

## 江苏大学 2010 年硕士研究生入学考试试题

科目代码: 851

科目名称: 数据结构

考生注意: 答案必须写在答题纸上,写在试卷、草稿纸上无效!

## 一、单项选择题(每小题 1 分,共 10 分)

1. 设有数组  $A[i,j]$ ,数组的每个元素长度为 3 字节, $i$  的值为 1 到 8, $j$  的值为 1 到 10,数组从内存首地址 SA 开始顺序存放,当以列为主存放时,元素  $A[5,8]$  的存储首地址为( )。  
(A) SA+141 (B) SA+180 (C) SA+222 (D) SA+225
2. 在双向链表指针  $p$  的结点前插入一个指针  $q$  的结点操作是( )。  
(A)  $p \rightarrow \text{Llink} = q; q \rightarrow \text{Rlink} = p; p \rightarrow \text{Llink} \rightarrow \text{Rlink} = q; q \rightarrow \text{Llink} = q;$   
(B)  $q \rightarrow \text{Rlink} = p; q \rightarrow \text{Llink} = p \rightarrow \text{Llink}; p \rightarrow \text{Llink} \rightarrow \text{Rlink} = q; p \rightarrow \text{Llink} = q;$   
(C)  $p \rightarrow \text{Llink} = q; p \rightarrow \text{Llink} \rightarrow \text{Rlink} = q; q \rightarrow \text{Rlink} = p; q \rightarrow \text{Llink} = p \rightarrow \text{Llink};$   
(D)  $q \rightarrow \text{Llink} = p \rightarrow \text{Llink}; q \rightarrow \text{Rlink} = q; p \rightarrow \text{Llink} = q; p \rightarrow \text{Llink} = q;$
3. 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( )结点。  
(A)  $2h$  (B)  $2h-1$  (C)  $2h+1$  (D)  $h+1$
4. 当在一个有序的顺序存储表上查找一个数据时,既可用折半查找,也可用顺序查找,但前者比后者的查找速度( )。  
(A) 取决于表递增还是递减 (B) 必定快 (C) 不一定 (D) 在大部分情况下要快
5. 串的长度是指( )。  
(A) 串中所含不同字母的个数 (B) 串中所含非空格字符的个数  
(C) 串中所含不同字符的个数 (D) 串中所含字符的个数
6. 下面说法错误的是( )。  
(A) 算法原地工作的含义是指不需要任何额外的辅助空间  
(B) 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  $O(2^n)$  的算法  
(C) 所谓时间复杂度是指最坏情况下,估算算法执行时间的一个上界  
(D) 同一个算法,实现语言的级别越高,执行效率就越低
7. 以下错误的是( )  
(i) 静态链表既有顺序存储的优点,又有动态链表的优点。所以,它存取表中第  $i$  个元素的时间与  $i$  无关。  
(ii) 静态链表中能容纳的元素个数的最大数在表定义时就确定了,以后不能增加。  
(iii) 静态链表与动态链表在元素的插入、删除上类似,不需做元素的移动。  
(A) (i), (ii) (B) (i) (C) (i), (ii), (iii) (D) (ii)
8. 已知广义表  $LS = ((a,b,c),(d,e,f))$ ,运用 head 和 tail 函数取出 LS 中原子 e 的运算是( )。  
(A)  $\text{head}(\text{tail}(\text{head}(\text{tail}(LS))))$  (B)  $\text{tail}(\text{head}(LS))$   
(C)  $\text{head}(\text{tail}(LS))$  (D)  $\text{head}(\text{tail}(\text{tail}(\text{head}(LS))))$
9. 对于链式存储的线性表,访问结点和增加结点的时间复杂度为( )。  
(A)  $O(n)$   $O(n)$  (B)  $O(n)$   $O(1)$  (C)  $O(1)$   $O(n)$  (D)  $O(1)$   $O(1)$
10. 若栈采用顺序存储方式存储,现两个栈共享空间  $V[1..m]$ ,  $\text{top}[j]$  表示第  $j$  个栈 ( $j=1,2$ ) 栈顶指针,栈 1 的底设在  $V[1]$ ,栈 2 的底设在  $V[m]$ ,则栈满的条件是( )。  
(A)  $\text{top}[2] - \text{top}[1] == m$  (B)  $\text{top}[2] - \text{top}[1] == 1$   
(C)  $\text{top}[2] - \text{top}[1] == 0$  (D)  $\text{top}[2] + \text{top}[1] == m$

## 二、填空题(每空 1 分,共 10 分)

1. 循环队列用下标范围是 0 到  $m$  的数组  $V$  存放其元素值, 已知其头、尾指针分别是  $f$  和  $r$ ,  $f$  是队首元素的前一个位置,  $r$  是队尾元素位置, 若用牺牲一个单元的办法来区分队满和队空, 则队满的条件为\_\_\_\_\_。
2. 有一个  $100 \times 200$  的元素值为整型的稀疏矩阵, 非零元素有 20 个, 设每个整型数占 2 字节, 则用三元组顺序表表示该矩阵时, 所需的字节总数是\_\_\_\_\_。
3. 如果按关键码值递增的顺序依次将 99 个关键码值插入到二叉排序树中, 则对这样的二叉排序树检索时, 在等概率情况下查找成功时的平均查找长度 ASL 为\_\_\_\_\_。
4. 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 ( $i < n$ ) { for ( $j=1$ ;  $j \leq n$ ;  $j++$ )  $x=x+1$ ;  $i=i*2$ ;
5. 对于长度为 54 的表, 若索引表和各块内均用顺序查找, 且每块长度为 6, 则在等概率情况下查找成功的平均查找长度为\_\_\_\_\_。
6. 将有关二叉树的概念推广到三叉树, 则一棵有 244 个结点的完全三叉树的高度是\_\_\_\_\_。
7. 两个字符串相等的充分必要条件是\_\_\_\_\_。
8. 构造  $n$  个结点的强连通图, 至少有\_\_\_\_\_条弧。
9. 表达式  $a*(b+c)-d$  的后缀表达式是\_\_\_\_\_。
10. 一棵完全二叉树上有 1000 个结点, 其中叶子结点的个数是\_\_\_\_\_。

## 三、应用题(共 80 分)

1. (12 分) 假设一棵二叉树的中序序列为 GLDHBIEIACJFK, 后序序列为 LGHDIEBJKFCA。要求:
  - (1) 画出该二叉树。
  - (2) 画出该二叉树所对应的森林。
  - (3) 画出该二叉树的中序全线索二叉树。
  - (4) 画出该二叉树所对应的森林中第一棵树的带双亲域的孩子链表。
2. (8 分) 给定一组数据 {15, 8, 10, 21, 6, 19, 3} 分别代表字符 A, B, C, D, E, F, G 出现的频度, 要求:
  - (1) 画出构造哈夫曼树的过程以及最后的哈夫曼树。
  - (2) 在哈夫曼树上对各字符进行编码, 并给出各字符的编码值。
  - (3) 计算哈夫曼树的带权路径长度 WPL。
3. (20 分) 设  $G=(V, E)$  为有向网, 其中  $V(G)=\{V1, V2, V3, V4, V5, V6\}$ 。现用三元组  $\langle x, y, z \rangle$  表示弧  $\langle x, y \rangle$  以及弧上的权值  $z$ , 则  $E(G)$  为  $E(G)=\{ \langle V6, V5, 100 \rangle, \langle V6, V4, 30 \rangle, \langle V4, V5, 60 \rangle, \langle V4, V3, 20 \rangle, \langle V3, V5, 15 \rangle, \langle V2, V6, 10 \rangle, \langle V2, V3, 50 \rangle, \langle V1, V2, 5 \rangle \}$ 。要求:
  - (1) 画出该有向网。
  - (2) 画出该有向网的邻接矩阵。
  - (3) 求基于你的邻接矩阵的从顶点  $V1$  出发的 DFS 序列以及 DFS 生成树。
  - (4) 给出该有向网的所有拓扑有序序列。
  - (5) 用 dijkstra 算法, 求从源点  $V1$  出发到其它各终点的最短路径以及长度, 请给出执行算法过程中各步的状态, 可用下面给出的表格形式表示。
4. (12 分) 有一结点的关键字序列  $F=\{26, 36, 41, 38, 44, 15, 68, 12, 06, 51, 25\}$ , 散列函数为:  $H(K)=K \% 11$ , 其中  $K$  为关键字, 散列地址空间为  $0 \sim 10$ 。要求:
  - (1) 画出相应的闭散列表。发生冲突时, 以线性探测法解决。
  - (2) 该散列表的装填因子  $\alpha$  是多少?
  - (3) 在等概率情况下查找成功时的平均查找长度 ASL 是多少?
  - (4) 用线性探测法解决冲突时, 如何处理散列表中删除的结点? 为什么?

5.(8分)已知长度为10的表{16,3,7,12,9,28,25,18,14,20},按表中元素顺序依次插入一棵初始时为空的平衡二叉排序树中,画出每一步插入后平衡二叉排序树的形态。若做了某种旋转,说明旋转的类型。

6.(20分)已知关键字序列  $F=\{22,12,26,40,18,38,14,20,30,16,28\}$ 。要求:

- (1) 将该序列调整为“小顶”堆,并给出调整过程。请从时间和空间两方面对简单选择排序、树形选择排序和堆排序作一比较。
- (2) 若采用链式基数排序方法排序,请写出第一趟“分配”之后各队列的状态和第一趟“收集”之后的关键字序列。并请简要说出基数排序方法和其他排序方法有什么区别?
- (3) 要求最后排序结果是按关键字从小到大的次序排列,请给出冒泡排序的前3趟排序结果。

#### 四、简答题(共30分)

- 1.(6分)线性表的顺序存储结构和链式存储结构各有哪些优缺点?
- 2.(8分)画出对长度  $n$  为10的递增有序表  $\{A[1], A[2], \dots, A[10]\}$  进行折半查找的判定树。当实现插入排序过程时,可以用折半查找来确定第  $i$  个元素在前  $i-1$  个元素中的可能插入位置,这样做能否改善插入排序的时间复杂度?为什么?
- 3.(8分)快速排序的平均时间复杂度是多少?快速排序在所有同数量级的排序方法中,其平均性能好。但在什么情况下快速排序将蜕化为起泡排序,此时的时间复杂度又是多少?为改进之,通常可以怎么做?
- 4.(8分)请简要叙述求最小生成树的 Prim 算法的思想(或步骤)。并指出对具有  $n$  个顶点和  $e$  条边的连通网而言,Prim 算法和 Kruskal 算法各自适合于什么情况下的连通网?其时间复杂度又各为多少?

#### 五、算法设计题(共20分)

- 1.(10分)给定(已生成)一个带头结点的单链表,设 head 为头指针,结点的结构为  $(data,next)$ ,data 为整型元素, next 为指针,试写出算法:按递减次序输出单链表中各结点的数据元素,并释放结点所占的存储空间。(要求:不允许使用数组作辅助空间)
- 2.(10分)以二叉链表作为二叉树的存储结构,试编写递归算法,求二叉树中以元素值为 item 的结点为根的子树的深度(假设树中一定存在值为 item 的结点)。

注意:

- (1)可用(类)Pascal 语言或(类)C 语言或 C++语言描述你的算法;
- (2)请简要描述你的算法思想;
- (3)若你的算法是(类)Pascal 或(类)C 语言编写,则请给出相应的存储结构描述;
- (4)若你的算法是用 C++语言描述,则可参考使用以下给出的相关存储结构的类定义,算法中可以使用类中已列出的成员函数。若在你的算法中使用了未列出的成员函数,则要写出该成员函数的完整算法描述。

//单链表的类定义

```
template <class type> class linklist; //单链表前视声明
template <class type> class node{//单链表结点类
    friend class linklist <type>; //定义单链表类 linklist <type>为结点类的友元
private:
    node <type> *next; //链指针域
public:
    type data; //数据域
    node (node <type> *pNext = NULL) {next = pnext;}//构造函数,用于构造头结点
};
template <class type> class linklist{ //单链表类定义
private:
    node <type> *head; //指向头结点的头指针
public:
    linklist () { head = new node <type> (); head->next=NULL; }//构造函数
    ~linklist (); //析构函数
};
```

//二叉链表的类定义:

```
template <class Type> class BinaryTree; //二叉链表类前视声明, 以便使用友元
template <class Type> class BinTreeNode { //结点类
friend class BinaryTree<Type>;
private:
    BinTreeNode<Type> *leftChild, *rightChild; //结点的左、右孩子指针域
    Type data; //结点的数据域
public:
    BinTreeNode () : leftChild (NULL), rightChild (NULL) { } //构造函数, 构造一个空结点
    BinTreeNode (Type d, BinTreeNode<Type> *lp = NULL, BinTreeNode<Type> *rp =NULL) : data
        (d), leftChild (lp), rightChild (rp) { }
        //构造函数, 构造一个数据域的值为 d 的结点
};
template <class Type> class BinaryTree { //二叉链表类
private:
    BinTreeNode <Type> *root; //二叉树根结点指针
public:
    BinaryTree () : root (NULL) { } //构造函数
    ~BinaryTree () { destroy ( root ); } //析构函数
};
```