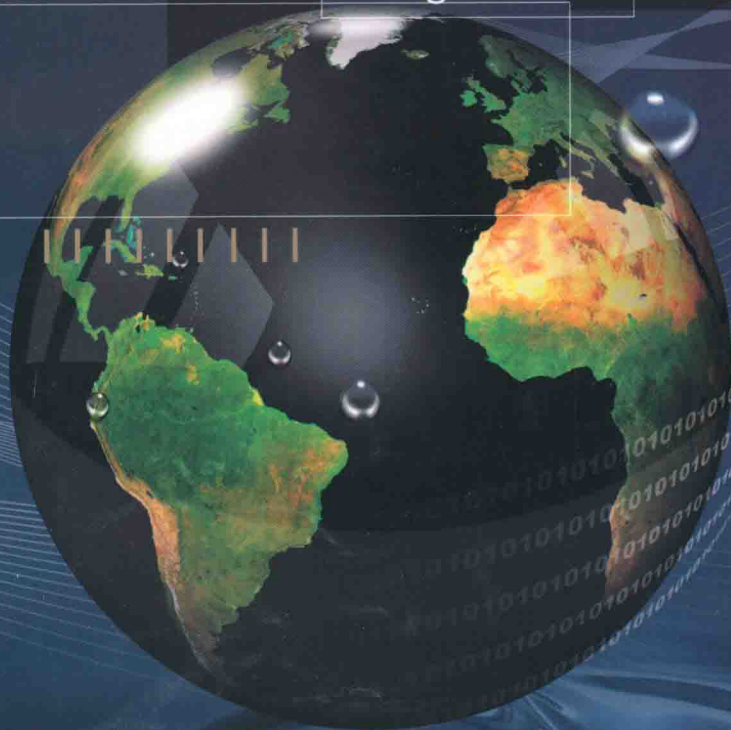
数据结构

©丛书编委会

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

Data Structure

Algorithm



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY



清华大学出版社
Tsinghua University Press

数据结构

第二版

清华大学出版社
Tsinghua University Press



清华大学出版社

工业和信息产业职业教育教学指导委员会“十二五”规划教材
全国高等职业教育计算机系列规划教材

数 据 结 构

丛书编委会

電子工業出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 简 介

本书主要包括数据结构绪论, 线性表, 栈和队列, 串、数组和广义表, 树, 图, 查找, 排序, 以及课程设计指导。在每章开始给出了本章导读和教学目标, 使学生在在学习之前就能明白要重点掌握的内容; 每章后附有习题及实训, 以便学生巩固所学知识。课程设计指导一章给出了几种设计题目及设计的思想供学生选择, 有助于教师指导学生完成课程设计任务。

本书适合高等职业院校、高等专科学校、民办本科院校, 以及软件职业教育学院、继续教育学院、技能紧缺人才培养使用, 还可供计算机专业人员和爱好者使用。

未经许可, 不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有, 侵权必究。

图书在版编目(CIP)数据

数据结构 / 《全国高等职业教育计算机系列规划教材》丛书编委会编. —北京: 电子工业出版社, 2011.6
工业和信息化部职业教育教学指导委员会“十二五”规划教材·全国高等职业教育计算机系列规划教材
ISBN 978-7-121-13751-8

I. ①数… II. ①全… III. ①数据结构—高等职业教育—教材 IV. ①TP311.12

中国版本图书馆 CIP 数据核字 (2011) 第 104486 号

策划编辑: 左 雅

责任编辑: 郝黎明 特约编辑: 赵海红 李云霞

印 刷: 北京建筑工业印刷厂
装 订:

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本: 787×1 092 1/16 印张: 17.25 字数: 442 千字

印 次: 2011 年 6 月第 1 次印刷

印 数: 4 000 册 定价: 31.00 元

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888。

质量投诉请发邮件至 zltz@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线: (010) 88258888。

丛 书 编 委 会

主 任 郝黎明 逢积仁

副 主 任 左 雅 方一新 崔 炜 姜广坤 范海波 敖广武 徐云晴 李华勇

委 员 (按拼音排序)

陈国浪 迟俊鸿 崔爱国 丁 倩 杜文洁 范海绍 何福男

贺 宏 槐彩昌 黄金栋 蒋卫祥 李 琦 刘宝莲 刘红军

刘 凯 刘兴顺 刘 颖 卢锡良 孟宪伟 庞英智 钱 哨

乔国荣 曲伟峰 桑世庆 宋玲玲 王宏宇 王 华 王晶晶

温丹丽 吴学会 邢彩霞 徐其江 严春风 姚 嵩 殷广丽

尹 辉 俞海英 张洪明 张 薇 赵建伟 赵俊平 郑 伟

周绯非 周连兵 周瑞华 朱香卫 邹 羚

本 书 编 委 会

主 编 乔国荣

副主编 王晶晶

丛书编委会院校名单

(按拼音排序)

保定职业技术学院	山东省潍坊商业学校
渤海大学	山东司法警官职业学院
常州信息职业技术学院	山东信息职业技术学院
大连工业大学职业技术学院	沈阳师范大学职业技术学院
大连水产学院职业技术学院	石家庄信息工程职业学院
东营职业学院	石家庄职业技术学院
河北建材职业技术学院	苏州工业职业技术学院
河北科技师范学院数学与信息技术学院	苏州托普信息职业技术学院
河南省信息管理学校	天津轻工职业技术学院
黑龙江工商职业技术学院	天津市河东区职工大学
吉林省经济管理干部学院	天津天狮学院
嘉兴职业技术学院	天津铁道职业技术学院
交通运输部管理干部学院	潍坊职业学院
辽宁科技大学高等职业技术学院	温州职业技术学院
辽宁科技学院	无锡旅游商贸高等职业技术学校
南京铁道职业技术学院苏州校区	浙江工商职业技术学院
山东滨州职业学院	浙江同济科技职业学院
山东经贸职业学院	

前 言

“数据结构”是计算机学科的核心课程，也是计算机专业一门重要的专业基础课。这门课程主要研究如何合理地组织数据；怎样在计算机中有效地表示数据和处理数据。学习这门课程的教学要求是：使学生学会分析、研究计算机加工的数据结构的特性，以便选择适当的逻辑结构、存储结构及相应的算法，并初步掌握算法的时间分析和空间分析技术。另外，学习本课程也是复杂程序设计的训练过程，训练学生编写的程序结构清楚、正确易读，符合软件工程的规范，为后继课程的学习打下良好的基础。

全书共分为9章。第1章介绍了数据结构和算法的基本概念和常用术语；第2章至第6章介绍了基本的数据结构，分别讨论了线性表，栈与队列，串、数组和广义表，树和图几种结构类型数据的逻辑结构和存储结构，以及相应的算法；第7章和第8章介绍了几种常用的查找和排序方法；第9章是本书的特色，增加了课程设计指导的内容，使学生在学完基本知识的同时，能够综合利用所学知识完成一些实际课题的设计与制作。另外，为了便于教学，每章后面还配有习题和实训，并提供了实训练习题的相应参考答案。全书概念表述清楚、简洁，内容由浅入深，强调实践环节，利于教学和自学。

全书采用C语言作为数据结构和算法的描述语言，书中的大部分程序学生上机就可以调试运行，个别程序需稍做补充。这是考虑到程序设计语言学习环节相对薄弱的同学也能学会数据结构，而不会为编写程序所难倒，从而放弃该门课程的学习。

本书可作为高等职业学校、高等专科学校、成人高等学校及本科院校举办的二级职业技术学院计算机类专业或信息类相关专业的教材，也可作为非计算机专业学生选修课或全国计算机等级二级考试理论辅修课的教材，还可作为计算机应用人员和工程技术人员的自学参考书。

本书由乔国荣主编，其中王晶晶编写了第5章部分内容，由乔国荣统编。本书作者是辽宁省《数据结构》课程精品课获得者，精品课程的相关配套资源读者可登录华信教育资源网（www.hxedu.com.cn）免费下载。

本书在编写过程中，作者得到了所在单位大连轻工业学院职业技术学院的领导和同行的支持。在此一并表示衷心的感谢。

由于编者水平有限，书中难免有一些不足之处，恳请读者批评指正。

编 者

目 录

第1章 绪论	(1)
1.1 数据结构的基本概念	(1)
1.1.1 数据结构的定义	(1)
1.1.2 数据的逻辑结构及存储结构	(3)
1.1.3 数据结构有关概念及术语	(4)
1.2 算法和算法描述	(5)
1.2.1 什么是算法	(5)
1.2.2 算法描述	(5)
1.3 算法分析	(6)
1.3.1 空间复杂度	(6)
1.3.2 时间复杂度	(6)
1.4 本章小结	(7)
习题1	(7)
第2章 线性表	(10)
2.1 线性表的逻辑结构	(10)
2.1.1 线性表的定义	(10)
2.1.2 线性表的基本操作	(11)
2.2 线性表的顺序存储结构	(11)
2.2.1 线性表的顺序存储——顺序表	(11)
2.2.2 顺序表基本操作的实现	(12)
2.2.3 顺序表的应用举例	(16)
2.3 线性表的链式存储结构	(19)
2.3.1 线性表的链式存储——链表	(19)
2.3.2 单链表	(20)
2.3.3 循环链表	(29)
2.3.4 双向链表	(30)
2.3.5 单链表应用举例	(32)
2.4 本章小结	(36)
习题2	(36)
实训1	(39)
第3章 栈与队列	(42)
3.1 栈	(42)
3.1.1 栈的定义	(42)
3.1.2 栈的顺序存储及其基本操作的实现	(43)
3.1.3 栈的链式存储及其基本操作的实现	(49)
3.1.4 栈的应用举例	(52)

3.2	队列	(55)
3.2.1	队列的定义	(55)
3.2.2	队列的顺序存储及其基本操作的实现	(55)
3.2.3	队列的链式存储及其基本操作的实现	(60)
3.2.4	队列的应用举例	(63)
3.3	本章小结	(64)
	习题 3	(64)
	实训 2	(67)
第 4 章	串、数组和广义表	(76)
4.1	串	(76)
4.1.1	串的定义和特性	(76)
4.1.2	串的顺序存储及其基本操作实现	(77)
4.1.3	串的链式存储及其基本操作实现	(85)
4.1.4	串的应用举例	(86)
4.2	数组	(86)
4.2.1	数组的定义和运算	(86)
4.2.2	数组的顺序存储结构	(87)
4.2.3	矩阵的压缩存储	(88)
4.2.4	稀疏矩阵	(89)
4.3	广义表	(95)
4.3.1	广义表的定义和特性	(95)
4.3.2	广义表的存储结构及其基本操作实现	(96)
4.4	本章小结	(97)
	习题 4	(97)
	实训 3	(99)
第 5 章	树	(103)
5.1	树	(103)
5.1.1	树的定义及基本术语	(103)
5.1.2	树的表示	(104)
5.2	二叉树及其遍历	(105)
5.2.1	二叉树的定义	(105)
5.2.2	二叉树的重要性质	(105)
5.2.3	二叉树的存储结构	(107)
5.2.4	二叉树的遍历	(108)
5.3	线索二叉树	(115)
5.3.1	线索二叉树的定义	(115)
5.3.2	线索二叉树的基本操作	(117)
5.4	树和森林	(118)
5.4.1	树的存储结构	(118)
5.4.2	二叉树与树之间的转换	(120)

5.4.3	森林与二叉树的转换	(120)
5.4.4	树与森林的遍历	(121)
5.5	二叉树应用实例	(122)
5.5.1	二叉排序树	(122)
5.5.2	平衡二叉树	(128)
5.5.3	B 树	(131)
5.5.4	哈夫曼树	(133)
5.6	本章小结	(135)
习题 5		(136)
实训 4		(140)
实训 4.1	二叉树的操作	(140)
实训 4.2	树的应用	(140)
第 6 章	图	(145)
6.1	图的基本概念	(145)
6.1.1	图的定义	(145)
6.1.2	图的基本术语	(146)
6.2	图的存储结构	(148)
6.2.1	邻接矩阵	(148)
6.2.2	邻接表	(150)
6.3	图的遍历	(152)
6.3.1	深度优先搜索	(152)
6.3.2	广度优先搜索	(154)
6.4	最小生成树	(157)
6.4.1	普里姆算法	(158)
6.4.2	克鲁斯卡尔算法	(161)
6.5	最短路径	(164)
6.5.1	单源最短路径	(165)
6.5.2	每对顶点之间的最短路径	(169)
6.6	拓扑排序	(173)
6.6.1	AOV 网	(173)
6.6.2	拓扑 (Topology) 排序的实现	(174)
6.7	本章小结	(177)
习题 6		(177)
实训 5		(180)
第 7 章	查找	(184)
7.1	查找的基本概念	(184)
7.2	顺序查找	(185)
7.3	二分查找	(186)
7.4	分块查找	(189)
7.5	哈希表查找	(192)

7.5.1 哈希表查找的基本概念	(192)
7.5.2 构造哈希函数的方法	(193)
7.5.3 哈希冲突的解决方法	(195)
7.5.4 哈希查找效率的分析	(199)
7.6 本章小结	(200)
习题 7	(200)
实训 6	(203)
第 8 章 排序	(206)
8.1 排序的基本概念	(206)
8.2 插入排序	(207)
8.2.1 直接插入排序	(208)
8.2.2 二分法插入排序	(209)
8.2.3 希尔排序	(210)
8.3 选择排序	(212)
8.3.1 简单选择排序	(212)
8.3.2 堆排序	(213)
8.4 交换排序	(216)
8.4.1 冒泡排序	(216)
8.4.2 快速排序	(219)
8.5 归并排序	(221)
8.6 基数排序	(223)
8.7 本章小结	(226)
习题 8	(226)
实训 7	(229)
第 9 章 课程设计指导	(234)
9.1 课程设计大纲	(234)
9.2 课程设计题目及设计要求	(235)
9.3 飞机售票系统实例	(237)
9.4 本章小结	(243)
附录 A 习题答案	(245)
习题 1	(245)
习题 2	(246)
习题 3	(249)
习题 4	(251)
习题 5	(253)
习题 6	(257)
习题 7	(259)
习题 8	(262)
参考文献	(266)

第1章 绪 论

本章导读

在深入学习数据结构之前,首先了解学习数据结构的意义、基本术语及一些相关概念等,这对学习后面的内容将有很大的帮助。本章介绍的数据结构研究对象和有关概念包括数据、数据元素、数据类型、逻辑结构、存储结构、算法描述(VC++语言描述)和算法分析等。

教学目标

通过本章学习,要求掌握以下内容:

- 理解和熟悉数据结构中的基本概念。
- 理解和掌握线性结构、树形结构和图形结构的概念。
- 熟悉算法评价的一般规则,算法时间复杂度、空间复杂度的概念和数量级的表示方法。

1.1 数据结构的基本概念

1.1.1 数据结构的定义

计算机是一种数据处理装置。用计算机处理实际问题时,一般要先对具体问题进行抽象化,建立起实际问题的求解模型,然后设计出相应的算法,编写程序并上机调试,直至得到最终结果。

在计算机处理数据过程中,大批量的数据并不是彼此孤立、杂乱无章的,它们之间有着某种内在的联系。只有利用这些内在的联系,把所有数据按照某种规则有机地组织起来,才能对数据进行有效的处理。因此,要设计出一个结构好、效率高的程序,必须研究数据的特性、数据间的相互关系及其对应的存储表示,并利用这些特性和关系设计出相应的算法和程序。

下面举出几个例子,来说明什么是数据结构。

例 1.1 学生成绩检索问题,见表 1-1。



表 1-1 学生成绩表

学 号	姓 名	性 别	视 频 技 术	C 程序设计	网 页 设 计	多媒体著作工具
2002001	李丽	女	76	88	78	80
2002002	乔丽娜	女	77	82	67	77
2002003	王龙龙	男	84	76	76	67
2002004	李琳	女	85	87	88	78
2002005	朱宏利	男	88	90	78	69

这个学生成绩表是一个二维表格，每一行表示一个学生的全部信息，每一列数据的类型相同。整个二维表形成学生成绩的一个线性序列，每个学生的信息按照学号次序存放，各学生之间形成一种线性关系。这是一种典型的数据结构，我们称这种数据结构为线性表。对这种线性表的主要操作是，当给出学生的姓名时，能在该表中快速找到学生每门课程的成绩，还有就是对该表如何添加一个新的学生，如何删除已经退学的学生，如何修改表中的数据等操作。这就是数据结构要研究的内容。

例 1.2 某高校的专业设置情况，如图 1-1 所示。

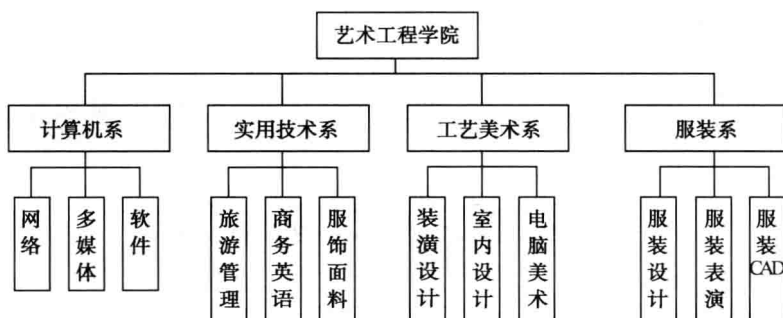


图 1-1 某高校专业设置情况

在图 1-1 中，艺术工程学院分 4 个系，每个系设 3 个专业。在这种数据结构中，数据之间的关系是一对多的非线性关系。这也是常用的一种数据结构，我们称为树形结构。

例 1.3 城镇之间的公路网，如图 1-2 所示。

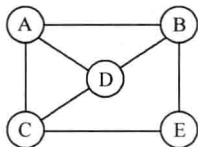


图 1-2 城镇之间的公路网

在城镇公路网中，每个顶点代表一个城镇，边表示城镇之间的道路。在这种数据结构中，数据之间的关系是多对多的非线性关系，我们称这种数据结构为图形结构。

综合 3 个例子可见，数据结构是相互之间存在一种或多种特定关系的数据元素的集合，是一门研究非数值计算的程序设计中，计算机的操作对象及它们之间的关系和操作的学科。



1.1.2 数据的逻辑结构及存储结构

1. 数据的逻辑结构

数据的逻辑结构与数据在计算机中的存储无关,因此,数据的逻辑结构可以看成是从具体的问题中抽象出来的数学模型。根据数据元素之间关系的不同特性,数据的逻辑结构可划分为下面4种。

(1) 集合

结构中各数据元素之间不存在任何关系。这是数据结构的一种特殊情况,不在本书讨论范围之内。

(2) 线性结构

该数据结构中的数据元素存在着一对一的关系。

(3) 树形结构

该数据结构中的数据元素存在着一对多的关系。

(4) 图形或网状结构

该数据结构中的数据元素存在着多对多的关系。

上述4类基本数据结构关系如图1-3所示。

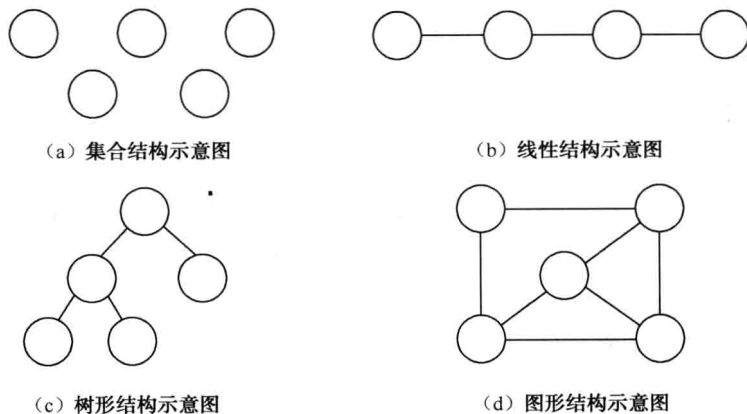


图 1-3 4 种基本数据结构

2. 数据的存储结构

数据的逻辑结构需要用计算机处理,要存入到计算机的存储单元中。数据的逻辑结构在计算机中的表示称为数据的存储结构,又称数据的物理结构。讨论数据结构时,不但要讨论数据的逻辑结构,还要讨论数据的存储结构。通常,计算机内的数据元素用一组连续的位串来表示,这个位串称为结点。数据元素之间的关系,又称结点之间的关系。在计算机内有以下4种存储数据元素的表示方法。

(1) 顺序存储方法

该方法是将逻辑上相邻的结点存储在物理位置上也相邻的存储单元中,结点之间的逻辑关系由存储单元的邻接关系来表示。用这种方法存储数据元素时,只存储结点的值,不存储结点之间的关系,这种存储表示称为顺序存储结构,主要应用于线性的数据结构;非线性的数据结构也可以通过某种线性化的过程后进行存储。



(2) 链式存储方法

链式存储方法不要求逻辑上相邻的结点在物理位置上也相邻, 结点间的关系由附加的指针来表示。通过指针指向结点的邻接结点, 将所有结点串联在一起, 这称为链式存储结构。也就是说, 链式存储方法不仅存储结点的值, 而且还存储结点之间的关系。所以, 链式存储方法中的结点由两部分组成, 一个是存储结点本身的值, 称为数据域; 另一个是存储该结点的各后继结点的存储单元地址, 称为指针域 (可包含一个或多个指针)。

(3) 索引存储方法

索引存储方法是在存储结点信息的同时, 再建立一个附加的索引表, 然后利用索引表中索引项的值来确定结点的实际存储单元地址。索引表中的每一项称为索引项, 索引项的一般形式为 (关键字, 地址), 关键字能唯一标识一个结点。

(4) 哈希存储方法

哈希存储方法的基本思想是根据结点的关键字直接计算出结点的存储地址。方法是把结点的关键字作为自变量, 通过一个称为哈希函数 (Hash) 的计算规则, 确定出该结点的确切存储单元地址。

上面这 4 种方法既可以单独使用, 也可以组合起来对数据结构进行存储。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。选取哪种存储结构来表示相应的逻辑结构视具体的情况而定, 具体要考虑数据的运算是否方便及相应算法的时间复杂度和空间复杂度的要求。

1.1.3 数据结构有关概念及术语

1. 数据 (Data)

数据是指能够输入到计算机中, 并能被计算机处理的一切对象。对计算机科学而言, 数据的含义极为广泛, 如整数、实数、字符、文字、图形、图像和声音等都是数据。

2. 数据元素 (Data Element)

数据元素是数据的基本单位, 在计算机程序中通常作为一个整体进行考虑和处理。例如, 在表 1-1 中, 学生成绩表中的一行就是一个数据元素。数据元素还可以分割成若干个具有不同属性的项 (字段)。一般由一个或多个数据项组成。

3. 数据项 (Data Item)

数据项是具有独立意义的最小数据单位, 是对数据元素属性的描述。在表 1-1 中, 每个数据元素由 7 个数据项组成, 其中“学号”数据项描述了顺序, “姓名”数据项描述了成绩所有者的名字, 其他几项描述了学生的具体成绩。

4. 数据类型 (Data Type)

数据类型是一组性质相同的值的集合及定义于这个集合上的一组操作的总称。每个数据项都属于某一确定的基本数据类型。在表 1-1 中, “学号”为数值型, “姓名”为字符型。

5. 数据对象 (Data Object)

数据对象是性质相同的数据元素的集合, 是数据的一个子集。例如, 整数数据对象的集合是 $\{0, \pm 1, \pm 2, \dots\}$; 字符数据对象的集合是 $\{'A', 'B', 'C', \dots, 'Z'\}$ 。



1.2 算法和算法描述

1.2.1 什么是算法

算法是对某一特定问题求解步骤的一种描述。在计算机系统中,算法是由若干条指令组成的有穷序列,其中每一条指令表示计算机的一个或多个操作。算法满足以下5个性质:

- 输入:一个算法可以有零个或多个输入量,在算法执行之前提供给算法。
- 输出:一个算法的执行结果要有一个或多个输出量,它是算法对输入数据处理的结果。
- 有穷性:一个算法必须在执行有穷步骤之后结束,即必须在有限时间内完成。
- 确定性:算法中的每一步骤都有明确的含义,没有二义性。
- 可行性:算法中的每一步都必须可行的,算法中描述的操作的每一步都能在有限次、有限时间内得以实现。

对于同一个问题,可以有很多种不同的算法,这就需要对算法有一个总的设计要求。一般来说,一个算法必须具有以下几个方面的基本特征。

1. 正确性

正确性是设计一个算法的首要条件,所设计的算法要满足具体问题的要求。在给算法输入合理的数据后,能在有限的时间内得出正确的结果。

2. 可读性

算法是对特定问题求解步骤的一种描述,它能转变成计算机可执行的程序,同时必须可以供他人使用。为了使所设计的算法让他人能看懂,在算法或程序中可以增加一些注释来提高可读性。

3. 健壮性

当输入的数据不符合要求时,算法应能判断出数据的非法性,并能进行适当的处理,比如暂停或终止程序的执行、显示错误信息等。算法不应允许产生不可预料的结果。

4. 高效性

算法的效率是指算法执行的时间和占用的存储空间。如果对于同一个问题有多个算法可供选择,应尽可能选择执行时间短、占用空间少的算法。

1.2.2 算法描述

算法的描述方法有很多。根据描述算法语言的不同,可将算法描述分为以下4种。

1. 框图算法描述

框图算法描述是采用传统流程图或N-S图等方式来描述算法,在算法研究的早期很流行。它的优点是直观、易懂,但用来描述比较复杂的算法时就显得不太方便,也不够清晰。

2. 自然语言描述

自然语言描述就是用人类自然语言(如中文、英文等),同时使用一些程序设计语言中的语句来描述算法。

3. 伪语言算法描述

如使用类C语言进行算法描述。这种算法不能直接在计算机上运行,但专业设计人员



经常使用类 C 语言来描述算法，因为它容易编写、易阅读，有统一的格式。

4. 计算机语言描述

这是可以在计算机上运行并获得结果的算法，通常这种算法也称为程序。本书中的大部分算法都是用 C 语言描述的，并且在 VC++ 6.0 环境下调试通过，而且尽可能给出一个完整的 C 语言程序。在本书所有实训的参考答案中，也给出了一个完整的 C 语言程序，以方便学生上机参考。

1.3 算法分析

求解一个给定的问题，往往可以设计出若干个算法。那么如何评价这些算法的优劣呢？首先，正确性是评价一个算法的首要条件，一个正确的算法是指在合法的数据输入下，能在有限的运行时间内得出正确的结果；此外，主要考虑执行算法所耗费的时间和执行算法所占用的存储空间。

1.3.1 空间复杂度

空间复杂度是指执行算法所需要的存储空间，包括算法本身所占用的存储空间、输入数据占用的存储空间及算法在运行过程中的工作单元和实现算法所需要的辅助空间。空间复杂度可以用 $S(n)=O(F(n))$ 表示。算法在运行过程中临时占用的辅助存储空间随算法的不同而异，有的算法只需占用少量的临时工作单元，而且不随问题规模的大小而改变；有的算法需要占用的临时工作单元数随着问题规模 n 的增大而增大，此时要按最坏情况分析。

1.3.2 时间复杂度

一个程序在计算机上运行时所耗费的时间由下列因素所决定：

- 程序运行时所需要输入的数据总量。
- 对源程序进行编译所需的时间。
- 计算机执行每条指令所需的时间。
- 程序中的指令重复执行的次数。

前三条取决于实现算法的计算机软、硬件系统；习惯上常常把语句重复执行的次数作为算法运行时间的相对量度，称为算法的时间复杂度。若解决一个问题的规模为 n ，那么算法的时间复杂度就是 n 的一个函数，通常记为 $T(n)$ ；同时，一般情况下，算法中基本操作重复执行的次数也是问题规模 n 的某个函数，通常记为 $f(n)$ ，因此，把算法的时间复杂度记为 $T(n)=O(f(n))$ 。该式表示随问题规模 n 的增大，算法执行时间的增长率和 $f(n)$ 的增长率相同，其中 $f(n)$ 和 $T(n)$ 是同数量级的函数，大写字母 O 表示 $f(n)$ 同 $T(n)$ 只差一个常数倍。时间复杂度往往不是精确的执行次数，而是估算的数量级，它着重体现的是随着问题规模 n 的增大，算法执行时间的变化趋势。

例如，在下列 3 个程序段中：

- (1) $i++; x=x+1;$
- (2) $\text{for}(i=1; i \leq n; i++) x=x+1;$
- (3) $\text{for}(i=1; i \leq n; i++)$