普通高等教育“十三五”规划教材

实用软件高级应用教程

◆李　慧　郁洪波　高明芳　等编著

中国工信出版集团
電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

普通高等教育“十三五”规划教材

实用软件高级应用教程

李　慧　郁洪波　高明芳　毕　野
樊　宁　杨　玉　孔媛媛　刘　飞　编著
张明霞　陈云平　陈加顺

電子工業出版社
Publishing House of Electronics Industry
北京 · BEIJING

内 容 简 介

本书根据教育部高等学校大学计算机课程教学指导委员会关于大学计算机基础课程教学基本要求，结合新形势下培养创新、创业型人才的需要及教学实践的具体情况编写而成。

本书以全国计算机等级考试(二级)的内容为主，以提升学生综合应用目标为辅，并扩充了计算机公共基础、计算机基础知识讲解及 Access 2010、Visio 2010 的操作指导，架构了一个完整而又实用的教材体系。

全书共分为 7 部分。第一部分为计算机公共基础；第二部分介绍了计算机基础知识；第三部分介绍了 Word 2010 文档编排；第四部分介绍了 Excel 2010 基础；第五部分介绍了 PowerPoint 2010 演示文稿制作；第六部分介绍了 Access 2010 基础；第七部分介绍了 Visio 2010 绘图操作。通过对本书的学习，可以提高学生综合应用和处理复杂办公事务的能力，并能学以致用。读者可登录华信教育资源网 www.hxedu.com.cn 免费下载本书教学资源。

本书可以作为管理、财经、信息等非计算机专业的教材或教学参考书，也可以作为办公自动化培训教材及自学考试相关科目的辅导读物，还可供有志于学习 Microsoft Office 实用技术、提高计算机操作技能的各方人士参考。

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

实用软件高级应用教程 / 李慧等编著. —北京：电子工业出版社，2019.2

ISBN 978-7-121-35399-4

I. ①实… II. ①李… III. ①电子计算机－高等学校－教材 IV. ①TP3

中国版本图书馆 CIP 数据核字(2018)第 253128 号

策划编辑：秦淑灵　杜　军
责任编辑：靳　平
印　　刷：北京京师印务有限公司
装　　订：北京京师印务有限公司
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：20.25　字数：544.6 千字
版　　次：2019 年 2 月第 1 版
印　　次：2019 年 2 月第 1 次印刷
定　　价：48.00 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)88254888，88258888。

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

本书咨询联系方式：qinshl@phei.com.cn。

前　言

随着办公自动化应用的不断推进，以及高校创新、创业教育改革的逐步深化，计算机应用已经深入各行各业，社会信息化不断向纵深发展。为了顺应社会信息化进程的发展，在大学计算机基础教育中实施分级、分类教学势在必行。本书根据教育部高等学校大学计算机课程教学指导委员会关于大学计算机基础课程教学基本要求，结合新形势下培养创新、创业型人才的需要及教学实践的具体情况编写而成，主要内容包括 Word 2010、Excel 2010、PowerPoint 2010、Access 2010 及 Visio 2010 的高级应用技术。

全书分为 7 部分，共 22 章，具体包括计算机公共基础、计算机基础知识、Word 2010 文档编排、Excel 2010 基础、PowerPoint 2010 演示文稿制作、Access 2010 基础、Visio 2010 绘图操作。其中，计算机公共基础部分介绍了数据结构与算法、程序设计基础、软件工程基础、数据库设计基础；计算机基础知识部分介绍了计算机及软/硬件系统、计算机中的数据、计算机网络；Word 2010 文档编排部分介绍了 Word 2010 文档的简单编辑、图文混排和表格、文档的高级排版、邮件合并、多文档编辑和宏；Excel 2010 基础部分主要介绍了 Excel 2010 概述、公式和函数、数据的图表化、数据的分析和处理；PowerPoint 2010 演示文稿制作部分介绍了演示文稿的创建与编辑、演示文稿整体效果的美化、演示文稿放映与共享；Access 2010 基础部分介绍了 Access 2010 概述、数据库的操作；Visio 2010 绘图操作部分介绍了 Visio 2010 的基本操作和形状、文本、图像、图表的使用等。

本书以培养计算思维能力为导向，以应用、实用、高级为主旨，采用案例驱动的方式组织内容。

本书内容紧扣全国计算机等级考试(二级)——Office 高级应用考试大纲，同时又兼顾不在大纲范围内但实际工作中难免遇到的内容。作为大学计算机公共基础教学的教材，如果仅局限于 Office 操作全国和计算机等级考试的内容又会过于简单和单一。通过大量的调研和考证，我们最终确定了以全国计算机等级考试(二级)的内容为主，以提升学生综合应用能力的内容为辅的教材体系架构，在 Office 2010 内容的基础上，又扩充了 Access 2010 及 Visio 2010 的内容。

在结构上，本书逻辑性强，以各个案例的实际处理进程为主线逐步展开，循序渐进，巧妙地将各个知识点串联起来。

在知识层次上，本书主次分明，基础知识点以归纳总结的形式一笔带过，主要介绍 Office 的高级应用部分，并对全国计算机等级考试(二级)的常考知识点和高级应用部分进行了重点讲解。

本书可以作为管理、财经、信息等非计算机专业的教材或教学参考书，也可以作为办公自动化培训教材及自学考试相关科目的辅导读物，还可供有志于学习 Office 实用技术、提高计算机操作技能的各方人士参考。

本书由淮海工学院计算机基础课程管理组策划，参与本书编著的有李慧、郁洪波、高明芳、毕野、樊宁、杨玉、孔媛媛、刘飞、张明霞、陈云平、陈加顺等老师。

因时间仓促和作者水平有限，书中难免有疏漏和不足之处，欢迎广大读者批评指正。

编著者

2018 年 10 月

此为试读，需要完整PDF请访问：www.ertongbook.com

目　录

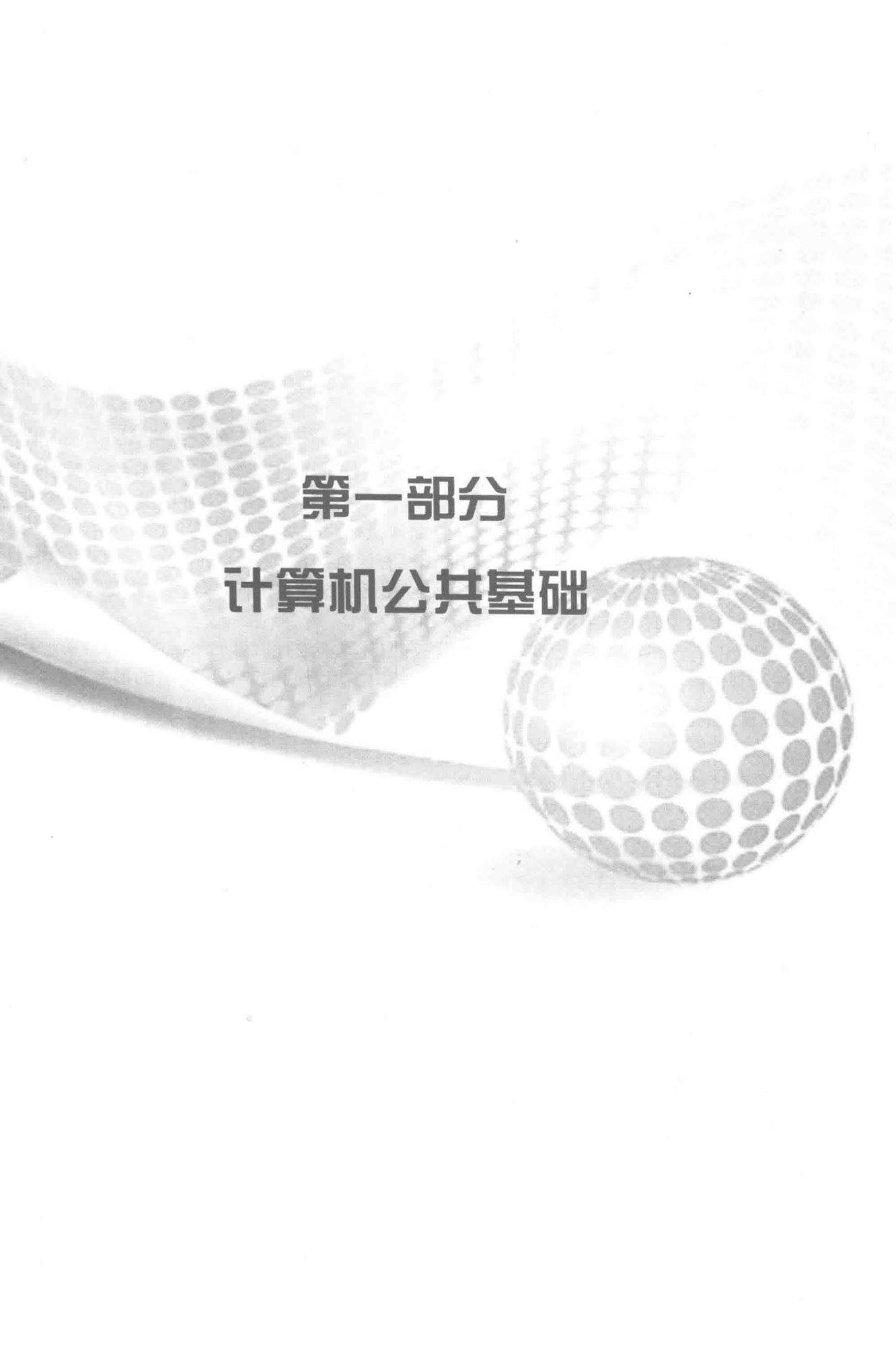
第一部分　计算机公共基础

第二部分　计算机基础知识

第三部分　Word 2010 文档编排

第六部分　Access 2010 基础

第七部分　Visio 2010 绘图操作



第一部分
计算机公共基础

第1章　数据结构与算法

1.1　算法的基本概念

1．算法的定义及特性

算法是对特定问题求解步骤的一种描述，是指令的有限序列。程序是算法在计算机中的实现。

一个算法必须满足以下5个重要特性。

(1)有限性：一个算法必须总是在执行有穷步后结束，且每一步都必须在有穷时间内完成。

(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，不会产生疑义，使算法的执行者或阅读者都能明确其含义及如何执行。

(3)可行性：算法中的每一步操作都应该能有效执行。

(4)输入：一个算法有零个或多个输入。当用函数描述算法时，输入往往是通过形参来表示，在它们被调用时，从主调函数获得输入值。

(5)输出：一个算法有一个或多个输出，它们是算法进行信息加工后得到的结果，无输出的算法没有任何意义。当用函数描述算法时，输出多用返回值或引用类型的形参表示。

2．算法的基本要素

算法的功能取决于两方面因素：选用的操作和各个操作之间的顺序。因此，一个算法通常由以下两种基本要素组成。

(1)对数据对象的运算和操作(包括算术运算、逻辑运算、关系运算、数据传输)。

(2)算法的控制结构。

3．算法设计基本方法

虽然算法设计是一件非常困难的工作，但是算法设计也不是无章可循的，人们经过实践，总结和积累了许多行之有效的方法。常用的算法设计方法有列举法、归纳法、递推法、递归法、减半递推法和回溯法6种。

4．算法设计要求

一个算法的优劣应该从以下几方面评价。

(1)正确性：不含有语法错误，对于各种合法的输入数据能够得到满足要求的结果。

(2)可读性：要求程序有较好的人机交互性，有助于人们对算法的理解。

(3)健壮性：对输入的非法数据能做出适当的响应或处理。

(4)高效率与低存储量：主要针对算法的执行时间和所需的存储空间，这两方面主要和问题的规模有关。

5．算法复杂度

一个算法复杂度的高低体现在运行该算法所需要的计算机资源的多少，所需的资源越多，

就说明该算法的复杂度越高；反之，所需的资源越少，则该算法的复杂度越低。计算机的资源，最重要的是时间和空间(即存储器)资源。因此，算法复杂度包括算法的时间复杂度和算法的空间复杂度。

1)算法的时间复杂度

算法的时间复杂度是指执行算法所需要的计算工作量。

值得注意的是，算法程序执行的具体时间和算法的时间复杂度并不一致。算法程序执行的具体时间受到所使用的计算机、程序设计语言及算法实现过程中许多细节的影响。而算法的时间复杂度与这些因素无关。

算法的计算工作量是用算法所执行的基本运算次数来度量，而算法所执行的基本运算次数是问题规模(通常用整数 n 表示)的函数，即算法的工作量为 $f(n)$，其中 n 为问题规模。

2)算法的空间复杂度

算法的空间复杂度是指在执行过程中算法所需要的内存空间。

算法执行期间所需的存储空间包括 3 个部分：输入数据所占的存储空间、程序本身所占的存储空间、算法执行过程中所需要的额外空间。

其中，额外空间包括算法程序执行过程中的工作单元及某种数据结构所需要的附加存储空间。

为了降低算法的空间复杂度，主要应减少输入数据所占的存储空间及额外空间，通常采用压缩存储技术。

1.2　数据结构的基本概念

1. 数据、数据元素、数据对象

数据：是客观事物的符号表示，是所有能输入计算机中并被计算机程序处理的符号总称。

数据元素：是数据的基本单位，可以由若干数据项组成。

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。

2. 数据结构

数据结构(Data Structure)是相互之间存在一种或多种特定关系的数据元素的集合。换句话说，数据结构是带“结构”的数据元素的集合，“结构”是指数据元素之间存在的关系。

数据结构包括逻辑结构、存储结构、数据的运算。

(1)逻辑结构：反映数据元素间的逻辑关系，包括线性结构(如线性表、栈、队列、串、数组、广义表)和非线性结构(如树、图)。

逻辑结构分类如下。

- 线性结构：每个节点有且只有一个前驱节点和一个后继节点(第一个和最后一个节点除外)。
- 树形结构：每个节点有且只有一个前驱节点(树根节点非线性结构除外)，但可以有任意多个后继节点。
- 图形结构：每个节点可以有任意多个前驱节点和任意多个后继节点。

(2)存储(又称物理)结构：反映数据元素及其关系在计算机存储器内的存储安排，包括顺序存储结构、链式存储结构、索引存储结构、散列存储结构。

此为试读，需要完整PDF请访问：www.ertongbook.com

- 顺序存储结构：将数据结构的数据元素按某种顺序存放在计算机存储器的连续存储单元中。其结构简单，存取方便，但需要连续的存储空间，当数据元素的数目不确定时，会造成存储空间的闲置，且插入与删除元素时要移动大量数据元素。
- 链式存储结构：为数据结构的每个节点附加一个数据项，其中存放一个与其相邻接的节点地址(指针)，通过指针找到下一个相关节点的实际存储地址。每个节点由数据域和指针域组成。其存储空间不必连续，在进行插入、删除操作时不必移动节点，但节点指针要占用额外的存储空间。

(3)数据的运算：对数据元素施加的操作，如插入、删除、修改、查找、排序等。

1.3 线 性 表

1. 线性表的定义

数据结构中，线性结构习惯称为线性表，线性表是最简单也是最常用的一种数据结构。

线性表是 $n(n=0)$ 个数据元素构成的有限序列，表中除第一个数据元素外的每个数据元素，有且只有一个前驱数据元素，除最后一个元素外，有且只有一个后继数据元素。

线性表要么是空表，要么可以表示为 $(a_1,a_2,\cdots,a_i,\cdots,a_n)$。

其中，$a_i(i=1,2,\cdots,n)$ 是线性表的数据元素，又称线性表的一个节点，同一线性表中的数据元素必定具有相同的特性，即属于同一数据对象。

每个数据元素的具体含义，在不同情况下各不相同，它可以是一个数或一个字符，也可以是一个具体事物，甚至其他更复杂的信息。

2. 线性表的特征

(1)数据元素个数 n 为线性表的表长，$n=0$ 时的线性表为空表。

(2)i 为 a_i 在线性表中的位序，$1<i<n$ 时，a_i 的前驱数据元素是 a_{i-1}，a_1 无前驱数据元素，a_i 的后继数据元素是 a_{i+1}，a_n 无后继数据元素。

(3)数据元素的结构相同，且不能出现缺项。

3. 线性表的顺序存储结构

通常，线性表可以采用顺序存储和链式存储两种结构。顺序存储结构是将线性表中的数据元素依次存放在一个连续的存储空间中。这种顺序表示的线性表又称顺序表。顺序存储结构如图 1-1 所示。

	1	2	3	4	5	6
data	25	34	57	16	48	09

图 1-1 顺序存储结构

顺序存储结构的特点：是随机存取的存储结构，只要确定了存储线性表的起始位置，线性表中的任一数据元素可随机存取。顺序存储结构的优点：逻辑相邻，物理相邻；可随机存取任一数据元素；存储空间使用紧凑，存储密度为 1。顺序存储结构的缺点：插入、删除操作要移动大量的数据元素；预先分配空间要按最大空间分配，利用不充分；表容量难以扩充。

4. 顺序表的插入运算(在第 i 个节点之前插入一个节点)

顺序表的插入运算是指在表的第 $i(1<i<n)$ 个位置上，插入一个新节点 x，使长度为 n 的

顺序表变成长度为 $n+1$ 的顺序表。在第 i 个节点之前插入一个新节点的操作步骤如下。

步骤 1：把原来第 i 个节点至第 n 个节点依次往后移一个节点位置。把新节点放在第 i 个位置上。

步骤 2：把新节点放在第 i 个位置上。

步骤 3：修正顺序表的节点个数。

顺序表的插入运算如图 1-2 所示。

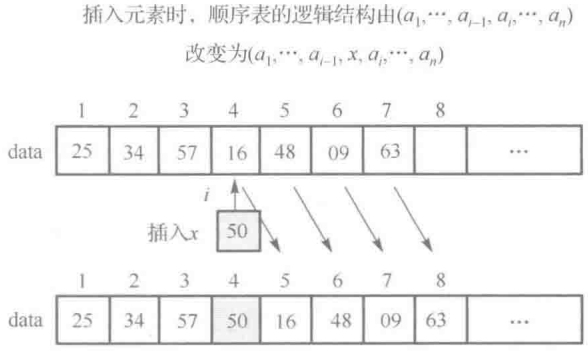


图 1-2　顺序表的插入运算

5. 顺序表的删除运算(删除第 i 个节点后面的一个节点)

顺序表的删除运算是指将表的第 $i(1<i<n)$ 个节点删除，使长度为 n 的顺序表变成长度为 $n-1$ 的顺序表。删除时应将第 $i+1$ 个节点至第 n 个节点依次向前移一个节点位置，共移动 $n-i$ 个元素，完成删除主要有以下几个步骤。

步骤 1：把第 i 个节点之后(不包含第 i 个节点)的 $n-i$ 个节点依次前移一个位置。

步骤 2：修正顺序表的节点个数。

顺序表的删除运算如图 1-3 所示。

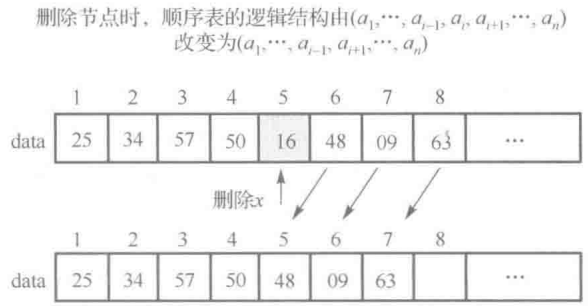


图 1-3　顺序表的删除运算

综上所述，线性表的顺序存储结构适用于小线性表，或者建立之后其中节点不常变动的线性表，而不适用于经常进行插入和删除运算的线性表和较长的线性表。

1.4　线 性 链 表

1. 线性链表的定义

所谓线性链表，就是指线性表的链接存储结构，简称链表。由于这种链表中，每个节点只有一个指针域，故又称单链表。

线性表链式存储结构的特点是可用一组不连续的存储单元存储线性表中的各个数据元素。因为存储单元不连续，数据元素之间的逻辑关系就不能依靠数据元素存储单元之间的物理关系来表示。为了表示每个数据元素与其后继数据元素之间的逻辑关系，每个数据元素除了要存储自身的信息外，还要存储一个指示其后继数据元素的信息(即后继数据元素的存储位置)。

线性表链接存储结构的基本单位称为存储节点，线性链表的一个存储节点如图 1-4 所示。每个存储节点包括以下两个组成部分。

(1)数据域：存放数据元素本身的信息。

(2)指针域：存放一个指向后继节点的指针，即存放下一个数据元素的存储地址。

图 1-4 线性链表的一个存储节点

假设一个线性表有 n 个数据元素，则这 n 个数据元素就通过所对应的 n 个节点指针链接成一个线性链表。

在线性链表中，第一个数据元素没有前驱数据元素，指向链表中的第一个节点的指针，是一个特殊的指针，称为这个链表的头指针(HEAD)。最后一个元素没有后继数据元素，因此，线性链表最后一个节点的指针域为空，用 NULL 或 0 表示。

2. 线性链表的插入运算

线性链表的插入运算是指在链式存储结构下的线性表中插入新节点。

首先，要给该新节点元素分配一个存储单元，存储单元可以从栈中取得，单链表的插入运算如图 1-5 所示。然后，将存放新元素值的节点链接到线性链表中指定的位置。

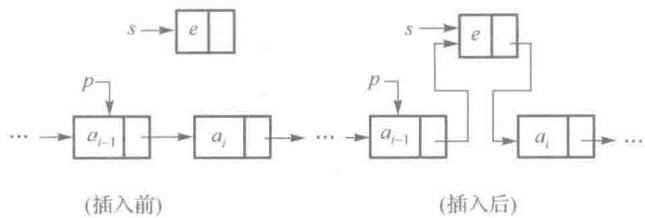


图 1-5 单链表的插入运算

在线性链表中数据域为 a_i 的节点之前插入一个新节点 e，其插入过程如下。

(1)在可利用栈的栈顶选取空闲节点，生成一个数据域为 e 的节点，将新节点的存储序号存放在指针变量 s 中。

(2)在线性链表中查找数据域为 a_i 的节点，将其前驱节点的存储序号存放在指针变量 p 中。

(3)将新节点 e 的指针变量 s 设置为指向数据域为 a_i 的节点。

(4)将指针变量 p 设置为指向新节点 e。

由于线性链表执行插入运算时，新节点的存储单元取自栈。因此，只要栈非空，线性链表总能找到存储插入的新节点，因而无须规定最大存储空间，也不会发生“上溢”的错误。此外，线性链表在执行插入运算时，无须移动节点，只要改动相关节点的指针变量即可，插入运算效率会大大提高。

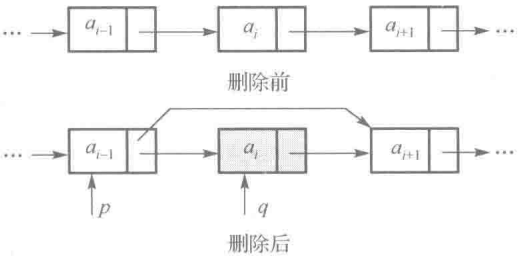
3. 线性链表的删除运算

线性链表的删除运算是指在链式存储结构下的线性表中删除指定的节点。单链表的删除运算如图 1-6 所示。在线性链表中删除数据域为 a_i 的节点，其过程如下。

(1) 在线性链表中查找包含 a_i 的节点，将该节点的存储序号存放在指针变量 q 中。

(2) 把 a_i 节点的前驱节点存储序号存放在指针变量 p 中，将 a_i 节点的后继节点存储序号存放在指针变量 q 中。

(3) 把数据域为 a_i 的节点“回收”到栈。



在单链表中删除含a_i的节点

图 1-6　单链表的删除运算

4. 循环链表

最后一个节点的指针域不为 NULL，而是指向了表的前端。其特点是：只要知道表中某一节点的地址，即可搜寻到所有其他节点的地址。

为了使空链表和非空链表的操作统一，在循环链表中往往加入头节点。循环链表的存储结构如图 1-7 所示。

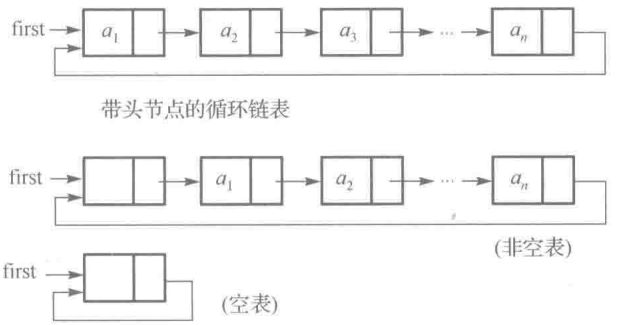


图 1-7　循环链表的存储结构

1.5　栈 和 队 列

栈和队列都是一种特殊的线性表，它们都有自己的特点，栈是“先进后出”的线性表，而队列是“先进先出”的线性表。

1. 栈的定义和特点

栈的定义：栈是一种特殊的线性表，它所有的插入与删除都限定在表的同一端进行。在栈中，一端是封闭的，既不允许插入元素，也不允许删除节点；另一端是开口的，允许插入和删除节点。

在栈中，允许插入与删除的一端称为栈顶，不允许插入与删除的另一端称为栈底。当栈中没有节点时，称为空栈。

特点：后进先出(Last In First Out，LIFO)或先进后出(First In Last Out，FILO)。

2．栈的基本运算

栈的基本运算有 3 种：入栈、出栈和读栈顶节点。

入栈：又称进栈，即在栈顶位置插入一个新节点。先将 top+1，然后将新节点插入 top 所指的位置；当 top 已指向栈存储空间的最后一个位置时，说明栈已满，若再进行进栈操作则出现“上溢”错误。

出栈：又称退栈，即取出栈顶节点并赋给一个指定的变量。先将栈顶节点赋值给一个指定的变量，然后将 top–1；当 top 为 0 时，说明栈已空，若再进行出栈操作则出现“下溢”错误。

读栈顶节点：将栈顶节点赋值给一个指定的变量，top 不变。

3．队列的定义和特点

队列也是一种特殊的线性表。队列是指允许在一端进行插入，而在另一端进行删除的线性表。

在队列中，允许进行删除运算的一端称为队头(或排头)，允许进行插入运算的一端称为队尾。习惯上将往队列的队尾插入一个节点称为入队运算，从队列的队头删除一个节点称为退队运算。若有队列：

$$Q=(q_1,q_2,\cdots,q_n)$$

那么，q_1 为队头节点(排头节点)，q_n 为队尾节点。队列中的节点是按照 $q_1, q_2, \cdots, q_n$ 的顺序进入的，退出队列也只能按照这个次序依次退出，也就是说，只有在 $q_1, q_2, \cdots, q_{n-1}$ 都退队之后，q_n 才能退出队列。因为最先进入队列的节点将最先出队，所以队列具有“先进先出”的特点，体现“先来先服务”的原则。

队头节点 q_1 是最先被插入的节点，也是最先被删除的节点。队尾节点 q_n 是最后被插入的元素，也是最后被删除的元素。因此，与栈相反，队列又称“先进先出”(First In First Out，FIFO)或“后进后出”(Last In Last Out，LILO)的线性表。

例如，火车进隧道，最先进隧道的是火车头，最后进的是火车尾，而火车出隧道的时候也是火车头先出，最后出的是火车尾。

特点：先进先出。

4．队列的基本运算

可以用顺序存储的线性表来表示队列，为了指示当前执行退队运算的队头位置，需要一个队头指针(排头指针)front，为了指示当前执行入队运算的队尾位置，需要一个队尾指针 rear。排头指针 front 总是指向队头节点的前一个位置，而队尾指针 rear 总是指向队尾节点。

队列的入队、退队原则如下。

(1) 入队时队尾指针先进一，rear = rear + 1，再将新节点按 rear 指示位置加入。

(2) 退队时队头指针先进一，front = front + 1，再将下标为 front 的节点取出。

(3) 队满时再入队将产生“溢出”错误。

(4) 队空时再退队将产生“下溢”错误。

队列的入队、退队运算如图 1-8 所示。

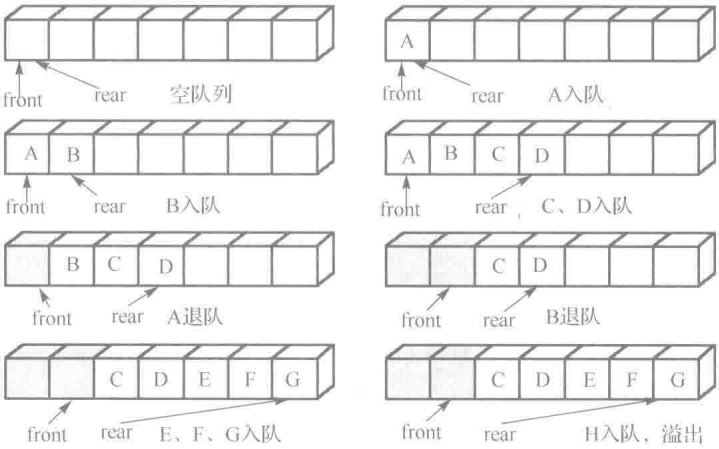


图 1-8　队列的入队、退队运算

5. 循环队列

所谓循环队列，就是将队列存储空间的最后一个位置绕到第一个位置，形成逻辑上的环状空间，供队列循环使用。

循环队列的运算规则如下。

(1) 每进行一次入队运算，队尾指针就进一，当 rear=m+1 时，置 rear=1。

(2) 每进行一次退队运算，队头指针就进一，当 front=m+1 时，置 front=1。

1.6　树与二叉树

1. 树的基本概念

树是 $n(n\geq 0)$ 个节点的有限集 T，其中：有且仅有一个特定的节点，称为树的根，当 $n>1$ 时，其余节点可分为 $m(m>0)$ 个互不相交的有限集 $T_1, T_2, \cdots, T_m$，其中每一个集合本身又是一棵树，称为根节点的子树。

节点：表示树中的元素，包括数据项及若干指向其子树的分支。

节点的度：节点拥有的子树数。

叶子：度为 0 的节点。

孩子：节点子树的根。

双亲：孩子节点的上层节点。

兄弟：同一双亲的孩子。

树的度：一棵树中最大的节点度数。

节点的层次：从根节点算起，根为第一层，它的孩子为第二层，依次类推。

深度：树中节点的最大层次数。

森林：$m(m\geq 0)$ 棵互不相交的树的集合。

2. 二叉树的定义与存储结构

1) 二叉树的定义

二叉树是 $n(n\geq 0)$ 个节点的有限集，它或为空树(n=0)，或由一个根节点和两棵分别称为左子树和右子树的互不相交的二叉树构成。树结构如图 1-9 所示。

此为试读，需要完整PDF请访问：www.ertongbook.com