

大学生热门考试必备用书馆配经典系列

大学生热门考试 必备用书馆配经典系列

——**考研计算机**

历年真题及强化训练1000题(上)

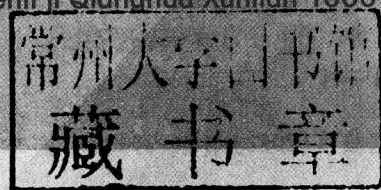
▶ 全国硕士研究生入学统一考试辅导用书编委会

大学生热门考试必备用书馆藏经典系列

大学生热门考试 必备用书馆藏经典系列 ——考研计算机 历年真题及强化训练1000题(上)

► 全国硕士研究生入学统一考试辅导用书编委会

Daxuesheng Remen Kaoshi Bibei Yongshu Guanpei Jingdian Xilie
——Kaoyan Jisuanji Linian Zheniti Ji Qianghua Xunlian 1000 ti (Shang)



高等教育出版社·北京
HIGHER EDUCATION PRESS BEIJING

图书在版编目 (C I P) 数据

考研计算机历年真题及强化训练 1000 题:全 2 册/
全国硕士研究生入学统一考试辅导用书编委会编. -- 北
京:高等教育出版社,2012. 5

(大学生热门考试必备用书馆配经典系列)

ISBN 978 - 7 - 04 - 035396 - 9

I. ①考… II. ①全… III. ①电子计算机-研究生-
入学考试-习题集 IV. ①TP3 - 44

中国版本图书馆 CIP 数据核字(2012)第 067394 号

策划编辑 刘 佳	责任编辑 柳秀丽	封面设计 赵 阳	版式设计 余 杨
插图绘制 尹 莉	责任校对 刘春萍	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 100120
印 刷 北京市密东印刷有限公司
开 本 787mm × 1092mm 1/16
总 印 张 24. 5
总 字 数 590 千字
购书热线 010 - 58581118

咨询电话 400 - 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>
版 次 2012 年 5 月第 1 版
印 次 2012 年 5 月第 1 次印刷
总 定 价 54. 00 元

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 35396 - 00

郑重声明

高等教育出版社依法对本书享有专有出版权。任何未经许可的复制、销售行为均违反《中华人民共和国著作权法》，其行为人将承担相应的民事责任和行政责任，构成犯罪的，将被依法追究刑事责任。为了维护市场秩序，保护读者的合法权益，避免读者误用盗版书造成不良后果，我社将配合行政执法部门和司法机关对违法犯罪的单位和个人给予严厉打击。社会各界人士如发现上述侵权行为，希望及时举报，本社将奖励举报有功人员。

反盗版举报电话：(010)58581897/58581896/58581879

传 真：(010)82086060

E-mail: dd@hep.com.cn

通信地址：北京市西城区德外大街4号

高等教育出版社法务部

邮 编：100120

购书请拨打读者服务部电话：(010)58581114/5/6/7/8

特别提醒：“中国教育考试在线”<http://www.eduexam.com.cn>是高教版考试用书专用网站。网站本着真诚服务广大考生的宗旨，为考生提供名师导航、下载中心、在线练习、在线考试、网上商城、网络课程等多项增值服务。高教版考试用书配有本网站的增值服务卡，该卡为高教版考试用书正版书的专用标识，广大读者可凭此卡上的卡号和密码登录网站获取增值信息，并以此辨别图书真伪。

目 录

第一部分 数据结构	1	第二章 进程管理	72
第一章 线性表	1	第三章 处理机调度与死锁	75
第二章 栈与队列、数组	4	第四章 存储器管理	77
第三章 树与二叉树	9	第五章 设备管理	79
第四章 图	14	第六章 文件管理	81
第五章 查找	18	第七章 操作系统接口	82
第六章 排序	23	第四部分 计算机网络	84
第二部分 计算机组成原理	28	第一章 计算机网络体系结构	84
第一章 计算机系统概论	28	第二章 物理层	86
第二章 数据的表示和运算	33	第三章 数据链路层	88
第三章 存储器层次结构	40	第四章 网络层	91
第四章 指令系统	52	第五章 传输层	94
第五章 中央处理器(CPU)	56	第六章 应用层	96
第六章 总线	61	第一部分答案及解析	99
第七章 输入/输出(I/O)系统	64	第二部分答案及解析	152
第三部分 操作系统原理	69	第三部分答案及解析	180
第一章 操作系统引论	69	第四部分答案及解析	204

第一部分 数据结构

数据结构科目对知识点的考查分为三种情况：

1. 掌握数据结构的基本概念、基本原理和基本方法。
2. 掌握数据的逻辑结构、存储结构及基本操作的实现,能够对算法进行基本的时间复杂度与空间复杂度的分析;比如,进栈和出栈操作,入队、出队操作,二叉树的遍历和线索化,图的存储与搜索,查找、排序等。
3. 对算法的考查,能够运用数据结构的基本原理和方法进行问题的分析与求解,具备采用 C、C++或 Java 语言设计与实现算法的能力。

第一章 线性表

知识点概要

1. 算法、数据结构

深刻理解算法、数据结构的概念,掌握数据结构的“三要素”:逻辑结构、物理(存储)结构及在这种结构上所定义的操作“运算”。

2. 时间复杂度、空间复杂度

理解时间复杂度和空间复杂度的概念。在数据结构中,常用计算语句频度来估算算法的时间复杂度,以下 6 种计算算法时间的多项式是最常用的。其关系为:

$$O(1) < O(\log_2 n) < O(n) < O(n \log_2 n) < O(n^2) < O(n^3)$$

指数时间的关系为:

$$O(2^n) < O(n!) < O(n^n)$$

3. 线性表

线性表是具有相同数据类型的 $n(n \geq 0)$ 个数据元素的有限序列,是最简单,最基本,也是最常用的一种线性结构。线性表的特点是表中的元素应具有相同的性质,除第一个结点和最后一个结点外,其他结点有且仅有一个前驱,一个后继。常用的存储方式有两种:顺序存储和链式存储。

4. 顺序表基本运算的实现

(1) 顺序表具有按数据元素的序号随机存取的特点,插入和删除需要移动大量元素,表的长度要事先设定。时间复杂度为 $O(1)$ 。

(2) 按值 x 查找:主要运算是比较,比较的次数与值 x 在表中的位置有关,也与表长有关,平均比较次数为 $(n+1)/2$,时间复杂度为 $O(n)$ 。

(3) 插入运算: 在第 i 个位置上插入 x , 从 a_n 到 a_i 都要向下移动一个位置, 共移动 $n-i+1$ 个元素。

在等概率情况下, 平均移动数据元素的次数为:

$$E_{in} = \sum_{i=1}^{n+1} p_i(n-i+1) = \frac{1}{n+1} \sum_{i=1}^{n+1} (n-i+1) = \frac{n}{2}$$

注意: 在顺序表上做插入操作需移动表中一半的数据元素, 时间复杂度为 $O(n)$ 。

(4) 删除运算: 删除第 i 个元素, 从 a_{i+1} 到 a_n 都要向上移动一个位置, 共移动 $n-i$ 个元素。

在等概率情况下, 平均移动数据元素的次数:

$$E_{de} = \sum_{i=1}^n p_i(n-i) = \frac{1}{n} \sum_{i=1}^{n+1} (n-i) = \frac{n-1}{2}$$

注意: 在顺序表上做删除运算时大约需要移动表中一半的数据元素, 时间复杂度为 $O(n)$ 。

5. 链表

链表需要动态地分配存储空间, 但每个结点要增加一个指针域。查找元素需要从头结点的指针域开始向后查找, 但容易插入和删除元素。注意链表的几种形式: 单链表、双链表、循环链表。

一、单项选择题

1. 下面程序段中带下划线的语句的执行次数的数量级是()。

```
i=1; WHILE i<n DO i=i*2;
```

A. $O(n)$ B. $O(\log_2 n)$ C. $O(n^2)$ D. $O(n \log_2 n)$

2. 算法的时间复杂度取决于()。

A. 问题的规模 B. 待处理数据的初态 C. A 和 B D. 以上都不正确

3. 一个算法应该是()。

A. 程序 B. 问题求解步骤的描述 C. 要满足 5 个基本特性 D. A 和 C

4. 下面说法错误的是()。

I. 算法原地工作的含义是指不需要任何额外的辅助空间

II. 在相同的规模 n 下, 复杂度 $O(n)$ 的算法在时间上总是优于复杂度 $O(2n)$ 的算法

III. 所谓时间复杂度, 是指在最坏情况下, 估算算法执行时间的一个上界

IV. 同一个算法, 实现语言的级别越高, 执行效率就越低

A. I B. I、II C. I、IV D. III

5. 从逻辑上可以把数据结构分为()两大类。

A. 动态结构、静态结构 B. 顺序结构、链式结构
C. 线性结构、非线性结构 D. 初等结构、构造型结构

6. 以下与数据的存储结构无关的术语是()。

A. 循环队列 B. 链表 C. 哈希表 D. 栈

7. 以下数据结构中, () 是线性数据结构。

A. 广义表 B. 二叉树 C. 稀疏矩阵 D. 串

8. 以下数据结构中, () 是非线性数据结构。

- A. 树 B. 字符串 C. 队 D. 栈
9. 连续存储设计时,存储单元的地址()。
- A. 一定连续 B. 一定不连续 C. 不一定连续 D. 部分连续,部分不连续
10. 以下属于逻辑结构的是()。
- A. 顺序表 B. 哈希表 C. 线性表 D. 单链表
11. 下面关于线性表的叙述中,错误的是()。
- A. 线性表采用顺序存储,必须占用一片连续的存储单元
- B. 线性表采用顺序存储,便于进行插入和删除操作
- C. 线性表采用链式存储,不必占用一片连续的存储单元
- D. 线性表采用链式存储,便于插入和删除操作
12. 若某线性表最常用的操作是存取任一指定序号的元素和在最后进行插入和删除运算,则利用()存储方式最节省时间。
- A. 顺序表 B. 双链表 C. 带头结点的双循环链表 D. 单循环链表
13. 若某表最常用的操作是在最后一个结点之后插入一个结点或删除最后一个结点。则采用()存储方式最节省运算时间。
- A. 单链表 B. 双链表 C. 单循环链表 D. 带头结点的双循环链表
14. 静态链表中指针表示的是()。
- A. 内存地址 B. 数组下标 C. 下一元素地址 D. 左、右孩子地址
15. 若长度为 n 的线性表采用顺序存储结构,在其第 i 个位置插入一个新元素的算法的时间复杂度为() ($1 \leq i \leq n+1$)。
- A. $O(0)$ B. $O(1)$ C. $O(n)$ D. $O(n^2)$
16. 线性表 $(a_1, a_2, \dots, a_n)$ 以链式存储方式存储时,访问第 i 位置元素的时间复杂度为()。
- A. $O(i)$ B. $O(1)$ C. $O(n)$ D. $O(i-1)$
17. 非空的循环单链表 head 的尾结点 p 满足()。
- A. $p \rightarrow \text{next} = \text{head}$ B. $p \rightarrow \text{next} = \text{NULL}$ C. $p = \text{NULL}$ D. $p = \text{head}$
18. 双向链表中有两个指针域,即 prior 和 next,分别指回前驱及后继,设 p 指向链表中的一个结点,q 指向一个待插入结点,现要求在 p 前插入 q,则正确的插入为()。
- A. $p \rightarrow \text{prior} = q; q \rightarrow \text{next} = p; p \rightarrow \text{prior} \rightarrow \text{next} = q; q \rightarrow \text{prior} = p \rightarrow \text{prior};$
- B. $q \rightarrow \text{prior} = p \rightarrow \text{prior}; p \rightarrow \text{prior} \rightarrow \text{next} = q; q \rightarrow \text{next} = p; p \rightarrow \text{prior} = q \rightarrow \text{next};$
- C. $q \rightarrow \text{next} = p; p \rightarrow \text{next} = q; p \rightarrow \text{prior} \rightarrow \text{next} = q; q \rightarrow \text{next} = p;$
- D. $p \rightarrow \text{prior} \rightarrow \text{next} = q; q \rightarrow \text{next} = p; q \rightarrow \text{prior} = p \rightarrow \text{prior}; p \rightarrow \text{prior} = q;$
19. 在单链表指针为 p 的结点之后插入指针为 s 的结点,正确的操作是()。
- A. $p \rightarrow \text{next} = s; s \rightarrow \text{next} = p \rightarrow \text{next};$
- B. $s \rightarrow \text{next} = p \rightarrow \text{next}; p \rightarrow \text{next} = s;$
- C. $p \rightarrow \text{next} = s; p \rightarrow \text{next} = s \rightarrow \text{next};$
- D. $p \rightarrow \text{next} = s \rightarrow \text{next}; p \rightarrow \text{next} = s;$
20. 对于一个头指针为 head 的带头结点的单链表,判定该表为空表的条件是()。

- A. head == NULL B. head->next == NULL
C. head->next == head D. head != NULL

二、综合应用题

1. 运算是数据结构的一个重要方面。举例说明两个数据结构的逻辑结构和存储方式完全相同,只是对于运算的定义不同,因而两个数据结构具有显著不同的特性,是两个不同的结构。

2. 将下列函数,按它们在 $n \rightarrow \infty$ 时的无穷大阶数,从小到大排序。

$$n, n - n^3 + 7n^5, n \log_2 n, 2^{n/2}, n^3, \log_2 n, n^{1/2} + \log_2 n, (3/2)^n, n!, n^2 + \log_2 n$$

3. 在单链表、双链表和单循环链表中,若仅知道指针 p 指向某结点,不知道头指针,能否将结点 $*p$ 从相应的链表中删去? 若可以,其时间复杂度各为多少?

4. 设有集合 A 和集合 B ,要求设计生成集合 $C = A \cap B$ 的算法,其中集合 A 、集合 B 和集合 C 用链式存储结构表示。

5. 已知单链表 L 是一个递增有序表,试写一高效算法,删除表中值大于 $\min$ 且小于 $\max$ 的结点(若表中有这样的结点),同时释放被删结点的空间,这里 $\min$ 和 $\max$ 是两个给定的参数。

6. 试写出在单链表中搜索元素 x 的算法。若 x 存在链表中,则输出它在表中的序号,否则将 x 插到表尾。

7. 设有一个双链表,每个结点中除有 prior 、 data 和 next 这 3 个域外,还有一个访问频度域 freq ,在链表被启用之前,其值均初始化为零。每当在链表进行一次 $\text{LocateNode}(L, x)$ 运算时,令元素值为 x 的结点中 freq 域的值加 1,并调整表中结点的次序,使其按访问频度的递减序排列,以便使频繁访问的结点总是靠近表头。试写一符合上述要求的 LocateNode 运算的算法。

8. 两个整数序列: $A = a_1, a_2, a_3, \dots, a_m$ 和 $B = b_1, b_2, b_3, \dots, b_n$,已经存入两个单链表中,设计一个算法,判断序列 B 是否是序列 A 的子序列。

9. 已知 3 个带头结点的线性链表 A 、线性链表 B 和线性链表 C 中的结点均依元素值自小至大非递减排列(可能存在两个以上值相同的结点),编写算法对链表 A 进行如下操作:使操作后的链表 A 中仅留下 3 个表中均包含的数据元素的结点,且没有值相同的结点,并释放所有无用结点。限定算法的时间复杂度为 $O(m+n+p)$,其中 m 、 n 和 p 分别为 3 个表的长度。

第二章 栈与队列、数组

知识点概要

栈和队列是限定插入和删除只能在表的“端点”进行的线性表。栈又称为后进先出的线性表,队列又称为先进先出的线性表。

1. 顺序栈形式描述

(1) 静态分配:

```
typedef struct {
    SElemType elem[STACKSIZE];
    int top;
```

```
} SqStack;
```

注意,非空栈时 top 始终在栈顶元素的位置。

(2) 动态分配:

```
typedef struct {
    SElemType * top;
    int stacksize;
} SqStack;
```

注意,非空栈时 top 始终在栈顶元素的下一个位置。

2. 链栈

因为栈中的主要运算是在栈顶插入、删除,显然在链表的头部做栈顶是最方便的,而且没有必要像单链表那样为了运算方便附加一个头结点。

3. 顺序队列

设队头指针指向队头元素前面一个位置,队尾指针指向队尾元素(这样的设置是为了某些运算的方便,并不是唯一的方法)。

(1) 入队:

```
sq->data[ ++sq->rear ] = x;
```

(2) 出队:

```
x = sq->data[ ++sq->front ];
```

4. 循环队列元素

循环队列元素个数:

$$(Q.rear - Q.front + MAXQSIZE) \% MAXQSIZE$$

5. 数组

一般采用顺序存储,是一个随机存取结构。

(1) 二维数组按行优先寻址计算方法,每个数组元素占据 d 个地址单元。

设数组的基址为 $LOC(a_{11})$: $LOC(a_{ij}) = LOC(a_{11}) + ((i-1) * n + j - 1) * d$

设数组的基址为 $LOC(a_{00})$: $LOC(a_{ij}) = LOC(a_{00}) + (i * n + j) * d$

(2) 二维数组按列优先寻址计算方法。

设数组的基址为 $LOC(a_{11})$: $LOC(a_{ij}) = LOC(a_{11}) + ((j-1) * n + i - 1) * d$

设数组的基址为 $LOC(a_{00})$: $LOC(a_{ij}) = LOC(a_{00}) + (j * n + i) * d$

6. 特殊矩阵的压缩存储(假设以行序为主序)

(1) 对称矩阵:将对称矩阵 A 压缩存储到 $SA[n(n+1)/2]$ 空间中, a_{ij} 的下标 i, j 与在 SA 中的对应元素的下标 k 的关系为 $k = i \times (i-1) / 2 + (j-1)$, $0 \leq k < n \times (n+1) / 2$ 。

(2) 三角矩阵:与对称矩阵类似,不同之处在于存储完下(上)三角中的元素之后,接着存储对角线上(下)方的常量,因为是同一个常数,所以存一个即可。将三角矩阵 A 压缩存储到 $SA[n(n+1)/2+1]$ 空间中, a_{ij} 的下标 i, j 与在 SA 中的对应元素的下标 k 的关系为:

当 $i=1$ 时, $k=j-1 (1 \leq j \leq 2)$;

当 $i>1$ 时, $k=2i+j-3 (|i-j| \leq 1)$ 。

(3) 三对角矩阵:将三对角矩阵 A 压缩存储到 $SA[3n-2]$ 空间中, a_{ij} 的下标 i, j 与在 SA 中的

对应元素的下标 k 的关系为 $k=2i+j-2$ 。

(4) 稀疏矩阵:三元组 (I, J, K) 行号、列号、元素值的保存方法。

一、单项选择题

1. 一个栈的输入序列为 $1, 2, 3, \dots, n$, 若输出序列的第一个元素是 n , 输出第 i ($1 \leq i \leq n$) 个元素是()。

- A. 不确定 B. $n-i+1$ C. i D. $n-i$

2. 若一个栈的输入序列为 $1, 2, 3, \dots, n$, 输出序列的第一个元素是 i , 则第 j 个输出元素是()。

- A. $i-j-1$ B. $i-j$ C. $j-i+1$ D. 不确定

3. 若已知一个栈的入栈序列是 $1, 2, 3, \dots, n$, 其输出序列为 $p_1, p_2, p_3, \dots, p_n$, 若 p_n 是 n , 则 p_i 是()。

- A. i B. $n-i$ C. $n-i+1$ D. 不确定

4. 有 6 个元素 $6, 5, 4, 3, 2, 1$ 的顺序进栈, 下列不合法的出栈序列是()。

- A. $5, 4, 3, 6, 1, 2$ B. $4, 5, 3, 1, 2, 6$ C. $3, 4, 6, 5, 2, 1$ D. $2, 3, 4, 1, 5, 6$

5. 若一个栈以向量 $V[n]$ 存储, 初始栈顶指针 top 为 $n+1$, 则下面 x 进栈的正确操作是()。

- A. $top=top+1; V[top]=x$ B. $V[top]=x; top=top+1$
C. $top=top-1; V[top]=x$ D. $V[top]=x; top=top-1$

6. 若栈采用顺序存储方式存储, 现两栈共享空间 $V[1..m]$, $top[i]$ 代表第 i 个栈 ($i=1, 2$) 栈顶, 栈 1 的底在 $V[1]$, 栈 2 的底在 $V[m]$, 则栈满的条件是()。

- A. $|top[2]-top[1]|=0$ B. $top[1]+1=top[2]$
C. $top[1]+top[2]=m$ D. $top[1]=top[2]$

7. 栈在()中应用。

- A. 递归调用 B. 子程序调用 C. 表达式求值 D. A, B, C

8. 一个递归算法必须包括()。

- A. 递归部分 B. 终止条件和递归部分
C. 迭代部分 D. 终止条件和迭代部分

9. 执行完下列语句段后, i 值为()。

```
int f(int x)
{ return ((x>0)? x * f(x-1); 2); }
int i;
i=f(f(1));
```

- A. 2 B. 4 C. 8 D. 无限递归

10. 表达式 $a * (b+c) - d$ 的后缀表达式是()。

- A. $abcd * + -$ B. $abc + * d -$ C. $abc * + d -$ D. $- + * abcd$

11. 递归过程或函数调用时, 处理参数及返回地址, 要用一种称为()的数据结构。

- A. 队列 B. 多维数组 C. 栈 D. 线性表

12. 若用一个大小为 6 的数组来实现循环队列,且当前 rear 和 front 的值分别为 0 和 3,当从队列中删除一个元素,再加入两个元素后,rear 和 front 的值分别为()。

- A. 1 和 5 B. 2 和 4 C. 4 和 2 D. 5 和 1

13. 设有一个 10 阶的对称矩阵 A,采用压缩存储方式,以行序为主存储, a_{11} 为第一元素,其存储地址为 1,每个元素占一个地址空间,则 a_{85} 的存储地址为()。

- A. 13 B. 33 C. 18 D. 40

14. $A[n][n]$ 是对称矩阵,将下面三角(包括对角线)以行序存储到一维数组

$T[n(n+1)/2]$ 中,则对任一上三角元素 $a[i][j]$ 对应 $T[k]$ 的下标 k 是()。

- A. $i(i-1)/2+j$ B. $j(j-1)/2+i$ C. $i(j-i)/2+1$ D. $j(i-1)/2+1$

15. 设二维数组 $A[m][n]$ (即 m 行 n 列)按行存储在数组 $B[1..m \times n]$ 中,则二维数组元素 $A[i][j]$ 在一维数组 B 中的下标为()。

- A. $(i-1) \times n + j$ B. $(i-1) \times n + j - 1$ C. $i \times (j-1)$ D. $j \times m + i - 1$

16. 有一个 100×90 的稀疏矩阵,非 0 元素有 10 个,设每个整型数占 2 字节,则用三元组表示该矩阵时,所需的字节数是()。

- A. 60 B. 66 C. 18 000 D. 33

二、综合应用题

1. 有 5 个元素,其入栈次序为:A,B,C,D,E,在各种可能的出栈次序中,以元素 C,D 最先出栈(即 C 第一个且 D 第二个出栈)的次序有哪几个?

2. 如果输入序列为 1,2,3,4,5,6,试问能否通过栈结构得到以下两个序列:4,3,5,6,1,2 和 1,3,5,4,2,6;请说明为什么不能或如何才能得到。

3. 利用两个栈 s1、s2 模拟一个队列时,如何用栈的运算实现队列的插入、删除以及判队空的运算。请简述这些运算的算法思想。

4. 顺序队列一般应该组织成为环状队列的形式,而且一般队列头或尾其中之一应该特殊处理。例如,队列为 listarray[n-1],队列头指针为 front,队列尾指针为 rear,则 listarray[rear] 表示下一个可以插入队列的位置。请解释其原因。

5. 一个 $n \times n$ 的对称矩阵,如果以行或列为主序存入内存,则其容量为多少?

6. 设有上三角矩阵 $(a_{ij})_{n \times n}$,将其上三角中的元素按先行后列的顺序存于数组 B[m] 中,使得 $B[k] = a_{ij}$ 且 $k = f_1(i) + f_2(j) + c$,请推导出函数 f_1 、 f_2 和常数 c ,要求 f_1 和 f_2 中不含常数项。

7. 设有两个栈 s1、s2 都采用顺序栈方式,并且共享一个存储区[maxsize-1],为了尽量利用空间,减少溢出的可能,可采用栈顶相向,迎面增长的存储方式。试设计 s1、s2 有关入栈和出栈的操作算法。

8. 请利用两个栈 s1 和 s2 来模拟一个队列。已知栈的三个运算定义如下:

PUSH(ST,x):元素 x 入 ST 栈;

POP(ST,x):ST 栈顶元素出栈,赋给变量 x;

Empty(ST):判 ST 栈是否为空。那么,如何利用栈的运算来实现该队列的 3 个运算:

(1) enqueue:插入一个元素入队列;

(2) dequeue:删除一个元素出队列;

(3) q_empty:判队列为空。

9. 双端队列(duque)是一个可以在任一端进行插入和删除的线性表。现采用一个一维数组作为双端队列的数据存储结构,使用 C 语言描述如下:

```
#define maxsize 32 { 数组中可容纳的元素个数 }
```

```
typedef struct
```

```
{   datatype elem[ maxsize ];
```

```
    int end1, end2;
```

```
}; duque;
```

试编写两个算法 add(duque Qu,datatype x,int tag) 和 delete(duque Qu,datatype &x,int tag) 用以在此双端队列的任一端进行插入和删除。当 tag=0 时在左端 end1 端操作,当 tag=1 时在右端 end2 端操作。

10. 已知 q 是一个非空队列,s 是一个空栈。仅用队列和栈的 ADT() 函数和少量工作变量,使用 Pascal 或 C 语言编写一个算法,将队列 Q 中的所有元素逆置。

(1) 栈的 ADT() 函数有:

makeEmpty(s); 置空栈

push(s,value); 新元素 value 进栈

pop(s); 出栈,返回栈顶值

isEmpty(s); 判栈空否

(2) 队列的 ADT() 函数有:

enqueue(q,value); 元素 value 进队

dequeue(q); 出队列,返回队头值

isEmpty(q); 判队列空否

11. 线性表中元素存放在数组 A[1..n] 中,元素是整型数。试写出递归算法求出数组 A 中的最大和最小元素。

12. 设数组 A[2n] 中存放有 n 个负数和 n 个正数,且随机存放。现要求按负数、正数相间存放,请写出实现此要求的算法。算法要求:不能使用额外的存储空间,但可使用少量工作单元,算法的时间复杂度应为 O(n)。

13. 设有一个长度为 n 的由“0”和“1”元素组成的输入序列,存于数组 A[n] 中。设计一个算法,依次让每个元素通过一个栈 s(容量 ≥ n) 而得到一个输出序列,使得输出序列中“0”元素都出现在“1”元素之前。输出序列存入数组 B[n] 中。

14. 给定一个整数数组 B[N],数组 B 中连续的相等元素构成的子序列称为平台。试设计算法,求出 B 中最长平台的长度。

15. 给定 $n \times m$ 矩阵 $A[a..b, c..d]$, 并设 $A[i, j] \leq A[i, j+1]$ ($a \leq i \leq b, c \leq j \leq d-1$) 和 $A[i, j] \leq A[i+1, j]$ ($a \leq i \leq b-1, c \leq j \leq d$)。设计一算法判定 x 的值是否在 A 中,要求时间复杂度为 O(m+n)。

16. 设二维数组 A[1..m, 1..n] 含有 $m \times n$ 个整数。

(1) 写出算法(Pascal 过程或 C 函数):判断二维数组 A 中所有元素是否互不相同并输出相关信息(yes/no)。

(2) 试分析算法的时间复杂度。

17. 已知两个定长数组 A、B, 它们分别存放两个非降序有序序列, 请编写程序把数组 B 序列中的数逐个插入到数组 A 序列中, 完成后两个数组中的数分别有序(非降序)并且数组 A 中所有的数都不大于数组 B 中的任意一个数。要求, 不能另开辟空间, 也不能对任意一个数组进行排序操作。例如,

数组 A 为: 4, 12, 28;

数组 B 为: 1, 7, 9, 29, 45

输出结果为: 1, 4, 7 (数组 A)

9, 12, 28, 29, 45 (数组 B)

18. 设数组 $A[n]$ 中, $A[n-2k+1..n-k]$ 和 $A[n-k+1..n]$ 中元素各自从小到大排好序, 试设计一个算法使 $A[n-2k+1..n]$ 按从小到大次序排好序。要求空间复杂度为 $O(1)$, 并分析算法所需的计算时间。

19. 设 $A[100]$ 是一个记录构成的数组, $B[100]$ 是一个整数数组, 其值介于 $1 \sim 100$, 现要求按 $B[100]$ 的内容调整 A 中记录的次序, 比如, 当 $B[1] = 11$ 时, 则要求将 $A[1]$ 的内容调整到 $A[11]$ 中去。规定可使用的附加空间为 $O(1)$ 。

20. 给定有 m 个整数的递增有序数组 $A[1..m]$ 和有 n 个整数的递减有序数组 $B[1..n]$, 试写出算法: 将数组 A 和 B 归并为递增有序数组 $C[1..m+n]$ 。(要求: 算法的时间复杂度为 $O(m+n)$)

第三章 树与二叉树

知识点概要

1. 二叉树的概念、性质

(1) 掌握树和二叉树的定义。

(2) 理解二叉树与普通双分支树的区别。二叉树是一种特殊的树, 这种特殊不仅仅在于其分支最多为 2 以及其他特征, 二叉树的一个最重要的特殊之处是在于: 二叉树是有序的。即二叉树的左右孩子是不可交换的, 如果交换了就成了另外一棵二叉树, 这样交换之后的二叉树与原二叉树是不相同的两棵二叉树。但是, 对于普通的双分支树而言, 不具有这种性质。

(3) 掌握满二叉树和完全二叉树的概念。

(4) 重点掌握二叉树的 5 个性质及证明方法, 并把这种方法推广到 K 叉树。普通二叉树的 5 个性质:

第 i 层的最多结点数; 深度为 k 的二叉树的最多结点数; $n_0 = n_2 + 1$ 的性质; n 个结点的完全二叉树的深度; 顺序存储二叉树时孩子结点与父结点之间的换算关系(序号为 i 的结点的左孩子的结点编号为: $2i$, 右孩子的结点编号为: $2i+1$)。

2. 二叉树的顺序存储结构、二叉树链表、三叉树链表存储结构

掌握二叉树的顺序存储结构和二叉链表、三叉链表存储结构的各自优缺点及适用场合, 以及二叉树的顺序存储结构和二叉链表存储结构相互转换的算法。

3. 二叉树的先序、中序和后序遍历算法以及按层次遍历

熟练掌握二叉树的先序、中序和后序3种遍历算法,划分的依据是视其每个算法中对根结点数据的访问顺序而定。不仅要熟练掌握这3种遍历的递归算法,还要理解其执行的实际步骤,并且应该熟练掌握3种遍历的非递归算法。重点掌握在3种基本遍历算法的基础上实现二叉树的其他算法,如求二叉树叶子结点总数,求二叉树结点总数,求度为1或度为2的结点总数,复制二叉树,建立二叉树,交换左右子树,查找值为 n 的某个指定结点,删除值为 n 的某个指定结点等。

4. 线索二叉树

线索二叉树的引出,是为避免如二叉树遍历时的递归求解。递归虽然形式上比较好理解,但是消耗了大量的内存资源,递归层次一多,势必带来资源耗尽的危险。二叉树线索化的实质是建立结点在相应序列中与其前驱和后继之间的直接联系。对于线索二叉树,应该掌握:线索化的实质,3种线索化的算法,线索化后二叉树的遍历算法,基本线索二叉树的其他算法问题(如查找某一类线索二叉树中指定结点的前驱或后继结点)。

5. 树与森林的遍历

树与森林的遍历,只有两种遍历算法:先根与后根(对于森林而言称作先序与中序遍历)。两者的先根与后根遍历与二叉树中的遍历算法是有对应关系的,先根遍历对应二叉树的先序遍历,而后根遍历对应二叉树的中序遍历。二叉树使用二叉链表分别存放它的左右孩子,树利用二叉链表存储孩子及兄弟(称孩子兄弟链表),而森林也是利用二叉链表存储孩子及兄弟。掌握树、森林和二叉树间的相互转换。

6. 哈夫曼树

哈夫曼树为了解决特定问题引出的特殊二叉树结构,它的前提是给二叉树的每条边赋予了权值,这样形成的二叉树按权相加之和是最小的,一般来说,哈夫曼树的形态不是唯一的。理解哈夫曼编码的基本原理,掌握基于哈夫曼树生成哈夫曼编码的方法。利用哈夫曼树可以构造一种不等长的二进制编码,并且构造所得的哈夫曼编码是一种最优前缀编码,即使所传电文的总长度最短。

一、单项选择题

1. 已知一算术表达式的中缀形式为 $A+B * C-D/E$, 后缀形式为 $ABC * +DE/-$, 其前缀形式为()。

A. $-A+B * C/DE$

B. $-A+B * CD/E$

C. $-+ * ABC/DE$

D. $-+A * BC/DE$

2. 设树 T 的度为4, 其中度为1、2、3和4的结点个数分别为4、2、1、1, 则树 T 中的叶子数为()。

A. 5

B. 6

C. 7

D. 8

3. 在下述结论中, 正确的是()。

- ① 只有一个结点的二叉树的度为0; ② 二叉树的度为2; ③ 二叉树的左右子树可任意交换; ④ 深度为 K 的完全二叉树的结点个数小于或等于深度相同的满二叉树。

A. ①②③

B. ②③④

C. ②④

D. ①④

4. 设森林 F 对应的二叉树为 B , 它有 m 个结点, 二叉树 B 的根为 p , p 的右子树结点数为

- n , 森林 F 中第一棵树的结点个数是()。
- A. $m-n$ B. $m-n-1$ C. $n+1$ D. 无法确定
5. 若一棵二叉树具有 10 个度为 2 的结点, 5 个度为 1 的结点, 则度为 0 的结点个数是()。
- A. 9 B. 11 C. 15 D. 不确定
6. 在一棵二叉树中度为 3 的结点数为 2 个, 度为 2 的结点数为 1 个, 度为 1 的结点数为 2 个, 则度为 0 的结点数为()个。
- A. 4 B. 5 C. 6 D. 7
7. 设森林 F 中有 3 棵树, 第一、第二、第三棵树的结点个数分别为 M_1, M_2 和 M_3 。与森林 F 对应的二叉树根结点的右子树上的结点个数是()。
- A. M_1 B. M_1+M_2 C. M_3 D. M_2+M_3
8. 一棵完全二叉树上有 1 001 个结点, 其中叶子结点的个数是()。
- A. 250 B. 500 C. 254 D. 501
9. 设给定权值总数有 n 个, 其哈夫曼树的结点总数为()。
- A. 不确定 B. $2n$ C. $2n+1$ D. $2n-1$
10. 若度为 m 的哈夫曼树中, 其叶结点个数为 n , 则非叶结点的个数为()。
- A. $n-1$ B. $n/m-1$ C. $(n-1)/(m-1)$ D. $(n+1)(m+1)-1$
11. 一个具有 1 025 个结点的二叉树的高度 h 为()。
- A. 11 B. 10 C. $11 \sim 1\,025$ D. $10 \sim 1\,024$
12. 一棵二叉树高度为 h , 所有结点的高度或为 0, 或为 2, 则这棵二叉树最少有() 结点。
- A. $2h$ B. $2h-1$ C. $2h+1$ D. $h+1$
13. 对于有 n 个结点的二叉树, 其高度为()。
- A. $n\log_2 n$ B. $\log_2 n$ C. $\log_2 n+1$ D. 不确定
14. 若二叉树采用二叉链表存储结构, 要交换其所有分支结点左右子树的位置, 利用() 遍历方法最合适。
- A. 前序 B. 中序 C. 后序 D. 按层次
15. 一棵二叉树的前序遍历序列为 ABCDEFG, 它的中序遍历序列可能是()。
- A. CABDEFG B. ABCDEFG C. DACEFBG D. ADCFEG
16. 某二叉树中序序列为 ABCDEFG, 后序序列为 BDCAFGE, 则前序序列是()。
- A. EGFACDB B. EACBDGF C. EAGCFBD D. 上面的都不对
17. 某二叉树中序序列为 ABCDEFG, 后序序列为 BDCAFGE, 该二叉树对应的森林包括多少棵树()。
- A. 1 B. 2 C. 3 D. 概念上是错误的
18. 一棵非空的二叉树的先序遍历序列与后序遍历序列正好相反, 则该二叉树一定满足()。
- A. 所有的结点均无左孩子 B. 所有的结点均无右孩子
- C. 只有一个叶子结点 D. 是任意一棵二叉树

19. 在二叉树结点的先序序列,中序序列和后序序列中,所有叶子结点的先后顺序()。
- A. 都不相同
B. 完全相同
C. 先序和中序相同,而与后序不同
D. 中序和后序相同,而与先序不同
20. 某二叉树的前序序列和后序序列正好相反,则该二叉树一定是()的二叉树。
- A. 空或只有一个结点
B. 任一结点无左子树
C. 高度等于其结点数
D. 任一结点无右子树
21. 一棵左子树为空的二叉树在先序线索化后,其中空的链域的个数是()。
- A. 不确定 B. 0 C. 1 D. 2
22. 一棵左右子树均不空的二叉树在先序线索化后,其中空的链域的个数是()。
- A. 0 B. 1 C. 2 D. 不确定
23. ()的遍历仍需要栈的支持。
- A. 前序线索树 B. 中序线索树 C. 后序线索树 D. 以上都不是
24. n 个结点的线索二叉树上含有的线索数为()。
- A. $2n$ B. $n-1$ C. $n+1$ D. n
25. 下面几个符号串编码集合中,不是前缀编码的是()。
- A. $\{0, 10, 110, 1111\}$ B. $\{11, 10, 001, 101, 0001\}$
C. $\{00, 010, 0110, 1000\}$ D. $\{b, c, aa, ac, aba, abb, abc\}$

二、综合应用题

1. 从概念上讲,树、森林和二叉树是3种不同的数据结构,说明将树、森林转化为二叉树的基本目的是什么,并指出树和二叉树的主要区别。

2. 一棵高度为 h 的满 k 叉树有如下性质:根据结点所在层次为0;第 h 层上的结点都是叶子结点;其余各层上每个结点都有 k 棵非空子树,如果按层次自顶向下,同一层自左向右,顺序从1开始对全部结点进行编号,试问:

- (1) 各层的结点个数是多少?
- (2) 编号为 i 的结点的双亲结点(若存在)的编号是多少?
- (3) 编号为 i 的结点的第 m 个孩子结点(若存在)的编号是多少?
- (4) 编号为 i 的结点有右兄弟的条件是什么? 其右兄弟结点的编号是多少?

3. 证明任一结点个数为 n 的二叉树的高度至少为 $O(\log_2 n)$ 。

4. 已知一棵满二叉树的结点个数为20~40的素数,此二叉树的叶子结点有多少个?

5. 一棵共有 n 个结点的树,其中所有分支结点的度均为 K ,求该树中叶子结点的个数。

6. 若一棵树中有度数为 $1 \sim m$ 的各种结点数为 $n_1, n_2, \dots, n_m$ (n_m 表示度数为 m 的结点数),请推导出该树中共有多少个叶子结点 n_0 的公式。

7. 高度为 k 的完全二叉树至少有多少个叶结点?