

2021 天勤计算机考研八套模拟卷 · 卷八

数据结构篇

一、选择题

1. 对顺序存储的线性表, 设其长度为 n , 且在任何位置上插入或删除操作都是等概率的。则插入一个元素时平均要移动表中的 () 个元素。
A. $n/2$
B. $(n+1)/2$
C. $(n-1)/2$
D. n
2. 某线性表用带头结点的循环单链表存储, 头指针为 $head$, 当 $head \rightarrow next \rightarrow next \rightarrow next == head$ 为真时, 线性表长度可能是 ()。
A. 0
B. 1
C. 2
D. 0 或 2
3. 和顺序栈相比, 链栈有一个比较明显的优势是 ()
A. 通常不会出现栈满的情况
B. 通常不会出现栈空的情况
C. 插入操作更容易实现
D. 删除操作更容易实现
4. 执行 () 操作时, 需要使用队列作为辅助存储空间。
A. 查找哈希 (hash) 表
B. 广度优先搜索图
C. 前序 (根) 遍历二叉树
D. 深度优先搜索图
5. 一棵有 n 个结点的二叉树, 按层次从上到下, 同一层从左到右顺序存储在一维数组 $A[1 \dots n]$ 中, 则二叉树中第 i 个结点 (i 从 1 开始用上述方法编号) 的右孩子在数组 A 中的位置是 ()
A. $A[2i]$ ($2i \leq n$)
B. $A[2i+1]$ ($2i+1 \leq n$)
C. $A[i-2]$
D. 条件不充分, 无法确定
6. 二叉树在线索化后, 仍不能有效求解的问题是 ()
A. 前序线索二叉树中求前序后继
B. 中序线索二叉树中求中序后继
C. 中序线索二叉树中求中序前驱

D. 后序线索二叉树中求后序后继

7. 对于二叉排序树, 下面说法 () 是正确的

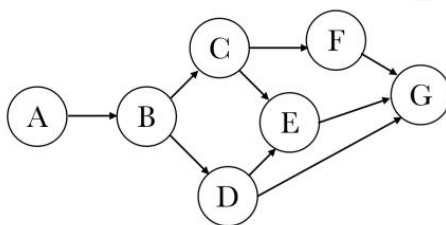
- A. 二叉排序树是动态树表, 查找不成功时插入新结点时, 会引起树的重新分裂和组合
- B. 对二叉排序树进行层序遍历可得到有序序列
- C. 用逐点插入法构造二叉排序树时, 若先后插入的关键字有序, 二叉排序树的深度最大
- D. 在二叉排序树中进行查找, 关键字的比较次数不超过结点数的 $1/2$

8. 分别以下列序列构造二叉排序树, 与用其它三个序列所构造的结果不同的是 ()

- A. (100, 80, 90, 60, 120, 110, 130)
- B. (100, 120, 110, 130, 80, 60, 90)
- C. (100, 60, 80, 90, 120, 110, 130)
- D. (100, 80, 60, 90, 120, 130, 110)

9. 下面有向图的所有拓扑排序序列共有 () 个。

- A. 4 B. 5 C. 6 D. 7



9 题图

10. 将 10 个元素散列到 100000 个单元的散列表中, 则 () 产生冲突。

- A. 一定会 B. 一定不会 C. 仍可能会

二、综合题

1. 已知一个长度为 12 的表 {Jan, Feb, Mar, Apr, May, June, July, Aug, Sep, Oct, Nov, Dec}: 按表中元素的顺序构造一棵平衡二叉树。

2. 设有向无环图 G 以邻接矩阵的方式存储, $G[i][j]$ 中存放的是从结点 i 出发到结点 j 的边权, $G[i][j]=0$ 代表从 i 到 j 没有直接的边, 试编写程序, 求 G 图中最长的路径长度。

(1) 给出算法的基本设计思想。

(2) 根据设计思想, 采用 C 或 C++ 语言描述算法, 关键之处给出注释。

(3) 给出算法的时间复杂度。

答案

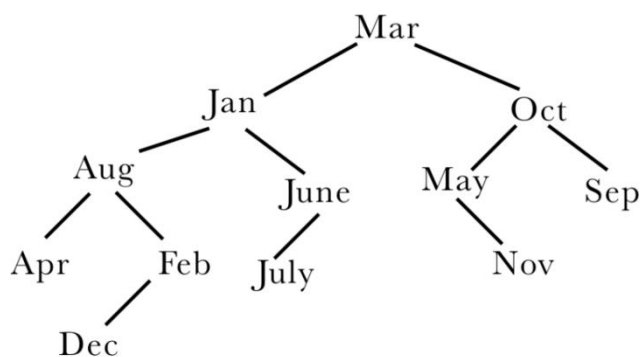
一、选择题答案

1.A 2.D 3.A 4.B 5.D 6.D 7.C 8.C 9.B 10.C

二、综合题答案

1.

平衡二叉树:



2.

(1) 算法思想:

基本设计思想: 我们知道可以利用弗洛伊德算法 (floyd) 来求得图中任意两点间的最短路径长度, 这里的边权是正数, 如果图中所有的边权均为负数, 那我们根据弗洛伊德算法求出的便是任意两点间最小的负权路径长度, 此时若把所有的边权取相反数, 则刚才求得的最短路径长度的相反数一定是现在的最长路径长度; 根据此思想, 将图 G 的边权改为它的相反数, 得到图 G', 然后用 floyd 算法对 G' 求出每对顶点间的最短路径, 那么图 G' 中最短路径的相反数即为原图 G 的最长路径长度。

(2)

代码:

```
int lengestPath(int MG[][maxSize], int Mn)
{
    int i,j,k;int max = 0;int A[][maxSize];
    for(i=0;i<Mn;++i)
        for(j=0;j<Mn;++j)
        {
            if(MG[i][j]<INF)
                A[i][j]=-MG[i][j];
            else
                A[i][j]=MG[i][j];
        }
    for(k=0;k<Mn;++k)
        for(i=0;i<Mn;++i)
```

```
for (j=0; j<Mn; ++j)
    if (A[i][j]>A[i][k]+A[k][j])
        A[i][j]=A[i][k]+A[k][j];
for (i=0; i<Mn; ++i)
    for (j=0; j<Mn; ++j)
    {
        if (A[i][j]<INF && max<-A[i][j])
            max = -A[i][j];
    }
return max;
}
```

(3) 时间复杂度: $O(n^3)$

全套模拟卷以及答案解析视频讲解来辉解读公众号获取:

