



杨章静 葛修娟 朱俊
飞思教育产品研发中心

编著
监制

2009-2010考试专用



全国计算机等级考试

National Computer Rank Examination

实用应试教程

——二级公共基础知识(最新版)



光盘内容

三大系统

练习系统、考试系统、辅导系统、自动
生成试卷、自动计时、试题评析

超量题库

全真笔试题库+全真上机题库+配书
辅导题库

两大模块

笔试模式、上机模式

教学支持

提供完整电子教案



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

杨章静 葛修娟 朱俊
飞思教育产品研发中心

编著
监制

2009-2010考试专用



全国计算机等级考试

National Computer Rank Examination

实用应试教程

——二级公共基础知识(最新版)

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内容简介

本书紧扣最新版考试大纲,以高教版教程为基础,结合编者多年从事命题、阅卷及培训辅导的实际经验编写而成。章节主体部分是知识点的讲解,精讲重点与难点,并运用特殊标记对重要考点进行标识;讲解过程中穿插真题和典型例题,并给出详细的解析;章节末安排适量习题并提供解答;书中附有数套笔试模拟试卷及解析,供考生考前实战演练;书末提供 2009 年 3 月~2008 年 4 月真题及解析。

本书配有上机光盘,包含电子教案,方便培训班老师教学,还包含配书辅导软件,便于读者自学自测;另外,光盘中提供数套全真上机达标试题,上机题的整个考试过程与真实考试的完全相同,并特别增加了试题评析功能及考点速记,便于读者考前上机演练以掌握相关知识点,手把手引领考生过关。

本书具有标准、严谨、实用、高效、考点全面、考题典型和练习丰富等特点,非常适合相关考生使用,也可作为高等院校或培训班的教材。

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

图书在版编目(CIP)数据

全国计算机等级考试实用应试教程:最新版. 二级公共基础知识 / 杨章静, 葛修娟, 朱俊编著. —北京: 电子工业出版社, 2009.7

(飞思考试中心)

ISBN 978-7-121-09119-3

I. 全… II. ①杨…②葛…③朱… III. 电子计算机—水平考试—教材 IV. TP3

中国版本图书馆 CIP 数据核字 (2009) 第 103066 号

责任编辑: 王树伟 李新承

印刷: 北京四季青印刷厂

装订: 涿州市桃园装订有限公司

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: 850×1168 1/16 印张: 10.5 字数: 336 千字

印次: 2009 年 7 月第 1 次印刷

印数: 5 000 册 定价: 24.50 元 (含光盘 1 张)

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。

知己知彼 百战百胜

全国计算机等级考试是目前我国规模最大、参加人数最多的全国性计算机类水平考试，它具有权威性、公平性和广泛性，在社会上享有良好的声誉，因此，越来越多的单位把获得计算机等级考试证书作为人事录用、职称评定及职务晋升的标准之一。

为了给广大考生提供一套高效实用的标准应试教材，编者在广泛调研和充分论证的基础上，听取资深专家及众多考生的建议，组织编写了这套《全国计算机等级考试实用应试教程》，其目的是引导考生在短时间内快速过关，并为广大培训学校提供一套规范实用的应试教材。

◆ 丛书书目

丛书第一批推出 5 本：

- ◆ 全国计算机等级考试实用应试教程——一级 MS Office/B
- ◆ 全国计算机等级考试实用应试教程——二级 C 语言程序设计
- ◆ 全国计算机等级考试实用应试教程——二级公共基础知识
- ◆ 全国计算机等级考试实用应试教程——三级网络技术
- ◆ 全国计算机等级考试实用应试教程——四级网络工程师

◆ 丛书特色

- (1) 紧扣最新考试大纲，以高教版教程为基础，涵盖所有大纲规定考点。
- (2) 在全面覆盖考点的基础上，精讲重点与难点，深入分析例题，并提供实战训练。
- (3) 章节主体部分是知识点的讲解，讲解过程突出重点和难点，并运用特殊标记对重要考点进行标识；讲解过程中穿插真题和典型例题，并给出详细的解析；章节末安排适量习题并提供解答。
- (4) 在正文中提供专门章节进行上机辅导。
- (5) 在正文中提供数套模拟题，供考生考前实战演练。
- (6) 配多媒体上机盘。特点如下：
 - ◆ 登录、抽题、答题、交卷等与真实上机考试完全一致，营造逼真的考试氛围。
 - ◆ 自动生成试卷、自动计时，并特别增加了试题评析功能，便于考生自学与提高。
 - ◆ 在光盘建立题库，提供“按章节”和“按题型”两种学习方式，读者既可以在学习过程中进行同步练习，还可以在考前进行题型的强化训练。
 - ◆ 提供电子教案，方便培训班老师教学。

◆ 读者对象

本丛书以全国计算机等级考试考生为主要读者对象，特别适合于要在较短时间内取得较大收获的广大应试考生，也可作为相关考试培训班的培训教材。

◆ 关于作者

本丛书由一线教学及考试研究专家共同编写。编写人员长期从事这方面的教学和研究工作，积累了丰富的经验，对等级考试颇有研究（其中大多数编写者多年参加真题阅卷及相关培训与辅导工作）。本书由杨章静、葛修娟、朱俊编写，此外参与本丛书组织、指导、编写、审校和资料收集及光盘开发的人员有（排名不分先后）：张居晓、尹静、史国川、姚昌顺、江家宝、吴婷、乔正洪、曹冬梅、李海、葛武滇、丁为民、唐瑞华、陈海燕、陈智、何光明等，在此对他们表示衷心的感谢。

Preface

◆ 特别致谢

在此,对丛书所引用试题的出题老师和相关单位表示真诚的感谢。感谢电子工业出版社对这套书的大力支持。由于时间仓促,学识有限,书中难免有不妥之处,敬请广大读者指正。

◆ 互动交流

读者的进步,是我们的心愿。如果您对本书有疑惑或者在备考的过程中有疑难问题,请与我们交流,本套图书的互动QQ群号是:90296585,请使用本套图书的读者主动加入进行互动交流。

飞思教育产品研发中心

联系方式

咨询电话: (010) 68134545 88254161-67

电子邮件: support@fecit.com.cn

服务网址: <http://www.fecit.com.cn> <http://www.fecit.net>

通用网址: 计算机图书、飞思、飞思教育、飞思科技、FECIT

第 1 章 数据结构与算法	1
1.1 算法	2
1.1.1 算法的基本概念	2
1.1.2 算法复杂度	3
1.2 数据结构的基本概念	5
1.2.1 什么是数据结构	5
1.2.2 数据结构的图形表示	6
1.2.3 线性结构与非线性结构	6
1.3 线性表及其顺序存储结构	6
1.3.1 线性表的基本概念	6
1.3.2 线性表的顺序存储结构	7
1.3.3 顺序表的插入运算	7
1.3.4 顺序表的删除运算	8
1.4 栈和队列	9
1.4.1 栈及其基本运算	9
1.4.2 队列及其基本运算	10
1.5 线性链表	12
1.5.1 线性链表的基本概念	12
1.5.2 线性链表的基本运算	13
1.5.3 栈和队列的链式存储结构	14
1.5.4 循环链表及其基本运算	15
1.6 树与二叉树	15
1.6.1 树的基本概念	16
1.6.2 二叉树及其基本性质	17
1.6.3 二叉树的存储结构	18
1.6.4 二叉树的遍历	19
1.7 查找技术	20
1.7.1 顺序查找	20
1.7.2 二分法查找	20
1.8 排序技术	21
1.8.1 交换类排序法	21
1.8.2 插入类排序法	22
1.8.3 选择类排序法	23
1.9 典型例题分析	25
1.9.1 经典例题分析	25
1.9.2 历年真题详解	29
1.10 过关练习与答案	33
1.10.1 过关练习	33

CONTENTS

1.10.2 参考答案	36
第2章 程序设计基础	37
2.1 程序设计的方法与风格	38
2.1.1 程序设计的方法	38
2.1.2 程序设计的风格	38
2.2 结构化程序设计	39
2.2.1 结构化程序设计的原则	39
2.2.2 结构化程序的基本结构与特点	40
2.2.3 结构化程序设计原则和方法的应用	41
2.3 面向对象的程序设计	41
2.3.1 关于面向对象方法	41
2.3.2 面向对象方法的基本概念	42
2.4 典型考题分析	44
2.4.1 典型例题分析	44
2.4.2 历年真题详解	45
2.5 过关练习与答案	47
2.5.1 过关练习	47
2.5.2 参考答案	48
第3章 软件工程基础	49
3.1 软件工程基本概念	50
3.1.1 软件定义与软件特点	50
3.1.2 软件危机与软件工程	51
3.1.3 软件工程过程与软件生命周期	51
3.1.4 软件工程的目标与原则	52
3.1.5 软件开发工具与软件开发环境	53
3.2 结构化分析方法	53
3.2.1 需求分析与需求分析方法	53
3.2.2 结构化分析方法	54
3.2.3 软件需求规格说明书	57
3.3 结构化设计方法	57
3.3.1 软件设计的基本概念	57
3.3.2 概要设计	60
3.3.3 详细设计	64
3.4 软件测试	67
3.4.1 软件测试的目的	67
3.4.2 软件测试的准则	67
3.4.3 软件测试技术与方法综述	67
3.4.4 软件测试的实施	69

3.5	程序的调试	73
3.5.1	基本概念	73
3.5.2	软件调试方法	74
3.6	典型考题分析	75
3.6.1	经典例题分析	75
3.6.2	历年真题详解	78
3.7	过关练习与答案	81
3.7.1	过关练习	81
3.7.2	参考答案	82
第4章	数据库设计基础	83
4.1	数据库系统的基本概念	84
4.1.1	数据、数据库、数据库管理系统	84
4.1.2	数据库系统的发展	86
4.1.3	数据库系统的基本特点	87
4.1.4	数据库系统的内部结构体系	88
4.2	数据模型	89
4.2.1	数据模型的基本概念	89
4.2.2	E-R 模型	90
4.2.3	层次模型	92
4.2.4	网状模型	93
4.2.5	关系模型	93
4.3	关系代数	95
4.4	数据库设计与管理	99
4.4.1	数据库设计概述	99
4.4.2	数据库设计的需求分析	100
4.4.3	数据库概念设计	100
4.4.4	数据库的逻辑设计	101
4.4.5	数据库的物理设计	102
4.4.6	数据库管理	102
4.5	典型考题分析	103
4.5.1	典型例题分析	103
4.5.2	历年真题详解	105
4.6	过关练习与答案	109
4.6.1	过关练习	109
4.6.2	参考答案	111
第5章	全真模拟试题及答案	113
5.1	全真模拟试题	113
5.1.1	全真模拟试题(一)	113

CONTENTS

5.1.2	全真模拟试题(二)	114
5.1.3	全真模拟试题(三)	115
5.1.4	全真模拟试题(四)	116
5.1.5	全真模拟试题(五)	116
5.2	全真模拟试题参考解析	117
5.2.1	全真模拟试题(一) 参考答案及解析	117
5.2.2	全真模拟试题(二) 参考答案及解析	119
5.2.3	全真模拟试题(三) 参考答案及解析	120
5.2.4	全真模拟试题(四) 参考答案及解析	122
5.2.5	全真模拟试题(五) 参考答案及解析	123
第6章	二级公共基础知识速记卡	125
附录A	2009年3月真题与解析	149
附录B	2008年9月真题与解析	153
附录C	2008年4月真题与解析	157



考纲分析

通过对最近3年考试真题的分析,可知本章题量和分值比较固定。大约有4道选择题和1道填空题,占二级公共基础知识部分分值的33%。表1-1统计了最近几次考试中本章考点的分布。

表1-1 历年考题知识点分布统计表

年份	试题分布	题型	分值	考核要点
2009.3	1、2、3、4	选择题	8	线性结构与非线性结构、各种数据结构的应用和特点、计算二叉树的叶子结点、冒泡排序的比较次数
	1	填空题	2	计算栈中的元素
2008.9	1~5	选择题	10	出栈顺序、循环队列的特点、二分查找的比较次数、顺序存储结构和链式存储结构的区别、对数据流图中部件的了解
	1	填空题	2	对二叉树中序遍历的结果
2008.4	5~7	选择题	6	算法的有穷性、排序方法的特点、栈的应用
	2、3	填空题	4	计算二叉树的叶子结点个数、计算循环队列的元素个数
2007.9	6~8	选择题	6	冒泡排序、计算二叉树的结点、数据结构、存储结构、逻辑结构
	3、4	填空题	4	顺序存储结构和链式存储结构的特征、二叉树的中序遍历
2007.4	1、5、6、7	选择题	8	算法的基本概念、算法的复杂度、队列的基本知识、队列组织数据的原则、二叉树的遍历结果、计算二叉树的叶子结点数
	1	填空题	2	计算二叉树的叶子结点数
2006.9	7、8、10	选择题	6	算法的空间复杂度和时间复杂度的关联、有序线性表的顺序查找、二叉树的遍历结果
	4、5	填空题	4	组织数据的原则、线性结构和非线性结构的区分
2006.4	4~7	选择题	8	组织数据的原则、线性结构和非线性结构的区分、线性链表和线性表的关系、对二叉树后序遍历的结果、计算二叉树的叶子结点
	1	填空题	2	对线性表冒泡排序, 计算比较的次数

本章数据结构与算法的学习是很重要的部分。需要重点掌握的内容包括: 线性结构和非线性结构、栈及基本运算、队列线性列表、二叉树的特性及查找、交换类排序法。其中, 线性结构与非线性结构、栈和队列、二叉树、交换类排序是考核的重点。对于各种数据结构的应用和特点、算法的基本概念、算法复杂度等内容一般性了解即可。

1.1 算法

**考核****说明:**

本节内容比较简单,在最近两次考试中没有出现,不是考核的重点,只做一般性了解即可。

1.1.1 算法的基本概念

确切地说,算法是指解题方案准确而完整的描述。对于一个问题,如果可以通过一个计算机程序,在有限的存储空间内运行有限时间而得到正确的结果,则称这个问题是算法可解的。算法不等于程序,也不等于计算方法。

1. 算法的基本特征

(1) 可行性 (effectiveness): 针对实际问题而设计的算法,执行后能够得到满意的结果。

(2) 确定性 (definiteness): 每一条指令的含义明确,无二义性。并且在任何条件下,算法只有唯一的一条执行路径,即相同的输入只能得到相同的输出。

(3) 有穷性 (finiteness): 算法必须在有限时间内完成。有两重含义,一是算法中的操作步骤为有限个,二是每个步骤都能在有限时间内完成。

(4) 拥有足够的情报: 算法中各种运算总是要施加到各个运算对象上,而这些运算对象又可能具有某种初始状态,这就是算法执行的起点或依据。因此,一个算法执行的结果总是与输入的初始数据有关,不同的输入将会有不同的结果输出。当输入不够或输入错误时,算法将无法执行或执行有错。一般说来,当算法拥有足够的情报时,此算法才是有效的;而当提供的情报不够时,算法可能无效。

综上所述,算法是一组严谨地定义运算顺序的规则,并且每一个规则都是有效的,同时是明确的,此顺序将在有限的次数后终止。

※重点提示: 算法是解题方案的准确而完整的描述,它不等于程序,也不等于计算方法。其基本特征包括可行性、确定性、有穷性和拥有足够的情报。

2. 算法的基本要素

1) 算法中对数据的运算和操作

每个算法实际上是按解题要求,从环境能进行的所有操作中选择合适的操作所组成的一组指令序列。

基本的运算和操作有以下4类: (1) 算术运算(包括加、减、乘、除等); (2) 逻辑运算(包括“与”、“或”、“非”等运算); (3) 关系运算(包括“大于”、“小于”、“等于”、“不等于”等); (4) 数据传输(包括赋值、输入、输出等操作)。

2) 算法的控制结构

一个算法的功能不仅仅取决于所选用的操作,还与各操作之间的执行顺序有关。算法中各操作之间的执行顺序称为算法的控制结构。

算法的控制结构给出了算法的基本框架,它不仅决定了算法中各操作的执行顺序,而且也直接反映了算法的设计是否符合结构化原则。描述算法的工具通常有传统流程图、N-S 结构化流程图、算法描述语言等。算法一般由顺序、选择和循环3种基本控制结构组合而成。

※重点提示: 算法由算法中对数据的运算和操作,以及算法的控制结构组成。

3. 算法设计的基本方法

计算机算法不同于人工处理的方法,工程上常用的算法有列举法、归纳法、递推法、减半递推法和回溯法等。在实际应用时,各种方法之间往往存在着一定的联系。

(1) 列举法:其基本思想是,根据提出的问题,列举所有可能的情况,并用问题中给定的条件检测哪些是需要的,哪些是不需要的。列举法常用于解决“是否存在”或“有多少种可能”等类型的问题。

(2) 归纳法:其基本思想是,通过列举少量的特殊情况,经过分析,最后找出一般的关系。从本质上讲,归纳就是通过观察一些简单而特殊的情况,最后总结出一般性的结论。

(3) 递推法:递推是指从已知的初始条件出发,逐次推出所要求的各中间结果和最后结果。其中初始条件或问题本身已经给定,或可以通过对问题的分析与化简而确定。递推本质上属于归纳法,工程上许多递推关系式实际上是通过分析实际问题的分析与归纳而得到的,因此,递推关系式往往是归纳的结果。对于数值型的递推算法必须要注意数值计算的稳定性问题。

(4) 递归法:人们在解决一些复杂问题时,为了降低问题的复杂程度(如问题的规模等),通常将问题逐层分解,最后归结为一些最简单的问题。这种将问题逐层分解的过程,实际上并没有对问题进行求解,而只是当解决了分解的简单问题后,再沿着原来的分解的逆过程逐步进行综合,这就是递归的基本思想。递归分为直接递归与间接递归两种。

(5) 减半递推法:实际问题的复杂程度往往与问题的规模有着密切的联系。因此,利用分治法解决这类实际问题是有效的。工程上常用的分治法是减半递推技术。所谓“减半”,是指将问题的规模减半,而问题的性质不变;所谓“递推”,是指重复“减半”的过程。

(6) 回溯法:在工程上,有些实际问题很难归纳出一组简单的递推公式或直观的求解步骤,并且也不能进行无限的列举。对于这类问题,一种有效的方法是“试”。通过对问题的分析,找出一个解决问题的线索,然后沿着这个线索逐步试探,若试探成功,就得到问题的答案,若试探失败,就逐步回退,换别的路线再逐步试探。

1.1.2 算法复杂度

※重点提示: 算法复杂度主要包括时间复杂度和空间复杂度。

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。

算法执行时间需通过依据该算法编制的程序,在计算机上运行时所消耗的时间来度量,因此,这种机器消耗时间与下列因素有关:

- (1) 书写算法的程序设计语言;
- (2) 编译产生的机器语言,代码质量;
- (3) 机器执行指令的速度;
- (4) 问题的规模。

这表明使用绝对的时间单位衡量算法的效率是不合适的。撇开这些与计算机硬件、软件有关的因素,可以认为一个特定算法“运行工作量”的大小,只依赖于问题的规模(通常用整数 n 表示),它是问题的规模函数。即:

$$\text{算法的工作量} = f(n)$$

为了便于比较同一个问题的不同算法,通常以算法中基本操作重复执行的频率作为度量标准。被看

做算法基本操作的一般是最深层循环内的语句。因为算法中基本操作重复执行的频率 $T(n)$, 是问题规模 n 的某个函数 $f(n)$, 记为:

$$T(n)=O(f(n))$$

记号“ O ”读做“大 O ”。它表示随问题规模 n 的增加, 算法执行时间的增长率和 $f(n)$ 相应增加。一个没有循环算法的基本操作的执行频率与问题规模 n 无关, 记为 $O(1)$, 也称做常数阶。一个只有一重循环算法的基本操作的执行频率随问题规模 n 呈线性增大关系, 记为 $O(n)$, 也称做线性阶。其余常用的还有平方阶 $O(n^2)$, 立方阶 $O(n^3)$, 对数阶 $O(\log n)$ 和指数阶 $O(2^n)$ 等。

例如, 下面算法是用来描述两个 $n \times n$ 矩阵相乘。在这个算法中, 乘法操作、加法操作和赋值操作是矩阵相乘问题的基本操作, 这些基本操作重复执行的频率为 n^3 , 即整个算法执行时间与问题规模 n 的三次方同阶, 算法的时间复杂度为 $O(n^3)$ 。

```
void Mult_matrix( int c[][], int a[][], int b[][], int n)
{
    // a、b 和 c 均为 n 阶方阵, 且 c 是 a 和 b 的乘积
    for (i=1; i<=n; ++i)
        for (j=1; j<=n; ++j) {
            c[i,j] = 0;
            for (k=1; k<=n; ++k)
                c[i,j] += a[i,k]*b[k,j];
        }
}
```

分析算法的工作量通过采用以下两种方法来分析。

(1) 平均性态 (Average Behavior): 所谓平均性态分析, 是指用各种特定输入下基本运算次数的加权平均值来度量算法的工作量。

(2) 最坏情况复杂性 (Worst-case Complexity): 所谓最坏情况分析, 是指在规模为 n 时, 算法所执行的基本运算的最大次数。

※重点提示: 算法的时间复杂度用来衡量算法执行过程中所需要的基本运算次数。

2. 算法的空间复杂度

算法的存储量是指算法执行过程中所需的最大存储空间。算法执行期间所需要的存储量应该包括以下三部分: (1) 算法程序所占用的空间; (2) 输入的初始数据所占用的存储空间; (3) 算法执行过程中所需要的额外空间。

类似于算法的时间复杂度, 通常以算法的空间复杂度作为算法所需存储空间的量度。算法空间复杂度可定义为:

$$S(n)=O(f(n))$$

表示随着问题规模 n 的增大, 算法运行所需辅助存储量的增长率与 $f(n)$ 的增长率相同。

如果一个算法所需辅助空间都只是若干简单变量, 和问题规模无关, 这类所需额外空间相对于输入数据量来说是常量的算法, 被称做是原地工作 (in place) 的算法, 记为 $O(1)$ 。在许多实际问题中, 为了减少算法所占的存储空间, 通常采用压缩存储技术, 以便尽量减少不必要的额外空间。

※重点提示: 算法的空间复杂度用来衡量算法执行过程中所需要的存储空间。

1.2 数据结构的基本概念



考核说明： 本节主要介绍数据结构的基本概念，不是重点考核内容，在最近几次考试中有出现，稍作了解即可。

1.2.1 什么是数据结构

1. 数据结构研究的主要内容

数据结构作为计算机的一门学科，主要研究和讨论以下三个方面的问题：

- (1) 数据集中各数据元素之间所固有的逻辑关系，即数据的逻辑结构；
- (2) 在对数据进行处理时，各数据元素在计算机中的存储关系，即数据的存储结构；
- (3) 对各种数据结构进行的运算。

2. 研究数据结构的目

研究数据结构的主要目的是为了提高数据处理的效率。所谓提高数据处理的效率，主要包括两个方面：一是提高数据处理的速

3. 数据结构的定义

数据结构是指相互有关联的数据元素的集合。数据元素之间的关系可以用前后件关系（或直接前驱与直接后继关系）来描述。一个数据结构应包含以下两方面信息：

- (1) 表示数据元素的信息；
- (2) 表示各数据元素之间的前后件关系。

4. 数据的逻辑结构

数据的逻辑结构是对数据元素之间的逻辑关系的描述，它可以用一个数据元素的集合和定义在此集合中的若干关系来表示。数据的逻辑结构只抽象地反映数据元素之间的逻辑关系，即数据元素之间的前后件关系，而不管它在计算机中的存储表示形式。

数据的逻辑结构有两个要素：一是数据元素的集合，通常记为 D ；二是 D 上的关系，它反映了数据元素之间的前后件关系，通常记为 R 。一个数据结构 B 可以表示为：

$$B = (D, R)$$

例如，一年四季的数据结构可以表示成： $B = (D, R)$ ， $D = \{\text{春, 夏, 秋, 冬}\}$ ， $R = \{(\text{春, 夏}), (\text{夏, 秋}), (\text{秋, 冬})\}$ 。再如，家庭成员数据结构可以表示为： $B = (D, R)$ ， $D = \{\text{父亲, 儿子, 女儿}\}$ ， $R = \{(\text{父亲, 儿子}), (\text{父亲, 女儿})\}$ 。

5. 数据的存储结构

数据的存储结构是数据的逻辑结构在计算机存储空间中的存放形式，也称数据的物理结构。一个数据结构中的各数据元素在计算机存储空间的位置与逻辑关系有可能不同。

一种数据结构可根据需要采用不同的存储结构。常用的存储结构有顺序、链接和索引等。采用不同的存储结构，其数据处理的效率是不同的。

※重点提示： 数据的逻辑结构反映数据元素之间的逻辑关系，数据的存储结构是数据的逻辑结构在计算机存储空间中的存放形式。同一种逻辑结构的数据可以采用不同存储结构，但会影响数据处理效率。

1.2.2 数据结构的图形表示

一个数据结构除了用二元关系表示外，还可以直观地用图形表示。在数据结构的图形表示中，对于数据集合 D 中的每一个数据元素用中间标有元素值的方框表示，一般称之为数据结点，并简称为结点。为了进一步表示各数据元素之间的前后件关系，对于关系 R 中的每一个二元组，用一条有向线段从前件结点指向后件结点。例如，图 1-1 所示的是上述一年四季的数据结构和家庭成员数据结构的图形表示。

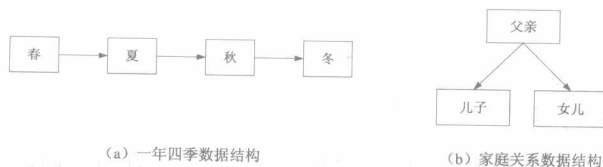


图 1-1 数据结构的图形表示

在数据结构中，没有前件的结点称为根结点；没有后件的结点称为终端结点（也称为叶子结点），例如，图 1-1 (a) 中的“春”结点是根结点，“冬”结点是终端结点；图 1-1 (b) 的中“父亲”结点是根结点，“儿子”结点和“女儿”结点是终端结点。

一个数据结构中的结点可能是在动态变化的。可以根据需要，在一个数据结构中增加新结点（称为插入运算），也可以删除数据结构中的某个结点（称为删除运算）。除此之外，对数据结构的运算还有查找、分类、合并、分解、复制和修改等。

1.2.3 线性结构与非线性结构

如果在一个数据结构中一个数据元素都没有，则称该数据结构为空的数据结构。

根据数据结构中各数据元素之间前后件关系的复杂程度，一般将数据结构分为线性结构与非线性结构两大类。

如果一个非空数据结构满足下列两个条件：(1) 有且只有一个根结点；(2) 每一个结点最多有一个前件，也最多有一个后件。则称该数据结构为线性结构，线性结构又称为线性表。例如，一年四季的数据结构就是一个线性结构。常见的线性结构有：线性表、栈与队列和线性链表。

如果一个数据结构不是线性结构，称之为非线性结构。例如，家庭成员的数据结构就是一个非线性结构。常见的非线性结构有：树、二叉树和图。

线性结构与非线性结构都可以是空的数据结构。对于空的数据结构，如果对该数据结构的运算是按线性结构的规则来处理的，则属于线性结构，否则属于非线性结构。

※重点提示：数据结构分为线性结构与非线性结构两大类。常见的线性结构有线性表、栈、队列和线性链表等；常见的非线性结构有树、二叉树和图等。

1.3 线性表及其顺序存储结构



考核

说明：

需求分析只在 08 年 9 月份出现过一次，考查的是顺序存储结构和链式存储结构的区别。从整体上讲，本节内容较为简单，不是考核的重点。

1.3.1 线性表的基本概念

线性表 (Linear List) 是最简单最常用也是最基本的一种数据结构，它是由 $n(n \geq 0)$ 个相同类型的数

据元素构成的有限序列,表示方法如下:

$$(a_1, a_2, \dots, a_i, \dots, a_n)$$

序列中数据元素的个数 n 定义为线性表的表长,当 $n=0$ 时线性表为空表,其中, i 称为线性表中的位序。

例如,英文大写字母表(A,B,C,D,E,F,...,X,Y,Z)是一个长度为26的线性表,其中的每一个大写字母就是一个数据元素。又如,同一花色的13张扑克牌(2,3,4,5,6,7,8,9,10,J,Q,K,A)是一个长度为13的线性表,其中每一张牌就是一个元素。

线性表有如下一些结构特征:

- (1) 数据元素在表中的位置由序号决定,数据元素之间的相对位置是线性的;
- (2) 对于一个非空线性表,有且只有一个根结点 a_1 ,它无前件,有且只有一个终端结点 a_n ,它无后件,除根结点与终端结点外,其他所有结点有且只有一个前件,也有且只有一个后件。

在计算机内,线性表有两种基本的存储结构:一种是顺序存储,另一种是链式存储。

1.3.2 线性表的顺序存储结构

在计算机内存放线性表,一种最简单的方法是顺序存储,也称为顺序分配。线性表的顺序存储结构具有以下两个基本特点:

- (1) 线性表中所有元素所占的存储空间是连续的;
- (2) 线性表中各数据元素在存储空间中是按逻辑顺序依次存放的。

由此可以看出,在线性表的顺序存储结构中,其前后元素在存储空间中是紧邻的,且前驱元素一定存储在后继元素的前面。

在顺序表的存储结构中,假设线性表中的第一个数据元素 a_1 的存储地址(指第一个字节的地址,即首地址)为 $Loc(a_1)$,每一个数据元素占 k 个字节,则序号为 i 的数据元素 a_i 的存储地址为:

$$Loc(a_i) = Loc(a_1) + (i-1) * k$$

图 1-2 所示的是一个长度为 n 的线性表 $(a_1, a_2, \dots, a_i, \dots, a_n)$ 在计算机中的顺序存储结构。

逻辑地址	数据元素	物理地址
	$\vdots$	
1	a_1	$Loc(a_1)$
2	a_2	$Loc(a_1) + k$
$\vdots$	$\vdots$	$\vdots$
i	a_i	$Loc(a_1) + (i-1) * k$
$\vdots$	$\vdots$	$\vdots$
n	a_n	$Loc(a_1) + (n-1) * k$
	$\vdots$	

图 1-2 线性表的顺序存储

在线性表的顺序存储结构下,可以对线性表做以下运算:(1)插入;(2)删除;(3)查找;(4)排序;(5)分解;(6)合并;(7)复制;(8)逆转等。

1.3.3 顺序表的插入运算

先通过一个例子来说明如何在顺序存储结构的线性表中插入一个新元素。图 1-3(a)所示的是一

一个长度为 8 的线性表，顺序存储在长度为 10 的存储空间中，现要求在第 2 个元素（即 30）之前插入一个新元素 83。插入过程如下：

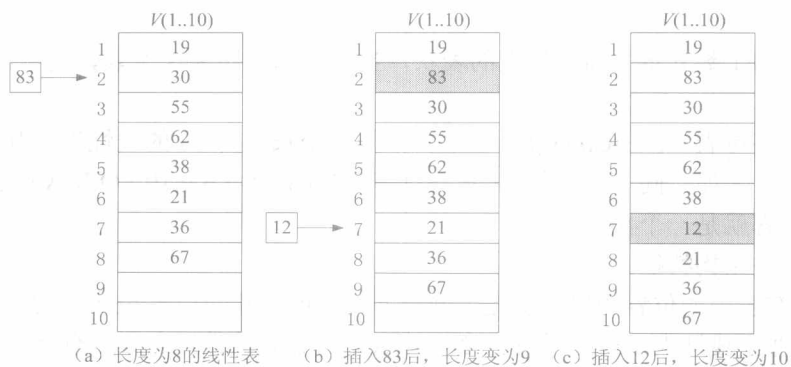


图 1-3 线性表在顺序存储下的插入

首先从最后一个元素开始直到第 2 个元素，将其中的每一个元素往后移动一个位置，然后将新元素 83 插入到第 2 个元素的位置。插入一个新元素后，线性表元素的个数变成 9 个。

如果在线性表的第 7 个元素之前插入一个新元素 12，则采用类似的方法：将第 9、8、7 个元素往后移动一个位置。将新元素 12 插入到第 7 个元素位置。插入后，线性表的长度变成了 10。现在，为线性表开辟的存储空间已经满了，不能再插入新的元素了，如果再要插入，则造成称为“上溢”的错误。

一般来说，设长度为 n 的线性表：

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

在表的第 i ($1 \leq i \leq n$) 个元素之前插入一个新结点 x ，插入后得到长度为 $n+1$ 的线性表：

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

在一般情况下，要在第 i ($1 \leq i \leq n$) 个元素之前插入一个元素，首先需从最后一个（即第 n 个）元素开始，直到第 i 个元素共 $(n-i+1)$ 个元素依次向后移动一个位置，空出第 i 个位置，然后将新元素插入到第 i 项，插入结束后，线性表的长度增长 1。所以，若顺序表中结点个数为 n ，在往每个位置插入的概率相等的情况下，插入一个结点的平均移动结点个数为 $n/2$ 。

※重点提示：顺序表的插入运算需要移动元素，在等概率情况下，平均需要移动 $n/2$ 个元素。

1.3.4 顺序表的删除运算

先通过一个例子来说明如何在顺序存储结构的线性表中删除一个元素。图 1-4 (a) 所示的是一个长度为 8 的线性表，顺序存储在长度为 10 的存储空间中，现要求删除线性表中的第 1 个元素（即删除元素 19）。删除过程如下：

从第 2 个元素开始直到最后一个元素，将其中的每一个元素均依次往前移动一个位置。此时，线性表的长度变成 7。如果再删除线性表中的第 6 个元素（即删除元素 36），则采用类似的方法，将第 7 个元素往前移动一个位置。此时，线性表的长度变成了 6。