

2004年版

全国计算机等级考试

考试要点、真题详解与训练

——二级C语言程序设计

主 编 欧 阳 副主编 李太勇



高等教育出版社
HIGHER EDUCATION PRESS

内容提要

本书是依据《全国计算机等级考试考试大纲(2004年版)》中的二级公共基础知识和C语言程序设计部分的要求编写的。本书分为文字部分和配套光盘,文字部分包括笔试部分(1~3章)和附录,其中笔试部分主要内容包括:公共基础知识(含数据结构与算法、程序设计基础、软件工程基础、数据库设计基础)、C语言程序设计以及4套全真笔试试卷;附录A为上机模拟考试软件的安装步骤,附录B为相应的考试大纲,附录C、D、E、F分别给出了C语言的关键字、运算符的优先级和结合性、常用字符与ASCII代码对照表、库函数等。试题均选自历届考题,具有针对性强、内容不超纲等特点。笔试试卷的题型、题型比例和难易程度都与真题一致。

配套光盘为上机考试模拟软件,该软件与真考环境完全一致;并且与正式考试共用一个题库,题目命中率高。

本书可作为全国计算机等级考试二级C语言程序设计考前指导书,同时也可作为C语言程序设计课程的学习参考书。

图书在版编目(CIP)数据

全国计算机等级考试(2004年版)考试要点、真题详解与训练. 二级C语言程序设计 / 欧阳主编. —北京: 高等教育出版社, 2005. 5

ISBN 7-04-016286-5

I. 全... II. 欧... III. ①电子计算机-水平考试-解题②C语言-程序设计-水平考试-解题
IV. TP3-44

中国版本图书馆CIP数据核字(2005)第043753号

策划编辑 刘 英 责任编辑 刘 英 封面设计 张申申 责任印制 韩 刚

出版发行 高等教育出版社
社 址 北京市西城区德外大街4号
邮政编码 100011
总 机 010-58581000

购书热线 010-58581118
免费咨询 800-810-0598
网 址 <http://www.hep.edu.cn>
<http://www.hep.com.cn>
网上订购 <http://www.landaco.com>
<http://www.landaco.com.cn>

经 销 北京蓝色畅想图书发行有限公司
印 刷 高等教育出版社印刷厂

开 本 787×1092 1/16
印 张 21.25
字 数 520 000

版 次 2005年5月第1版
印 次 2005年5月第1次印刷
定 价 35.70元(含光盘)

本书如有缺页、倒页、脱页等质量问题,请到所购图书销售部门联系调换。

版权所有 侵权必究

物料号 16286-00

前 言

全国计算机等级考试是由教育部考试中心主办,用于考查计算机应用知识与能力的等级水平考试,是用人单位录用和考核工作人员的评价标准。参加全国计算机等级考试的人员主要由在校大学生、机关公务员、部队官兵等构成,等级证书的获得与否与他们的就业、晋升有着密切的关系。

全国计算机等级考试自从 1994 年举办以来,得到了全国各行各业从事计算机工作与学习人士的积极参与和各用人单位的普遍认可,是目前国内影响最大、参加人数最多的计算机类考试。

为了使广大考生能顺利地通过全国计算机等级考试,编者根据《全国计算机等级考试考试大纲(2004 年版)》的要求,结合多年从事全国计算机等级考试培训和基础教学工作的实践经验,编写了“全国计算机等级考试 考试要点、真题详解与训练”系列丛书。本套丛书共包括以下 13 本:

全国计算机等级考试 考试要点、真题详解与训练——一级 MS Office

全国计算机等级考试 考试要点、真题详解与训练——一级 B

全国计算机等级考试 考试要点、真题详解与训练——二级 C 语言程序设计

全国计算机等级考试 考试要点、真题详解与训练——二级 C++ 语言程序设计

全国计算机等级考试 考试要点、真题详解与训练——二级 Java 语言程序设计

全国计算机等级考试 考试要点、真题详解与训练——二级 Visual Basic 语言程序设计

全国计算机等级考试 考试要点、真题详解与训练——二级 Visual FoxPro 数据库程序设计

全国计算机等级考试 考试要点、真题详解与训练——二级 Access 数据库程序设计

全国计算机等级考试 考试要点、真题详解与训练——三级 PC 技术

全国计算机等级考试 考试要点、真题详解与训练——三级信息管理技术

全国计算机等级考试 考试要点、真题详解与训练——三级数据库技术

全国计算机等级考试 考试要点、真题详解与训练——三级网络技术

全国计算机等级考试 考试要点、真题详解与训练——四级

丛书具有以下特色:

- (1) 按考试大纲要求列举了考试必备知识。
- (2) 丛书中所列举的试题都选自历届全国计算机等级考试试题,按考试时间由现在到过去将历届考题归类。
- (3) 丛书中的试题都是真题,具有典型性和针对性,并给出了试题分析和参考答案。
- (4) 丛书配有上机考试光盘,光盘中的上机考试系统与真考系统一致,其登录、抽题、答题、提交,与正式上机考试一模一样,自动生成试卷、自动计时、自动评分,与正式考试共用一个题库,命中率高。

本套丛书将会令考生更易于理解全国计算机等级考试的基本要求和解答试题的思路,使读者在较短的时间内取得较大的收获,为参加等级考试和使用计算机打下良好的基础,为

考生通过考试增添一分把握。

下面介绍一下如何准备二级公共基础知识 C 语言程序设计考试:

(1) 公共基础知识在笔试试卷中占 30 分,多数考题要求记忆。因此在本书的第 1 章中,根据考试大纲要求,将考试中心教材中的要点进行了归纳,并按考点设计了许多真题,考生只要把本书第 1 章的要点掌握,并将试题看懂、记住,就应该能满足公共基础知识部分的测试要求。

(2) C 语言程序设计在笔试试卷中占 70 分,要求考生掌握 C 语言的语法和简单算法。通过对历届 C 语言考题的分析和总结,发现多数考题在以往曾经出现过(由于 C 语言已经考了很多次)。针对这种特点,本书将 2004 年 9 月至 2000 年 4 月的笔试试卷按考试大纲和教材归类,将题型相同的考题放在一起,这样考生很容易看出出题的规律。同时在相应章节还给出了考试要点。

(3) C 语言上机考试题型有程序填空题、程序修改题和程序设计题,考试时间是 60 分钟。其中程序填空题占 30 分,只要求把程序中的空白处填上合适的语句即可;程序修改题占 30 分,将程序中的错误语句改正确即可;程序设计题占 40 分,要求按试题要求将程序空白函数补上,并调试、运行正确(一般要有运行结果,否则没有成绩)。考生只要掌握 C 语言的简单算法和查错、排错的基本方法,应该能通过上机考试的要求。上机考试有这样一个特点:有些考生很快就能交卷,而有些考生思考很久也做不出来。这是因为,有的考生平时重视上机操作,操作熟练,而有些考生平时很少上机练习,对上机考试也采取背的方式,所以很难通过考试。针对这种情况,我们开发了“全国计算机等级考试上机考试模拟系统”并制作成光盘。本系统与真考系统一致,其登录、抽题、答题、提交,与正式上机考试完全一致,自动生成试卷、自动计时、自动评分,并且与正式考试共用一个题库,题目的命中率高。

合理分配复习时间也很重要,我们认为笔试用 70% 的时间、上机用 30% 的时间比较合适。

总之,全国计算机等级考试其实不难,只要按照上面介绍的办法去做,通过率会很高。最后祝广大考生顺利通过考试,获得证书。

本书的第 1 章由欧阳编写,第 2 章由李太勇、沙开波、吴江、毛昶、程伟、吴艳玲编写,第 3 章由廖勇、马增辉编写,其他部分由郭新明编写。

由于时间仓促和作者水平有限,书中错误难免,敬请读者指正。

编 者

(guoxm3@vip.163.com)

2005 年 4 月

目 录

第1章 公共基础知识	(1)	2.3.3 最新考题解答	(140)
1.1 数据结构与算法	(1)	2.3.4 全真试题训练	(184)
1.1.1 考试必备知识	(1)	2.3.5 全真试题参考答案	(197)
1.1.2 全真试题解答	(7)	2.4 编译预处理与指针	(199)
1.1.3 全真试题训练	(14)	2.4.1 大纲解读	(199)
1.1.4 全真试题参考答案	(17)	2.4.2 考试必备知识	(199)
1.2 程序设计基础	(18)	2.4.3 最新考题解答	(205)
1.2.1 考试必备知识	(18)	2.4.4 全真试题训练	(227)
1.2.2 全真试题解答	(20)	2.4.5 全真试题参考答案	(237)
1.2.3 全真试题训练	(24)	2.5 结构体、共用体、位运算	
1.2.4 全真试题参考答案	(26)	及文件操作	(237)
1.3 软件工程基础	(26)	2.5.1 大纲解读	(237)
1.3.1 考试必备知识	(26)	2.5.2 考试必备知识	(238)
1.3.2 全真试题解答	(32)	2.5.3 最新考题解答	(247)
1.3.3 全真试题训练	(37)	2.5.4 全真试题训练	(268)
1.3.4 全真试题参考答案	(40)	2.5.5 全真试题参考答案	(275)
1.4 数据库设计基础	(41)	第3章 全真笔试试卷	(277)
1.4.1 考试必备知识	(41)	3.1 全真笔试试卷(一)	(277)
1.4.2 全真试题解答	(45)	3.2 全真笔试试卷(一)参考答案	(284)
1.4.3 全真试题训练	(50)	3.3 全真笔试试卷(二)	(285)
1.4.4 全真试题参考答案	(53)	3.4 全真笔试试卷(二)参考答案	(295)
第2章 C语言程序设计	(54)	3.5 全真笔试试卷(三)	(296)
2.1 C语言的基础知识	(54)	3.6 全真笔试试卷(三)参考答案	(304)
2.1.1 大纲解读	(54)	3.7 全真笔试试卷(四)	(305)
2.1.2 考试必备知识	(54)	3.8 全真笔试试卷(四)参考答案	(314)
2.1.3 最新考题解答	(61)	附录A 如何安装和使用上机考试系统	(315)
2.1.4 全真试题训练	(75)	附录B 二级C语言程序设计考试大纲	
2.1.5 全真试题参考答案	(77)	(2004年版)	(321)
2.2 C语言程序设计	(78)	B.1 公共基础知识	(321)
2.2.1 大纲解读	(78)	B.2 C语言程序设计	(322)
2.2.2 考试必备知识	(79)	附录C 关键字及双目运算转换规律	(325)
2.2.3 最新考题解答	(85)	C.1 C语言的关键字	(325)
2.2.4 全真试题训练	(116)	C.2 双目算术运算中两边运算	
2.2.5 全真试题参考答案	(132)	量类型转换规律	(325)
2.3 数组与函数	(133)	附录D 运算符的优先级和结合性	(326)
2.3.1 大纲解读	(133)	附录E 常用字符与ASCII代码对照表	(327)
2.3.2 考试必备知识	(134)	附录F 库函数	(328)

第1章 公共基础知识

1.1 数据结构与算法

1.1.1 考试必备知识

1. 算法

(1) 算法的基本概念

算法是指解题方案的准确而完善的描述。它具有可行性 (Effectiveness)、确定性 (Definiteness)、有穷性 (Finiteness) 和必需的信息 (Necessary Information) 等基本特征。

算法的基本要素包括算法中对数据的运算和操作以及算法的控制结构。基本的运算和操作包括算术运算、逻辑运算、关系运算和数据传输;基本的控制结构包括顺序结构、选择结构、循环结构。

算法设计的基本方法有:列举法、归纳法、递推、递归、减半递推技术和回溯法。

(2) 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。

① 算法的时间复杂度:指执行算法所需要的计算工作量。算法的工作量可以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的函数,即

$$\text{算法的工作量} = f(n)$$

其中 n 是问题的规模。

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的输入时,可以用两种方法分析算法的工作量:平均性态 (Average Behavior) 和最坏情况复杂性 (Worst - Case Complexity)。

所谓平均性态分析,是指用各种特定输入下的基本运算次数的加权平均值来度量算法的工作量,可以用以下公式表示

$$A(x) = \sum_{x \in D_n} p(x) t(x)$$

其中 x 是所有可能输入中的某个特定输入, $p(x)$ 是 x 出现的概率, $t(x)$ 是算法在输入为 x 时所执行的基本运算次数, D_n 表示当问题规模为 n 时,算法执行时所有可能输入的集合。

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,它定义为

$$W(n) = \max_{x \in D_n} \{t(x)\}$$

② 算法的空间复杂度:指执行这个算法所需要的内存空间大小。一个算法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用的存储空间以及算法执行过程中

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以及某种数据结构所需要的附加存储空间。

2. 数据结构的基本概念

(1) 数据结构学科的研究对象

数据结构学科主要研究如下3个方面的内容:

- 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) 数据结构学科的研究目的

数据结构学科研究的目的是提高数据处理的效率,主要包括:

- ① 数据处理速度。
- ② 尽量节省在数据处理过程中所占用的计算机存储空间。

(3) 数据结构的定义

数据处理:指对数据集中的各元素以各种方式进行运算。

数据元素:在数据处理领域中,每一个需要处理的对象都可以抽象为数据元素。

数据结构:指相互关联的数据元素的集合。

① 数据的逻辑结构:指反映数据元素之间逻辑关系的数据结构。它包含两个要素: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趋、后继关系,通常记为 R 。形式表示如下:

$$B = (D, R)$$

其中 B 表示数据结构。

② 数据的存储结构:指数据的逻辑结构在计算机存储空间中的存放形式。

一种数据结构的逻辑结构根据需要可以表示成多种存储结构。常用的有顺序、链接、索引等存储结构。

(4) 数据结构的图形表示

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,简称为结点;对于关系 R 中的每一个二元组,用一条有向线段从前趋结点指向后继结点,以表示数据元素之间的前趋、后继关系。

例如,用图形表示数据结构 $B = (D, R)$,其中:

$$D = \{d_i | 1 \leq i \leq 6\} = \{d_1, d_2, d_3, d_4, d_5, d_6\}$$

$$R = \{(d_1, d_2), (d_1, d_3), (d_3, d_4), (d_5, d_4), (d_5, d_6)\}$$

这个数据结构的图形如图 1.1 所示。

(5) 线性结构和非线性结构

线性结构满足如下条件:

- ① 有且仅有一个根结点。
- ② 每一个结点最多有一个前趋,也最多有一个后继。

③ 在一个线性结构中插入或删除任何一个结

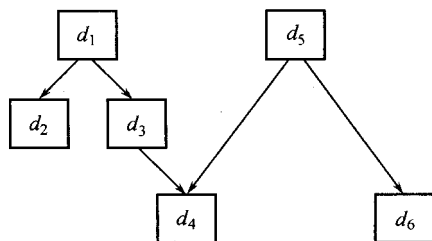


图 1.1 数据结构的图形表示

点后还是线性结构。

如果一个数据结构不是线性结构,则称之为非线性结构。

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,有且只有一个前趋,除了最后一个外,有且只有一个后继。即线性表或是一个空表,或是可以表示为

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

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

非空线性表有如下结构特征:

- ① 有且只有一个根结点 a_1 ,它无前趋。
- ② 有且只有一个终端结点 a_n ,它无后继。
- ③ 除根结点与终端结点外,其他所有结点有且只有一个前趋,也有且只有一个后继。

线性表中结点的个数 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的基本特点:

- ① 线性表中所有元素所占的存储空间是连续的。
- ② 线性表中的数据元素在存储空间中是按逻辑顺序一次存放的。

线性表的随机存取地址的计算公式:

$$\text{ADD}(a_i) = \text{ADD}(a_1) + (i-1)k$$

线性表的主要操作包括:插入、删除、查找、排序、分解、合并、复制和逆转。

(3) 线性表的插入运算

线性表插入操作前后元素的对应关系:

设长度为 n 的线性表为

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

在线性表的第 i 个元素 a_i 之前插入一个新元素 b 后得到的长度为 $n+1$ 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:

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

线性表插入操作的实现方法:

一般地,要在第 $i(1 \leq i \leq n)$ 个元素之前插入一个新元素时,首先要从最后一个(即第 n 个)元素开始,直到第 i 个元素之间的 $n-i+1$ 个元素依次向后移动一个位置,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,线性表的长度就增加了 1。

线性表插入操作算法的复杂度分析:

由线性表插入算法可知:如果插入的位置在第 $i(1 \leq i \leq n)$ 个元素之前,则原来第 i 个元素之后的所有元素都必须向后移动。在平均情况下,要在线性表中插入一个新元素,需要移

动表中一半的元素。

(4) 线性表的删除运算

线性表删除操作前后元素的对应关系:

设长度为 n 的线性表为

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

删除线性表的第 i 个元素 a_i 后得到的长度为 $n-1$ 的线性表为

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

则删除后两线性表中的元素满足如下关系:

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

线性表删除操作的实现方法:

通常要删除第 i ($1 \leq i \leq n$) 个元素时,首先要从第 $i+1$ 个元素开始,直到最后一个(即第 n 个)元素之间的 $n-i$ 个元素依次向前移动一个位置。删除结束后,线性表的长度就减少了1。

线性表删除操作算法的复杂度分析:

由线性表插入算法可知:如果删除的是第 i ($1 \leq i \leq n$) 个元素,则原来第 i 个元素之后的所有元素都必须向前移动一个位置。在平均情况下,要在线性表中删除一个已有的元素,需要移动表中一半的元素。

(5) 线性表顺序存储结构的适用场合

线性表的顺序存储结构对于小线性表或者其中元素不常变动的线性表来说是合适的,因为顺序存储的结构比较简单,但这种顺序存储的方式对于元素需要变动的大线性表就不太合适了,因为插入和删除的效率比较低。

4. 栈和队列

(1) 栈及其基本运算

栈(Stack)是限定在一端进行插入和删除的线性表,它按照“后进先出”的原则组织数据。

在程序设计语言中,一般是用一维数组 $S(1:m)$ 作为栈的顺序存储空间,其中 m 为栈的最大容量。

栈的基本运算包括:

① 入栈运算:首先将栈顶指针加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,此时发生栈“上溢”错误。

② 退栈运算:首先将栈顶元素赋给一个指定的变量,然后将栈顶指针减1。当栈顶指针为0时,说明栈空,不可能进行退栈操作,此时,发生栈“下溢”错误。

③ 读栈顶元素:将栈顶元素赋给一个指定的变量,栈指针不变。当栈顶指针为0时,说明栈空,读栈顶元素操作失败。

(2) 队列及其基本运算

队列(Queue)是指允许在一端进行插入,而在另一端进行删除的线性表,它按照“先进先出”的原则组织数据。

循环队列空的状态: $s = 0$, 且 $\text{front} = \text{rear} = m$ 。

循环队列满的状态: $s = 1$, 且 $\text{front} = \text{rear}$ 。

循环队列的基本运算包括:

① 入队运算: 首先将队尾指针加 1 ($\text{rear} = \text{rear} + 1$), 并当 $\text{rear} = m + 1$ 时置 $\text{rear} = 1$; 然后将新元素插入到指针指向的位置。当循环队列满时 ($s = 1$, 且 $\text{front} = \text{rear}$), 入队操作将引起“上溢”错误。

② 退队运算: 首先将排头指针加 1 ($\text{front} = \text{front} + 1$), 并当 $\text{front} = m + 1$ 时置 $\text{front} = 1$; 然后将排头指针指向的元素赋给指定的变量。当循环队列为空时 ($s = 0$, $\text{front} = \text{rear}$), 退队操作将引起“下溢”错误。

5. 线性链表

(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。

① 线性链表。

单链表或线性链表的链式存储结构: 单链表的每个结点中数据域存放数据元素的值, 指针域存放后继结点的存储地址。

双向链表的链式存储结构: 双向链表的链式存储结构比线性链表的链式存储结构多出一个指针域, 它用来存放前趋结点的存储地址。

② 带链的栈。栈的链式结构基本上和线性链表的链式存储结构相同。只是线性链表的链式存储结构的头指针变成了栈的链式结构的栈顶指针。

③ 带链的队列。队列的链式结构和线性链表的链式存储结构也基本相同。只不过队列的链式结构保持有两个指针: 一个指向队列头的头指针和一个指向队列尾的尾指针。

(2) 线性链表的基本运算

线性链表的主要运算有: 线性链表的查找、线性链表的插入、线性链表的删除、线性链表的合并、线性链表的分解、线性链表的逆转、线性链表的复制、线性链表的排序。

① 线性链表的查找: 线性链表的查找过程是从头指针指向的结点开始往后沿指针进行扫描, 直到后面已没有结点或下一个结点的数据域为搜索值 x 为止。

② 线性链表的插入: 线性链表的插入是先从栈中为新元素分配一个新的结点 p , 并赋值, 利用线性链表的查找算法找到待插入位置的前一个结点的指针 q , 将 p 指向 q 的后继, 然后将 p 悬挂在 q 结点后面。

③ 线性链表的删除: 利用线性链表的查找算法找到待删除元素的前一个结点 p , 用另一个指针 q 暂时保存 p 的后继结点 (即待删除结点), 然后把 q 结点的后继链直接悬挂在 p 的后面, 最后归还 q 结点所分配的栈空间。

(3) 循环链表及其基本运算

循环链表有如下几个特点:

① 在循环链表中增加一个表头结点, 使得循环链表对空表和非空表的操作实现了统一。

② 循环链表中最后一个结点的指针域不是空, 而是指向表头结点。

③ 判断循环链表是否为空的办法不是看表头指针为空, 而是看表头结点的后继结点是否还是表头结点。

④ 在循环链表中, 从任何一个结点出发可以访问到表中其他所有的结点。

6. 树与二叉树

(1) 树的基本概念

树的递归定义:

树是 $n(n \geq 0)$ 个结点的有限集 T , T 为空时称为空树, 否则它满足如下两个条件:

- ① 有且仅有一个特定的称为根的结点。
- ② 其余的结点可分为 $m(m \geq 0)$ 个互不相交的子集 $T_1, T_2, \dots, T_m$, 其中每个子集本身又是一棵树, 并称其为根的子树。

基本术语: 父结点, 子结点, 根结点, 叶子结点, 结点的度, 树的度, 树的深度, 子树。

(2) 二叉树及其基本性质

二叉树具有两个特点: 非空二叉树只有一个根结点; 每一个结点最多有两棵子树。

二叉树的基本性质:

- ① 在二叉树的第 k 层上, 最多有 $2^{k-1} (k \geq 1)$ 个结点。
- ② 深度为 m 的二叉树是指二叉树最多有 $2^m - 1$ 个结点。
- ③ 在任意一棵二叉树中, 度为 0 的结点 (即叶子结点) 总是比度为 2 的结点多一个。
- ④ 具有 n 个结点的二叉树, 其深度至少为 $\lceil \log_2 n \rceil + 1$, 其中 $\lceil \log_2 n \rceil$ 表示 $\log_2 n$ 的整数部分。

(3) 满二叉树和完全二叉树

- ① 满二叉树的第 k 层上有 2^{k-1} 个结点, 且其深度为 m 的满二叉树有 $2^m - 1$ 个结点。
- ② 具有 n 个结点的完全二叉树的深度为 $\lceil \log_2 n \rceil + 1$ 。
- ③ 设完全二叉树共有 n 个结点。如果从根结点开始, 按层序 (每一层从左到右) 用自然数 $1, 2, \dots, n$, 给结点进行编号, 则对与编号为 $k (k = 1, 2, \dots, n)$ 的结点有以下结论:

若 $k = 1$, 则该结点为根结点, 它没有父结点; 若 $k > 1$, 则该结点的父结点编号为 $\lceil k/2 \rceil$ 。

若 $2k \leq n$, 则编号为 k 的结点的左子结点编号为 $2k$; 否则该结点无左子结点 (显然也没有右子结点)。

若 $2k + 1 \leq n$, 则编号为 k 的结点的右子结点编号为 $2k + 1$; 否则该结点无右子结点。

(4) 二叉树的存储结构

L(i)	V(i)	R(i)
------	------	------

L(i): 左指针域, 指向该结点的左子结点, 存放左子结点的存储地址。

R(i): 右指针域, 指向该结点的右子结点, 存放右子结点的存储地址。

V(i): 数据域, 存放结点的值。

(5) 二叉树的遍历

二叉树的遍历集中用到了递归的思想, 它主要有三种遍历方式。

- ① 前序遍历 (DLR): 先访问根结点, 后前序遍历左子树, 再前序遍历右子树。
- ② 中序遍历 (LDR): 先序遍历左子树, 后访问根结点, 再中序遍历右子树。
- ③ 后序遍历 (LRD): 先后序遍历左子树, 后后序遍历右子树, 再访问根结点。

7. 查找技术

(1) 顺序查找

- ① 查找方法: 从表头到表尾逐个比较, 若相同则结束查找, 否则一直继续比较下一个表

中元素直到整个表都遍历完。

② 适用场合:无序表或链式存储的有序表。

(2) 二分查找

① 查找方法:每次把待查找值与表中间元素比较,以减半的方式缩小搜索范围。

② 适用场合:顺序存储的有序表。

8. 排序技术

(1) 交换类排序

常用的交换类排序有:冒泡排序法和快速排序法。

(2) 插入类排序

常用的插入类排序有:简单插入排序法和希尔排序法。

(3) 选择类排序

常用的选择类排序有:简单选择排序法和堆排序法。

1.1.2 全真试题解答

一、选择题

(1) 算法的时间复杂度是指_____。

- A) 执行算法程序所需要的时间
- B) 算法程序的长度
- C) 算法执行过程中所需要的基本运算次数
- D) 算法程序中的指令条数

答案: C

分析: 本题考察考生对算法的时间复杂度概念的理解,如果考生注意到了“所谓算法的时间复杂度,是指执行算法所需要的计算工作量。算法的工作量可以用算法所执行的基本运算次数来度量”这句定义,就很快可以做出对 C) 项的选择了。A) 项的迷惑性比较大,考生容易望文生义地把它选上去。B) 项和 D) 项的意思差不多都是想误导考生以算法程序的长短论它的时间复杂度。实际上,算法程序长的算法不一定比算法程序短的算法的时间复杂度高,如一个 10 000 行的顺序执行的算法和一个只有 2 000 行,但其中有 1 000 行循环执行了 50 次的算法比较,前者的时间复杂度是 10 000,后者的则是 51 000。显然后者的时间复杂度更高。

(2) 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) 算法程序所占的存储空间
- D) 算法执行过程中所需要的存储空间

答案: D

分析: 本题考察考生对算法的空间复杂度概念的理解。算法的空间复杂度是指算法在执行过程中所需要的存储空间,这包括静态地存储算法程序指令的存储空间,还包括算法程序动态地执行过程中为中间结果和数据结构开辟的数据存储空间。题中 A)、B)、C) 三项都只考虑到了影响算法程序静态存储空间大小的因素。

(3) 下列叙述中正确的是_____。

- A) 线性表是线性结构 B) 栈和队列是非线性结构
C) 线性链表是非线性结构 D) 二叉树是线性结构

答案: A

分析: 线性结构和非线性结构是一对互斥的概念。通常,一个数据结构不是线性结构,就一定是非线性结构。在数据结构中线性表、线性链表、栈和队列都是线性结构;树和图是非线性结构。

(4) 数据的存储结构是指_____。

- A) 数据所占的存储空间量 B) 数据的逻辑结构在计算机中的表示
C) 数据在计算机中的顺序存储方式 D) 存储在外存中的数据

答案: B

分析: 本题考查考生对数据存储结构和逻辑结构的概念及区别和联系的理解。数据的逻辑结构是指数据元素之间的逻辑关系的数据结构。数据的存储结构则是数据的逻辑结构在计算机中的物理实现,有时也称做数据的物理结构。两者的区别是数据的逻辑结构只涉及数据之间抽象的数学关系。存储结构则涉及如何在计算机中通过对数据的物理存储进行组织来表达数据元素之间的逻辑关系,如在线性表的顺序存储中是利用物理存储空间上的连续性来表达线性表中数据的前趋、后继关系;在线性表的链式存储中是通过指针域构成的逻辑链条来表达数据的前趋、后继关系。通常一种数据的逻辑结构对应的物理实现,即数据的存储结构不止一种。

(5) 长度为 10 的顺序表的首地址是从 1023 开始的,顺序表中每个元素的长度为 2,在第 4 个元素前面插入一个元素和删除第 7 个元素后,顺序表的总长度不变。问在执行插入和删除操作前,顺序表中第 5 个元素在执行插入和删除操作后的存储地址是_____。

- A) 1028 B) 1029 C) 1031 D) 1033

答案: D

分析: 本题考查了线性表随机存取地址的计算公式以及线性表中的插入和删除操作前后元素的对应关系。由于问的是原来顺序表中的第 5 个元素,它在插入操作后变成了第 6 个元素(因为插入的元素在它前面)。由于删除的第 7 个元素在它后面,不会影响它在顺序表中的排位。因此在执行插入和删除操作后,原先顺序表中的第 5 个元素变成了新顺序表中的第 6 个元素。再按照线性表的随机存取地址的计算公式:

$$\text{ADD}(a_i) = \text{ADD}(a_1) + (i - 1) \times k$$

计算:

$$\begin{aligned} \text{ADD}(a_6) &= \text{ADD}(a_1) + (6 - 1) \times 2 \\ &= 1023 + 5 \times 2 \\ &= 1033 \end{aligned}$$

因此答案选 D。

(6) 下列关于线性表的两种存储结构叙述正确的是_____。

- A) 存储相同数目的元素,线性链表比顺序表要节省存储空间
B) 对无序表的查找,顺序表和线性链表的效率是一样的
C) 顺序表适用于插入、删除等更新操作频繁的场所
D) 线性链表适用于查询操作比较频繁的场所

答案: B

分析: 对线性表的两种存储结构的比较能够加深对顺序表和线性链表这两种存储结构的理解。顺序表通过在内存中分配一块连续的空间,用物理上的相邻关系表达数据间的逻辑相邻关系;而线性链表则是通过增加一个指针域来把相邻的数据元素链接成一个线性序列。因此存储相同数目的元素线性链表比顺序表要多消耗一些存储空间来存放后继指针。但是线性链表的这种结构使得它存储数据的空间可以是离散的,并不像顺序表那样一定要求物理上的连续空间。因此在它里面增加和删除元素的操作很简单,只需要分配一个数据结点,然后把它挂接到正确的地方就行了。而在顺序表中要增加和删除一个数据元素,为了保证数据在物理空间上的连续性,不得不大量移动数据元素。因此 C)、D) 两项也可以排除了。至于顺序表和线性链表在查询效率上的比较问题,对于无序表的查找,不管是用顺序表还是线性链表,都只能用顺序查找的方法把线性表遍历一遍,因此效率大致是相同的。顺序表只是在数据组织成一种有序的结构时采用二分查找方法要比线性链表的速度快。

(7) 下列关于栈的叙述中不正确的是_____。

- A) 在栈中只能在一端插入、删除数据
- B) 在栈中只能在一端插入数据,在另一端删除数据
- C) 栈是先进后出的线性表
- D) 栈是后进先出的线性表

答案: B

分析: 栈是限制在一端对数据进行插入和删除的线性表,它对数据的操作顺序是先进后出,也就是后进先出。考生在做这道题时如果不明白先进后出和后进先出指的是同一回事,很可能在 C) 和 D) 中选择。至于 B) 项说的是队列的特性。

(8) 在线性链表的插入算法中,若要把结点 q 插在结点 p 后面,下列操作正确的是_____。

- A) 使结点 p 指向结点 q,再使结点 q 指向结点 p 的后继结点
- B) 使结点 q 指向 p 的后继结点,再使结点 p 指向结点 q
- C) 使结点 q 指向结点 p,再使结点 p 指向结点 q 的后继结点
- D) 使结点 p 指向 q 的后继结点,再使结点 q 指向结点 p

答案: B

分析: 在修改结点指针域的操作中,有一个操作顺序的问题。比较 A) 和 B) 只是操作顺序颠倒了一下。A) 中先使结点 p 指向 q 后,q 就成为 p 新的后继结点了,原先通过结点 p 指向的后继结点与结点 p 脱节了。那么后面的一步操作没有任何意义,因为使结点 q 指向 p 的后继结点就是使结点 q 成为自己的后继结点。按照 B) 指定的顺序操作就不会出现在引用结点 p 的指针域之前已经把它值修改了的情形。至于 C) 和 D) 项是命题者设计的干扰项,想让考生把 p 和 q 的顺序搞混。

(9) 下列叙述中错误的是_____。

- A) 循环链表中,通过表中的任何一个结点可以访问到表中其他所有的结点
- B) 线性链表插入和删除效率比顺序表的插入和删除效率高
- C) 线性链表与顺序表相比,它容易实现动态增长
- D) 在线性链表中查找一个元素要比在顺序表中查找一个元素快

(13) 下面排序算法中,平均排序速度最快的是_____。

A) 冒泡排序法 B) 选择排序法 C) 交换排序法 D) 堆排序法

答案: D

分析: 在各种排序方法中,快速排序法和堆排序法的平均速度是最快的,因为它们的时间复杂度都是 $O(n \log_2 n)$,其他的排序算法的时间复杂度大都是 $O(n^2)$ 。

二、填空题

(1) 已知被查项 x 在数组中出现的概率为 q ,且需要查找的 x 在数组中每个位置上的可能性是一样的。试分析采用顺序搜索方法,在长度为 n 的一维数组中查找值为 x 的元素的算法中,算法的平均时间复杂度是____【1】____,最坏情况下,它的时间复杂度是____【2】____。

答案: 【1】 $(n+1)q/2 + (1-q)n$ 【2】 n

分析: 在搜索算法中,基本运算是被查值 x 与数组元素的比较操作。当被查值 x 为数组中的第 i 个元素时,则在查找过程中需要做 i 次比较,当需要查找的 x 不在数组中时,则需要与数组中所有的元素进行比较,即

$$t_i = \begin{cases} i, & 1 \leq i \leq n \\ n, & i = n+1 \end{cases}$$

由于需要查找的 x 出现在数组中每一个位置上的概率是一样的,因此 x 出现在数组中每一个位置上的概率为 q/n ,而 x 不在数组中的概率为 $1-q$,即

$$p_i = \begin{cases} q/n, & 1 \leq i \leq n \\ 1-q, & i = n+1 \end{cases}$$

因此,用顺序搜索法在长度为 n 的一维数组中查找值为 x 的元素,在平均情况下需要做的比较次数为

$$A(n) = \sum_{i=1}^{n+1} p_i t_i = \sum_{i=1}^n (q/n)i + (1-q)n = (n+1)q/2 + (1-q)n$$

最坏情形时,查找的 x 是数组的最后一个元素或是不在数组中,此时显然有

$$W(n) = \max\{t_i \mid 1 \leq i \leq n+1\} = n$$

(2) 假如刚开始时栈为空,依次有 A、B、C、D 4 个元素入栈,此时栈底指针指向元素____【1】____,栈顶指针值为____【2】____(假设每个元素的长度为 1)。执行 4 次出栈操作后把 E、F、G 压入栈,问此时栈底指针指向元素____【3】____,此时栈的长度为____【4】____。

答案: 【1】 A 【2】 4 【3】 E 【4】 3

分析: 这种题目用图解法最方便。用 $()$ 表示空栈,当 A 入栈时表示为 (A) ,当 B 入栈时表示为 (A,B) ,总之这种表示法中最右边的元素为栈顶元素,最左边元素为栈底元素,栈的长度为括号中元素的个数,栈顶指针的值等于栈的长度。入栈时把元素插在最右边,出栈时把最右边的元素剔除。当 A、B、C、D 依次入栈时,栈的状态为 (A,B,C,D) ,栈底指针指向 A,栈顶指针值为 4。执行 4 次出栈后栈为空,这个过程可以表示为

$$(A,B,C,D) \rightarrow (A,B,C) \rightarrow (A,B) \rightarrow (A) \rightarrow ()$$

最后 E、F、G 入栈后栈的状态为 (E,F,G) ,此时栈底指针指向 E,栈的长度为 3。

(3) 一个容量为 8 的循环队列,当它的队首和队尾指针相等时 ($front = rear$),队列中有_____个有效数据。

答案: 0 或 8

分析：对于循环队列来说，队首和队尾指针相等时对应有两个状态：队列空和队列满（队尾指针转了一圈又赶上了队首指针）。因此在循环队列中不得不另外设一个标志 s 来表示队列是否为空，若 $s=0$ 则队列为空，若 $s=1$ ，且 $front=rear$ 时队列为满。要判断队列是否为空或满是不能单看 $front$ 等不等于 $rear$ 。

(4) 图 1.2 中二叉树表示的表达式为_____。

答案： $A \times (B/C) + A$

分析：按树的中序遍历就得到了计算序列，只要注意一下运算顺序的优先级，在有的地方要加上括号。

(5) 拥有奇数个结点的完全二叉树中有 4 个内部结点（非叶子结点），它的叶子结点数是_____。

答案：5

分析：此题综合考查了完全二叉树的性质和二叉树的基本性质。由于完全二叉树是自上而下，自左而右地从 1 开始连续编码的，因此完全二叉树要么不存在一度结点（当结点个数为奇数个时），要么存在一个一度结点，而且惟一的一个一度结点就是最后编号为 n （ n 为偶数）的叶子结点的父结点。而在二叉树中零度结点个数总比二度结点个数多 1，因此拥有 4 个二度结点的二叉树的叶子结点的个数是 $4+1=5$ 。

(6) 如图 1.3 所示，二叉树的前序遍历序列是_____【1】_____，中序遍历序列是_____【2】_____，后序遍历序列是_____【3】_____。

答案：【1】前序序列：A, B, D, E, G, H, C, F

【2】中序序列：D, B, G, E, H, A, C, F

【3】后序序列：D, G, H, E, B, F, C, A

分析：在重点掌握二叉树的中序遍历的同时需要了解其他两种遍历方式。本题若是一道选择题，要求从答案提供的 3 个序列中选出二叉树的中序遍历序列，就应该能够通过前序遍历首先访问的是根结点和后序遍历最后才访问根结点的特性对前序序列和后序序列进行快速排除。

(7) 用二分查找法在有序顺序表(1, 2, 3, 4, 6, 8, 9, 11)中查找 3 的比较序列为_____。

答案：4, 2, 3

分析：可采用擦去法做这类二分法查找序列的题。每次从序列中找出中间元素，刚开始时是 4，由于 3 比 4 小，只能存在于 4 之前的序列中，于是把 4 以后的序列擦去，只剩下序列(1, 2, 3)，重复以上过程直到找到查找元素或是序列为空。

(8) 请填写表 1.1 中的空白部分。

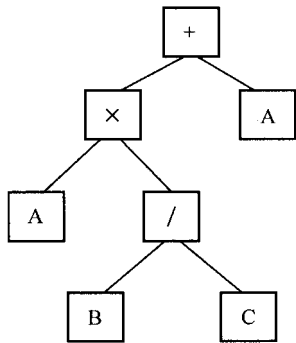


图 1.2 二叉树示例一

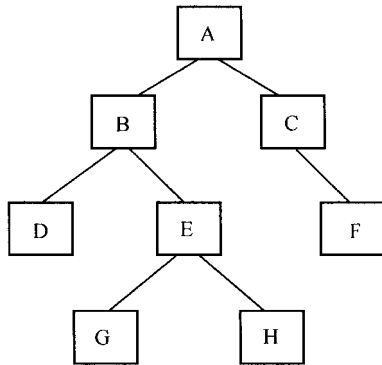


图 1.3 二叉树示例二