



全国计算机等级考试命题研究中心 / 编著
飞思数字创意出版中心 / 监制

全国计算机等级考试

National Computer Rank Examination

飞思考试中心
Feicit Examination Center



考点分析、 题解与模拟

二级C(最新版)

含

无纸化真题库
二级公共基础知识

新大纲 新题型 新题库

考点精编 专家分析最新大纲与真题库，权威揭秘各考点考核概率

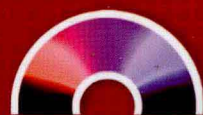
真题题解 精选真考试题，权威分析真考出题角度与考试难度

同步练习 习题源自真题库，练习即为真考，以练促学步步通关

真题库 全套真题库，与真考环境完全一致，系统智能评分



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>



考试模拟软件

全国计算机等级考试命题研究中心 / 编著
飞思数字创意出版中心 / 监制

全国计算机等级考试

National Computer Rank Examination

飞思考试中心
Fecit Examination Center

考点分析、 题解与模拟

二级C(最新版)



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发布的《全国计算机等级考试考试大纲》，在《全国计算机等级考试考点分析、题解与模拟（2012版）》的基础上修订而成。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、讲解，并选取经典考题进行了深入剖析；另一方面配有同步练习、无纸化考试试题，以逐步向考生详尽透析考试中的所有知识要点。“一书在手，通关无忧”。

本书配有“全国计算机等级考试模拟软件”。其中智能化的答题系统按照本书的顺序循序渐进、逐步编排；完全模拟真实考试，考试步骤、考试界面、考试方式、题目形式与真实考试完全一致，并可以自动评分。“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

全国计算机等级考试考点分析、题解与模拟：最新版. 二级 C / 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，2013.1

（飞思考试中心）

ISBN 978-7-121-19122-0

I. ①全… II. ①全… III. ①电子计算机—水平考试—自学参考资料②C 语言—程序设计—水平考试—自学参考资料 IV. ①TP3

中国版本图书馆 CIP 数据核字（2012）第 286208 号

责任编辑：王树伟

特约编辑：赵树刚

印 刷：涿州市京南印刷厂

装 订：涿州市京南印刷厂

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：14.75 字数：566.4 千字

印 次：2013 年 1 月第 1 次印刷

定 价：33.80 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。

丛书编委会

主编 朱 伶

编委 (排名不分先后)

丁海艳	万克星	马立娟	朱爱彬
王 伟	王 亮	王强国	王 磊
卢文毅	卢继军	任海艳	乔 影
刘之夫	刘金丽	刘春波	孙小稚
张 迪	张仪凡	张海刚	李 静
范二朋	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	赵苡萱
王倩愉	倪海宇	高志军	高雪轩
董国明	谢公义	韩峻余	樊 钰

全国计算机等级考试自1994年由国家教育部考试中心推出以来,为评测全社会非计算机专业人员的计算机知识与技能,培养各行业的计算机应用人才开辟了一条新的道路,受到了用人单位和学习人员的热烈欢迎。全国计算机等级考试通过数年的发展,已经成为我国最大型的计算机类考试。

为了帮助更多的学习者顺利地通过考试,并掌握相应的操作技能,我们在深入调研、详尽分析考试大纲的基础上,组织国内著名高校的计算机专家和一线教师编写了本书。

本书共分为三大部分,同时配有一张学习软件光盘。

※ 考点分析/真题题解/同步练习

“考点分析”结合最新考试大纲、教材,对教材中考核的重点和难点进行了讲解,内容涵盖了大纲中所有的选择题和操作题的考点。

“真题题解”选取极具代表性的经典例题。例题符合考试命题规律的特征,对题目的讲解深入、透彻,循序渐进,极有条理。

“同步练习”提供了大量习题,对前面所学的理论知识进行温习和巩固,以练促学、学练结合。

※ 无纸化考试试题

结合最新考试大纲,筛选与演绎出的典型试题集,不论在形式上还是难度上,都与真题类似,使学习者熟悉整个考试过程,了解考试的题型、题量,并配有详细的解析,使学习者既能知其然,也能知其所以然。

※ 配套学习软件

本书配套光盘具有如下特色:

- 超大量仿真考试模拟试卷,自动组卷,即时评分,由专家对答题结果进行“现场指导”。
- 从抽题、答题到交卷完全模拟真实考试,唯一不同之处是可以对上机作答进行评分。

本书所有操作题都经过上机调试通过。由于时间仓促,书中难免有不当之处,敬请指正。

全国计算机等级考试命题与研究中心
飞思数字创意出版中心
未来教育教学与研究中心

第0章 公共基础知识

0.1 数据结构与算法	1	0.4 数据库设计基础	20
0.2 程序设计基础	11	0.5 同步练习	26
0.3 软件工程基础	13	0.6 同步练习答案	27

第1章 C语言概述

1.1 C语言基础知识	28	1.3 同步练习	35
1.2 常量、变量和数据类型	30	1.4 同步练习答案	36

第2章 运算符与表达式

2.1 C语言运算符简介	37	2.4 位运算	42
2.2 算术运算符和算术表达式	38	2.5 同步练习	43
2.3 赋值运算符和赋值表达式	40	2.6 同步练习答案	45

第3章 基本语句

3.1 C语句概述	46	3.5 数据格式的输入与输出	49
3.2 赋值语句	47	3.6 同步练习	53
3.3 输入/输出的概念及其实现	48	3.7 同步练习答案	56
3.4 单个字符的输入/输出	48		

第4章 选择结构

4.1 关系运算符和关系表达式	57	4.4 switch语句和goto语句	62
4.2 逻辑运算符和逻辑表达式	58	4.5 同步练习	64
4.3 if语句和用if语句构成的选择结构	59	4.6 同步练习答案	68

第5章 循环结构

5.1 概述	69	5.6 几种循环的比较	75
5.2 while语句	69	5.7 break语句和continue语句	75
5.3 do...while语句	70	5.8 同步练习	77
5.4 for语句	71	5.9 同步练习答案	80
5.5 循环的嵌套	74		

第6章 数 组

6.1 一维数组的定义和引用	81	6.4 同步练习	87
6.2 二维数组的定义和引用	83	6.5 同步练习答案	90
6.3 字符数组	84		

第7章 函 数

7.1 概述	91	7.3 函数定义的一般形式	92
7.2 库函数	92	7.4 函数参数和函数返回值	94

7.5 函数的调用	95	7.9 变量的存储类别	100
7.6 函数的嵌套调用与递归调用	97	7.10 内部函数和外部函数	103
7.7 数组作为函数参数	99	7.11 同步练习	103
7.8 全局变量和局部变量	99	7.12 同步练习答案	107

第8章 指 针

8.1 关于地址和指针	108	8.6 返回指针值的函数	116
8.2 变量的指针和指向变量的指针变量	109	8.7 指针数组和指向指针的指针	117
8.3 数组与指针	110	8.8 同步练习	118
8.4 字符串与指针	113	8.9 同步练习答案	121
8.5 指向函数的指针	115		

第9章 编译预处理和动态存储分配

9.1 宏定义	122	9.4 关于动态存储的函数	126
9.2 文件包含	123	9.5 同步练习	127
9.3 条件编译	125	9.6 同步练习答案	129

第10章 结构体与共用体

10.1 用 typedef 说明一种新类型名	130	10.6 指向结构体类型数据的指针	134
10.2 结构体类型	130	10.7 链表	135
10.3 结构体类型变量的定义	131	10.8 共用体	139
10.4 结构体变量的引用	132	10.9 同步练习	141
10.5 结构体数组	133	10.10 同步练习答案	143

第11章 文 件

11.1 C 语言文件的概念	144	11.5 文件的定位	148
11.2 文件类型指针	145	11.6 同步练习	149
11.3 文件的打开与关闭	145	11.7 同步练习答案	151
11.4 文件的读写	147		

第12章 操作题高频考点

12.1 C 程序设计基础	152	12.5 数组	159
12.2 C 语言的基本结构	154	12.6 字符串	162
12.3 函数	157	12.7 结构体、共用体和用户定义类型	164
12.4 指针	158	12.8 文件	166

第13章 无纸化考试试题

13.1 无纸化考试试题(1)	168	13.5 无纸化考试试题(5)	200
13.2 无纸化考试试题(2)	176	13.6 参考答案及解析	208
13.3 无纸化考试试题(3)	184	13.7 无纸化考试试题及解析(6)~(105)(见光盘)	224
13.4 无纸化考试试题(4)	192		

附 录

附录 A 常用字符与 ASCII 码对照表	225	附录 C 运算符的优先级与结合性	227
附录 B C 语言关键字	226		

第0章 公共基础知识

考核知识点

- 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

重要考点提示

- 栈和队列
- 树与二叉树
- 结构化分析方法
- 数据库系统的基本概念
- 数据模型
- 关系代数

0.1 数据结构与算法

考核概率 100%

考点 1 算 法

1. 算法的基本概念

算法是指一系列解决问题的清晰指令。

(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,即必须有一个或多个输出,即使在数学理论上是正确的,如果在实际的计算工具上不能执行,则该算法也是不具有可行性的。

• 确定性:是指算法中每一步骤都必须是有明确定义的。

• 有穷性:是指算法必须能在有限的时间内做完。

• 拥有足够的情报:一个算法是否有效,还取决于为算法所提供的情报是否足够。

(2) 算法的基本要素。

算法一般由两种基本要素构成:

• 对数据对象的运算和操作。

• 算法的控制结构,即运算和操作时间的顺序。

算法中对数据的运算和操作:算法就是按解题要求从指令系统中选择合适的指令组成的指令序列。因此计算机算法就是计算机能执行的操作所组成的指令序列。不同的计算机系统,指令系统是有差异的,但一般的计算机系统中包括的运算和操作有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、减半递推技术和回溯法。

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。

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。

一般情况下,算法的工作量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:

算法的工作量 $= f(n)$

其中 n 表示问题的规模。该表达式表示随着问题规模 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。

算法的空间复杂度一般是指执行这个算法所需要的内存空间。算法执行期间所需要的存储空间包括以下 3 个部分:

- 算法程序所占的空间。
- 输入的初始数据所占的存储空间。
- 算法执行过程中所需要的额外空间。

在实际操作中,为了减少算法所占的存储空间,通常采用压缩存储的技术,用于减少不必要的额外空间。

考点 2 数据结构的基本概念

1. 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,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(即前后件关系)的数据结构。它包括两个要素,数据元素的集合和数据元素之间的关系。

(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(也称为数据的物理结构)。数据结构的存储方式包括顺序存储方法、链式存储方法、索引存储方法和散列存储方法。而采用不同的存储结构,其数据处理的效率是不同的。因此,在进行数据处理时,选择合适的存储结构是很重要的。

数据结构研究的内容主要包括 3 个方面:

- 数据集合中各数据元素之间的逻辑关系,即数据的逻辑结构。
- 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。
- 对各种数据结构进行的运算。

2. 数据结构的图形表示

数据元素之间最基本的关系是前后件关系。前后件关系,即每一个二元组,都可以用图形来表示。用中间标有元素值的方框表示数据元素,一般称为数据结点,简称为结点。对于每一个二元组,用一条有向线段从前件指向后件。

图形表示数据结构具有直观易懂的特点,在不引起歧义的情况下,前件结点到后件结点连线上的箭头可以省略。例如,树型结构中,通常都是用无向线段来表示前后件关系的。

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件关系的复杂程度,一般将数据结构分为两大类型,即线性结构和非线性结构。

如果一个非空的数据结构满足有且只有一个根结点,并且每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,又称线性表。如果不满足上述条件的数据结构则称为非线性结构。

考点 3 线性表及其顺序存储结构**1. 线性表的基本概念**

在数据结构中,线性结构又称为线性表,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,除表中的第一个元素外,有且只有一个前件;除了最后一个元素外,有且只有一个后件。

线性表或者是个空表,或者可以表示为:

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

其中, $a_i (i=1, 2, \dots, n)$ 是线性表的数据元素,也称为线性表的一个结点。

每个数据元素在不同情况下其具体含义各不相同,它可以是一个数或一个字符,也可以是一个具体的事物,甚至是更复杂的信息。但是需要注意的是,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。

TIPS 小提示

非空线性表具有以下一些结构特征:

- 只有一个根结点,即头结点,它无前件;
- 有且只有一个终结点,即尾结点,它无后件;
- 除头结点与尾结点外,其他所有结点有且只有一个前件,也有且只有一个后件。结点个数 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

将线性表中的元素逐个存储在一片相邻的存储区域中。这种顺序表示的线性表也称为顺序表。

线性表的顺序存储结构具有以下两个基本特点:

- 元素所占的存储空间必须是连续的。
- 元素在存储空间的位置是按逻辑顺序存放的。

从上述特点也可以看出,线性表是用元素在计算机内物理位置上的相邻关系来表示元素之间逻辑上的相邻关系。只要确定了首地址,线性表内任意元素的地址都可以方便地计算出来。

3. 线性表的插入运算

在线性表的插入运算中,在第 i 个元素之前插入一个新元素,完成插入操作主要有以下 3 个步骤。

- (1) 把原来第 n 个结点至第 i 个结点依次往后移一个元素位置。
- (2) 把新结点放在第 i 个位置上。
- (3) 修正线性表的结点个数。

TIPS 小提示

一般会为线性表开辟一个大于线性表长度的存储空间,经过多次插入运算,可能出现存储空间已满的情况,如果此时仍继续插入运算,将会产生错误,此类错误称为“上溢”。

如果需要在线性表末尾进行插入运算,则只需在表的末尾增加一个元素即可,不需要移动线性表中的元素。

如果在第一个位置插入新的元素,则需要移动表中所有的元素。

4. 线性表的删除运算

在线性表的删除运算中,删除第 i 个位置的元素,则要从第 $i+1$ 个元素开始,直到第 n 个元素之间共

$n-i$ 个元素依次向前移一个位置,完成删除运算主要有以下几个步骤。

- (1) 把第 i 个元素之后 (不包括第 i 个元素) 的 $n-i$ 个元素依次前移一个位置。
- (2) 修正线性表的结点数。

综上所述,如果删除运算在线性表的末尾进行,即删除第 n 个元素,则不需要移动线性表中的元素。如果要删除第 1 个元素,则需要移动表中的所有数据。

TIPS 小提示

由线性表的上述性质可以看出,线性表的顺序存储结构适用于小线性表,或者建立之后其中元素不常变动的线性表,而不适用于需要经常进行插入和删除运算的线性表和长度较大的线性表。

考点 4 栈和队列

1. 栈及其基本运算

(1) 栈的基本概念。

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,其插入运算与删除运算都只在线性表的一端进行。

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)。若栈中没有元素称为空栈,栈也被称为“先进后出”表,或“后进先出”表。

(2) 栈的特点。

根据栈的上述定义,栈具有以下特点:

- 栈顶元素总是最后被插入的元素,也是最早被删除的元素。
- 栈底元素总是最早被插入的元素,也是最后才能被删除的元素。
- 栈具有记忆作用。
- 在顺序存储结构下,栈的插入和删除运算都不需要移动表中其他数据元素。
- 栈顶指针 top 动态反映了栈中元素的变化情况。

(3) 栈的顺序存储及其运算。

栈的状态如图 0-1 所示。

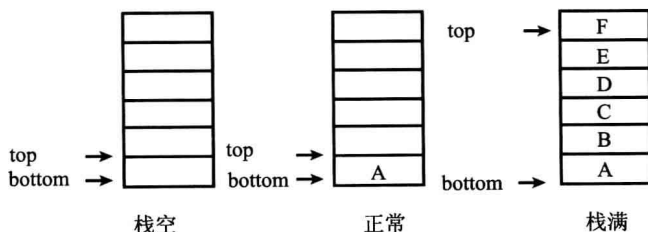


图 0-1 栈的状态

根据栈的状态,可以得知栈的基本运算有以下 3 种。

- 入栈运算:在栈顶位置插入一个新元素。
- 退栈运算:取出栈顶元素并赋给一个指定的变量。
- 读栈顶元素:将栈顶元素赋给一个指定的变量。

2. 队列及其基本运算

(1) 队列的基本概念。

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。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;允许删除的一端称为排头,通常用一个头指针(front)指向头元素的前一个位置。

因此,队列又称为“先进先出”(First In First Out, FIFO)的线性表。插入元素称为入队运算,删除元素称为退队运算。

队列的基本结构如图 0-2 所示。

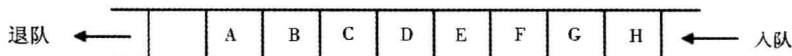


图 0-2 队列

(2) 循环队列及其运算。

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。

在循环队列中,用尾指针(rear)指向队列的尾元素,用头指针(front)指向头元素的前一个位置。因此,从头指针(front)指向的后一个位置,直到尾指针(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 $rear = front$ 。

循环队列的基本运算主要有两种:入队运算与退队运算。

- 入队运算是指在循环队列的队尾加入一个新的元素。
- 退队运算是指在循环队列的排头位置退出一个元素,并赋给指定的变量。

TIPS 小提示

栈是按照“先进后出”或“后进先出”的原则组织数据的,而队列是按照“先进先出”或“后进后出”的原则组织数据。这就是栈和队列的不同。

考点 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。

为了存储线性表中的每一个元素,一方面要存储数据元素的值;另一方面要存储各数据元素之间的前后件关系。为此,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,用于存放下一个数据元素的存储序号,即指向后件结点。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。

线性表链式存储结构的特点是用一组不连续的存储单元存储线性表中的各个元素。因为存储单元不连续,所以数据元素之间的逻辑关系就不能依靠数据元素的存储单元之间的物理关系来表示。

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

(1) 循环链表的定义。

在单链表的第一个结点前增加一个表头结点,队头指针指向表头结点,在最后一个结点的指针域的值由 NULL 改为指向表头结点,这样的链表称为循环链表。在循环链表中,所有结点的指针构成了一个环状链。

(2) 循环链表与单链表的比较。

对单链表的访问是一种顺序访问,从其中某一个结点出发,只能找到它的直接后继,但无法找到它的直接前驱。而且对于空表和第一个结点的处理必须单独考虑,空表与非空表的操作不统一。

在循环链表中,只要指出表中任何一个结点的位置,就可以从它出发访问到表中其他所有的结点。由于表头结点是循环链表所固有的结点,因此,即使在表中没有数据元素的情况下,表中也至少有一个结点存在,从而使空表和非空表的运算统一。

考点 6 树和二叉树

1. 树的基本概念

树是一种简单的非线性结构,直观地来看树是以分支关系定义的层次结构。树是由 $n(n \geq 0)$ 个结点构成的有限集合,当 $n=0$ 时,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。

在树的结构中主要涉及下面几个概念。

- 每一个结点只有一个前件,称为父结点,没有前件的结点只有一个,称为树的根结点,简称树的根。
- 每一个结点可以有多个后件,称为该结点的子结点。没有后件的结点称为叶子结点。
- 一个结点所拥有的后继个数称为该结点的度。
- 所有结点最大的度称为树的度。
- 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,是一个有限的结点集合,该集合或者为空,或者由一个根结点及其两棵互不相交的左右二叉子树组成。当集合为空时,称该二叉树为空二叉树。

二叉树具有以下特点。

- 二叉树可以为空,空的二叉树没有结点,非空二叉树有且只有一个根结点。
- 每一个结点最多有两棵子树,且分别称为该结点的左子树与右子树。

(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,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($k \geq 1$)。
- 深度为 m 的满二叉树中有 $2^m - 1$ 个结点。
- 对于任何一棵二叉树,度为 0 的结点(即叶子结点)总是比度为 2 的结点多一个。
- 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

在计算机中,二叉树通常采用链式存储结构,用于存储二叉树中各元素的存储结点由数据域和指针域组成。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。因此,二叉树的链式存储结构也称为二叉链表。

对于满二叉树与完全二叉树可以按层次进行顺序存储。

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。二叉树的遍历主要是针对非空二叉树的,对于空二叉树,则结束返回。

二叉树的遍历有前序遍历、中序遍历和后序遍历。

(1) 前序遍历(DLR)。

首先访问根结点,然后遍历左子树,最后遍历右子树。

(2) 中序遍历(LDR)。

首先遍历左子树,然后访问根结点,最后遍历右子树。

(3) 后序遍历(LRD)。

首先遍历左子树,然后遍历右子树,最后访问根结点。

TIPS 小提示

已知一棵二叉树的前序遍历序列和中序遍历序列,可以唯一确定这棵二叉树。已知一棵二叉树的后序遍历序列和中序遍历序列,也可以唯一确定这棵二叉树。已知一棵二叉树的前序遍历序列和后序遍历序列,不能唯一确定这棵二叉树。

考点 7 查找技术

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。

在最好的情况下,第一个元素就是要查找的元素,则比较次数为1次。

在最坏的情况下,顺序查找需要比较 n 次。

在平均情况下,需要比较 $n/2$ 次。因此查找算法的时间复杂度为 $O(n)$ 。

在下列两种情况下只能够采取顺序查找:

- 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。

2. 二分查找

使用二分法查找的线性表必须满足以下两个条件:

- 顺序存储结构。
- 线性表是有序表。

所谓有序表,是指线性表中的元素按值非递减排列(即从小到大,但允许相邻元素值相等)。

对于长度为 n 的有序线性表,利用二分法查找元素 x 的过程如下:

(1) 将 x 与线性表的中间项进行比较。

(2)若中间项的值等于 x ,则查找成功,结束查找。

(3)若 x 小于中间项的值,则在线性表的前半部分以二分法继续查找。

(4)若 x 大于中间项的值,则在线性表的后半部分以二分法继续查找。

这样反复进行查找,直到查找成功或子表长度为 0(说明线性表中没有这个元素)为止。

当有序线性表为顺序存储时采用二分查找的效率要比顺序查找高得多。对于长度为 n 的有序线性表,在最坏的情况下,二分查找只需比较 $\log_2 n$ 次,而顺序查找则需要比较 n 次。

考点 8 排序技术

1. 交换类排序法

交换类排序法是指借助数据元素的“交换”进行排序的一种方法。本节介绍的冒泡排序法和快速排序法就属于交换类排序法。

(1) 冒泡排序法。

冒泡排序的基本思想:在线性表中依次查找相邻的数据元素,将表中最大的元素不断往后移动,反复操作直到消除所有逆序。此时,该表已经排序结束。

冒泡排序法的基本过程如下。

①从表头开始往后查找线性表,在查找过程中逐次比较相邻两个元素的大小。若在相邻两个元素中,前面的元素大于后面的元素,则将它们交换。

②从后向前查找剩下的线性表(除去最后一个元素),同样,在查找过程中逐次比较相邻两个元素的大小。若在相邻两个元素中,后面的元素小于前面的元素,则将它们交换。

③对剩下的线性表重复上述过程,直到剩下的线性表变空为止,线性表排序完成。

假设线性表的长度为 n ,则在最坏的情况下,冒泡排序需要经过 $n/2$ 遍从前往后的扫描和 $n/2$ 遍从后往前扫描,需要比较 $n(n-1)/2$ 次,其数量级为 n^2 。

(2) 快速排序法。

快速排序法的基本思想:在线性表中逐个选取元素,将线性表进行分割,直到所有元素全部选取完毕,此时线性表已经排序结束。

快速排序法的基本过程如下。

①从线性表中选取一个元素,设为 T ,将线性表后面小于 T 的元素移动到前面,而将大于 T 的元素移到后面,这样就将线性表分成了两部分(称为两个子表)。 T 就是处于分界线的位置,将线性表分成了前后两个子表,且前面子表中的所有元素均不大于 T ,而后子表中的所有元素均不小于 T ,此过程称为线性表的分割。

②对分割后的子表再按上述原则进行反复分割,直到所有子表为空结束,则此时的线性表就变成有序。

2. 插入类排序法

插入排序是指将无序序列中的各元素依次插入已经有序的线性表中。下面将主要介绍简单插入排序法和希尔排序法。

(1) 简单插入排序法。

简单插入排序是把 n 个待排序的元素看成一个有序表和一个无序表,开始时,有序表只包含一个元素,而无序表包含 $n-1$ 个元素,每次取无序表中的第一个元素插入到有序表中的正确位置,使之成为增加一个元素的新的有序表。插入元素时,插入位置及其后的记录依次向后移动。最后有序表的长度为 n ,而无序表为空,此时排序完成。

在简单插入排序中,每一次比较后最多移除一个逆序,因此,该排序方法的效率与冒泡排序法相同。一般简单插入排序需要 $n(n-1)/2$ 次比较。

(2) 希尔排序法。

希尔排序法是将整个无序序列分割成若干个小的子序列并分别进行插入排序。

分割方法如下：

①将相隔某个增量 h 的元素构成一个子序列。

②在排序过程中,逐次减少这个增量,直到 h 减到 1 时,进行一次插入排序,排序即可完成。

希尔排序的效率与所选取的增量序列有关。

3. 选择类排序法

选择排序是通过每次从待排序序列中选出的最小值元素,顺序放在已排好序的有序子表的后面,直到全部序列满足排序要求为止。下面就介绍选择类排序法中的简单选择排序法和堆排序法。

(1) 简单选择排序法。

进行简单选择排序,首先从所有 n 个待排序的数据元素中选择最小的元素,将该元素与第一个元素交换,再从剩下的 $n-1$ 个元素中选出最小的元素与第二个元素交换。重复这样的操作直到所有的元素有序为止。

简单选择排序需要比较 $n(n-1)/2$ 次。

(2) 堆排序法。

堆排序的方法如下：

①将一个无序序列建成堆。

②将堆顶元素与堆中最后一个元素交换。忽略已经交换到最后的那个元素,考虑前 $n-1$ 个元素构成的子序列,只有左、右子树是堆,可以将该子树调整为堆。这样反复做第二步,直到剩下的子序列空为止。

堆排序需要比较的次数为 $O(n\log_2 n)$ 。

真题题解

【例1】下列叙述中正确的是()。

- 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

答案:A)

解析:在计算机中,数据的存储结构对数据的执行效率有较大影响,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很高。

【例2】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

答案:B)

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

【例3】在一个长度为 n 的顺序表中,向第 i 个元素 ($1 \leq i \leq n+1$) 的位置插入一个新元素,需要从后向前依次移动()个元素。

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

答案:D)

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,所以共需要移动 $n-i+1$ 个元素。

【例4】下列对队列的叙述正确的是()。

- A) 队列属于非线性表
C) 队列在队尾删除数据

- B) 队列按“先进后出”原则组织数据
D) 队列按“先进先出”原则组织数据

答案:D)

解析:队列是一种特殊的线性表,它只能在一端进行插入,在另一端进行删除。允许插入的一端称为队尾,允许删除的另一端称为队头。队列又称为“先进先出”或“后进后出”的线性表,体现了“先到先服务”的原则。

【例 5】下列关于栈的描述正确的是()。

- A) 在栈中只能插入元素而不能删除元素
B) 在栈中只能删除元素而不能插入元素
C) 栈是特殊的线性表,只能在一端插入或删除元素
D) 栈是特殊的线性表,只能在一端插入元素,而在另一端删除元素

答案:C)

解析:栈是一种特殊的线性表。在这种特殊的线性表中,其插入和删除操作只能在线性表的一端进行。

【例 6】下列叙述中正确的是()。

- 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

答案:A)

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,也称线性表。若不满足上述条件,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。

【例 7】对图 0-3 中的二叉树进行后序遍历的结果为()。

- A) ABCDEF
B) DBE AFC
C) ABDECF
D) DEBFCA

答案:D)

解析:执行后序遍历,依次执行如下操作:

- ① 首先按照后序遍历的顺序遍历根结点的左子树。
- ② 然后按照后序遍历的顺序遍历根结点的右子树。
- ③ 最后访问根结点。

【例 8】下列数据结构中,能用二分法进行查找的是()。

- A) 顺序存储的有序线性表
B) 线性链表
C) 二叉链表
D) 有序线性链表

答案:A)

解析:二分法查找只适用于顺序存储的有序表。所谓有序表是指线性表中的元素按值非递减排列(即从小到大,但允许相邻元素值相等)。

【例 9】对于长度为 n 的线性表,下列各排序法所对应的比较次数中正确的是()。

- A) 冒泡排序为 $n/2$
B) 冒泡排序为 n
C) 快速排序为 n
D) 快速排序为 $n(n-1)/2$

答案:D)

解析:假设线性表的长度为 n ,则冒泡排序需要经过 $n/2$ 遍的从前往后扫描和 $n/2$ 遍的从后往前扫描,需要比较次数为 $n(n-1)/2$ 。快速排序法在最坏的情况下,比较次数也是 $n(n-1)/2$ 。

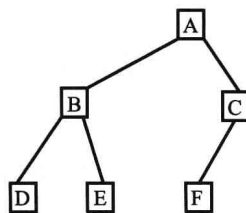


图 0-3