

中山大学

二00五年攻读硕士学位研究生入学考试试题

科目代码: 817

科目名称: 数据结构

考试时间: 1 月 23 日 下午

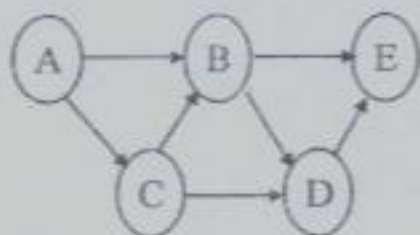
考生须知

全部答案一律写在答题纸上,
答在试题纸上的不得分! 请用
蓝、黑色墨水笔或圆珠笔作答。
答题要写清题号, 不必抄题。

一. 单项选择题(每题1分, 16分)

1. 可以用 (), 数据关系和基本操作定义一个完整的抽象数据类型:
A. 数据元素 B. 数据对象 C. 原子类型 D. 存储结构
2. 四种基本的数据结构为: 集合, 线性结构, 树形结构和 ()。
A. 广义表 B. 平衡树 C. 图形结构 D. 复合结构
3. 假定 1, 2, 3, 4 和 5 依次进入一个栈, 并且中途可以出栈, 则可能的出栈次序为 ()。
A. 3 4 1 2 5 B. 2 1 5 3 4 C. 4 5 3 1 2 D. 2 1 4 5 3
4. () 的先序遍历序列和后序遍历序列正好相反。
A. 二叉排序树 B. 三个结点的二叉树
C. 平衡二叉树 D. 无右孩子的二叉树
5. N 个结点的强连通图至少有 () 条弧。
A. N B. $N-1$ C. $2N$ D. $N(N-1)$
6. 对于折半查找法, 一般选择 () 作为其存储结构。
A. 顺序表 B. 双链表 C. 循环链表 D. 单链表
7. 对二叉排序树进行 () 遍历, 可以得到所有结点的一个有序序列。
A. 先序 B. 中序 C. 后序 D. 逐层
8. 以下四种排序方法中, () 是稳定的排序方法。
A. 希尔排序 B. 快速排序 C. 归并排序 D. 堆排序
9. 三维数组 $A[8, 8, 10]$ 用行主序的方式从地址 $A[0, 0, 0]$ 开始存放, 问: $A[3, 2, 8]$ 的存放位置为 ()。假设每个数组元素占用存储空间的大小为 L 。
A. $A[0, 0, 0] + 198 * L$ B. $A[0, 0, 0] + 108 * L$
C. $A[0, 0, 0] + 268 * L$ D. $A[0, 0, 0] + 13 * L$
10. () 是表示稀疏矩阵的一种方法。
A. 十字链表 B. 循环链表 C. 邻接表 D. 邻接多重表
11. 建立 () 是为了将二叉树进行线性化。
A. 线索二叉树 B. 平衡二叉树 C. 二叉排序树 D. 最优二叉树

12. 若可利用空间表共有三个空闲块, 其大小依次为 12000, 30000 和 8000, 若申请 6000 的一块空闲块, 采用 () 进行分配, 可利用空间表的空闲块变为 12000, 30000 和 2000。
A. 首次拟合法 B. 最佳拟合法 C. 最差拟合法 D. 伙伴系统。
13. 一个无序序列 12, 36, 41, 20, 80, 55 采用顺序表存储表示, 采用堆排序算法建立的初始(大顶)堆为:
A. 80, 12, 55, 20, 30, 41 B. 80, 30, 20, 12, 55, 41
C. 80, 30, 55, 20, 12, 41 D. 80, 12, 55, 20, 30, 41
14. 下图为 AOV 网, 其可能的拓扑有序序列为()。



- A. ABDCE B. ABCDE C. ACBDE D. ACDBE
15. 将上图看作无向图, 其从 A 出发的广度优先遍历结果可以为()。
A. ABECD B. ACBDE C. ACDBE D. ABDEC
16. 若一个 n 个结点和 e 条边的图采用邻接表作为其存储结构, 其深度优先遍历的时间复杂性为 ()。
A. $O(n^2)$ B. $O(e^2)$ C. $O(n+e)$ D. $O(e)$

二. 填空题(每题 1 分,12 分)

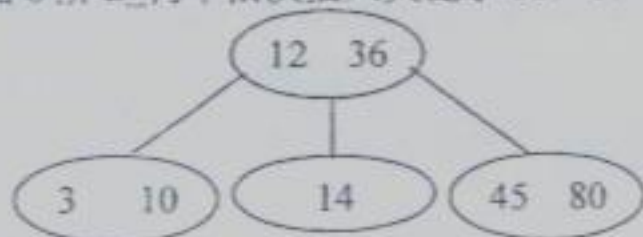
1. 当输入非法错误时, 一个“好”的算法会进行适当处理, 而不会产生难以理解的输出结果。这称为算法的_____。
2. 栈为后进先出, 而队列为_____。
3. 具有 N (N 为奇数) 个叶子结点的完全二叉树总共有_____个结点。
4. 假设结点的 next 域指向其直接后继, head 指向一个循环单链表的头结点, 表达式_____可以用来判定该循环单链表是否为空。
5. 哈希函数的基本构造原则是_____。
6. 克鲁斯卡尔算法和普里姆算法相比, 从时间复杂性来讲, _____适合求边稀疏的网的最小生成树。
7. 对 n 个无序数进行归并排序的时间复杂性为_____。
8. 广义表 $((a), b, c)$ 的表尾是_____。
9. 对序列 80, 30, 50, 92, 36, 81, 95, 10 进行快速排序, 若以 85 为支点, 第一趟排序的结果为_____。
10. 求 $n!$ 的递归函数是 `int factor (int n){_____}`。
11. 将一棵有 100 个结点的完全二叉树从根这一层开始, 每一层上从左到右依次对结点进行编号, 根结点的编号为 1, 则编号为 49 的结点的双亲编号为_____。
12. 树的孩子兄弟表示法是以_____作为树的存储结构。

三. 作图或简答题(每题 6 分, 共 72 分)

- 假设字符 a,b,c,d,e,f 的使用频度分别是 0.07, 0.09, 0.12, 0.22, 0.23, 0.27, 画出赫夫曼树并写出 a,b,c,d,e,f 的赫夫曼编码。
- 广义表 (A(B(E, F), C, D(G))) 表示一棵树, 请画出该树并给出其中序遍历序列。
- 一个 7×7 的稀疏矩阵的三元组表示为 $\langle 1, 2, 15 \rangle, \langle 3, 2, 9 \rangle, \langle 5, 7, 18 \rangle, \langle 5, 6, 6 \rangle$, 请采用十字链表表示之。
- 依次插入结点(关键字)10, 20, 18, 15, 12 构造一棵平衡二叉树, 并标出结点的平衡因子。
- 假设一棵二叉树的先序遍历序列为 ABCDEFGH 和中序遍历序列为 CBDAEGHF。请画出该二叉树。
- 画出有序表 (11, 18, 28, 40, 45, 56, 88) 的折半查找判定树(按下取整, 数组下标从 0 开始)。
- 如下图已知哈希表为空, 哈希函数为 $H(\text{Key}) = \text{Key} \bmod 11$, 用二次探测再散列作为冲突解决方法, 请填入在依次插入关键字 5, 37, 14, 26 之后的情况。

0	1	2	3	4	5	6	7	8	9	10

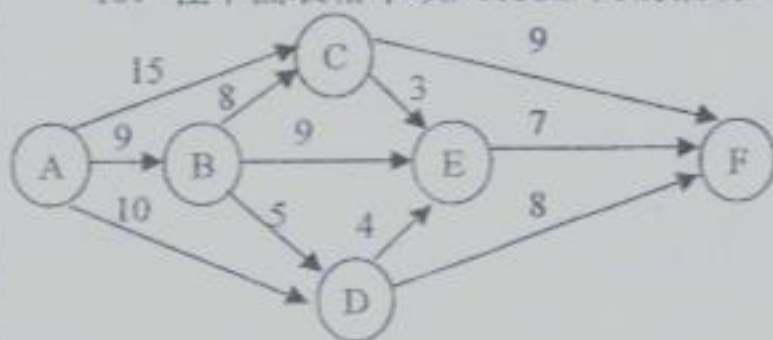
- 画出下面 3 阶 B₊ 树中依次插入关键字 38, 6, 16 后所得的结果:



- 已知无向图 G 的邻接矩阵如下 (其中 1 和 0 分别表示有边相连和无边相连), 画出图 G 及其邻接多重表。

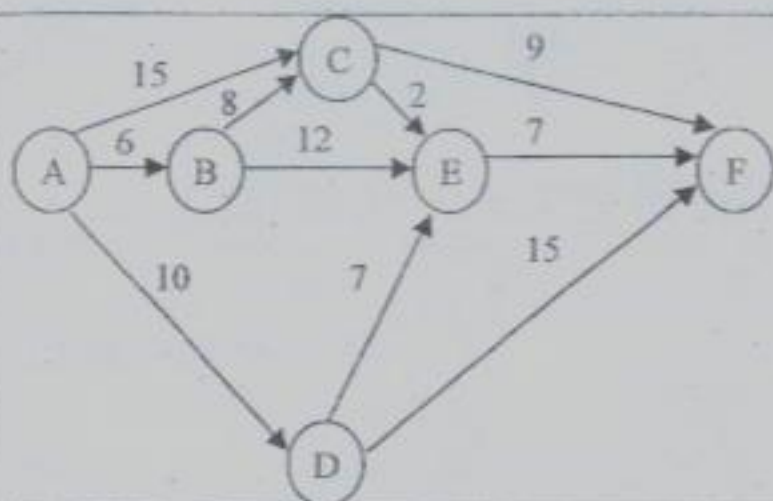
	A	B	C	D	E
A	0	1	1	0	0
B	1	0	0	1	0
C	1	0	0	1	1
D	0	1	1	0	0
E	0	0	1	0	0

- 在下面表格中填入 AOE 网的活动 (边) 的最早开始时间和最迟开始时间。



	AB	AC	AD	BC	BD	BE	CE	CF	DE	DF	EF
最早开始时间											
最迟开始时间											

- 给定一个无向图, 需要判定图中任意两个结点是否处于一个连通分量, 请给出一种算法。
- 按 Dijkstra 最短路径算法的计算顺序依次给出从顶点 v_0 到所有其它结点的最短路径的中间计算结果和最终结果。



	i=1	i=2	i=3	i=4	i=5
A					
B					
C					
D					
E					

四. 编程题(共 50 分)

1. (10 分)以下为用顺序存储结构表示的循环队列的出队列操作的类 C 语言程序, 请填空。

```

#define OK          1
#define ERROR       0
#define MAXSIZE     100 // 最大队列长度
typedef int          Status;
typedef int          QElemType;
typedef struct {
    QElemType *base; // 指向队列的存储空间
    ① front; // 头指针, 若队列不空, 指向队列头元素
    ② rear; // 尾指针, 若队列不空, 指向队列尾元素的下一个位置
} SqQueue;

```

```

Status DeQueue (SqQueue &Q, QElemType e){
    if (③) return ERROR;
    e = ④;
    Q.front = ⑤;
    return OK;
}

```

2. (20) 用类 C 语言写出复制广义表的算法:

```

#define OK          1
#define ERROR       0
#define OVERFLOW    -2

```

```

typedef int Status;
typedef char AtomType;
typedef enum {ATOM, LIST} ElemTag;
// 广义表的扩展线性链表存储表示
typedef struct GLNode{
    ElemTag tag;
    union {
        AtomType atom;
        struct GLNode * hp;
    };
    struct GLNode * tp;
} * GList;
Status CopyGList(GList &T, GList L){
    // 由广义表 L 复制得到广义表 T
    .....
}

```

3. (20 分) 用类 C 语言写出中序遍历建立中序线索链表的算法:

```

// 二叉树的二叉线索存储表示
#define OK 1
#define ERROR 0
#define OVERFLOW -2
typedef int Status;
typedef int TElemType;
typedef enum {Link, Thread} PointerTag; // Link 表示指针, Thread 表示线索
typedef struct BiThrNode {
    TElemType data;
    struct BiThrNode * lchild, * rchild; // 左右孩子指针
    PointerTag LTag, RTag; // 左右标志
} BiThrNode, *BiThrTree;
BiThrNode * pre; // 全局变量, 可以用来指向上一个访问的结点

// InOrderThreading 的参数 Thrt 指向头结点

```



```

// 若二叉树 T 为空，头结点的左右孩子指向头结点自己；
// 若二叉树 T 非空，头结点的左孩子指向二叉树的根而右孩子指向线索链表的
// 最后一个结点；
// 中序线索链表的第一结点的前驱和最后一个结点的后继均为头结点。
Status InOrderThreading(BiThrTree &Thrt, BiThrTree T){
    // 设置与头结点相关的指针和线索，并调用 InThreading 为 T 建立中序线索树
    if (!(Thrt = (BiThrTree)malloc(sizeof(BiThrNode)))) exit(OVERFLOW);
    .....

}

void InThreading(BiThrTree p){
    //建立二叉树 p 的中序线索树的递归算法
    .....
}

```