

全国计算机等级考试

National Computer Rank Examination

2009
考试专用

优化全解



(二级 Visual FoxPro)



全国计算机等级考试命题研究中心 编著
飞思教育产品研发中心
未来教育教学与研究中心 联合监制



新 大 纲

► 超媒体教学软件

多媒体课堂精析、精讲各个考点、难点，模拟真实的教学课堂，视频讲解中演示软件环境、机考误区、解析重点、难点，把等考老师请回家。

► 模拟考试软件

模拟考场，模拟真实的考试环境、考试过程，练习、模拟考试一步到位。智能评分，自动显示错误原因，提供详细解析，提高应试技巧。

► 优化全解

考点速记+经典例题+真题汇编+全真模拟+上机指导，优化设计、全方位讲解等考核心知识点，体验高效的备考方式，让学习变得更轻松。



电子工业出版社

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

飞思考试中心
Fecit Examination Center

全国计算机等级考试

National Computer Rank Examination

优化全解

(二级Visual FoxPro)

全国计算机等级考试命题研究中心 编著
飞思教育产品研发中心 联合监制
未来教育教学与研究中心



电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内容简介

本书依据教育部考试中心最新考试大纲，结合多年考试经验和众多图书的优点，精心编制而成。

在编写过程中，一方面结合最新大纲和数套真卷，对重要考点进行了分析、总结和讲解，并选取经典例题与真题进行深入剖析；另一方面配有历年真题汇编、全真试题训练、笔试全真模拟试卷和上机考试指导，以逐步向考生详尽透析考试中的所有知识要点。可谓“一书在手，通关无忧”。

随书光盘提供了视频讲解、多媒体课堂和全真模拟考试环境，全方位学与练相结合，即“书+光盘=物超所值”。

本书适合作为全国计算机等级考试考前培训班辅导用书，也可作为应试人员的自学用书。

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优化全解. 二级 Visual FoxPro / 全国计算机等级考试命题研究中心编著.

北京: 电子工业出版社, 2008.11

(飞思考试中心)

ISBN 978-7-121-07501-8

I. 全… II. 全… III. ①电子计算机—水平考试—自学参考资料②关系数据库—数据库管理系统, Visual FoxPro—水平考试—自学参考资料 IV.TP3

中国版本图书馆 CIP 数据核字 (2008) 第 153823 号

责任编辑: 杨 鹂

印 刷: 北京市通州大中印刷厂

装 订: 三河市鹏成印业有限公司

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: 880×1230 1/16 印张: 14.25 字数: 603.2 千字

印 次: 2008 年 11 月第 1 次印刷

定 价: 29.00 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。

如何才能顺利通过计算机等级考试

“全国计算机等级考试”在各级考试中心、各级考试专家和各考点的精心培育下,现已得到社会各界的广泛认可,并有了很高的知名度和权威性。虽然计算机等级考试作为应用能力考试,难度并不高,但参加考试的都是非计算机专业的考生,其中大部分是依靠自学,因此历次考试的通过率都不高。在考生计算机基础不是很好的情况下,如何顺利通过计算机等级考试呢?本书将给出答案。

全国计算机等级考试专业研究机构——未来教育教学与研究中心联合飞思教育产品研发中心历时7年,累计对2万余名考生的备考情况进行了跟踪研究,通过对最新考试大纲、命题规律和历年真题的分析,结合考生的复习规律和备考习惯,精心设计并研发了本系列图书。

1. 考点速记、经典题解、历年真题、全真试题

- ※ 考点速记:将教材化繁为简,完全针对考试,涵盖大纲要求全部考试考点,多考多讲、少考少讲、不考不讲。
- ※ 经典题解:选择极具代表性的例题和真题,帮助考生深入理解考点内容,掌握解题技巧。
- ※ 历年真题:按考点归纳详解历年真题,准确把握真考难度、考查重点和命题规律。
- ※ 全真试题:对每章所学知识进行温习和巩固,以练促学、学练结合。

2. 笔试全真模拟、上机考试指导、最新笔试真题

- ※ 笔试全真模拟:对最新大纲、历年真题和命题规律进行研究,设计数套典型全真模拟试卷,无论从形式还是难度上都与真题类似。
- ※ 上机考试指导:揭秘上机考试特点,详细分析上机考点及解题技巧,帮助考生全面把握上机过关秘诀。
- ※ 最新笔试真题:提供最新笔试真题和详尽解析,使考生了解最新考试动向,把握考试难易程度。

3. 模考软件、考试题库、多媒体教学

- ※ 模考软件:登录、抽题、答题、交卷与真考一模一样,评分系统、评分原理与真考一模一样,让考生在真考环境下综合训练、模拟考试。
- ※ 精选题库:模拟考试系统采用考试题库试题,考试中原题出现率高,且提供详细的试题解析和参考答案,错题库、学习笔记等辅助功能亦可使复习事半功倍。
- ※ 多媒体教学:结合例题详细讲解笔试考点、要点,以视频教学的形式全面介绍上机考试的过程和易错环节,以及上机考试软件的使用方法。

本书内容编排繁简有度,同时汇集了计算机等级考试最实用的复习资料,充分体现了“优化设计 全解速学”的理念。大量考生备考实例表明:结合“3S 学习法”的优化思路,合理使用好本书及配套智能考试模拟软件,以较短的时间复习便能顺利通过计算机等级考试。

联系方式

电 话: (010)88552266 66134545 88254160

电子邮件: support@fecit.com.cn eduwin@sina.com

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

通用网址: 计算机图书、飞思、飞思教育、飞思科技、FECIT

编 著 者

丛书编委会

丁海艳	万克星	马立娟	亢艳芳
王 伟	王 亮	王强国	王 磊
卢文毅	卢继军	任海艳	伍金凤
刘之夫	刘金丽	刘春波	孙小稚
张 迪	张仪凡	张海刚	李 静
李明辉	李志红	杨 力	杨 闯
杨生喜	花 英	陈秋彤	周 辉
孟祥勇	欧海升	武 杰	范海双
郑 新	姜 涛	姜文宾	胡 杨
胡天星	赵 亮	赵东红	赵艳平
侯俊伯	倪海宇	高志军	高雪轩
董国明	谢公义	韩峻余	熊化武

目 录

第1章 数据结构与算法 1

考点1 算法	1
考点2 数据结构的基本概念	3
考点3 线性表及其顺序存储结构	4
考点4 栈和队列	5
考点5 线性链表	7
考点6 树与二叉树	8
考点7 查找技术	10
考点8 排序技术	12
历年真题汇编	13
全真试题训练	14
历年真题参考答案及解析	17
全真试题参考答案	18

第2章 程序设计基础 19

考点1 程序设计方法与风格	19
考点2 结构化程序设计	20
考点3 面向对象的程序设计	21
历年真题汇编	23
全真试题训练	23
历年真题参考答案及解析	24
全真试题参考答案	24

第3章 软件工程基础 25

考点1 软件工程基本概念	25
考点2 结构化分析方法	27
考点3 结构化设计方法	28
考点4 软件测试	30
考点5 程序的调试	32
历年真题汇编	33
全真试题训练	34
历年真题参考答案及解析	35
全真试题参考答案	36

第4章 数据库设计基础 37

考点1 数据库系统的基本概念	37
考点2 数据模型	39

考点3 关系代数	41
考点4 数据库设计与管理	42
历年真题汇编	44
全真试题训练	45
历年真题参考答案及解析	47
全真试题参考答案	48

第5章 Visual FoxPro 数据库基础 49

考点1 数据库系统	49
考点2 数据模型	51
考点3 关系模型	52
考点4 关系运算	53
考点5 项目管理器	54
考点6 向导和设计器简介	56
历年真题汇编	56
全真试题训练	57
历年真题参考答案及解析	59
全真试题参考答案	60

第6章 Visual FoxPro 程序设计基础 61

考点1 常量	61
考点2 变量	62
考点3 表达式	64
考点4 数值函数	66
考点5 字符函数	67
考点6 日期和时间函数	69
考点7 数据类型转换函数和测试函数	70
考点8 程序文件概述	72
考点9 程序基本结构	73
考点10 多模块程序设计	78
历年真题汇编	80
全真试题训练	82
历年真题参考答案及解析	88
全真试题参考答案	90

第7章 Visual FoxPro 数据库及其操作 91

考点1 数据库和表的基本概念	91
考点2 数据库的基本操作	92

考点3 建立数据库表	94	考点2 菜单设计	159
考点4 表的基本操作	96	历年真题汇编	161
考点5 索引	99	全真试题训练	161
考点6 数据完整性	101	历年真题参考答案及解析	162
考点7 多个表的同时使用	103	全真试题参考答案	163
历年真题汇编	104		
全真试题训练	105	第12章 报表的设计与应用	164
历年真题参考答案及解析	108	考点1 创建报表	164
全真试题参考答案	109	考点2 设计报表	165
		考点3 输出报表	167
第8章 关系数据库标准语言 SQL	111	历年真题汇编	168
考点1 SQL 概述	111	全真试题训练	168
考点2 SQL 的数据查询功能	112	历年真题参考答案及解析	169
考点3 SQL 操作功能	116	全真试题参考答案	169
考点4 定义功能	118		
历年真题汇编	121	第13章 应用程序的开发和生成	170
全真试题训练	125	考点1 连编应用程序	170
历年真题参考答案及解析	129	考点2 主程序设计	171
全真试题参考答案	132	历年真题汇编	172
		全真试题训练	172
第9章 查询与视图	133	历年真题参考答案及解析	173
考点1 查询	133	全真试题参考答案	173
考点2 视图	135		
历年真题汇编	136	第14章 笔试全真模拟试卷	174
全真试题训练	137	笔试全真模拟试卷(1)	174
历年真题参考答案及解析	138	笔试全真模拟试卷(2)	178
全真试题参考答案	139	笔试全真模拟试卷(3)	182
		笔试全真模拟试卷(4)	187
第10章 表单的设计与应用	140	参考答案	191
考点1 面向对象的概念	140		
考点2 Visual FoxPro 中的类	141	第15章 上机考试指导	194
考点3 事件、方法和属性	142	15.1 上机考试环境及流程	194
考点4 创建和运行表单	144	15.2 Visual FoxPro 6.0 使用简介	197
考点5 表单设计器	145	15.3 上机考试题型剖析	203
考点6 表单数据环境	146	附录	207
考点7 基本型控件	148	附录1 最新大纲专家解读	207
考点8 容器型控件	151	附录2 Visual FoxPro 常用命令	211
历年真题汇编	153	附录3 Visual FoxPro 常用函数	213
全真试题训练	154	附录4 Visual FoxPro 常用对象	214
历年真题参考答案及解析	156	附录5 Visual FoxPro 文件类型	215
全真试题参考答案	157	附录6 2008年9月笔试试卷	216
		参考答案及解析	219
第11章 菜单的设计与应用	158		
考点1 Visual FoxPro 系统菜单	158		

第1章 数据结构与算法

	考查知识点	考核几率	分值
考点1	算法	50%	2~4分
考点2	数据结构的基本概念	50%	2~4分
考点3	线性表及其顺序存储结构	50%	1~2分
考点4	栈和队列	100%	2~4分
考点5	线性链表	20%	1~2分
考点6	树与二叉树	100%	2~6分
考点7	查找技术	80%	1~2分
考点8	排序技术	80%	1~2分

考点1 算 法



考点速记

1. 算法的基本概念

算法是对解题方案的准确而完整的描述,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和明确的,此顺序将在有限的次数下终止。

(1) 算法的基本特征

- 可行性:针对实际问题而设计的算法,执行后能够得到满意的结果;
- 确定性:算法中的每一个步骤都必须有明确的定义,不允许有模棱两可的解释和多义性;
- 有穷性:算法必须在有限时间内做完,即算法必须能在执行有限个步骤之后终止;
- 拥有足够的情报: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。

(2) 算法的基本要素

算法一般由两种基本要素构成:一是对数据对象的运算和操作,二是算法的控制结构。

- 算法中对数据的运算和操作:每个算法实际上是按照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。通常基本的运算和操作有4类:算术运算、逻辑运算、关系运算和数据传输;
- 算法的控制结构:算法中各操作之间的执行顺序称为算法的控制结构。算法的功能不仅取决于所选用的操作,还与各操作之间的执行顺序有关。基本的控制结构包括顺序结构、选择结构和循环结构。

(3) 算法设计的基本方法

算法设计的基本方法有列举法、归纳法、递推法、递归法和减半递推技术。

2. 算法复杂度

算法的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。

一般情况下,算法所执行的基本运算次数是问题规模 n 的某个函数 $f(n)$,即算法的工作量 $=f(n)$,通常记作:

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

它表示随着问题规模 n 的增大,算法执行时间的增长率和 $f(n)$ 的增长率相同。

在同一个问题规模下,如果算法执行所需的基本运算次数取决于某一特定输入时,可以用两种方法来分析算法的工作量:平均性态分析和最坏情况分析。

(2) 算法的空间复杂度

算法的空间复杂度是指执行这个算法所需要的内存空间。算法执行所占用的存储空间包括算法程序所占的空间、输入的初始数据所占的存储空间,以及算法执行过程中所需要的额外空间。



经典题解

【例题】下列叙述中正确的是()。

- A) 算法的效率只与问题的规模有关,而与数据的存储结构无关
- B) 算法的时间复杂度是指执行算法所需要的计算工作量
- C) 数据的逻辑结构与存储结构是一一对应的
- D) 算法的时间复杂度与空间复杂度一定相关

解析:数据的结构,直接影响算法的选择和效率。而数据结构包括数据的逻辑结构和数据的存储结构两方面。因此,数据的逻辑结构和存储结构都影响算法的效率。因此,选项 A) 错误。

算法的时间复杂度是指算法在计算机内执行时所需时间的变量;与时间复杂度类似,空间复杂度是指算法在计算机内执行时所需存储空间的度量。因此,选项 B) 正确。

数据之间的相互关系称为逻辑结构,通常分为 4 类基本逻辑结构,即线性结构、树形结构、图状结构和网状结构。存储结构是逻辑结构在存储器中的映像,它包含数据元素的映像和关系的映像。存储结构在计算机中有两种,即顺序存储结构和链式存储结构。由此可见,逻辑结构和存储结构不是一一对应的。因此,选项 C) 错误。

有时为了提高算法的时间复杂度,而以牺牲空间复杂度为代价。但是,这两者之间没有必然的联系。因此,选项 D) 错误。

答案: B)

【真题】下列叙述中正确的是()。

【2006 年 9 月】

- A) 一个算法的空间复杂度大,则其时间复杂度也必定大
- B) 一个算法的空间复杂度大,则其时间复杂度必定小
- C) 一个算法的时间复杂度大,则其空间复杂度必定小
- D) 上述三种说法都不对

解析:算法在运行过程中所需要存储空间的大小称为算法空间复杂度;算法的时间复杂度是执行该算法所需要的计算工作量,即算法执行过程中所需要的基本运算次数。为了能比较客观地反映出一个算法的效率,在度量一个算法的工作量时,与所使用的计算机、程序设计语言及程序编制者无关,而且还与算法实现过程中的许多细节无关。但可用算法在执行过程所需基本运算的执行次数来度量算法的工作量。

答案: D)

考点2 数据结构的基本概念



考点速记

1. 数据结构的定义

数据结构是指反映数据元素之间关系的数据元素集合的表示,即数据的组织形式。

(1) 数据的逻辑结构

所谓数据的逻辑结构,是指反映数据元素之间逻辑关系的数据结构。它包括两个要素:一是数据元素集合,通常记为 D ;二是数据元素之间的关系,它反映了 D 中各数据元素之间的前后关系,通常记为 R 。于是,一个数据结构可表示为 $B = (D, R)$,其中 B 表示数据结构。

(2) 数据的存储结构

数据的逻辑结构在计算机存储空间中的存放形式称为数据的存储结构(也称数据的物理结构)。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。数据结构的存储方式有:顺序存储方法、链式存储方法、索引存储方法和散列存储方法。

2. 数据结构的图形表示

一个数据结构除了可用二元关系表示外,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。

3. 线性结构与非线性结构

根据数据结构中各数据元素之间前后关系的复杂程度,可以将数据结构分为两大类型:线性结构和非线性结构。

如果一个非空的数据结构满足下列两个条件:

- 有且只有一个根结点;
- 每个结点最多有一个直接前驱,也最多有一个直接后继。

则称该数据结构为线性结构,又称线性表。

需要注意的是,在一个线性结构中插入或删除任何一个结点后还应该是线性结构。否则,不能称为线性结构。如果一个数据结构不满足上述两个条件之一,则称之为非线性结构。



经典题解

【例题】下列叙述中正确的是()。

- A) 数据的逻辑结构与存储结构必定是一一对应的
- B) 由于计算机在存储空间上是向量式的存储结构,因此利用数据只能处理线性结构
- C) 程序设计语言中的数组一般是顺序存储结构,因此利用数组只能处理线性结构
- D) 以上说法都不对

解析:一般来说,数据的逻辑结构根据需要表示成多种存储结构。数组是数据的逻辑结构,可以用多种存储结构来表示,因此选项 B)、C) 错误。数据的逻辑结构与存储结构之间没有必然的联系。

答案: D)

【真题1】下列叙述中正确的是()。

【2007年9月】

- A) 程序执行的效率与数据的存储结构密切相关
- B) 程序执行的效率只取决于程序的控制结构
- C) 程序执行的效率只取决于所处理的数据量
- D) 以上三种说法都不对

解析: 在计算机中处理数据时,数据的存储结构对程序的执行效率影响很大,比如,在有序存储的表中查找某个数值比在无序存储的表中查找的效率要高很多。

答案: A)

【真题2】数据结构分为线性结构和非线性结构,带链的队列属于_____结构。

【2006年9月】

解析: 与栈类似,队列也是线性表,可以采用链式存储结构,所以带链的队列属于线性结构。

答案: 线性

考点3 线性表及其顺序存储结构



考点速记

1. 线性表的基本概念

线性表是由 $n(n \geq 0)$ 个数据元素 $a_1, a_2, \dots, a_n$ 组成的一个有限序列,表中的每一个数据元素,除了第一个外,有且只有一个前件,除了最后一个外,有且只有一个后件,那么线性表或是一个空表,或者也可以表示为:

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

其中 $a_i (i = 1, 2, \dots, n)$ 是数据对象的元素,通常也称为线性表的一个结点。

2. 线性表的顺序存储结构

线性表的顺序存储是指在内存中用地址连续的一块存储空间顺序存放线性表的各元素,用这种存储形式存储的线性表称其为顺序表。线性表的顺序存储结构具有以下两个基本特点:

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

由此可知,在线性表的顺序存储结构中,其前后件两个元素在存储空间中是紧邻的,且前件元素一定存储在后件元素的前面。

3. 顺序表的插入运算

设长度为 n 的线性表为:

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

若在线性表的第 j 个元素 a_j 之前插入一个新元素 b ,插入后得到长度为 $n+1$ 的线性表为:

$$(a'_1, a'_2, \dots, a'_j, a'_{j+1}, \dots, a'_n, a'_{n+1})$$

则插入前后的两线性表中的元素满足如下关系:

$$a'_j = \begin{cases} a_j & 1 \leq j \leq i-1 \\ b & j = i \\ a_{j-1} & i+1 \leq j \leq n+1 \end{cases}$$

通常,要在第 $i (1 \leq i \leq n)$ 个元素之间插入一个新元素时,首先要从最后一个(即第 n 个)元素开始,直到第 i 个元素之间共 $n-i+1$ 个元素依次向后移动一位,移动结束后,第 i 个位置就被空出,然后将新元素插入到第 i 项。插入结束后,线性表的长度增加了1。

4. 顺序表的删除运算

设长度为 n 的线性表为:

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

若删除第 j 个元素,删除后得到长度为 $n-1$ 的线性表为:

$$(a'_1, a'_2, \dots, a'_j, \dots, a'_{n-1})$$

则删除前后的两线性表中的元素满足如下关系:

$$a'_j = \begin{cases} a_j & 1 \leq j \leq i-1 \\ a_{j+1} & i+1 \leq j \leq n-1 \end{cases}$$

通常,当删除第 i ($1 \leq i \leq n$) 个元素时,则要从第 $i+1$ 个元素开始,直到第 n 个元素之间共 $n-i$ 个元素依次向前移动一位,删除结束后,线性表的长度减小了 1。



经典题解

【例题 1】下列有关顺序存储结构的叙述,不正确的是()。

- A) 存储密度大
- B) 逻辑上相邻的结点物理上不必邻接
- C) 可以通过计算机直接确定第 i 个结点的存储地址
- D) 插入、删除操作不方便

解析:顺序存储结构要求逻辑上相邻的元素物理上也相邻,所以只有选项 B) 叙述错误。

答案: B)

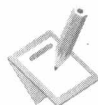
【例题 2】在一个长度为 n 的顺序表中,向第 i 个元素 ($1 \leq i \leq n+1$) 位置插入一个新元素时,需要从后向前依次移动()个元素。

- A) $n-i$
- B) i
- C) $n-i-1$
- D) $n-i+1$

解析:根据顺序表的插入运算的定义知道,在第 i 个位置上插入 x ,从 a_i 到 a_n 都要向后移动一个位置,共需要移动 $n-i+1$ 个元素。

答案: D)

考点 4 栈和队列



考点速记

1. 栈及其基本运算

(1) 栈的基本概念

栈是一种只允许在一端进行插入和删除运算的线性表,它是一种操作受限的线性表。

在栈中,允许插入与删除的一端称为栈顶(top),另一端称为栈底(bottom)。栈顶元素总是最后被插入的元素,栈底元素总是最先被插入的元素,即栈是按照“先进后出”(FILO)或“后进先出”(LIFO)的原则组织数据的,因此栈也被称为“先进后出”表,或“后进先出”表。

(2) 栈的顺序存储及其运算

- 入栈运算:入栈运算是指在栈顶位置插入一个新元素。

首先将栈顶指针加 1 (即 top 加 1),然后将新元素插入到栈顶指针指向的位置。当栈顶指针已经指向存储空间的最后一

个位置时,说明栈空间已满,不可能再进行入栈操作,这种情况称为栈“上溢”错误,如图 1.1 所示。

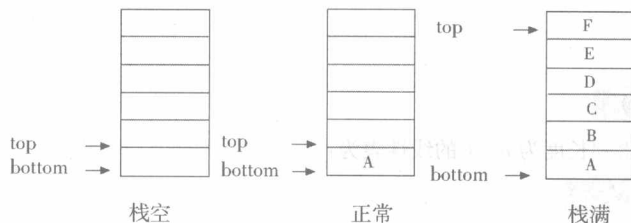


图 1.1 栈的状态

- 退栈运算:退栈是指取出栈顶元素并赋给一个指定的变量。

首先将栈顶元素(栈顶指针指向的元素)赋给一个指定的变量,然后将栈顶指针减 1(即 top 减 1)。当栈顶指针为 0 时,说明栈空,不可进行退栈操作,这种情况称为栈的“下溢”错误,如图 1.1 所示。

- 读栈顶元素:读栈顶元素是指将栈顶元素赋给一个指定的变量。

这个运算不删除栈顶元素,只是将它赋给一个变量,因此栈顶指针不会改变。当栈顶指针为 0 时,说明栈空,读不到栈顶元素。

2. 队列及其基本运算

(1) 队列的基本概念

队列是一种只允许在一端进行插入,而在另一端进行删除的线性表,它也是一种操作受限的线性表。

在队列中,允许插入的一端称为队尾,通常用一个称为尾指针(rear)的指针指向队尾元素,即尾指针总是指向最后被插入的元素;允许删除的一端称为队头,通常也是一个排头指针(front)指向排头元素的前一个位置。因此,队列又称为“先进先出”(FIFO—First In First Out)的线性表。

队列的基本结构如图 1.2 所示。

向队列的队尾插入元素称为入队运算,从队列的队头删除元素称为退队运算。

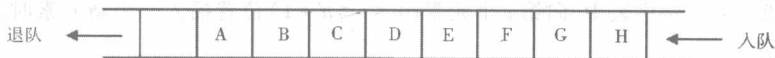


图 1.2 队列

(2) 循环队列及其运算

在实际应用中,队列的顺序存储结构通常采用循环队列的形式。

所谓循环队列,是指将队列存储空间最后一个位置绕到第一个位置,形成逻辑上的形状空间,供队列循环使用。

在循环队列中,用队尾指针(rear)指向队列的队尾元素,用排头指针(front)指向排头元素的前一个位置,因此,从排头指针(front)指向的后一个位置直到队尾指针(rear)指向的位置之间所有的元素均为队列中的元素。

循环队列的初始状态为空,即 $rear = front$ 。



经典题解

【例题 1】下列关于栈的描述中错误的是()。

- A) 栈是先进后出的线性表
- B) 栈只顺序存储
- C) 栈具有记忆作用
- D) 在对栈的插入与删除操作中,不需要改变栈底指针

解析:栈是一种特殊的线性表,该类线性表只能在固定的一端进行插入和删除操作。允许插入和删除的一端称为栈顶,另一端称为栈底。一个新元素只能从栈顶一端进入;删除时,只能删除栈顶的元素,即刚刚被插入的元素。所以栈又叫做“先进后出”表(FILO—First In Last Out)。线性表可以顺序存储,也可以链式存储,而栈是一种线性表,也可采用链式存储结构。对栈进行插入、删除操作时,栈顶指针会加 1 或减 1。

答案: B)

【例题 2】设栈 S 初始状态为空,元素 a、b、c、d、e、f 依次通过栈 S,若出栈的顺序为 c、f、e、d、b、a,则栈 S 的容量至少应该为()。

- A) 6
- B) 5
- C) 4
- D) 3

解析: 根据题中给定的条件, 可做如下模拟操作: ① 元素 a、b、c 进栈, 栈中有 3 个元素, 分别为 a、b、c; ② 元素 c 出栈后, 元素 d、e、f 进栈, 栈中有 5 个元素, 分别为 a、b、d、e、f; ③ 元素 f、e、d、b、a 出栈, 栈为空。进栈的顺序为 a、b、c、d、e、f, 出栈的顺序为 c、f、e、d、b、a, 满足题中所提出的要求。注意, 在每一次进栈操作后, 栈中最多有 5 个元素, 因此为了顺利完成这些操作, 栈的容量应至少为 5。

答案: B)

【真题 1】线性表的储存结构主要分为顺序储存结构和链式储存结构。队列是一种特殊的线性表, 循环队列是队列的 _____ 存储结构。 [2007 年 9 月]

解析: 队列的顺序存储结构通常采用循环队列的形式。循环队列, 就是将队列存储空间最后一个位置绕到第一个位置, 形成逻辑上的环状空间。

答案: 链式

【真题 2】按“先进后出”原则组织数据的数据结构是 _____。 [2006 年 9 月]

解析: 栈和队列都是操作受限的线性表, 只允许在端点处进行插入和删除。二者的区别: 栈只允许在表的一端进行插入和删除操作, 是一种“先进后出”的线性表; 而队列只允许在一端进行插入操作, 在另一端进行删除操作, 是一种“先进先出”的线性表。

答案: 栈

考点 5 线性链表



考点速记

1. 线性链表的基本概念

线性表的链式存储结构称为线性链表。假设数据结构中的每个数据结点对应于一个存储单元, 这种存储单元称为存储结点(结点)。线性链表分为单链表、双向链表和循环链表 3 种类型。

在链式存储方式中, 每个结点由两部分组成: 一部分称为数据域, 用于存放数据元素值。另一部分称为指针域, 用于存放指针。其中指针域用于指向该结点的前一个或后一个结点, 这种结构的线性链表叫做单链表。在链式存储结构中, 存储数据结构的存储空间可以不连续, 各数据结点的存储顺序与数据元素之间的逻辑关系可以不一致, 数据元素之间的逻辑关系由指针域来确定。链式存储结构既可以表示线性结构, 也可以表示非线性结构。

2. 线性链表的基本运算

线性链表主要包括以下几种运算:

- 在线性链表中包含指定元素的结点之前插入一个新元素;
- 在线性链表中删除包含指定元素的结点;
- 将两个线性链表按要求合并成一个线性链表;
- 将一个线性链表按要求进行分解;
- 逆转线性链表;
- 复制线性链表;
- 线性链表的排序;
- 线性链表的查找。

3. 循环链表及其基本运算

与线性链表相比, 循环链表具有以下两个特点:

- 在循环链表中增加了一个表头结点, 其数据域为任意或者可根据需要来设置, 指针域指向线性表第一个元素的结点,

循环链表的头指针指向表头结点;

- 循环链表中最后一个结点的指针域不为空,而是指向表头结点,即所有结点的指针构成了一个环状链。



经典题解

【例题】在一个容量为 20 的循环队列中,若头指针 $\text{front} = 3$,尾指针 $\text{rear} = 10$,则该循环队列中共有()个元素。

- (A) 7 (B) 6 (C) 8 (D) 20

解析:在循环队列中,从排头指针 front 指向的后一个位置直到队尾指针 rear 指向的位置间的所有元素均为队列中的元素,所以循环队列中元素的个数 $= \text{rear} - \text{front}$,本题中元素个数 $= 10 - 3 = 7$ 。

答案: A)

【真题】下列叙述中正确的是()。

【2006 年 4 月】

- (A) 线性链表是线性表的链式存储结构 (B) 栈与队列是非线性结构
(C) 双向链表是非线性结构 (D) 只有根结点的二叉树是线性结构

解析:根据数据结构中各数据元素之间前后关系的复杂程度,可将数据结构分为两大类型:线性结构与非线性结构。如果一个非空的数据结构满足下列两个条件:① 有且只有一个根结点;② 每个结点最多有一个前驱,也最多有一个后继,则称该数据结构为线性结构,也叫做线性表。若一个数据结构不是线性结构,则称之为非线性结构。线性表、栈与队列、线性链表都是线性结构,而二叉树是非线性结构。

答案: A)

考点 6 树与二叉树



考点速记

1. 树的基本概念

树是一种简单的非线性结构。树是由 $n(n \geq 0)$ 个结点构成的有限集合, $n = 0$ 的树称为空树;当 $n \neq 0$ 时,树中的结点应该满足以下两个条件:

- 有且仅有一个没有前驱的结点称之为根;
- 其余结点分成 $m(m > 0)$ 个互不相交的有限集合 $T_1, T_2, \dots, T_m$,其中每一个集合又都是一棵树,称 $T_1, T_2, \dots, T_m$ 为根结点的子树。

在树结构中,一个结点所拥有的后继个数称为该结点的度。

树结构具有明显的层次关系,是一种层次结构。根结点在第 1 层。同一层上所有结点的所有子结点在下一层。树的最大层次称为树的深度。

2. 二叉树及其基本性质

(1) 二叉树的定义

二叉树是一种非线性结构,是个有限元素的集合,该集合或者为空,或者由一个称为根的元素及两个不相交的、被分别称为左子树和右子树的二叉树组成。当集合为空时,称该二叉树为空二叉树。

二叉树具有以下两个特点:

- 非空二叉树只有一个根结点;
- 每一个结点最多有两棵子树,且分别称为该结点的左子树和右子树。

(2) 满二叉树和完全二叉树

满二叉树:除最后一层外,每一层上的所有结点都有两个子结点,也就是说,每一层上的结点数都达到最大值,即在满二

叉树的第 k 层上有 2^{k-1} 个结点,且深度为 m 的满二叉树中有 $2^m - 1$ 个结点。

完全二叉树:除最后一层外,每一层上的结点数都达到最大值;在最后一层上只缺少右边的若干结点。

满二叉树与完全二叉树的关系:满二叉树一定是完全二叉树,但完全二叉树不一定是满二叉树。

(3) 二叉树的主要性质

- 一棵非空二叉树的第 k 层上最多有 2^{k-1} 个结点($k \geq 1$)。
- 一棵深度为 m 的二叉树中,最多有 $2^m - 1$ 个结点。
- 对于一棵非空的二叉树,如果叶子结点数为 n_0 ,度数为 2 的结点数为 n_2 ,则有 $n_0 = n_2 + 1$ 。
- 具有 n 个结点的完全二叉树的深度 k 为 $\lceil \log_2 n \rceil + 1$ 。

3. 二叉树的存储结构

二叉树的存储结构有顺序存储结构和链式存储结构两种。二叉树通常采用链式存储结构。

与线性链表类似,用于存储二叉树中各元素的存储结点也由数据域和指针域两部分组成。但在二叉树中,由于每一个元素可以有两个后件(即两个子结点),所以用于存储二叉树的存储结点的指针域有两个:一个指向该结点的左子结点的存储地址,称为左指针域;另一个指向该结点的右子结点的存储地址,称为右指针域。

4. 二叉树的遍历

二叉树的遍历是指不重复地访问二叉树中的所有结点。

二叉树的遍历分为 3 种:前序遍历、中序遍历和后序遍历。各个遍历过程描述如下。

(1) 前序遍历(DLR)

若二叉树为空,则结束返回;否则,首先访问根结点,然后前序遍历左子树,最后前序遍历右子树。

(2) 中序遍历(LDR)

若二叉树为空,则结束返回;否则,首先中序遍历左子树,然后访问根结点,最后中序遍历右子树。

(3) 后序遍历(LRD)

若二叉树为空,则结束返回;否则,首先后序遍历左子树,然后后序遍历右子树,最后访问根结点。



经典题解

【例题 1】一棵二叉树中共有 70 个叶子结点与 80 个度为 1 的结点,则该二叉树的总结点数为()。

A) 219

B) 221

C) 229

D) 231

解析:由二叉树的性质可知,在任意一棵二叉树中,度为 0 的结点(叶子结点)总是比度为 2 的结点多一个。本题中,度为 0 的结点数为 70,因此度为 2 的结点为 69 个,再加上度为 1 的结点 80 个,总共是 219 个结点。

答案: A)

【例题 2】已知一棵二叉树先序遍历为 ABDEGCFH,中序遍历为 DBGEACHF,则该二叉树的后序遍历为()。

A) GEDHFBCA

B) DGEBHFC A

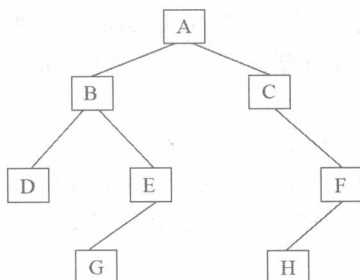
C) ABCDEFGH

D) ACBFEDHG

解析:利用先序和中序遍历相结合的方法可以确定一棵二叉树,其步骤如下:

- ① 先序遍历的第 1 个结点 A 就是树的根;
- ② 中序遍历中 A 的左边的结点为 A 的左子树,其右边结点为 A 的右子树;
- ③ 再分别对 A 的左、右子树进行上述两步处理,直到所有结点都找到正确的位置。

利用上述方法可以得到如下图所示的二叉树。



由上图中的二叉树,可得该树的后序遍历序列为 DGEBHFC A,所以选项 B)正确。

答案: B)

【真题 1】某二叉树中有 n 个度为 2 的结点,则该二叉树中的叶子结点数为()。【2007 年 4 月】

A) $n+1$

B) $n-1$

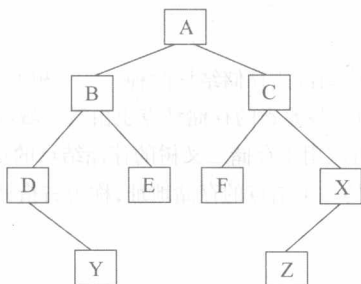
C) $2n$

D) $n/2$

解析: 由二叉树的性质可知,对于任意一棵二叉树,度为 0 的结点(即叶子结点)总是比度为 2 的结点多一个。本题中,度为 2 的结点数为 n ,故叶子结点数为 $n+1$ 个。

答案: A)

【真题 2】对下列二叉树进行前序遍历的结果为()。【2007 年 4 月】



A) DYBEAFCZX

B) YDEBFZXC A

C) ABDYECFXZ

D) ABCDEFXYZ

解析: 前序遍历首先访问根结点,然后遍历左子树,最后遍历右子树。

答案: C)

【真题 3】在深度为 7 的满二叉树中,叶子结点的个数为()。【2006 年 4 月】

A) 32

B) 31

C) 64

D) 63

解析: 所谓满二叉树是指二叉树满足:除最后一层外,每层上的所有结点都有两个子结点。也就是说,在满二叉树中,每一层上的结点数都达到最大值,即在满二叉树的第 k 层上有 2^{k-1} 个结点,且深度为 m 的满二叉树有 $2^m - 1$ 个结点。树的最大层次称为树的深度。本题中所给深度为 7,故叶子结点数为 $2^{7-1} = 64$ 。

答案: C)

考点 7 查找技术



考点速记

1. 顺序查找

顺序查找一般是指在线性表中查找指定的元素。其基本思路是:从表中的第一个元素开始,依次将线性表中的元素与被查找元素进行比较,直到两者相符,查到所要找的元素为止。否则,表中没有要找的元素,查找不成功。

在最坏的情况下,顺序查找需要比较 n 次。

程序进行了成功的测试之后进入调试阶段,程序调试是诊断和改正程序中潜在的错误。调试主要在开发阶段进行。

程序的调试活动由两部分组成,一是根据错误的迹象确定程序中错误的确切性质、原因和位置;二是对程序进行修改,排除错误。

在下列两种情况下只能采取顺序查找:

- 如果线性表中元素的排列是无序的,则无论是顺序存储结构还是链式存储结构,都只能用顺序查找;
- 即便是有序线性表,若采用链式存储结构,只能进行顺序查找。