

西 南 大 学

二零零六年攻读硕士学位研究生入学考试试题

学科、专业：计算机应用技术 研究方向：

计算机软件与理论

考试科目：数据结构含程序设计 编 号：471

(答题一律做在答题纸上，并注明题目番号，否则答题无效。)

一、 选择题 (每题 3 分，共 24 分)

1、若对 n 阶对称矩阵 A 以行序为主序方式将其下三角形的元素(包括主对角线上所有元素)依次存放于一维数组 $B[1..n*(n+1)/2]$ 中，则在 B 中确定 $a_{ij}(i < j)$ 的位置 k 的关系为_____。

- (A) $i*(i-1)/2+j$ (B) $j*(j-1)/2+i$
(C) $i*(i+1)/2+j$ (D) $j*(j+1)/2+i$

2、深度为 h 的满 m 叉数的第 k 层有_____ 个结点。($1 \leq k \leq h$)

- (A) m^{k+1} (B) m^{k-1} (C) m^{h-1} (D) m^{h+1}

3、具有 10 个叶结点的二叉树中有_____ 个度为 2 的结点。

- (A) 8 (B) 9 (C) 10 (D) 11

4、下面给出的四种排序方法中，最稳定的排序方法是_____。

- (A) 希尔排序法 (B) 快速排序法
(C) 基数排序法 (D) 堆排序法

5、若在线性表中采用折半查找，则该线性表应该_____。

- (A) 采用顺序存储结构
(B) 元素按值有序
(C) 元素按值有序且采用链式存储结构
(D) 元素按值有序且采用顺序存储结构

6、设有一个顺序栈 S ，元素 $s_1, s_2, s_3, s_4, s_5, s_6$ 依次进栈，如果 6 个元素的出栈顺序为 $s_2, s_3, s_4, s_6, s_5, s_1$ ，则顺序栈的容量至少应为_____。

- (A) 3 (B) 5
(C) 4 (D) 6

7、已知一算术表达式的中缀形式为 $A+B*C-D/E$ ，后缀形式为 $ABC*+DE/-$ ，则其前缀形

式为_____。

- (A) $-A+B*C/DE$ (B) $-A+B*CD/E$
(C) $-+*ABC/DE$ (D) $-+A*BC/DE$

8、具有_____条边的无向图称为完全图。

- (A) $(n-1)/2$ (B) $n*(n-1)/2$
(C) $n*(n-1)$ (D) $n*n$

二、综合题：（共 72 分）

1、（12 分）设在地址空间为 $0 \sim 11$ 的散列区中，对关键字序列 {21, 8, 12, 16, 19, 26, 5, 7, 22, 27} 分别按不同冲突处理方法构造哈希表。哈希函数为：

$H(key) = key \% 11$ ； 注：%是取模运算（ $= \text{mod}$ ）

- ① 用线性探测开放地址法处理冲突，并计算等概率情况下搜索成功的平均搜索长度 ASL。
② 用链地址法解决冲突，并计算等概率情况下搜索成功的平均搜索长度 ASL。

2、（20 分）对于序列 (47, 36, 89, 93, 12, 109, 71, 29, 84, 32, 80, 49)，写出用快速排序法和堆排序进行排列(升序)的过程，以及构造二叉排序树和平衡二叉树的过程。

3、（10 分）已知一字符串 ebcdbdccbdbdbcbdebc，设计其哈夫曼编码并画出相应的哈夫曼树。

4、（8 分）假设一棵二叉树的层序序列为 ABHCDIJEFKG，中序序列为 CBEDFGAIKHJ，请画出该树。

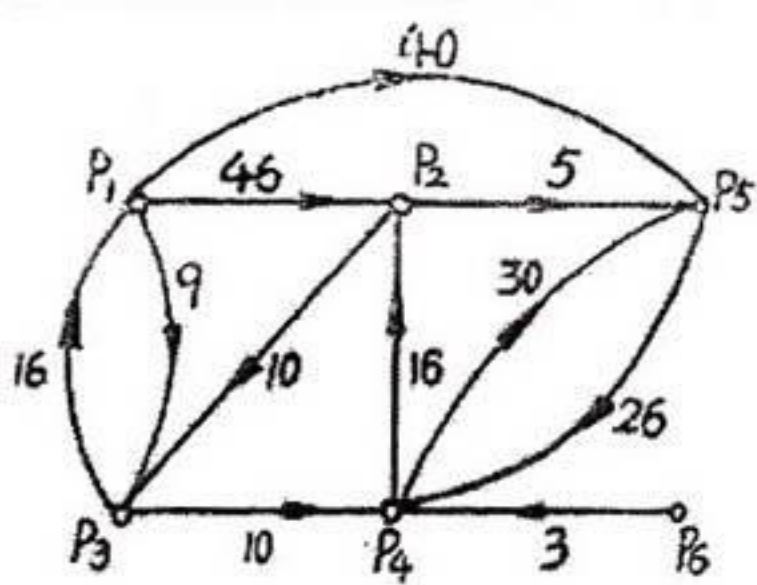
5、（7 分）设有一个工程包含了 8 个子工程，这些子工程之间有如下优先关系：

$1 > 2, 4$ $2 > 3, 5$ $3 > 6$ $4 > 3, 5$ $5 > 7$ $6 > 8$

（这里 $1 > 2, 4$ 表示子工程 1 需要在子工程 2 和子工程 4 开始之前完成，其它依次类推。）

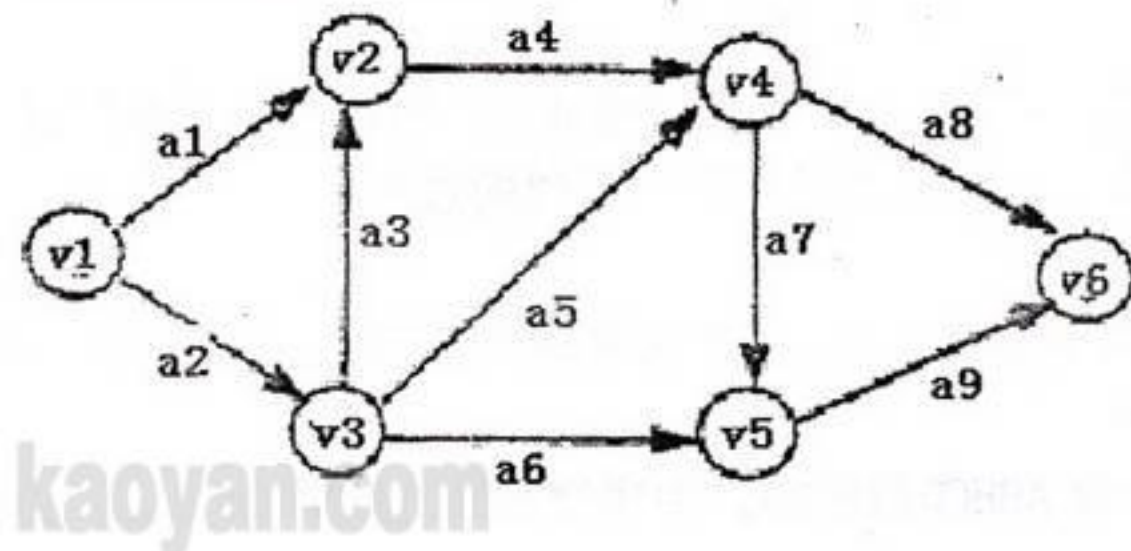
画出该图，若用邻接链表存储该图，每个定点的邻接点按定点序号从小到大排列，试并说明该工程的可行性，给出可行方案。

6、（10 分）有如下带权有向图，请用 Dijkstra 算法求从 P1 到其他各结点的最短路径。



7、(5 分) 根据下图给出的 AOE 网, 其中: $a_1=2, a_2=3, a_3=3, a_4=5, a_5=9, a_6=4, a_7=6, a_8=2, a_9=3$, 计算如下各问题:

- (1) 每个事件的最早发生时间 $Ve(j)$ 和最迟发生时间 $VI(j)$;
- (2) 每个活动的最早开始时间 $e(i)$ 和最迟开始时间 $l(i)$;
- (3) 找出其关键路径。



三、 程序填空: (共 24 分)

1、(14 分) 有两个带有表头结点的循环链表 A 和 B, 其中的结点值均为字母, A 表非递减有序。现在希望构造一个递增有序的循环链表 C, C 中结点的值为同时在 A, B 两链表中出现的字母, 且 C 中字母不重复。(设 A, B 的结点数分别为 m, n)。

```
typedef struct LNode{
    char data;
    struct LNode *next ;
} *Link;

Link ABC( Link A, Link B ){
    Link p, q, r, s;
    C = ( Link )malloc( sizeof( LNode ));
    C->next = C;
    q = A; p = A->next;
```



```

while ( _____ (1) _____ ) { //当 A 链表还没有结束时, 进行循环
    //跳过相同字母
    while ( p->data == p->next->data ) {
        _____ (2) _____
        p = p->next;
    }
    r = B->next;
    //找出相同字母并将该字母加入链表 C 中
    while ( r->data != p->data && r != B ) r = r->next;
    if ( r != B ) {
        s = p; q->next = p->next;
        _____ (3) _____;
        _____ (4) _____;
        _____ (5) _____;
    }
    else {
        _____ (6) _____;
        _____ (7) _____;
    }
    C = C->next;
}
return C ;
}

```

2、(10 分) 本程序完成将二叉树中左、右孩子交换的操作, 交换的结果如下图所示。本程序采用非递归的方法, 设立一个堆栈 stack 存放还没有转换过的结点, 它的栈顶指针为 tp。交换左、右子树的算法为:

- (1) 把根结点放入堆栈。
- (2) 当堆栈不空时, 取出栈顶元素, 交换它的左、右子树, 并把它的左、右子树分别入栈。
- (3) 重复 (2) 直到堆栈为空时为止。

程序:

```

typedef struct node *tree;
struct node{
    int data;
    tree lchild, rchild;
}

```

```
Exchange( tree t ){
```

```
    tree  r,  p;
```

```
    tree  stack [500];
```

```
    int  tp=0;
```

```
    _____(1)_____;
```

```
    while (  tp>=0 ){
```

```
        _____(2)_____;
```

```
        if _____(3)_____ {
```

```
            r = p→lchild;    p→lchild = p→rchild;    p→rchild = r;
```

```
            _____(4)_____ = p→lchild;
```

```
            _____(5)_____ = p→rchild;
```

```
        }//if 结束
```

```
    }//while 结束
```

```
}
```

四、 算法设计：（）

1、（10 分）已知 Q 是一个空队列，S 是一个非空栈。仅用队列和栈的 ADT 函数和少量工作变量，使用 C 语言编写一个算法，将队列栈 S 中的所有元素逆置。

栈的 ADT 函数有：

ClearStack(SqStack &S); 置空栈

Push(SqStack &S, SElemType e); 新元素 e 进栈

Pop(SqStack &S, SElemType &e); 出栈，返回栈顶值

StackEmpty(SqStack S); 判栈空否

队列的 ADT 函数有：

EnQueue(LinkQueue &Q, QElemType e); 元素 e 进队列

DeQueue(LinkQueue &Q, QElemType &e); 出队列，返回队头值

QueueEmpty(LinkQueue Q); 判队列空否

2、（20 分）已知一棵二叉树以二叉链表的形式存储，其结点结构说明如下：

```
struct node {          int data;    // 结点的数据域
                        struct node *left;    // 结点的左子结点的地址
                        struct node *right;   // 结点的右子结点的地址
};
```

① 写出求以 T 为根的二叉树或子树的结点个数的递归函数；

② 写出求以 T 为根的二叉树或子树的高度的递归函数；

③ 编写一个非递归算法，将以 T 为根的二叉树中所有叶结点自左至右链接成一个单链表，算法返回该链表头结点的地址。

（本套题结束，共四大题，满分 150 分）