

江苏大学 2007 年硕士研究生入学考试试题

科目代码: 445

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 2 分, 共 20 分)

- 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) 双循环链表 (C) 带头结点的双循环链表 (D) 单循环链表
- 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
- 若栈采用顺序存储方式存储, 现两个栈共享一个下标范围是 1 到 m 数组 A 的空间, $top[i]$ 表示第 i 个栈($i=1,2$)栈顶, 栈 1 的底设在 $A[1]$, 栈 2 的底设在 $A[m]$, 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) $top[1] == top[2]$
- 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, 则当前队列中的元素个数是()。
(A) $(r-f+m) \% m$ (B) $(r-f+m+1) \% m$
(C) $(r-f+m+1) \% (m+1)$ (D) $(r-f+m) \% (m+1)$
- 下面关于串的叙述中, 哪一个是不正确的? ()
(A) 串是字符的有限序列
(B) 空串是由空格构成的串
(C) 模式匹配是串的一种重要运算
(D) 串既可以采用顺序存储, 也可以采用链式存储
- 有一个 100×90 的元素值为整型的稀疏矩阵, 非零元素有 10 个, 设每个整型数占 2 字节, 则用三元组顺序表表示该稀疏矩阵时, 所需的字节总数是()。
(A) 60 (B) 66 (C) 18000 (D) 33
- 已知广义表 $L = ((a,b,c), d, (e, f, g))$, 从 L 表中取出原子项 f 的运算是()。
(A) $head(tail(tail(tail(L))))$ (B) $tail(head(head(tail(L))))$
(C) $head(tail(head(tail(L))))$ (D) $head(tail(head(tail(tail(L))))$
- 设 n 是图的顶点数, e 是图的边数, 则在图采用邻接~~表~~^{矩阵}存储时, 求最小生成树的 Prim 算法的时间复杂度为()。
(A) $O(n)$ (B) $O(n^2)$ (C) $O(n+e)$ (D) $O(e^2)$
- 在待排序的元素序列基本有序的前提下, 效率最高的排序方法是()。
(A) 插入排序 (B) 选择排序 (C) 快速排序 (D) 归并排序
- 双向循环链表中有两个指针域, $next$ 和 $priou$, 分别指向前驱及后继, 设 p 指向链表中的一个结点, s 指向一待插入结点, 现要求在 p 后插入 s , 则正确的插入为()。
(A) $p \rightarrow next = s; s \rightarrow priou = p; p \rightarrow next \rightarrow priou = s; s \rightarrow next = p \rightarrow next;$
(B) $p \rightarrow next \rightarrow priou = s; p \rightarrow next = s; s \rightarrow priou = p; s \rightarrow next = p \rightarrow next;$
(C) $s \rightarrow priou = p; s \rightarrow next = p \rightarrow next; p \rightarrow next = s; p \rightarrow next \rightarrow priou = s;$
(D) $s \rightarrow priou = p; s \rightarrow next = p \rightarrow next; p \rightarrow next \rightarrow priou = s; p \rightarrow next = s;$

二、填空题（每空 2 分, 共 20 分）

- 下列程序段的时间复杂度是 $O(\quad)$ 。

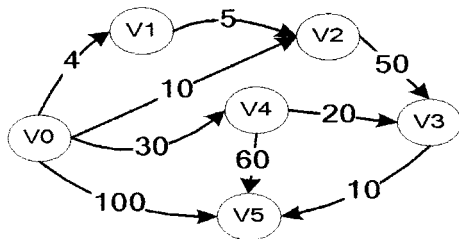
```
int i=1,j=1;
while (i<=n&& j<=n){ i=i+1; j=j+i; }
```
- 线性表 $L=(a_1, a_2, \dots, a_n)$ 用数组表示, 假定删除表中任一元素的概率相同, 则删除一个元素平均需要移动元素的个数是 \quad 。
- 若已知一个栈的入栈序列是 $1, 2, 3, \dots, n$, 其输出序列为 $p_1, p_2, p_3, \dots, p_n$, 若 p_1 是 n , 则 p_i 是 \quad 。
- 模式串 $P='abaabcac'$ 的失败函数 next 的函数值序列为 \quad 。
- 在一棵完全二叉树中, 编号为 i 和 j 的两个结点处在同一层的条件是 \quad 。
- 具有 N 个结点的二叉树, 采用二叉链表存储, 共有 \quad 个空链域。
- 有一份电文中共使用 6 个字符: a, b, c, d, e, f, 它们的出现频率依次为 2, 3, 4, 7, 8, 9, 若构造一棵哈夫曼树, 则其加权路径长度 WPL 为 \quad 。
- G 是一个非连通无向图, 共有 30 条边, 则该图至少有 \quad 个顶点。
- 在 n 个记录的有序顺序表中进行折半查找, 最大比较次数是 \quad 。
- 若不考虑基数排序, 则在排序过程中, 主要进行的两种基本操作是关键字的 \quad 和记录的 \quad 。

三、应用题（共 90 分）

- (5 分)按照四则运算加、减、乘、除优先关系的惯例, 画出对表达式 $8*(5-6/2)$ 求值时运算符栈、操作数栈的变化过程。(按以下格式填写, 其中 OPTR 栈为运算符栈, OPND 栈为操作数栈, #表示表达式开始或结束符)

步骤	OPTR 栈	OPND 栈	输入字符
初始	#		$8*(5-6/2)\#$

- (5 分)对于二叉树 BT 的两个结点 n_1 和 n_2 , 我们应该选择二叉树 BT 结点的前序、中序和后序中哪两个序列来判断结点 n_1 必定是结点 n_2 的祖先, 并请给出判断的方法 (不需证明判断方法的正确性)。
- (20 分)假设一棵二叉树的层次序列为 ABCDEFGHIJ, 中序序列 DBGEHJACIF。要求:
 - 画出该二叉树。
 - 画出该二叉树所对应的森林。
 - 给出对该森林进行后根遍历的序列。
 - 画出该二叉树的中序全线索二叉树。
 - 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。
- (15 分)已知带权有向图如图所示。要求:
 - 画出该带权有向图的邻接表。
 - 给出该有向图的所有拓扑有序序列。
 - 用 dijkstra 算法, 求从源点 V_0 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。



终点	从源点 V_0 到各终点的最短路径以及最短路径长度的求解过程				
	i=1	i=2	i=3	i=4	i=5
V1					
V2					
V3					
V4					
V5					
V_j					

5. (13 分)有一结点的关键字序列 $F=\{26,36,41,38,44,15,68,12,06,51,25\}$,利用除留余数法构造哈希函数,假设装填因子(负载系数) $\alpha=0.75$ 。要求:
- (1) 请设计出最合适的哈希函数。
 - (2) 若以线性探测(再散列)方法解决冲突,请画出相应的散列表。并求在等概率情况下查找成功时的平均查找长度ASL()。
 - (3) 用线性探测(再散列)方法解决冲突时,如何处理被删除的结点?为什么?
6. (22 分)请完成以下要求:
- (1) 已知关键字序列 $F=\{12,2,16,30,8,28,4,10,20,6,18\}$,要求最后排序结果是按关键字从小到大的次序排列,请给出增量序列 $D=\{5,3,2,1\}$ 的希尔排序第一趟排序结果。
 - (2) 已知关键字序列 $F=\{321,156,57,46,28,7,331,33,34,63\}$,请给出链式基数排序第一次分配和收集的过程。
 - (3) 已知关键字序列 $F=\{29,18,25,47,58,12,51,10\}$,请把该序列变成一个最大堆序列,并按照先后发生比较的次序,给出建堆时关键字比较的情况,比如,第一次的比较情况是47和10发生比较。
 - (4) 请从时间和空间两方面对简单选择排序、树形选择排序(也称为锦标赛排序)和堆排序作一比较。
7. (10 分)在二叉排序树中删除结点 p 可能有这样几种情况:
- (1) 结点 p 是叶子结点。
 - (2) 结点 p 只有左子树或只有右子树。
 - (3) 结点 p 的左子树和右子树都不空。
- 请针对以上三种情况讨论在二叉排序树中如何删除结点 p。

四、算法设计题 (共 20 分)

1. (10 分)设有一带头结点、不循环的头指针为 head 的单链表 L (至少有 1 个结点),请编写算法将该单链表 L 就地逆置,即利用单链表原有的结点空间将线性表 (a_1,a_2,\dots,a_n) 逆置为 (a_n,a_{n-1},\dots,a_1) 。
2. (10 分)试基于图的深度优先搜索策略编写一算法,判别以邻接表方式存储的有向图中是否存在编号为 i 的顶点到编号为 j 的顶点的路径 ($i \neq j$)。注意:在你所编写的算法中若引入了辅助的存储结构,要简要说明,此外,算法中涉及的图的基本操作必须在邻接表这种存储结构上实现。

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
```

```
    type data; //数据域
```

```

public:
    node ( );                //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 ( ){              //构造函数
        head = new node <type> ( );    //令头指针指向头结点
        head->next=NULL;}
    ~linklist ( );            //析构函数
};

```

//有向图的邻接表类定义:

```

const int MaxVertexes = 20; //最大的顶点数
template <class vertexType, class arcType> class Graph
template < class arcType> struct ArcNode { // 定义弧结点
    friend class Graph <class vertexType, class arcType>;
    int adjvex; //和弧相关联的另一个顶点序号
    arcType weight; //弧上的信息（权）
    ArcNode<arcType> *nextarc ; //指向下一条弧结点的指针
};
template < class arcType , class vertexType > struct VertexNode {
// 定义顶点结点
    friend class Graph <class vertexType, class arcType>;
    vertexType data; //顶点的信息
    ArcNode<arcType> *firstarc ; //指向依附该顶点的弧链表
};
template <class vertexType, class arcType> Graph{
private:
    VertexNode < arcType , vertexType > * VertexesTable; //顶点表
    int CurrentNumVertexes; //当前的顶点数
    int  CurrentNumArcs; //当前的弧数
public:
    Graph : CurrentNumVertexes (0), CurrentNumArcs (0){} //构造函数
    Graph ( vertexType v[ ], int num = MaxVertexes ); //构造函数
    ~Graph ( ); //析构函数
};

```