

## 五邑大学 试卷参考答案及评分标准

学期： 2015 至 2016 学年度 第 2 学期

课程： 数据结构 课程代号： 0800610

使用班级： 140801~140808

### 一、 单项选择题（10 小题，每小题 1 分，共 10 分）

1. 以下数据结构中，（ A ）是非线性数据结构。

A. 树 B. 字符串 C. 队 D. 栈

2. 程序段

```
for(i=n-1;i>=1;i--)
```

```
for(j=1;j<=i;j++)
```

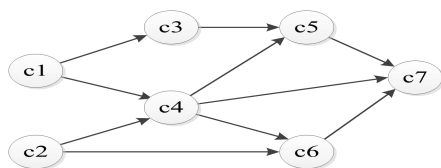
```
if(A[j]>A[j+1])
```

A[j]与 A[j+1]对换；

其中 n 为正整数，则最后一行的语句频度在最坏情况下是（ D ）。

A.  $O(n)$  B.  $O(n\log n)$  C.  $O(n^3)$  D.  $O(n^2)$

3. 拓扑序列是无环有向图中所有顶点的一个线性序列，图中任意路径中的各个顶点在该图的拓扑序列中保持先后关系，（ B ）不是下图所示有向图的一个拓扑序列。



A. c1 c2 c3 c4 c5 c6 c7

B. c1 c2 c4 c5 c6 c3 c7

C. c1 c2 c4 c6 c3 c5 c7

D. c2 c1 c4 c6 c3 c5 c7

4. 对于 n 个关键字序列  $\{k_1, k_2, \dots, k_n\}$ ，当且仅当满足关系  $k_i \leq k_{2i}$  且  $k_i \leq k_{2i+1}$  ( $2i \leq n, 2i+1 \leq n$ ) 称其为小根堆，反之，当且仅当满足关系  $k_i \geq k_{2i}$  且  $k_i \geq k_{2i+1}$  ( $2i \leq n, 2i+1 \leq n$ ) 则为大根堆，以下序列中（ D ）符合堆的定义。

A. (4,10,23,72,39,15,18)

B. (58,27,23,12,8,36,9)

C. (4,10,18,72,39,23,15)

D. (58,36,27,12,8,23,9)

5. 在若长度为 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（ C ） ( $1 \leq i \leq n$ )。

A.  $O(0)$  B.  $O(1)$  C.  $O(n)$  D.  $O(n^2)$

6. 当在一个有序的顺序存储表上查找一个数据时，即可用折半查找，也可用顺序查找，但前者比后者的查找速度（ C ）。

A. 必定快

B. 不一定

C. 在大部分情况下要快

D. 取决于表递增还是递减

7. 在采用二次探测法处理冲突所构成的闭散列表上进行查找，可能要探测多个位置，在查找成功的情况下，所探测的这些位置的键值（ A ）。

A. 不一定是同义词

B. 一定都是同义词

C. 一定都不是同义词

D. 都相同

8. 表达式  $a*(b+c)-d$  的后缀表达式是( B )。

A.  $abcd*+-$

B.  $abc+*d-$

C.  $abc*+d-$

D.  $-+*abcd$

9. 数组  $A[0..5,0..6]$  的每个元素占 5 个字节, 将其按列优先次序存储在起始地址为 1000 的内存单元中, 则元素  $A[5, 5]$  的地址是( A )。

A. 1175

B. 1180

C. 1205

D. 1210

10. 下面关于求关键路径的说法不正确的是( A )。

A. 求关键路径是以拓扑排序为基础的

B. 一个事件的最早开始时间与以该事件为尾的弧的活动最早开始时间相同

C. 一个事件的最迟开始时间为以该事件为尾的弧的活动最迟开始时间与该活动的持续时间的差

D. 关键活动一定位于关键路径上

## 二、简单填空题(10 小题, 20 个空, 每空 1 分, 共 20 分)

1. 当两个栈共享一存储区时, 栈利用一维数组  $data[1..n]$  表示, 栈 1 由低向高延伸, 栈 2 由高向低延伸, 栈 1 栈顶指针为  $top1$ , 栈 2 为  $top2$ , 则当栈 1 空时,  $top1$  为 -1, 栈满时为  $top1+1==top2$ 。

2. 对于一个具有  $n$  个结点的单链表, 在已知的结点  $p$  后插入一个新结点的时间复杂度为  $O(1)$  在给定值为  $x$  的结点后插入一个新结点的时间复杂度为  $O(n)$ 。

3. 按照书本定义的循环队列, 设  $front$  指向队头元素的前一个位置,  $rear$  指向队尾元素的位置, 设实现循环队列的空间大小为  $QueueSize$ , 那么循环队列的空的条件为  $front==rear$ , 满的条件为  $(rear+1)\%QueueSize==front$ 。

4. 用一维数组  $B$  与列优先存放带状矩阵  $A$ ,  $A$  中的非零元素  $A[i,j]$  ( $1 \leq i \leq n, i-2 \leq j \leq i+2$ ),  $B$  中的第 8 个元素是  $A$  中的第 1 行, 第 3 列的元素。

5. 深度为  $H$  的完全二叉树至少有  $2^{H-1}$  个结点; 至多有  $2^H-1$  个结点。

6. 快速排序法在 基本有序或(正序或逆序) 情况下最不利于发挥其长处, 在 每次划分元素个数都基本相等(相同意思均可) 情况下最易发挥其长处。

7. 顺序查找  $n$  个元素的顺序表, 若查找成功, 则比较关键字的次数最多为  $n$  次; 当使用监视哨时, 若查找失败, 则比较关键字的次数为  $n+1$ 。

8. 平衡二叉树又称 AVL 树, 最小不平衡子树是指在平衡二叉树的构造过程中, 以距离插入点最近的、且平衡因子的绝对值 大于 1 或  $>1$  或  $=2$  的结点为根的子树。

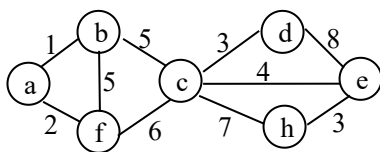
9. 设无向图  $G=(V,E)$  有  $n$  个顶点,  $e$  条边, 若采用邻接矩阵存储图, 其空间复杂度为  $O(n^2)$ , 在此基础上进行广度优先搜索的时间复杂度为  $O(n^2)$ 。

10. 先根次序遍历森林的结果正好等同于按 先序 遍历对应的二叉树, 后根次序遍历森林正好等同于按 中序 遍历对应的二叉树。

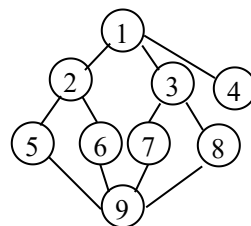
三、应用题（10 小题，每小题 6 分，共 60 分）

评分：每小题 6 分，每空 3 分

1. 每一棵树都能唯一的转换为它所对应的二叉树。若已知一棵二叉树的前序序列是 BEFCGDH，中序序列是 FEBGCHD，则它的后序序列是（ FEGHDCB ）；设上述二叉树是由某棵树转换而成，则该树的先根次序序列是（ BEFCGDH ）。
2. 已知有序表为(12,18,24,35,47,50,62,83,90,115,134)，当用二分法查找 90 时，需（ 2 ）次查找成功，查 100 时，需（ 3 ）次才能确定不成功。
3. 给定一组数据{6, 2, 7, 10, 3, 12}以它构造一棵哈夫曼树，则树高为（ 5 ），带权路径长度 WPL 的值为（ 113 ）。
4. 利用逐点插入法建立序列（50, 72, 43, 85, 75, 20, 35, 45, 65, 30）对应的二叉排序树，该二叉排序树的深度为（ 5 ），查找成功的平均查找长度为（ 3 ）。
5. 已知一个线性表（38, 25, 74, 63, 52, 48），假定采用散列函数  $h(\text{key}) = \text{key} \% 12$  计算散列地址，并散列存储在散列表 A[0..11]中，若采用线性探测方法解决冲突，查找 63 需要比较（ 2 ）次，该散列表的装填因子  $\alpha$  为（ 0.5 ）。
6. 题 6 图表示一个地区的通讯网，边表示城市间的通讯线路，边上的权表示架设线路花费的代价，选择能沟通每个城市且总代价最省的  $n-1$  条线路，总代价是（ 18 ）；选择的线路是（ (a,b) 1,(a,f)2,(b,c)5,(c,d)3,(c,e)4,(c,h)3 ）。

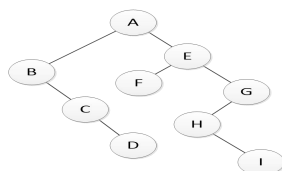


题 6 图



题 7 图

7. 如图题 7 图所示的无向图，如果采用邻接表存储该图，则从顶点 1 出发按深度优先遍历的顶点序列是（ 1 2 5 9 6 7 3 8 4 ）按广度优先遍历的顶点序列是（ 1 2 3 4 5 6 7 8 9 ）。
8. 设有输入序列(2,3,1,5,4,9,8,7,6)，借助辅助栈或辅助队列得到序列(1,2,3,4,5,6,7,8,9)，问最少需要（ 2 ）个辅助栈，或最少需要（ 3 ）个辅助队列。
9. 如图题 9 图所示的二叉树，将其转换成对应的森林，问：该森林中共有（ 3 ）棵树，（ A ）结点的度最大。



题 9 图

10. 设用序列(23,20,11,31,13,27)，构造一个小顶堆，该小顶堆的叶子元素包括（ 31, 20,27 ）；对该小顶堆进行堆排序，第一趟排序并调整后的顺序表为（ 13,20,23,31,27,11 ）。

四、算法设计题 （2 小题，每小题 5 分，共 10 分）

1. 设计一个算法判断单链表是否为有序表，假设有序表是升序的。

```
template <class T>
bool IsOrder(Node<T> *head)
{
    Node <T> *p;
    p=head->next; //1 分
    while(p && p->next) //1 分
    {
        if(p->data>p->next->data)
            return false; //1 分
        p=p->next; //1 分
    }
    return true; //1 分
}
```

2. 假设二叉树的根为 root，左孩子为 lchild，右孩子为 rchild，数据域为 data。设计一个算法判断二叉树是否为平衡二叉树，设已有算法 depth(BiNode \*root)，该算法返回树 root 的深度。

```
void IsAVL(BiNode *root)
{

    if(Root==NULL)
        return true;
    if(IsAVL(Root->lchild)==false)
        return false;
    if(IsAVL(Root->rchild)==false)
        return false;
    if(abs(depth(Root->lchild)-depth(Root->rchild))<=1)
        return true;
    else
        return false;

}
一个判断 1 分
```