

目录

2020 年南京大学计算机科学与技术系45 数据结构和算法、操作系统、计算机系统基础、计算机网络之数据结构（用面向对象方法C++语言描述）考研仿真模拟五套题（一）.....	2
2020 年南京大学计算机科学与技术系45 数据结构和算法、操作系统、计算机系统基础、计算机网络之数据结构（用面向对象方法C++语言描述）考研仿真模拟五套题（二）.....	14
2020 年南京大学计算机科学与技术系45 数据结构和算法、操作系统、计算机系统基础、计算机网络之数据结构（用面向对象方法C++语言描述）考研仿真模拟五套题（三）.....	27
2020 年南京大学计算机科学与技术系45 数据结构和算法、操作系统、计算机系统基础、计算机网络之数据结构（用面向对象方法C++语言描述）考研仿真模拟五套题（四）.....	37
2020 年南京大学计算机科学与技术系45 数据结构和算法、操作系统、计算机系统基础、计算机网络之数据结构（用面向对象方法C++语言描述）考研仿真模拟五套题（五）.....	48

2020年南京大学计算机科学与技术系45数据结构和算法、操作系统、计算机系统基础、计算机网络之数据结构（用面向对象方法与++语言描述）考研仿真模拟五套题（一）

特别说明：

1-本资料为2020考研初试学员使用，严格按照该科目历年常考题型及难度仿真模拟；

2-资料仅供考研复习参考，与目标学校及研究生院官方无关，如有侵权、请联系我们立即处理。

一、单项选择题

1. 循环队列A[0..m-1]存放其元素值 用 front 和 rear 分别表示队头和队尾 则当前队列中的元素数是（ ）。

- A. $(\text{rear} - \text{front} + m) \% m$
- B. $\text{rear} - \text{front} + 1$
- C. $\text{rear} - \text{front} - 1$
- D. $\text{rear} - \text{front}$

【答案】A

【解析】对于循环队列，需要深刻理解队头(front)和队尾(rear)的概念，在队头进行出队操作，在队尾进行进队操作。rear-front可能为正也可能为负，为正时元素个数= $(\text{rear}-\text{front})$ ；如果为负则元素的个数= $(\text{rear}-\text{front}+m)$ ，所以统一的公式就是 $(\text{rear}-\text{front}+m)\%m$ 。

2. 若一个有向图具有拓扑排序序列那么它的邻接矩阵必定为（ ）。

- A. 对称矩阵
- B. 稀疏矩阵
- C. 三角矩阵
- D. 一般矩阵

【答案】C

【解析】在图论中，由一个有向无环图的顶点组成的序列，当且仅当满足下列条件时，称为改图的一个拓扑排序①每个顶点出现且出现一次；②若顶点在序列中排在顶点B的前面，则在图中不存在从顶点B到顶点A的路径。由拓扑排序的性质知，有向图的邻接矩阵必定为三角矩阵。

3. 已知有向图 $G=(V, E)$, 其中 $V=\{V_1, V_2, V_3, V_4, V_5, V_6, V_7\}$,

$E=\{ \langle V_1, V_2 \rangle, \langle V_1, V_3 \rangle, \langle V_1, V_4 \rangle, \langle V_2, V_5 \rangle, \langle V_3, V_5 \rangle, \langle V_3, V_6 \rangle, \langle V_4, V_6 \rangle, \langle V_5, V_7 \rangle, \langle V_6, V_7 \rangle \}$,

G的拓扑序列是（ ）。

- A. $V_1, V_3, V_4, V_6, V_2, V_5, V_7$
- B. $V_1, V_3, V_2, V_6, V_4, V_5, V_7$
- C. $V_1, V_3, V_5, V_2, V_6, V_7$
- D. $V_1, V_2, V_5, V_3, V_4, V_6, V_7$

【答案】A

【解析】设 $G=(V, E)$ 是一个具有 n 个顶点的有向图, V 中顶点序列 V_1, V_2, \dots, V_n 能被称为拓扑序列的条件:若 $\langle V_i, V_j \rangle$ 是图中的边(即从顶点 V_i 到 V_j 有一条路径), 则在序列中顶点 V_i 必须排在顶点 V_j 之前。根据上面拓扑序列的定义, 就可以得出 G 的拓扑序列是 $V_1, V_3, V_4, V_5, V_2, V_5, V_7$ 。

4. 一组记录的关键码为46, 79, 56, 38, 40, 84, 则利用快速排序的方法 以第一个记录为基准得到的一次划分结果为 ()。

- A.(38, 40, 46, 56, 79, 84)
- B.(40, 38, 46, 79, 56, 84)
- C.(40, 38, 46, 56, 79, 84)
- D.(40, 38, 46, 84, 56, 79)

【答案】C

【解析】快速排序是将待排记录分割成独立的两部分, 其中一部分的关键字均比另一部分记录的关键字小。

第一次比较:46比84小, 不交换;

第二次比较:40比46小, 交换, 此时为(40, 79, 56, 38, 46, 84);

第三次比较:46比79小, 交换, 此时为(40, 46, 56, 38, 79, 84);

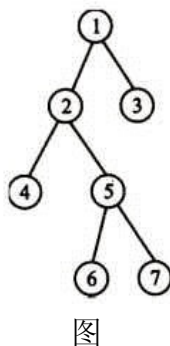
第四次比较:38比46小, 交换, 此时为(40, 38, 56, 46, 79, 84);

第五次比较:56比46大, 交换, 此时为(40, 38, 46, 56, 79, 84);

一次划分结束。

5. 给定二叉树如下图所示设 N 代表二叉树的根 L 代表根结点的左子树 R 代表根结点的右子树.若遍历后的结点序列为3, 1, 7, 5, 6, 2, 4, 则其遍历方式是 ()。

- A.LRN
- B.NRL
- C.RLN
- D.RNL



【答案】D

【解析】对“二叉树”而言, 一般有三条搜索路径:

①先上后下的按层次遍历;

②先左(子树)后右(子树)的遍历;

③先右(子树)后左(子树)的遍历.

其中第 1 种搜索路径方式就是常见的层次遍历, 第 2 种搜索路径方式包括常见的先序遍历 NLR、中序遍历 LNR、后序遍历 LRN, 第 3 种搜索路径方式则是不常使用的 NRL、RNL、RLN. 本题考查的是第 3 种搜索路径方式的一种情况根据遍历的序列以及树的结构图, 可以分析出该遍历的顺序是先右子树再跟结点最后左子树, 故答案为 D.

6. 循环队列存储在数组 $A[0..m]$ 中, 则入队时的操作为 ()。

A. $rear = rear + 1$

B. $rear = (rear + 1) \bmod (m - 1)$

C. $rear = (rear + 1) \bmod m$

D. $rear = (rear + 1) \bmod (m + 1)$

【答案】D

7. 用数组 r 存储静态链表, 结点的 $next$ 域指向后继, 工作指针 j 指向链中结点, 使 j 沿链移动的操作为 ()。

A. $j = r[j].next$

B. $j = j + 1$

C. $j = j - \rightarrow next$

D. $j = r[j] - \rightarrow next$

【答案】A

【解析】因为是用数组存储, 这里所说的工作指针 j 相当于数组的下标, 结点是存储一个值域和 $next$ 域, $next$ 域就是存放下一个结点的下表, 所以只要将 $next$ 域中的值赋给 j 就可以实现 j 沿链移动。

8. 若线性表最常用的操作是存取第 i 个元素及其前驱和后继元素的值为节省时间应采用的存储方式 ()。

A. 单链表

B. 双向链表

C. 单循环链表

D. 顺序表

【答案】D

【解析】线性表采用顺序表, 便于进行存取任一指定序号的元素。

9. 一棵二叉树高度为 h , 所有结点的度或为 0 或为 2, 则这棵二叉树最少有 () 个结点。

A. $2h$

B. $2h - 1$

C. $2h+1$ D. $h+1$ **【答案】** B**【解析】** 此树满足哈夫曼树，除根节点外每层有两个节点。

10. 每个结点的度或者为0或者为2的二叉树称为正则二叉树。n个结点的正则二叉树中有 () 个叶子。

A. $\lceil \log_2 n \rceil$ B. $\frac{n-1}{2}$ C. $\lceil \log_2(n+1) \rceil$ D. $\frac{n+1}{2}$ **【答案】** D

【解析】 二叉树结点总数 $n = n_0 + n_1 + n_2$ (n_0 , n_1 , n_2 分别代表度为 0, 度为 1, 度为 2 的结点数)。又在非空二叉树中: $n_0 = n_2 + 1$, 且本题所给树为正则二叉树, $n_1 = 0$, 所以 $n = 2 * n_0 - 1$, 因此 $n_0 = (n+1)/2$ 。

二、应用题

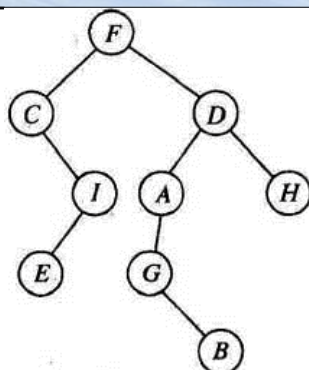
11. 阅读下面的算法 说明算法实现的功能。

```
node *link(node *head1, *head2)
{
    node *p, *q;    p=head1;
    while (p->next!=head1)    p=p->next;
    q=head2;
    while (q->next!=head2)    q=q->next;
    p->next=head2;            q->next=head1;
    return(head1);
}
```

【答案】 本算法功能是将两个无头结点的循环链表合并为一个循环链表。head1 最后一个结点的链域指向 head2, head2 最后一个结点的链域指向 head1, head1 为结果循环链表的指针。

12. 已知一棵二叉树的后序遍历序列为 EICBGAHDE, 同时知道该二叉树的中序遍历序列为 CEIFGBADH, 试画出该二叉树。

【答案】 该二叉树如图所示:



图

13. 外排序中为何采用 k 路($k>2$)合并而不用 2—路合并?这种技术用于内排序有意义吗为什么?

【答案】外排序用 k -路归并($k>2$)是因为 k 越小, 归并趟数越多, 读写外存次数越多, 时间效率越低, 故一般应大于最少的 2-路归并。若将 k -路归并的败者树思想单纯用于内排序, 因其由胜者树改进而来, 且辅助空间大, 完全可由堆排序取代, 故将其用于内排序效率并不高。

14. 写出下列排序算法的基本思想并写出对序列(23, 12, 35, 47, 16, 25, 36, 19, 21, 16)进行排序时每一趟的结果。

```

PROC  bbsort(VAR  r: sequence;  n: integer);
{r是一个数组}
  d:=1;  pos[-1]:=1; pos[1]:=n;  i:=1;  exchanged:= true;
  WHILE  exchanged  DO
    [ exchanged:= false;
      WHILE  i<>pos[d]  DO
        [ IF (r[i]-r[i+d])*d>0 THEN [r[i]与r[i+d]交换; exchanged:=true;]
          i:=i+d;
        ]
      pos[d]:=pos[d]-d; i:=pos[d]; d:=-d;
    ]
  ]
ENDP;
    
```

【答案】此排序为双向起泡排序:从前向后一趟排序下来得到一个最大值, 若其中发生交换, 则再从后向前一趟排序, 得到一个最小值。

第一趟:12, 23, 35, 16, 25, 36, 19, 21, 16, 47

第二趟:12, 16, 23, 35, 16, 25, 36, 19, 21, 47

第三趟:12, 16, 23, 16, 25, 35, 19, 21, 36, 47

第四趟:12, 16, 16, 23, 19, 25, 35, 21, 36, 47

第五趟:12, 16, 16, 19, 23, 25, 21, 35, 36, 47

第六趟:12, 16, 16, 19, 21, 23, 25, 35, 36, 47

第七趟:12, 16, 16, 19, 21, 23, 25, 35, 36, 47