



普通高等教育
“十一五”国家级
规划教材

FOUNDATIONS OF COMPUTER
SOFTWARE TECHNOLOGY

计算机软件技术基础

李金 编著



普通高等教育“十一五”国家级规划教材

计算机软件技术基础

李 金 编著
刘晓胜 刘 胜 主审



机械工业出版社

本书为普通高等教育“十一五”国家级规划教材。本书以软件基础知识为中心，以提高学生的综合素质为宗旨，目的是通过有限的篇幅，使学生掌握开发应用软件所必备的基础知识、方法和技能，建立开发软件系统的总体思路。在内容取材上既注重基础，又吸收了软件技术发展的最新成果，少而精，重点突出，层次性强。

本书的主要内容包括：数据结构的基础知识和应用；计算机系统体系结构的发展和 Windows 编程的核心技术；操作系统的基本原理；软件的定义和特征、软件开发的工程化方法和测试方法；数据库管理技术和常见的数据库系统；新型数据库和数据仓库等软件新技术。

本书配有电子课件，欢迎选用本书作教材的老师登录 www.cmpedu.com 下载或发邮件到 wbj@cmpbook.com 索取。

本书可作为高等院校理工科非计算机专业的教材，也可作为计算机软件设计人员的参考用书。

图书在版编目 (CIP) 数据

计算机软件技术基础/李金编著. —北京：机械工业出版社，2008.7
普通高等教育“十一五”国家级规划教材
ISBN 978-7-111-24628-2

I. 计… II. 李… III. 软件—高等学校—教材 IV. TP31

中国版本图书馆 CIP 数据核字 (2008) 第 103751 号

机械工业出版社 (北京市百万庄大街 22 号 邮政编码 100037)
策划编辑：王保家 责任编辑：罗子超 责任校对：张晓蓉
封面设计：王洪流 责任印制：杨 曦
北京机工印刷厂印刷 (北京樱花印刷厂装订)
2008 年 9 月第 1 版第 1 次印刷
184mm × 260mm · 15 印张 · 370 千字
标准书号：ISBN 978-7-111-24628-2
定价：26.00 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换
销售服务热线电话：(010) 68326294
购书热线电话：(010) 88379639 88379641 88379643
编辑热线电话：(010) 88379727
封面无防伪标均为盗版

全国高等学校电气工程与自动化系列教材 编审委员会

主任委员 汪樾生 浙江大学

副主任委员 (按姓氏笔画排序)

王兆安 西安交通大学

王孝武 合肥工业大学

田作华 上海交通大学

刘 丁 西安理工大学

陈伯时 上海大学

郑大钟 清华大学

赵光宙 浙江大学

赵 曜 四川大学

韩雪清 机械工业出版社

委 员 (按姓氏笔画排序)

戈宝军 哈尔滨理工大学

王钦若 广东工业大学

吴 刚 中国科技大学

张纯江 燕山大学

张晓华 哈尔滨工业大学

邹积岩 大连理工大学

陈庆伟 南京理工大学

夏长亮 天津大学

萧蕴诗 同济大学

韩 力 重庆大学

熊 蕊 华中科技大学

方 敏 合肥工业大学

白保东 沈阳工业大学

张化光 东北大学

张 波 华南理工大学

杨 耕 清华大学

陈 冲 福州大学

范 瑜 北京交通大学

章 兢 湖南大学

程 明 东南大学

雷银照 北京航空航天大学

序

随着科学技术的不断进步，电气工程与自动化技术正以令人瞩目的发展速度，改变着我国工业的整体面貌。同时，对社会的生产方式、人们的生活方式和思想观念也产生了重大的影响，并在现代化建设中发挥着越来越重要的作用。随着与信息科学、计算机科学和能源科学等相关学科的交叉融合，它正在向智能化、网络化和集成化的方向发展。

教育是培养人才和增强民族创新能力的基础，高等学校作为国家培养人才的主要基地，肩负着教书育人的神圣使命。在实际教学中，根据社会需求，构建具有时代特征、反映最新科技成果的知识体系是每个教育工作者义不容辞的光荣任务。

教书育人，教材先行。机械工业出版社几十年来出版了大量的电气工程与自动化类教材，有些教材十几年、几十年长盛不衰，有着很好的基础。为了适应我国目前高等学校电气工程与自动化类专业人才培养的需要，配合各高等学校的教学改革进程，满足不同类型、不同层次的学校在课程设置上的需求，由中国机械工业教育协会电气工程及自动化学科教育委员会、中国电工技术学会高校工业自动化教育专业委员会、机械工业出版社共同发起成立了“全国高等学校电气工程与自动化系列教材编审委员会”，组织出版新的电气工程与自动化类系列教材。这类教材基于“**加强基础，削枝强干，循序渐进，力求创新**”的原则，通过对传统课程内容的整合、交融和改革，以不同的模块组合来满足各类学校特色办学的需要。并力求做到：

1. 适用性：结合电气工程与自动化类专业的培养目标、专业定位，按技术基础课、专业基础课、专业课和教学实践等环节进行选材组稿。对有的具有特色的教材采取一纲多本的方法。注重课程之间的交叉与衔接，在满足系统性的前提下，尽量减少内容上的重复。

2. 示范性：力求教材中展现的教学理念、知识体系、知识点和实施方案在本领域中具有广泛的辐射性和示范性，代表并引导教学发展的趋势和方向。

3. 创新性：在教材编写中强调与时俱进，对原有的知识体系进行实质性的改革和发展，鼓励教材涵盖新体系、新内容、新技术，注重教学理论创新和实践创新，以适应新形势下的教学规律。

4. 权威性：本系列教材的编委由长期工作在教学第一线的知名教授和学者组成。他们知识渊博，经验丰富。组稿过程严谨细致，对书目确定、主编征集、资料申报和专家评审等都有明确的规范和要求，为确保教材的高质量提供了有

力保障。

此套教材的顺利出版，先后得到全国数十所高校相关领导的大力支持和广大骨干教师的积极参与，在此谨表示衷心的感谢，并欢迎广大师生提出宝贵的意见和建议。

此套教材的出版如能在转变教学思想、推动教学改革、更新专业知识体系、创造适应学生个性和多样化发展的学习环境、培养学生的创新能力等方面收到成效，我们将会感到莫大的欣慰。

全国高等学校电气工程与自动化系列教材编审委员会

汪植生 陈永时 郑大钟

前 言

本书为普通高等教育“十一五”国家级规划教材。

在当今社会中，计算机技术的发展已经成为各行各业、各种门类科学技术发展的先导。在计算机技术的发展过程中，硬件和软件的发展相互促进，把软件的发展推向了一次又一次的高潮。处于信息技术核心地位的软件技术，在经济发展和社会进步中发挥着越来越大的作用，甚至是不可缺少的。因此，掌握计算机应用技能，成为时代对大学生素质的基本要求。作者在总结近几年来教学实践经验并结合当前教学要求的基础上，编写了本书，全书共分6章。

第1章介绍了数据结构，包括表、栈和队列等基本的线性数据结构，树、图等非线性数据结构，以及递归、排序和查找等基本的程序操作。

第2章介绍了计算机系统体系结构和 Windows 编程机制，包括客户端/服务器计算机、事件驱动的程序设计、消息循环和处理机制等。

第3章介绍了操作系统，包括操作系统的形成和发展，操作系统的作用与类型，处理器管理，存储管理，设备管理和文件管理等。

第4章介绍了软件工程，包括软件的生命周期，软件开发的工程化方法，软件的测试策略与测试方法，以及软件质量的度量等。

第5章介绍了数据库技术，包括数据库管理技术的发展历程，数据库的安全性、完整性和并发控制，常用数据库系统概述，关系数据库理论基础及关系数据库管理系统的设计等。

第6章介绍了软件新技术，包括 Internet 与 Intranet，多媒体技术，数据库技术与其他学科的有机结合所产生的新一代数据库技术，如分布式数据库、并行数据库、多媒体数据库、面向对象数据库、工程数据库和空间数据库等，以及办公自动化软件等。

各章之间在知识层次上有一定的联系，但在内容上都独立成章，可以根据实际情况安排教学。书中安排的具有可操作性的示例程序均在计算机上调试通过。

本书由哈尔滨工程大学李金教授编写，由哈尔滨工业大学刘晓胜教授和哈尔滨工程大学刘胜教授主审。

本书配有免费电子课件，欢迎选用本书作教材的老师登录 www.cmpedu.com 下载或发邮件到 wbj@cmpbook.com 索取。

由于软件技术发展很快，而作者水平有限，书中难免存在疏漏，敬请广大读者和专家批评指正。

编 者

目 录

序	
前言	
第1章 数据结构	1
1.1 绪论	1
1.1.1 数据结构产生的背景	1
1.1.2 什么是数据结构	1
1.1.3 数据结构的重要性	2
1.1.4 数据结构的基本概念和术语	3
1.1.5 算法和算法分析	4
1.2 线性数据结构	6
1.2.1 线性表的逻辑结构定义	6
1.2.2 顺序存储的线性表及其运算	6
1.2.3 链式存储的线性表及其运算	22
1.3 递归与非线性数据结构	45
1.3.1 递归	45
1.3.2 树	47
1.3.3 图	55
1.4 内部排序	70
1.4.1 内部排序简介	70
1.4.2 插入排序	71
1.4.3 快速排序	75
1.4.4 堆排序	78
1.4.5 基数排序	82
1.5 查找	86
1.5.1 基本概念	86
1.5.2 线性表查找	86
1.5.3 哈希表查找	91
习题	99
第2章 计算机系统体系结构与Windows 编程机制	101
2.1 计算机系统体系结构	101
2.1.1 批处理阶段	101
2.1.2 中心主机远程处理阶段	101
2.1.3 共享资源服务器阶段	102
2.1.4 客户端/服务器阶段	102
2.2 Windows 编程机制	105
2.2.1 面向对象的程序设计	105
2.2.2 控制和对象的概念	106
2.2.3 封装	106
2.2.4 类	106
2.2.5 继承	107
2.2.6 事件驱动的程序设计	107
2.2.7 消息循环和处理机制	108
2.2.8 事务的完整性	108
习题	109
第3章 操作系统	110
3.1 操作系统概述	110
3.1.1 操作系统的地位	110
3.1.2 操作系统的基本概念和术语	111
3.1.3 操作系统的形成和发展	112
3.1.4 操作系统的作用	113
3.1.5 现代操作系统的新特性	114
3.1.6 操作系统的类型	115
3.2 处理器管理	119
3.2.1 作业调度	119
3.2.2 进程调度	120
3.2.3 调度算法	121
3.2.4 交通控制	122
3.3 存储管理	123
3.3.1 实存储器管理技术	124
3.3.2 虚拟存储器管理技术	125
3.4 设备管理	128
3.4.1 外部设备的种类	128
3.4.2 计算机访问外设的方式	128
3.4.3 设备管理的任务	128
3.5 文件管理	129
3.5.1 文件和文件系统	129
3.5.2 文件分类	129
3.5.3 文件系统的功能	130
3.5.4 文件的逻辑组织和物理组织	130
3.5.5 文件目录	130
3.5.6 文件的共享与文件系统的安全性	131
习题	131

第4章 软件工程	132	5.1 数据库技术的重要性	150
4.1 软件的定义及软件产品的特征	132	5.2 数据库技术的基本概念	150
4.1.1 软件的定义	132	5.2.1 信息	150
4.1.2 软件产品的特征	133	5.2.2 数据	151
4.2 软件危机及软件工程学的形成	134	5.2.3 信息与数据的关系	151
4.2.1 软件开发技术的发展历程	134	5.2.4 数据处理	151
4.2.2 软件危机	134	5.2.5 数据管理	151
4.2.3 软件工程学的形成	135	5.3 数据库管理技术的发展历程	151
4.2.4 软件工程的定义及基本原则	135	5.3.1 人工管理阶段	152
4.3 软件的生命周期	136	5.3.2 文件系统阶段	153
4.3.1 问题的定义	136	5.3.3 数据库系统阶段	153
4.3.2 可行性研究	137	5.4 数据库管理系统	156
4.3.3 需求分析	137	5.5 数据库的安全与保护	156
4.3.4 规格说明书	137	5.5.1 安全性	156
4.3.5 软件设计	137	5.5.2 完整性	158
4.3.6 编码	138	5.5.3 并发控制	159
4.3.7 软件测试	138	5.5.4 数据库的恢复	161
4.3.8 软件维护	138	5.6 数据模型及数据库的基本类型	162
4.4 软件开发的工程化方法	139	5.6.1 什么是数据模型	162
4.4.1 软件开发的工程化方法简介	139	5.6.2 常见数据模型	162
4.4.2 系统流程图法	140	5.6.3 数据库的基本类型	163
4.4.3 结构化分析方法	140	5.7 常用数据库系统概述	164
4.4.4 结构化设计方法	141	5.7.1 FoxPro	164
4.4.5 结构化程序设计	142	5.7.2 Visual FoxPro	164
4.4.6 面向对象的分析方法和面向对象的 设计方法	142	5.7.3 SQL	166
4.5 软件的测试策略与测试方法	143	5.7.4 Oracle	167
4.5.1 软件的测试策略	143	5.7.5 SYBASE	168
4.5.2 软件的测试方法	143	5.7.6 Power Builder	169
4.5.3 白盒测试法	144	5.8 关系数据库理论基础及关系数据库管理 系统 FoxPro	172
4.5.4 黑盒测试法	145	5.8.1 关系数据库理论基础	172
4.6 软件开发工具与开发环境	146	5.8.2 关系数据库管理系统 FoxPro	174
4.6.1 软件开发工具	146	习题	200
4.6.2 软件开发环境	146	第6章 软件新技术	201
4.7 软件文档	147	6.1 Internet 与 Intranet	201
4.7.1 系统文档	147	6.1.1 Internet 简介	201
4.7.2 用户文档	147	6.1.2 Internet 的地址	202
4.8 软件质量的度量	148	6.1.3 Internet 的域名服务	203
4.8.1 软件质量	148	6.1.4 超文本和超媒体	203
4.8.2 软件质量的度量标准	148	6.1.5 什么是 WWW	204
4.8.3 软件质量保证	149	6.1.6 Intranet 简介	204
习题	149	6.2 多媒体技术	204
第5章 数据库技术	150	6.2.1 多媒体技术与多媒体计算机	204

6.2.2 多媒体技术的特点	205	6.4.3 数据仓库与数据库的区别	218
6.2.3 多媒体技术中的关键问题	205	6.4.4 统计分析软件包 SAS	219
6.2.4 多媒体计算机的应用	206	6.5 办公自动化	221
6.3 数据库研究和应用的新领域	207	6.5.1 群件的概念	221
6.3.1 数据库技术研究的新特点	207	6.5.2 什么是 Lotus Notes	221
6.3.2 分布式数据库	207	6.5.3 Lotus Notes 的主要特点	221
6.3.3 并行数据库	210	6.6 程序设计语言	222
6.3.4 多媒体数据库	212	6.6.1 程序设计语言的发展	222
6.3.5 面向对象数据库	213	6.6.2 Visual Basic	223
6.3.6 对象—关系数据库	214	6.6.3 Visual C ++	223
6.3.7 工程数据库	215	6.6.4 BORLAND C ++	224
6.3.8 空间数据库	216	6.6.5 MATLAB	224
6.4 数据仓库	217	6.6.6 Java	225
6.4.1 什么是数据仓库	217	6.6.7 Delphi	226
6.4.2 操作型数据与分析型数据的 区别	217	习题	226
		参考文献	228

第1章 数据结构

1.1 绪论

1.1.1 数据结构产生的背景

计算机科学和软件工程的迅速发展使计算机的应用已渗透到各个领域，有效地解决了数值计算的各种问题。

但是，解决数值计算问题的许多理论、方法和技术一般不用于解决诸如数据的分类与查找、情报检索、数据库、企业管理、系统工程、图形图像处理、人工智能及日常生活等领域的非数值处理问题。计算机若要有效地处理这些非数值问题，就需要探索新的方法和技术。“数据结构”就是为研究和解决这些非数值问题而提出的理论和方法，是解决这些问题的重要基础知识。

1.1.2 什么是数据结构

从提出一个实际问题到计算机计算出答案需要经过下列步骤：

- 1) 从实际问题中抽取出数学模型。
- 2) 设计相应的算法。
- 3) 编程、调试并求解。

抽取数学模型，就是提取操作的对象及这些操作对象之间的关系。但并不是所有的实际问题都能用抽取数学模型的方法，来看下面的3个例子。

例 1.1 学生入学情况简表（见表 1.1）。

这是一张二维表，其中每行均是一名学生的有关情况。要查找某人的情况只能用顺序检索的方法，这显然不是一个好方法。

表 1.1 学生入学情况简表

准考证号	姓名	性别	年龄	总分	来自何地
------	----	----	----	----	------

例 1.2 学校组织体系（见图 1.1）。

在这种情况下，各数据元素之间的逻辑关系不是线性的，所以应采用树形结构分级管理。

要查找某位教师，即可顺着“系—教研室—一个人”的顺序往下查找，显然，这种结构便于检索。

例 1.3 部分城市公路交通网（见图 1.2）。

因为任意两个城市之间都可能由公路相通，所以，这是一种比树形结构更复杂的数据结构，称其为图结构。

从上面的例子可以看出，描述这类问题的数学模型不再是数值方程，而是诸如表、树、

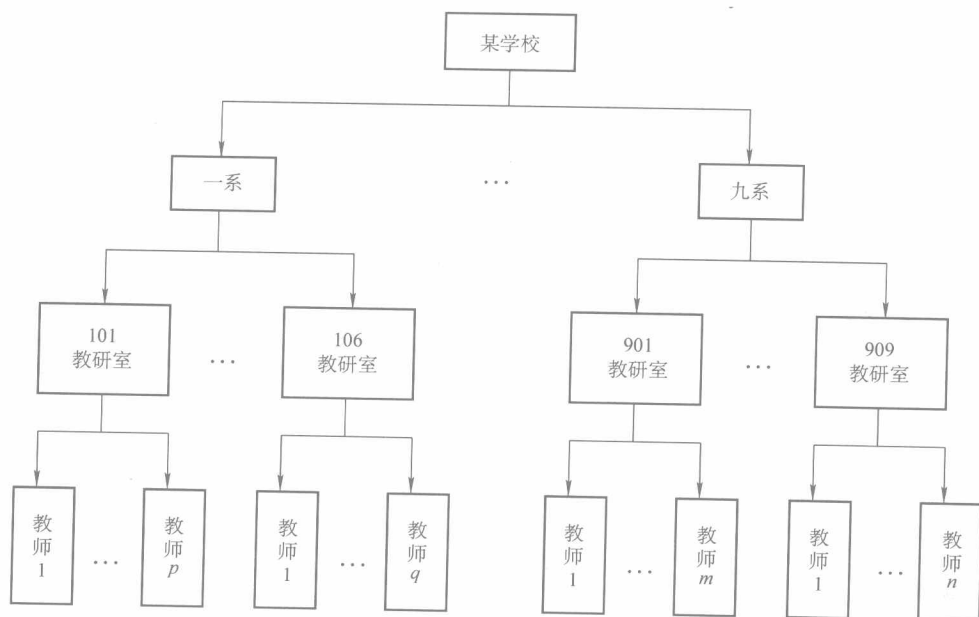


图 1.1 学校组织体系示意图

图等非数值处理的数据结构及其运算。对数据进行不同的组织，便可以形成不同的结构。

因此，简单地说，数据结构是一门研究非数值应用型程序设计中的对象，以及它们之间的关系和运算等的学科，即研究数据的逻辑结构、物理结构，适合这些结构的运算方法，以及各种结构在计算机科学中的应用等问题。

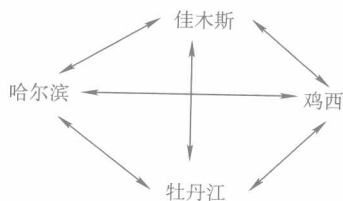


图 1.2 部分城市公路交通网

1.1.3 数据结构的重要性

程序设计语言大体可以分成两类。一类以程序为中心，这种语言侧重于建立程序。程序是在简单的数据结构上进行复杂的运算，适用于数值计算问题。

另一类以数据为中心，它把数据结构作为问题的中心部分（如数据库），而程序则围绕数据结构进行加工。它有时查询，有时对数据进行修改。这类系统要求采用复杂的数据结构来描述系统的状态。

可以说，程序设计中以数据为中心的语言对数据结构的发展起到了积极的推动作用。

近几十年来，由于社会生产力的高速发展、新技术的不断涌现和信息量的急剧膨胀，迫使人们对信息的处理和数据的利用实行自动化、网络化和社会化。

计算机在企业的管理工作中发挥了巨大的作用，如情报检索、企业管理、数据处理、数据库、网络通信和人工智能等。在这些领域中，数据结构是重要的基础和有效的手段。

计算机所处理的信息决不是杂乱无章的数据堆积，而是存在着内在联系的。只有分析清楚它们之间的内在联系并采取适当的结构，才能对数据进行行之有效的处理。因此，单纯依

靠程序设计人员的经验和技巧已不能编制出高效率的处理程序，必须对程序的设计方法进行系统的研究。这不仅涉及到程序的结构和算法，也包含了研究程序加工的对象——数据的数据结构。也就是说，为解决给定的问题而设计的程序，其有效性、清晰性和复杂性与程序中用到的数据的组织形式，即其数据结构有着密切的联系。

应该指出的是，算法和数据结构之间存在着本质的联系。因为研究某种类型的数据结构时总要讨论加于这种数据结构之上的运算（如插入、删除等），只有通过某种算法实现这些运算，数据结构才能体现出它的意义和作用；反之，当研究某种算法时也要考虑该算法处理对象所采取的数据结构。

数据结构的研究不仅涉及到计算机硬件（特别是编码理论、存储器和存取方法等）的研究范围，而且与计算机软件的研究有着更密切的联系。另外，在研究信息检索时也必须考虑如何组织数据。因此，可以认为数据结构是介于数学、计算机硬件和计算机软件三者之间的一门核心课程，如图 1.3 所示。

数据结构不仅是一般程序设计（特别是非数值应用型程序设计）的基础，而且是设计和实现编译程序、操作系统，以及其他系统程序和大型应用程序的重要基础。

近年来，数据库和人工智能的发展大大促进了数据结构的研究，但对非计算机专业的学生来说，数据结构主要侧重于实践和应用。本章列举了许多应用实例并附有上机调试通过的程序，以帮助理解和应用。

1.1.4 数据结构的基本概念和术语

1. 数据

数据是描述客观事物的数字、字符，以及所有能输入到计算机中，并被计算机程序处理的符号的集合。例如，计算所用的数据是数字，事务处理所用的数据可能既有数字也有字符，编译程序所使用的数据则是程序员编写的源程序。因此，对计算机来说，数据的含义极为广泛，如图形、图像和声音等都属于数据的范畴。

2. 数据项

数据项是数据的最小单位，是不可再分的。例如，表 1.1 中每一行的每一项如准考证号、姓名等均为数据项。

3. 数据元素

数据元素是数据的基本单位，即数据集合中的一个个体。一个数据元素可由若干个数据项组成。例如，表 1.1 中每一行为一个数据元素，它由准考证号、姓名等数据项组成。

4. 数据对象

数据对象是性质相同的数据元素的集合。例如，表 1.1 就是一个数据对象，它由每一个学生的情况组成。又如，整数的集合 $N = \{0, \pm 1, \pm 2, \dots\}$ ；一周 7 天的集合 $Date =$

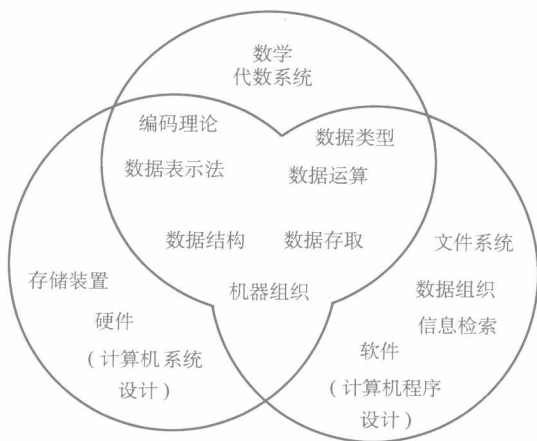


图 1.3 数据结构的地位

{Sunday, Monday, ..., Saturday}。N 和 Date 都是数据对象，而其中的单个项：0, ±1 或 Sunday, Monday 等都是数据元素，这种情况下数据元素由单个数据项组成。

5. 数据结构

通常数据对象中的元素不是孤立的，而是彼此之间存在着某种联系，数据元素相互之间的这种关系称为数据结构。

(1) 数据的逻辑结构

用计算机处理数据时，首先应该研究数据元素之间的相互关系，选用一个合适的结构将这些数据组织起来，以便于提高程序编制的质量和计算机运行的效率。数据元素之间的逻辑关系称为数据的逻辑结构。

数据的逻辑结构有多种形式，大体上分线性和非线性两种。线性结构有表、堆栈、队列和链表等；非线性结构有树和图等。对于各种逻辑结构要定义相应的运算，常见的运算有插入、删除、检索、更新和排序等。

(2) 数据的物理结构

也称为存储结构，它是数据的逻辑结构在计算机中的映像（或表示）。

由于映像的方法不同，数据元素之间的关系在计算机中有两种不同的表示，即顺序映像和非顺序映像，并由此得到两种不同的存储结构，即顺序存储结构和非顺序存储结构（也称为链式存储结构）。

顺序存储结构是把逻辑上相邻的结点存储在物理上相邻的存储单元上。

链式存储结构是给每个结点附加一个指针域，用于存储其后继结点的地址（即指针），由指针把各数据元素链接起来，故指针也就是链。后继结点可以是一个也可以是多个，也就是说链可以是单链，也可以是多链，随数据的逻辑结构而定。在链式存储结构中，由于有了指针，物理结构不一定与该数据的逻辑结构相同。

任何一个算法在编制程序之前首先要选择合适的存储结构，另一方面，同一运算在不同的存储结构中实现的方法不同。

对程序设计而言，数据的逻辑结构和物理结构是紧密相连的两个方面，因此，在有些数据结构的书中并不严格区分它们。

1.1.5 算法和算法分析

1. 算法

算法（Algorithm）是对特定问题求解步骤的一种描述，它是指令的有限序列，其中每一条指令表示一个或多个操作。

算法具有有穷性、确定性、可行性、输入和输出等特性。

2. 算法语言的说明

由于研究数据结构的目的是为了更好地进行程序设计，所以本章在讲解各种数据结构的基本运算的算法时都给出了用 Turbo C 实现的程序。

C 语言是一种结构化的语言，便于进行模块化程序设计，它在处理非数值型数据方面有较强的功能，并且，其数据类型丰富，特别是指针型变量在处理非线性的数据结构（如树、图等）方面和链式存储结构都很方便。

3. 评定良好算法的标准

前面已经讲过,研究数据结构的目的是为了能够更好地组织数据,寻求合理的存储结构和算法。因而,评定一个良好算法的标准是:

- 1) 正确性 (Correctness), 即设计或选择的算法应当能正确地反映某种需求。
- 2) 可读性 (Readability), 即结构清晰, 易懂的程序易于隐藏较多的错误而难于调试和修改。
- 3) 健壮性 (Robustness), 即当输入数据非法时, 算法也能适当地做出反应或进行处理, 而不会产生莫名其妙的输出结果。
- 4) 效率 (Efficiency), 即应考虑依据算法编制成的程序在计算机中运行时所耗费的时间。

程序在计算机上运行时所耗费的时间取决于下列因素:

- ① 程序运行时所需输入的数据总量, 即问题的规模;
- ② 计算机执行每条指令所需的时间;
- ③ 程序中的指令重复执行的次数。

前两条依赖于计算机的软、硬件资源, 如果计算机可以进行脱机输入, 则输入数据的时间可以忽略不计。因此, 习惯上常常把语句重复执行的次数, 即语句的频度 (Frequency Count), 作为算法的时间量度。

下面举例说明如何通过判定程序段中重复执行次数最多的语句的频度来估算算法的时间复杂度, 记作 $T(n)$, n 为问题的规模。

- ① $x = x + 1;$ 1
- ② FOR ($i = 1; i \leq n; i++$) $x = x + 1;$ n
- ③ FOR ($i = 1; i \leq n; i++$)
 FOR ($j = 1; j \leq n; j++$) $x = x + 1。$ n^2

其中, 语句的频度如上右列所示。

引入记号“ O ”, $O(n^2)$ 则表示当 n 较大时, 该程序的运行时间和 n^2 成正比, 或者说 $T(n)$ 的数量级和 n^2 的数量级相同。

因此, 上述 3 个程序段的时间复杂度分别为 $O(1)$ 、 $O(n)$ 和 $O(n^2)$, 分别称为常量阶、线性阶和平方阶。此外, 还可以有指数阶 $O(2^n)$ 的算法等。显然, 人们并不希望用指数阶的算法。

对于较复杂的算法, 可以将它分隔成容易估算的几个部分, 然后利用“ O ”的求和法则, 得到整个算法的时间复杂度。

例如, 若算法的两个部分的时间复杂度分别为 $T_1(n) = O(f(n))$, $T_2(n) = O(g(n))$, 而这两部分之间的关系又是平行的, 则总的时间复杂度为 $T(n) = T_1(n) + T_2(n) = O(\max f(n), g(n))$ 。

然而, 很多程序的运行时间不仅依赖于问题的规模, 而且, 随着处理的数据集的不同而不同。

例如, 有的排序算法对某些原始数据 (如由小到大排序的原始数据), 其 $T(n) = O(n)$; 而对另一些数据, 则 $T(n) = O(n^2)$ 。对这类算法, 除特别指明外, 均依据各种可能出现的数据集中最坏的情况, 来估算算法的时间复杂度。

5) 算法的存储空间需求。类似于算法的时间复杂度, 以空间复杂度作为算法所需存储空间的度量, 即在计算机中所占存储量的大小, 记作

$$S(n) = O(f(n)) \quad (1.1)$$

本教材仅对给定算法的时间复杂度做了简单的讨论。

1.2 线性数据结构

在计算机科学与应用中，数据是加工处理的主要对象，但数据并不都是作为单个的数据项来处理的。要提高计算机处理数据的效率，就必须研究数据与数据之间的组织结构以及数据在计算机内的存储形式。根据不同的组合形式，数据之间的关系可分为线性和非线性两种。数据之间的线性关系又称为表，而非线性关系又称为树、图等。

这一节主要介绍线性数据结构。

1.2.1 线性表的逻辑结构定义

在实际问题的处理过程中，经常遇到有先后顺序的数据结构，称为线性表（Linear List），它是应用最广泛且最简单的一种数据结构。

按照表中数据元素（也称为结点）在计算机内的不同存储形式，可分为线性表和链接表；按照数据元素在表中的存取方式来分，可有链、栈和队列等形式。

一个线性表可以抽象地写成

$$(a_1, a_2, \dots, a_{i-1}, a_i, a_{i+1}, \dots, a_n)$$

它是 n 个 ($n \geq 0$) 数据元素的有限序列，其中的数据元素取自某个与具体应用问题有关的集合，它可以是一个数、一个符号或其他更复杂的信息。

线性表中数据元素的个数 n 定义为线性表的长度，当 $n=0$ 时，称为空表。

例如，英文大写字母表

$$(A, B, \dots, Z)$$

是一个按字母顺序排列的线性表，共有 26 个数据元素。表中的数据元素由单个数据项——字母组成。

在稍复杂的线性表中，一个数据元素可以由若干个数据项组成，通常把这种情况下的数据元素称为记录，把含有大量记录的线性表称为文件。例如，表 1.1 构成一个文件，表中每个学生情况就是一个记录（数据元素），每条记录由“准考证号”、“姓名”、“性别”、“年龄”、“总分”和“来自何地”这几个数据项构成。

虽然线性表中的数据元素可根据实际意义的不同而不同，但同一线性表中的元素必定具有相同的特性，因此属于同一数据对象。

在线性表中， a_1 是第一个数据元素， a_n 是最后一个元素； a_{i-1} 领先于 a_i ，称 a_{i-1} 是 a_i 的直接前趋元素， a_i 是 a_{i-1} 的直接后继元素。当 $i=1, 2, \dots, n-1$ 时， a_i 有且仅有一个直接后继元素 a_{i+1} ；当 $i=2, 3, \dots, n$ 时， a_i 有且仅有一个直接前趋元素 a_{i-1} 。换句话说， a_1 没有直接前趋元素， a_n 没有直接后继元素。因此，线性表是一个线性的结构。

1.2.2 顺序存储的线性表及其运算

1. 一般的线性表

线性表在计算机内的存储方式（即线性表的物理结构）可以是顺序的和非顺序的两种。

首先介绍线性表的顺序存储结构。

线性表的顺序存储结构是用一组地址连续的存储单元按照线性表的逻辑顺序来存放数据元素的。

线性表中每个数据元素所占的位数（即结点的长度）必须是相等的。假设每个数据元素占用 c 个存储单元，并以其所占的第一个单元的存储地址作为该数据元素的存储位置，则线性表中第 $i+1$ 个数据元素的存储位置 $Loc(a_{i+1})$ 与第 i 个数据元素的存储位置 $Loc(a_i)$ 之间满足下列关系。

$$Loc(a_{i+1}) = Loc(a_i) + c \quad (1.2)$$

一般来说，线性表的第 i 个元素 a_i 的存储位置为

$$Loc(a_i) = Loc(a_1) + (i - 1) \times c \quad (1.3)$$

式中， $Loc(a_1)$ 是线性表的第一个数据元素 a_1 的存储位置，通常称作线性表的起始位置或基地址。

根据上述函数把线性表中的每个元素映射成为内存的实际地址的方法称为顺序分配或顺序映像，其特点是以元素在计算机内物理位置上的紧邻来表示线性表中数据元素之间相邻的逻辑关系，如图 1.4 所示，其中 $base$ 代表 $Loc(a_1)$ 。

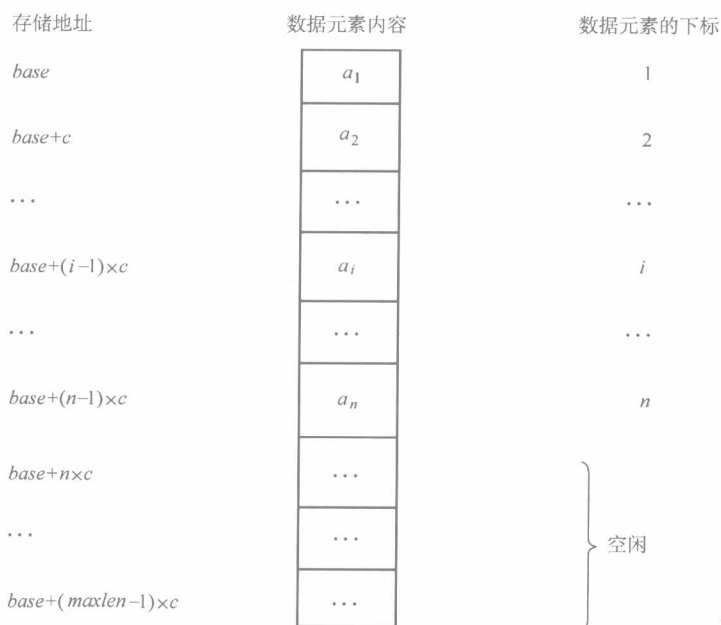


图 1.4 线性表的顺序分配示意图

因为数据元素之间的关系是线性的，故寻址函数也是线性的。任给一个 i ，根据式 (1.3) 即可很快算出 $Loc(a_i)$ 。

由此，只要确定了线性表的起始位置，线性表中任一数据元素都可随机存取，所以线性表的顺序存储结构是一种随机存取的存储结构。

由于程序设计语言中的一维数组（向量）在机内的表示也是顺序的映像，因此，可以