



高等院校“十一五”规划教材

最新全国计算机等级考试

Visual FoxPro

程序设计笔试题考点与解析

主 编 陈 超 常革新 胡 斌



中国水利水电出版社
www.waterpub.com.cn

高等院校“十一五”规划教材

最新全国计算机等级考试 Visual FoxPro 程序设计笔试题考点与解析

主 编 陈 超 常革新 胡 斌



中国水利水电出版社

www.waterpub.com.cn

内 容 提 要

随着最新修订的《全国计算机等级考试二级考试大纲 (Visual FoxPro 程序设计)》的推出, 考试内容也做出了较大的调整。我们组织多年从事全国计算机等级考试教学的专家教师在对多年的考试题目深刻分析、认真研究的基础上, 编写出《最新全国计算机等级考试 Visual FoxPro 程序设计笔试题考点与解析》一书。其内容与实际笔试题题型完全一致, 可以使考生有的放矢地进行练习。

本书以最新教学大纲和考试大纲为依据, 由具有丰富教学经验的高等学校教师编写。本书所有题目均结合考试重点、难点, 由浅入深、通俗易懂。书中所有题目不仅给出了全部答案, 同时对涉及到的考点都做了非常清晰简捷的解析。所有 Visual FoxPro 习题均上机验证, 方便考生分析和掌握。本书配套上机书提供全国计算机等级考试 Visual FoxPro 程序设计笔试题题解析及上机真实环境模拟试题解析练习软件下载, 每个题目均有解析和正确答案, 方便考生进行实战演练, 可到万水书苑以及中国水利水电出版社网站下载, 网址为: <http://www.wsbookshow.com> 和 <http://www.waterpub.com.cn/softdown/>。

本书适合所有参加全国计算机等级考试 Visual FoxPro 程序设计的考生强化训练使用。本书既可作为计算机等级考试应试者的必备辅导书, 又可作为各类大中专院校学生及有关工程技术人员学习参考用书。

图书在版编目 (CIP) 数据

最新全国计算机等级考试 Visual FoxPro 程序设计笔试题考点与解析 /
陈超等主编. —北京: 中国水利水电出版社, 2009
高等院校“十一五”规划教材
ISBN 978-7-5084-6428-2

I. 最… II. 陈… III. 关系数据库—数据库管理系统,
Visual FoxPro—程序设计—高等学校—教学参考资料
IV. TP311.138

中国版本图书馆 CIP 数据核字 (2009) 第 049409 号

策划编辑: 石永峰 责任编辑: 张玉玲 加工编辑: 许波林 封面设计: 李 佳

书 名	高等院校“十一五”规划教材 最新全国计算机等级考试 Visual FoxPro 程序设计笔试题考点与解析
作 者	主 编 陈 超 常革新 胡 斌
出版 发行	中国水利水电出版社 (北京市海淀区玉渊潭南路 1 号 D 座 100038) 网址: www.waterpub.com.cn E-mail: mchannel@263.net (万水) sales@waterpub.com.cn 电话: (010) 68367658 (营销中心)、82562819 (万水)
经 售	全国各地新华书店和相关出版物销售网点
排 版	北京万水电子信息有限公司
印 刷	北京蓝空印刷厂
规 格	184mm×260mm 16 开本 22.25 印张 577 千字
版 次	2009 年 4 月第 1 版 2009 年 4 月第 1 次印刷
印 数	0001—3000 册
定 价	34.00 元

凡购买我社图书, 如有缺页、倒页、脱页的, 本社营销中心负责调换
版权所有·侵权必究

前 言

全国计算机等级考试极大地促进了各高等学校的计算机教学工作,广大学生通过参加全国计算机考试,进一步掌握了计算机的应用知识,大大提高了计算机应用能力和实际操作水平,为今后的学习、工作打下了坚实基础。全国计算机等级考试是面向全社会的计算机应用能力水平考试,深受社会各界的欢迎。

本书以最新教学大纲和考试大纲为依据,由具有丰富经验的高校教师编写。全书内容包括:公共基础知识理论考点辅导和习题解析、全国计算机等级考试 Visual FoxPro 程序设计理论考点辅导和笔试习题解析、Visual FoxPro 笔试模拟试题及解析、历年全国计算机等级考试二级 Visual FoxPro 程序设计试卷、全国计算机等级考试二级 Visual FoxPro 程序设计考试大纲五部分。

本书以通俗易懂的语言全面介绍了 Visual FoxPro 程序设计的核心点及相关知识点。在每章的安排上,先对本章的考试内容进行系统而详细的考点辅导,然后给出典型的例题解析,重在突出解题思路、传授解题方法,以便帮助读者切实领会和掌握 Visual FoxPro。书中配有大量的习题与模拟试题及解析,供考生学习、自我检测时参考。本书既可作为计算机等级考试应试者的必备辅导书,又可作为各类大中专院校学生学习计算机应用基础课程的辅导资料。

本书由陈超、常革新、胡斌任主编,王晓明、李金山、王野任副主编。其中王莉军编写了 Visual FoxPro 理论辅导和笔试习题部分的第 1、2 章;张旭编写了第 4 章;常革新编写了第 5、6 章;陈超编写了第 7、8 章;宋微编写了第 3、9 章;林英建编写了第 10、11 章。王晓明编写了笔试模拟试题及解析第 1~4 套题目;李金山编写了第 5~8 套题目;王彬编写了第 9、10 套题目。王野编写了 2005—2007 年全国计算机等级考试二级 Visual FoxPro 程序设计笔试试卷的解析内容;胡斌编写了公共基础知识理论辅导和习题部分的第 1、2、4 章;赵焱编写了第 3 章。

与本书配套的《最新全国计算机等级考试 Visual FoxPro 程序设计上机模拟试题与解析》一书同时出版,上机考试题型与全国计算机等级考试题型完全一致,并且对上机考试的各种题型作了非常详尽的解答,并提供最新全国计算机等级考试二级 Visual FoxPro 程序设计笔试考题解析和全真上机环境模拟试题套题解析练习软件及运行结果下载,供读者进行实战演练。

由于时间仓促及编者水平有限,书中疏漏和不足之处在所难免,恳请广大读者批评指正。

编 者

2009 年 1 月

目 录

前言

第一部分 公共基础知识理论辅导和习题

第1章 数据结构与算法	1
1.1 考点辅导	1
1.2 例题分析	13
思考题	17
思考题参考答案	21
第2章 程序设计基础	22
2.1 考点辅导	22
2.2 例题分析	24
思考题	25
思考题参考答案	27
第3章 软件工程基础	28
3.1 考点辅导	28
3.2 例题分析	34
思考题	38
思考题参考答案	41
第4章 数据库设计	42
4.1 考点辅导	42
4.2 例题分析	45
思考题	48
思考题参考答案	52

第二部分 Visual FoxPro 理论辅导和笔试习题

第1章 Visual FoxPro 基础	53
1.1 考点辅导	53
1.2 例题分析	56
思考题	58
思考题参考答案	60
第2章 Visual FoxPro 系统概述	61
2.1 考点辅导	61
2.2 例题分析	62
思考题	64

思考题参考答案	65
第 3 章 数据库的基本操作	66
3.1 考点辅导	66
3.2 例题分析	69
思考题	72
思考题参考答案	76
第 4 章 Visual FoxPro 基本操作	78
4.1 考点辅导	78
4.2 例题分析	81
思考题	85
思考题参考答案	89
第 5 章 关系数据库标准语言 SQL	91
5.1 考点辅导	91
5.2 例题分析	99
思考题	104
思考题参考答案	114
第 6 章 查询与视图	116
6.1 考点辅导	116
6.2 例题分析	118
思考题	122
思考题参考答案	125
第 7 章 表单设计与应用	126
7.1 考点辅导	126
7.2 例题分析	129
思考题	132
思考题参考答案	134
第 8 章 程序设计	135
8.1 考点辅导	135
8.2 例题分析	154
思考题	161
思考题参考答案	170
第 9 章 菜单设计与应用	171
9.1 考点辅导	171
9.2 例题分析	173
思考题	175
思考题参考答案	176
第 10 章 报表设计与应用	177
10.1 考点辅导	177
10.2 例题分析	178

思考题.....	181
思考题参考答案.....	182
第 11 章 应用程序开发	183
11.1 考点辅导	183
11.2 例题分析	184
思考题.....	185
思考题参考答案.....	186

第三部分 二级 Visual FoxPro 笔试模拟 试题及解析

第一套笔试模拟试题及解析	187
第二套笔试模拟试题及解析	197
第三套笔试模拟试题及解析	207
第四套笔试模拟试题及解析	216
第五套笔试模拟试题及解析	224
第六套笔试模拟试题及解析	232
第七套笔试模拟试题及解析	240
第八套笔试模拟试题及解析	249
第九套笔试模拟试题及解析	259
第十套笔试模拟试题及解析	268

第四部分 2005—2009 年二级 Visual FoxPro 笔试试题及答案

2005 年 4 月全国计算机等级考试二级笔试试卷 Visual FoxPro 数据库程序设计	277
2005 年 9 月全国计算机等级考试二级笔试试卷 Visual FoxPro 数据库程序设计	287
2006 年 4 月全国计算机等级考试二级笔试试卷 Visual FoxPro 数据库程序设计	297
2006 年 9 月全国计算机等级考试二级笔试试卷 Visual FoxPro 数据库程序设计	305
2007 年 4 月全国计算机等级考试二级笔试试卷 Visual FoxPro 数据库程序设计	314
2007 年 9 月全国计算机等级考试二级笔试试卷 Visual FoxPro 数据库程序设计	323
2008 年 4 月全国计算机等级考试二级笔试试卷 Visual FoxPro 数据库程序设计	332
2008 年 9 月全国计算机等级考试二级笔试试卷 Visual FoxPro 数据库程序设计	337
2009 年 3 月全国计算机等级考试二级笔试试卷 Visual FoxPro 数据库程序设计	341
附录 全国计算机等级考试二级 Visual FoxPro 程序设计考试大纲	347

第一部分 公共基础知识理论辅导和习题

第1章 数据结构与算法

1.1 考点辅导

1.1.1 算法

1. 算法的定义

算法是指解题方案的准确而完整的描述。算法不等于程序，也不等于计算机方法，程序的编制不可能优于算法的设计。算法的基本特征如下：

(1) 可行性。

(2) 确定性：算法中每一步骤都必须有明确定义，不允许有模棱两可的解释，不允许有多义性。

(3) 有穷性：算法必须能在有限的时间内做完，即能在执行有限个步骤后终止，包括合理的执行时间的含义。

(4) 拥有足够的情报。一般来说，当算法拥有足够的情报时，此算法才会有效，而提供的情报不够时，算法可能无效。

2. 算法的基本要素

一个算法通常由两种基本要素组成：一是对数据对象的运算和操作；二是算法的控制结构。

基本运算和操作包括：算术运算、逻辑运算、关系运算、数据传输。

算法的控制结构包括：顺序结构、选择结构、循环结构。

算法设计基本方法包括：列举法、归纳法、递推、递归、减半递推技术、回溯法。

3. 算法的复杂度

算法的复杂度主要包括时间复杂度和空间复杂度。

算法的时间复杂度：是指执行算法所需要的计算工作量（或算法执行过程中所需要的基本运算次数）。

算法的空间复杂度：是指执行这个算法所需要的内存空间。

【例1】算法的空间复杂度是指（ ）。

A. 算法程序的长度

B. 算法程序中的指令条数

C. 算法程序所占用的存储空间

D. 算法执行过程中所需要的存储空间

解析：本题属于基础知识性题目。算法的空间复杂度是指执行这个算法所需要的内存空间。

答案：D

【例2】下列叙述中正确的是（ ）。

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：算法的复杂度包括时间复杂度和空间复杂度。算法时间复杂度是指执行算法所需要的计算工作量；算法空间复杂度是指执行这个算法所需要的内存空间。A、B、C 三种说法都不对。

答案：D

1.1.2 数据结构的基本概念

1. 什么是数据结构

数据结构是研究数据元素及其之间相互关系和数据运算的一门学科。

数据结构研究的 3 个方面包括：

- (1) 数据集合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。

一般情况下，在具有相同特征的数据元素集合中，各个数据元素之间存在着某种关系，这种关系反映了该集合中的数据元素所固有的一种结构。在数据处理领域中，通常把数据元素之间这种固有的关系简单地用前后件关系（或直接前驱与直接后继关系）来描述。

例如，在数据集合{春,夏,秋,冬}中，“春”是“夏”的前件，而“夏”是“春”的后件。同样，“夏”是“秋”的前件，“秋”是“夏”的后件；“秋”是“冬”的前件，“冬”是“秋”的后件。

2. 数据的逻辑结构

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。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 D 中各数据元素之间的前后件关系，通常记为 R，即一个数据结构可以表示成

$$B=(D,R)$$

其中，B 表示数据结构。为了反映 D 中各数据元素之间的前后件关系，一般用二元组来表示。

例如，一年四季的数据结构可以表示成

$$\begin{aligned} B &= (D, R) \\ D &= \{\text{春, 夏, 秋, 冬}\} \\ R &= \{(\text{春, 夏}), (\text{夏, 秋}), (\text{秋, 冬})\} \end{aligned}$$

例如，家庭成员数据结构可以表示成

$$\begin{aligned} B &= (D, R) \\ D &= \{\text{父亲, 儿子, 女儿}\} \\ R &= \{(\text{父亲, 儿子}), (\text{父亲, 女儿})\} \end{aligned}$$

3. 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）。

一般来说，一种数据结构根据需要可以表示成多种存储结构。常用的存储结构有顺序、链接、索引等，而采用不同的存储结构，其数据处理的效率是不同的；一个数据结构中的各数据元素在计算机存储空间中的位置关系与逻辑关系是有可能不同的。

在数据的存储结构中，不仅要存放各数据元素的信息，还需要存放各数据元素之间的前后件关系的信息。

在链式存储结构中,存储数据结构的存储空间可以是连续的,也可以是不连续的,各数据节点的存储顺序与数据元素之间的逻辑关系可以不一致。

4. 数据结构的图形表示

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,一般称之为数据节点,并简称为节点;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节点指向后件节点。

例如,一年四季的数据结构可以用图 1-1 所示的图形来表示。

又如,反映家庭成员间辈份关系的数据结构可以用图 1-2 所示的图形表示。

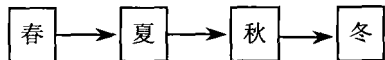


图 1-1 一年四季数据结构的图形表示

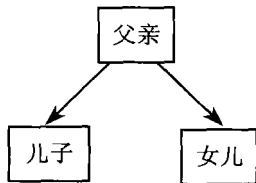


图 1-2 家庭成员辈份关系数据结构的图形表示

在数据结构中,没有前件的节点称为根节点;没有后件的节点称为终端节点(也称叶子节点);除了根节点和终端节点之外的其他节点称为内部节点。

例如,在一年四季的结构中,元素“春”节点为根节点,“冬”节点为终端节点,“夏”和“秋”为内部节点。在家庭成员的结构中,元素“父亲”为根节点,元素“儿子”和“女儿”为终端节点。

5. 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。

一般将数据结构分为两大类型:线性结构和非线性结构。

线性结构的条件为:①有且只有一个根节点;②每一个节点最多有一个前件,也最多有一个后件。

非线性结构的条件为:不满足线性结构条件的数据结构。

在一个线性结构中插入或删除任何一个节点后还应是线性结构。

【例 1】以下关于数据的逻辑结构的叙述中,()是不正确的。

- A. 数据的逻辑结构抽象地反映数据元素间的逻辑关系
- B. 数据的逻辑结构是数据间关系的描述
- C. 数据逻辑结构具体反映数据在计算机中的存储方式
- D. 数据的逻辑结构分为线性结构和非线性结构

解析:数据的逻辑结构是数据间关系的描述,抽象地反映数据元素间的逻辑关系,而不管其在计算机中的存储方式。数据的逻辑结构分为线性结构和非线性结构。

答案: C

【例 2】以下关于数据的存储结构的叙述中()是正确的?

- A. 数据的存储结构是数据间关系的抽象描述
- B. 数据的存储结构是逻辑结构在计算机存储器中的实现
- C. 数据的存储结构分为线性结构和非线性结构
- D. 数据的存储结构对数据运算的具体实现没有影响

解析: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。选项 A 中“数据间关系的抽象描述”是对数据的逻辑结构的定义;选项 C 中数据的逻辑结构分为线性结构

和非线性结构，数据的存储结构应分为顺序存储和链式存储；选项 D 中数据的存储结构与数据运算是紧密结合的，不同的存储结构对数据运算会产生较大影响。通过以上分析可知，本题的正确答案是 B。

答案：B

1.1.3 线性表及其顺序存储结构

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。即线性表或是一个空表，或可以表示为 $(a_1, a_2, \dots, a_n)$ 。

线性表是一种线性结构，数据元素在线性表中的位置只取决于它们自己的序号，即数据元素之间的相对位置是线性的。

非空线性表的结构特征为：①有且只有一个根节点 a_1 ，它无前件；②有且只有一个终端节点 a_n ，它无后件；③除根节点与终端节点外，其他所有节点有且只有一个前件，也有且只有一个后件。节点个数 n 称为线性表的长度，当 $n=0$ 时，称为空表。

2. 线性表的顺序存储结构

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。

假设线性表中的第一个数据元素的存储地址为 $ADR(a_1)$ ，每一个数据元素占 k 个字节，则线性表中第 i 个元素 a_i 在计算机存储空间中的存储地址为：

$$ADR(a_i) = ADR(a_1) + (i-1)k$$

【例】有长度为 20 的线性表，第一个元素存放位置是 100，每个元素占 4 个字节，若按顺序存储，求第 8 个元素的存储位置。

$$ADR(a_8) = ADR(a_1) + (8-1) \times 4 = 100 + 28 = 128$$

3. 顺序表的运算

插入运算：在线性表的指定位置加入一个新的元素。

删除运算：在线性表中删除指定的元素。

查找运算：在线性表中查找某个（或某些）选定的元素。

排序运算：对线性表中的元素进行整序。

在平均情况下，要在线性表中插入或删除一个元素，需要移动表中一半的元素。

1.1.4 栈和队列

1. 栈及其基本运算

(1) 什么是栈。

栈实际上也是线性表，只不过是一种特殊的线性表。在这种特殊的线性表中，其插入与删除都只在线性表的一端进行。

栈 (stack) 是限定在一端进行插入与删除的线性表，允许插入与删除的一端称为栈顶，不允许插入与删除的一端称为栈底。图 1-3 所示为栈的示意图。

栈按照“先进后出”(FILO)或“后进先出”(LIFO)组织数据，栈具有记忆作用。用 top 表示栈顶，用 bottom 表示栈底。

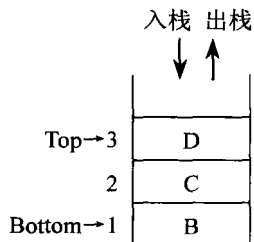


图 1-3 栈的示意图

(2) 栈的顺序存储及运算。

在程序设计语言中,用一维数组 $S(1:m)$ 作为栈的顺序存储空间,其中 m 为栈的最大容量。通常,栈底指针指向栈空间的低地址一端(即数组的起始地址这一端)。

在栈的顺序存储空间 $S(1:m)$ 中, $S(\text{bottom})$ 通常为栈底元素, $S(\text{top})$ 为栈顶元素。 $\text{Top}=0$ 表示栈空, $\text{top}=m$ 表示栈满。

(3) 栈的基本运算。

1) 入栈运算:是指在栈顶位置插入一个新元素。

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。

2) 退栈运算:是指取出栈顶元素并赋给一个指定的变量,栈顶指针退一(即删除栈顶元素)。

当栈顶指针为 0 时,说明栈空,不可能进行退栈操作。这种情况称为栈“下溢”错误。

3) 读栈顶元素:是指将栈顶元素赋给一个指定的变量,栈顶指针不改变。

【例 1】下列关于栈的叙述中正确的是()。

A. 在栈中只能插入数据

B. 在栈中只能删除数据

C. 栈是先进先出的线性表

D. 栈是先进后出的线性表

解析:在栈中可以进行插入和删除操作,只是限定在一端进行插入与删除的线性表。栈是先进后出的线性表。队列是先进先出的线性表。

答案: D

【例 2】若进栈序列为 1, 2, 3, 4, 假定进栈和出栈可以穿插进行,则可能的出栈序列是()。

A. 2, 4, 1, 3

B. 3, 1, 4, 2

C. 3, 4, 1, 2

D. 1, 2, 3, 4

解析:栈是限定在一端进行插入与删除的线性表。栈顶元素总是最后被插入的元素,从而也是最先能被删除的元素;栈底元素总是最先被插入的元素,从而也是最后被删除的元素。即栈是按照“先进后出”或“后进先出”的原则组织数据的。

因此,除 D 项外,其他情况均不可能出现。D 项中对栈的操作序列为:1 入栈→1 出栈→2 入栈→2 出栈→3 入栈→3 出栈→4 入栈→4 出栈。

答案: D

2. 队列及其基本运算

(1) 什么是队列。

队列是指允许在一端(队尾)进行插入,而在另一端(队头)进行删除的线性表。

用尾指针(rear)指向队尾元素;用头指针(front)指向排头元素的前一个位置。图 1-4 所示为队列的示意图。

队列是“先进先出”(FIFO)或“后进后出”(LILO)的线性表。

在程序设计语言中,用一维数组 $Q(1:m)$ 作为队列的顺序存储空间。

(2) 队列运算。

队列运算包括:①入队运算;②退队运算。

(3) 什么是循环队列。

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,供队列循环使用。

循环队列的初始状态为空,即 $\text{rear}=\text{front}=m$ 。图 1-5 所示为循环队列的示意图。

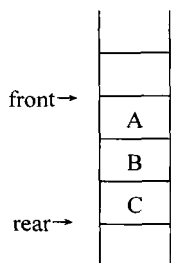


图 1-4 队列示意图

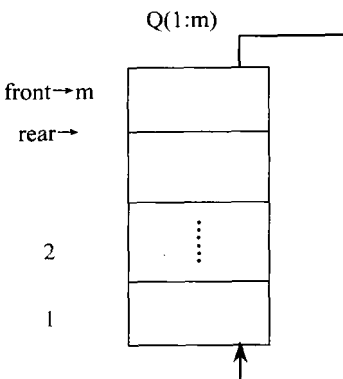


图 1-5 循环队列存储空间示意图

使用循环队列时，为了能区分队列满还是队列空，通常增加一个标志 s ， s 值的定义如下：

$$s = \begin{cases} 0 & \text{表示队列空} \\ 1 & \text{表示队列非空} \end{cases}$$

队列空与队列满的条件如下：

- 1) 队列空的条件为 $s=0$ 。
- 2) 队列满的条件为 $s=1$ 且 $\text{front}=\text{rear}$ 。

循环队列中元素的数目：

- 1) 当 $\text{rear} \neq \text{front}$ 时，为 $(\text{rear}-\text{front}+m) \bmod m$ 。
- 2) 当 $\text{rear}=\text{front}$ 且 $s=0$ 时，为 0。
- 3) 当 $\text{rear}=\text{front}$ 且 $s=1$ 时，为 m 。

3. 栈和队列的共同特点

栈和队列的共同特点是只允许在端点处插入和删除元素。

【例 1】下列关于队列的叙述中正确的是 ()。

- | | |
|----------------|---------------|
| A. 在队列中只能插入数据 | B. 在队列中只能删除数据 |
| C. 队列是先进先出的线性表 | D. 栈是先进后出的线性表 |

解析：队列的运算有插入数据和删除数据，它是一个先进先出的线性表，栈是先进后出的线性表。

答案：C

【例 2】数组 $Q[0 \cdots n-1]$ 作为一个环队列， f 为当前队头元素的前一位置， r 为队尾元素的位置，假定队列中元素的个数总小于 n ，队列中元素的个数是 ()。

- | | | | |
|----------|------------|------------|----------------------|
| A. $r-f$ | B. $n+f-r$ | C. $n+r-f$ | D. $(n+r-f) \bmod n$ |
|----------|------------|------------|----------------------|

解析：通过数组长度及头尾指针即可计算出循环链表的元素个数为 $(n+r-f) \bmod n$ ，因此本题正确答案为 D。

答案：D

1.1.5 线性链表

1. 线性链表的基本概念

节点：假设数据结构中的每一个数据节点对应于一个存储单元，这种存储单元称为存储节点，简称节点。

在链式存储方式中，要求每个节点由两部分组成：一部分用于存放数据元素值，称为数

据域；另一部分用于存放指针，称为指针域，如图 1-6 所示。其中指针用于指向该节点的前一个或后一个节点（即前件或后件）。

V(i)	NEXT(i)
------	---------

图 1-6 线性链表的一个存储节点

链表采用的是链式存储结构（链式存储中每个节点由两部分组成：数据域和指针域），它克服了顺序存储结构的缺点，它的节点空间可以动态申请和释放；它的数据元素的逻辑次序靠节点的指针来指示，不需要移动数据元素。

在链式存储结构中，存储数据结构的存储空间可以不连续，各数据节点的存储顺序与数据元素之间的逻辑关系可以不一致，而数据元素之间的逻辑关系是由指针域来确定的。

链式存储方式既可用于表示线性结构，也可用于表示非线性结构。

【例】线性表若采用链表存储结构时，要求内存中可用存储单元的地址（ ）。

- A. 必须是连续的
- B. 部分地址必须是连续的
- C. 一定是不连续的
- D. 连续不连续都可以

解析：为了适应线性表的链式存储结构，计算机存储空间被划分为一个一个小块，每一小块占若干字节，通常称这些小块为存储节点。存储节点由两部分组成：一部分用于存储数据元素的值，称为数据域；另一部分用于存放下一个数据元素的存储序号（即存储节点的地址），称为指针域。

一般来说，在线性表的链式存储结构中，各数据节点的存储序号是不连续的，并且各节点在存储空间中的位置关系与逻辑关系也不一致。

答案：D

2. 线性链表

线性表的链式存储结构称为线性链表。

在线性链表中，用指针 HEAD 指向线性链表中第一个数据元素的节点，最后一个节点的指针域为空（用 NULL 或 0 表示），表示链表终止。

对于线性链表，可以从头指针开始，沿各节点的指针扫描到链表中的所有节点。

用链表表示线性表的突出优点是便于插入和删除操作。

3. 循环链表

循环链表的特点：①在循环链表中增加了一个表头节点，其数据域为任意或者根据需要来设置，指针域指向线性表的第一个元素。循环链表的头指针指向表头节点；②循环链表中最后一个节点的指针域不是空，而是指向表头节点。

在循环链表中，只要指出表中任何一个节点的位置，就可以从它出发访问到表中其他所有的节点。

在循环链表中设置一个表头节点的优点是方便运算的实现，使空表与非空表的运算统一。

1.1.6 树与二叉树

1. 树的基本概念

树是一种简单的非线性结构。在树结构中，每一个节点只有一个前件，称为父节点，没有前件的节点只有一个，称为树的根节点，简称为树的根。在树结构中，每一个节点可以有多个后件，它们都称为该节点的子节点。没有后件的节点称为叶子节点。在树结构中，一个节点所拥有的后件个数称为该节点的度。叶子节点的度为 0。在树中，所有节点中最大的度称为树的度。树的最大层次称为树的深度。

例如，如图 1-7 所示是一棵树。

由图可以看出, 节点 A 为根节点, 其度为 3; 节点 C、E、F、I、H 为叶子节点, 其度为 0; 节点 B 的度为 2, 节点 G 的度为 1; 树的度为 3; 树的深度为 4。

【例】树最适合用来表示 ()。

- A. 有序数组元素
- B. 无序数组元素
- C. 元素之间具有分支层次关系的数据通信
- D. 元素之间无联系的数据

解析: 树是一种简单的非线性结构。在树这种数据结构中, 所有数据元素之间的关系具有明显的层次特性, 即层次模型, 层次模型是最早发展起来的数据库模型。层次模型的基本结构是树形结构。

答案: C

2. 二叉树及其基本性质

二叉树是一种很有用的非线性结构。二叉树具有以下两个特点: ①非空二叉树只有一个根节点; ②每一个节点最多有两棵子树, 且分别称为该节点的左子树与右子树。

由以上特点可以看出, 在二叉树中, 每一个节点的度最大为 2, 即所有子树 (左子树或右子树) 也均为二叉树, 而树结构中的每一个节点的度可以是任意的。另外, 二叉树中的每一个节点的子树被明显地分为左子树与右子树。可以没有其中的一个, 也可以全没有。

二叉树的基本性质:

性质 1: 在二叉树的第 k 层上, 最多有 2^{k-1} ($k \geq 1$) 个节点。

【例 1】在一棵二叉树上第 5 层的节点数最多是 ()。

- A. 8
- B. 16
- C. 32
- D. 15

解析: 对于一个二叉树来说, 每一个节点最多有两棵子树, 根据二叉树的性质 1, 第 5 层上最多有 2^4 个节点, 即 16 个节点。

答案: B

性质 2: 深度为 m 的二叉树最多有 $2^m - 1$ 个节点。

深度为 m 的二叉树是指二叉树共有 m 层。

【例 2】在深度为 7 的二叉树中, 总的节点数最多为 ()。

- A. 128
- B. 127
- C. 64
- D. 63

解析: 在深度为 7 的二叉树中最多有 $2^7 - 1$ 个节点, 即 $2^7 - 1 = 127$ 。

答案: B

性质 3: 在任意一棵二叉树中, 度为 0 的节点 (即叶子节点) 总是比度为 2 的节点多一个, 即 $n_0 = n_2 + 1$ 。

【例 3】设一棵二叉树中, 叶子节点数为 9, 则该二叉树中度为 2 的节点数是 ()。

- A. 10
- B. 11
- C. 8
- D. 不确定

解析: 叶子节点数为 9, 即 $n_0 = 9$, 根据性质 3 $n_0 = n_2 + 1$, 可求得 $n_2 = n_0 - 1 = 9 - 1 = 8$ 。

答案: C

性质 4: 具有 n 个节点的二叉树, 其深度至少为 $\lfloor \log_2 n \rfloor + 1$, 其中 $\lfloor \log_2 n \rfloor$ 表示取 $\log_2 n$ 的整数部分。

【例 4】以下数据结构中不属于线性结构的是 ()。

- A. 队列
- B. 线性表
- C. 二叉树
- D. 栈

解析: 线性结构必须满足如下两个条件: ① 有且只有一个根节点; ② 每一个节点最多

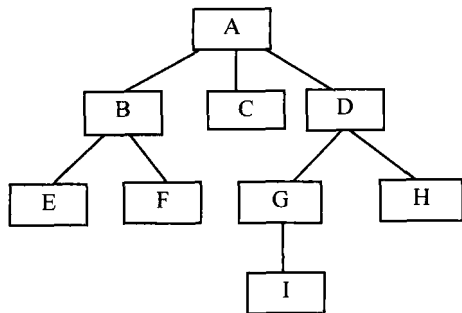


图 1-7 一般的树

有一个前件,也最多有一个后件。而对于二叉树来说,每一个节点最多可以有两棵子树,因此二叉树不是线性结构。

答案: C

3. 满二叉树与完全二叉树

满二叉树与完全二叉树是两种特殊形态的二叉树。

(1) 满二叉树。所谓满二叉树是指这样的一种二叉树:除最后一层外,每一层上的所有节点都有两个子节点。这就是说,在满二叉树中,每一层上的节点数都达到最大值,即在满二叉树的第 k 层上有 2^{k-1} 个节点,且深度为 m 的满二叉树有 2^m-1 个节点。

【例】在深度为 7 的满二叉树中,叶子节点的个数为 ()。

A. 32

B. 31

C. 64

D. 63

解析:满二叉树的第 k 层上有 2^{k-1} 个节点,在深度为 7 的满二叉树中,第 7 层上的节点数是 $2^6=64$ 。

答案: C

(2) 完全二叉树。所谓完全二叉树是指这样的二叉树:除最后一层外,每一层上的节点数均达到最大值;在最后一层上只缺少右边的若干节点。

更确切地说,如果从根节点起,对二叉树的节点自上而下、自左至右用自然数进行连续编号,则深度为 m 且有 n 个节点的二叉树,当且仅当其每一个节点都与深度为 m 的满二叉树中编号从 1 到 n 的节点一一对应时,称之为完全二叉树。

由满二叉树与完全二叉树的特点可以看出,满二叉树也是完全二叉树,而完全二叉树一般不是满二叉树。

(3) 完全二叉树还具有以下两个性质:

性质 5:具有 n 个节点的完全二叉树的深度为 $\lfloor \log^{2n} \rfloor + 1$ 。

性质 6:设完全二叉树共有 n 个节点。如果从根节点开始,按层序(每一层从左到右)用自然数 1, 2, ..., n 给节点进行编号,则对于编号为 k ($k=1, 2, \dots, n$) 的节点有以下结论:

- 若 $k=1$, 则该节点为根节点,它没有父节点;若 $k>1$, 则该节点的父节点编号为 $\text{INT}(k/2)$ 。
- 若 $2k \leq n$, 则编号为 k 的节点的左子节点编号为 $2k$; 否则该节点无左子节点(显然也没有右子节点)。
- 若 $2k+1 \leq n$, 则编号为 k 的节点的右子节点编号为 $2k+1$; 否则该节点无右子节点。

【例】将含有 100 个节点的完全二叉树从根这一层开始,每层从左到右依次对节点编号,根节点的编号为 1。编号为 71 的节点的双亲的编号为 ()。

A. 34

B. 35

C. 36

D. 无法确定

解析:若 i 为父节点,则 $2i$ 为它的左子节点, $2i+1$ 为它的右子节点。编号为 71 的节点与父节点编号的关系是: $2i+1=71$, $i=35$ 。

答案: B

(4) 二叉树的存储结构。在计算机中,二叉树通常采用链式存储结构,如图 1-8 所示。

Lchild	Value	Rchild
L(i)	V(i)	R(i)

图 1-8 二叉树存储节点的结构

(5) 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节点。

二叉树的遍历可以分为 3 种:

1) 前序遍历(DLR)是指先访问根节点,然后遍历左子树,最后遍历右子树。

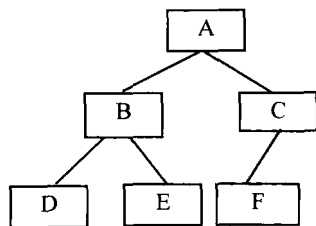
- 2) 中序遍历 (LDR) 是指先遍历左子树, 然后访问根节点, 最后遍历右子树。
 3) 后序遍历 (LRD) 是指先遍历左子树, 然后遍历右子树, 最后访问根节点。

【例 1】设有下列二叉树, 对此二叉树中序遍历的结果为 ()。

- A. ABCDEF B. DBEAFC
 C. ABDECF D. DEBFCA

解析: 中序遍历二叉树的过程是先遍历左子树, 访问根节点, 最后遍历右子树。根据这一规则访问节点的顺序依次是 D, B, E, A, F, C。

答案: B



【例 2】对上题所示的二叉树进行后序遍历的结果为 ()。

- A. ABCDEF B. DBEAFC C. ABDECF D. DEBFCA

解析: 后序遍历是首先遍历左子树, 然后遍历右子树, 最后访问根节点。根据后序遍历的规则对此二叉树进行后序遍历的顺序是 D→E→B→F→C→A。

答案: D

那么对上述二叉树进行前序遍历的结果又是什么呢?

【例 3】如果一棵二叉树中任一节点的值都大于其左子树中所有节点的值, 且小于其右子树中所有节点的值, 现欲得到各节点值的递增序列, 试问应采用的遍历方法是 ()。

- A. 前序遍历 B. 中序遍历 C. 后序遍历 D. 层次遍历

解析: 如果一棵二叉树中任一节点的值都大于其左子树中所有节点的值, 且小于其右子树中所有节点的值, 那么这棵二叉树叫做二叉排序树。二叉树的后序遍历是先遍历左子树, 再遍历右子树, 最后访问根节点, 因此二叉排序树的中序遍历直接有序。

答案: B

【例 4】某二叉树节点的前序序列为 EACBDGF, 中序序列为 ABCDEFG。该二叉树节点的后序序列为 ()。

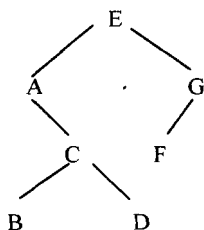
- A. BDCAFGE B. BDCFAGE C. EGFACDB D. EGACFDB

解析: 前序序列中以 E 开始, 则可以确定 E 为根节点, 则参考中序序列中 E 的位置可以判断 ABCD 属于 E 的左子树, FG 属于 E 的右子树。

分析 ABCD 的关系: 在前序序列中 A 排在第一个位置, 则 A 是 E 的左子树的根, 从中序序列中 A 排在第一位, 则可判断 A 没有左子树, 只有右子树, BCD 应为 A 的右子树成员; 再来看 BCD 的关系, 在前序序列中 C 排在第一位, 则 C 是 A 的右子树的根; 从中序序列中 C 排在 BD 中间, 说明 B 为 C 的左子树, D 为 C 的右子树。

分析 FG 的关系: 在前序序列中 G 排在 F 的前面, 则 G 是 E 的右子树的根, 中序序列中 G 排在 F 的后面, 则 F 为 G 的左子树。

那么整个二叉树的结构应如下图所示。



根据此图可以看出后序序列为 BDCAFGE。