

## 一、填空题（每空1分，共10分）

1. 向一个长度为 $n$ 的顺序表的第 $i$ 个元素 ( $1 \leq i \leq n+1$ ) 之前插入一个元素，需向后移动元素的平均数目为  $n/2$  个(设在各个位置插入的概率相同)。
2. 设 $p$ 指向一个带有头结点的循环单链表 $head$ 中的某个结点，则 $p$ 指向表尾的条件是  $p->next = head$ 。
3. 已知如图1所示的带权无向图，从顶点 $V_3$ 出发用Prim算法求该图的最小生成树时产生的第3条边为 ( $V_4, V_6$ )。

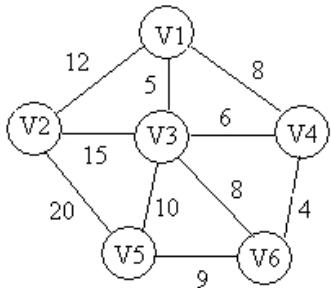


图 1

4. 设 $s='TEACHER'$ ，其长度为4的子串共有 4 个。
5. 一个 $6 \times 8$  的二维数组 $A$ ，假设其每个元素需占据4个存储单元，则存储其全部元素需要的总存储单元数为 192。
6. 假设在有序线性表 $A[1..20]$ 上进行二分查找，则找到元素 $A_{12}$ 时的比较次数为 3。
7. 已知二叉树上共有101个结点，其中度为1的结点为6个，则树中叶子结点数目为 48。
8. 含 $n$ 个顶点， $e$ 条边的有向图 $G$ ， $G$ 的邻接表中含有的边结点总数为  $e$ 。
9. 执行广义表操作  $\text{GetTail}[\text{GetHead}[(a,b),(c,d)]]$ 后的结果为 ( $b$ )。
10. 分析下面算法（程序段），该算法的时间复杂度是  $O(n^2)$ 。

```
for (i=0;i<n;i++)  
    for (j=0; j<i; j++)  
        A[i][j]=0;
```

## 二、选择题（每小题1分，共10分）

1. 在数据结构中，从逻辑上可以把数据结构分成（ C ）  
A. 动态结构和静态结构                      B. 紧凑结构和非紧凑结构

C. 线性结构和非线性结构

D. 内部结构和外部结构

2. 设单链表中指针p指向结点<sup>3</sup>，若要删除<sup>3</sup>之后的结点（若存在），则需修改指针的操作为( A )

- A. p->next=p->next->next;
- B. p=p->next;
- C. p=p->next->next;
- D. p->next=p;

3. 假设以I和O分别表示入队列和出队列操作，队列的初态和终态均为空，入队和出队的操作序列可表示为仅由I和O组成的序列。下列序列 ( B ) 是合法的。

- A. IIIIOIOIO
- B. IOIII0I00
- C. I00I0II0
- D. OIII0I0IO

4. 假设用块链存储结构表示串，如果每一个块的大小为4个字符（每个字符占用一个字节），一个指针占4个字节，则一个长为15的串需要多少字节的存储空间 ( C )

- A. 15
- B. 20
- C. 32
- D. 40

5. 以行序为主序存储一个10阶的上三角矩阵A[1...10,1...10]于一维数组B[0...54]中，那么A[5][8]存储在B中的下标k为( D )。

- A. 20
- B. 30
- C. 32
- D. 37

6. 一棵具有35个结点的完全二叉树的深度为 ( B )。

- A. 7
- B. 6
- C. 5
- D. 8

7. 已知一个图如图2所示，若从顶点a出发按深度优先搜索法遍历，则可能得到的一种顶点序列为 ( B )。

- A. a b c d e f
- B. a b c e f d
- C. a e b c f d
- D. a e d f c b

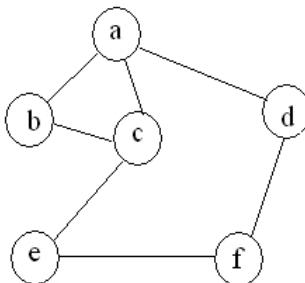


图 2

8. 对图3所示的含8个元素的二叉排序树进行查找，在等概率情况下查找成功的平均查找长度为( C )

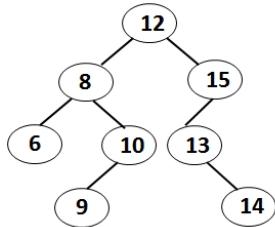


图 3

- A. 2.5      B. 3.0      C. 2.75      D. 3.4

9. 从未排序序列中依次取出元素与已排序序列（可选第一个记录构成初始序列）中的元素进行比较，将其放入已排序序列的正确位置的方法是什么？（**A**）

- A. 插入排序      B. 快速排序      C. 堆排序      D. 归并排序

10. 数据结构DS可以被形式地定义为二元组 $DS = (D, R)$ ，其中D是数据元素的有限集合，R是D上的（**D**）有限集合。

- A. 操作      B. 映象      C. 存储      D. 关系

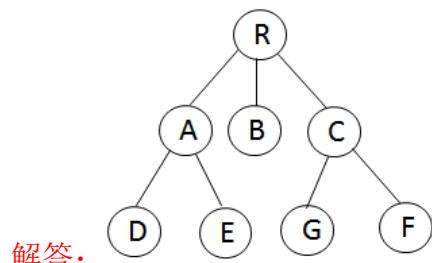
三. 简答题（每小题分值见题首，共40分）

1. (8分) 设初始序列 {12, 22, 16, 30, 2, 28, 4, 10, 20}。将原序列调整为最小堆，请写出调整过程中经过每一次筛选后得到的序列。

12 22 16 10 2 28 4 30 20  
 12 22 4 10 2 28 16 30 20  
 12 2 4 10 22 28 16 30 20  
 2 10 4 12 22 28 16 30 20

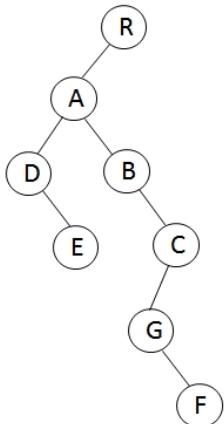
2. (12分) 回答下面三个问题：

(1) 设某棵树的先序遍历序列为RADEBCGF，后序序列为DEABGFCR，画出对应的树；**4分**



(2) 将这棵树转换成二叉树，并画出来。**4分**

解答：



(3) 由4个权值{3, 9, 5, 7}构造一棵哈夫曼树，并画出来。4分

3. (10分) 设加权有向图的如图2所示：

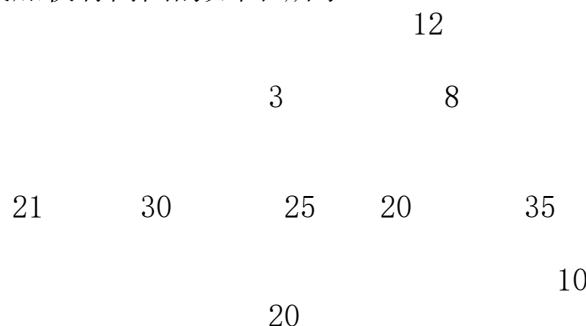


图 2  
(1) 写出该图的邻接矩阵 (顶点按A, B, C, D, E的顺序)；3分

$$A = \begin{bmatrix} \infty & 1 & 2 & 3 & 3 & 0 \\ 0 & \infty & \infty & 1 & 0 & \infty \\ 0 & 8 & 25 & \infty & & \\ 0 & 0 & 0 & 3 & 5 & 2 & 0 \\ 2 & & & \infty & 2 & \infty & 0 & \infty \end{bmatrix}$$

(2) 利用Dijkstra算法求出A到其余各点的最短路径，填表1，7分

表1

终点	i=1	i=2	i=3	i=4
B	12 (A, B)	11 (A, C, B)		
C	3 (A, C)			
D	$\infty$	$\infty$	21 (A, C, B, D)	

E	30 (A, E)	28 (A, C, E)	28 (A, C, E)	28 (A, C, E)
Vj	C	B	D	E
S	{A, C}	{A, C, B}	{A, C, B, D}	{A, C, B, D, E}

4. (10分) 设关键字序列为 {19, 14, 22, 1, 66, 21, 83, 27, 56, 15, 11, 50, 72} , 哈希地址空间为0~16, 哈希函数为  $H(key) = key \% 13$ 。

(1) 用线性探测再散列解决冲突, 构造哈希表 (填表2); 7分

(2) 求等概率下查找成功的平均查找长度ASL。3分

表2

哈希地址	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
关键字值		14	1	66	27	83	19	56	21	22	15	11	50	72			
比较次数		1	2	3	4	1	1	4	1	1	9	1	2	7			

ASL<sub>success</sub>=~~1+6+2+3+4+2+7+2+8+5~~

#### 四. 算法分析题 (每小题分值见题首, 共20分)

1. (6分) 试分析链表L的结构并简述以下算法的功能。

```
LinkList A(LinkList L)
{
    LinkList p=L;
    if(L->next==L) return NULL;
    else
    {
        while(p->next !=L) p=p->next;
        return p;
    }
}
```

答案:

1. L为带头结点的单向循环链表。(3分)

2. 该算法的功能是: 如果表不空, 则查找表尾结点并返回指向它的指针; 否则返回空指针。(3分)

2. (6分) 试分析链表L的结构并简述以下算法的功能。

```
void R(LinkList &L, ElemtType x, ElemtType y)
{
    LNode *p, *r;
    p=L; q=L->next;
    while (q!=null && q->data!=x) {
        p=q;
        q=q->next;
```

```

    }
    if(q==NULL)
        printf( “x 不存在！ ” );
    else {
        r=(LinkList)malloc(sizeof(LNode));
        r->data=y;
        r->next=q;
        p->next=r;
    }
}

```

**答案：**

1. L为带头结点的单向链表。 (3分)
2. 该算法的功能是：如果表L中存在值为x的数据，则在其之前插入值为y的数据。 (3分)

3. (4分) 试简述下面算法的功能。

```

void AI (int n )
{
    stack S;    int data;
    InitStack(S);
    for(i=0;i<n;i++)
    {
        scanf( “%d” ,&data);
        Push(S,data);
    }
    while (!StackEmpty(S)) {
        Pop(S,data);
        printf( “%d” ,data);
    }
}

```

**参考答案：** 该算法的功能是利用栈S将输入的n个数据逆序输出。

4. (4分) 下面代码实现排序功能，请仔细阅读并回答这是哪种排序算法。

```

void sort(ElemType  x[], int n)
{
    int i, j, small;
    ElemtType temp;
    for(i=0;i<n-1;i++)
    {
        small=i;
        for(j=i+1;j<n;j++)
            if(x[j]<x[small])
                small=j;

        if(small!=i)
        {
            temp=x[i];
            x[i]=x[small];
            x[small]=temp;
        }
    }
}

```

```

        x[i]=x[small];
        x[small]=temp;
    }
}

```

**参考答案：这是简单选择排序算法。**

五. 算法设计题（第1小题12分，第2小题8分，共20分）

1. 假设用**不带头结点的循环链表**表示队列Q，并且只设一个指针Q指向队尾元素（注意不设头指针）。

(1) 请写出队列为空的条件；**2分**

**Q==NULL**

(2) 写出入队列操作；**5分**

(3) 写出出队列操作，并返回出队的元素。**5分**

```

typedef struct node{
    ElemType data;
    struct node *next;
}Node, *Que;

```

```

Status EnQue (Que &Q, ElemType e) {
    //将元素e入队Q
    s=(struct node *)malloc(sizeof(Node));
    s->data=e;
    if(Q==NULL) {Q=s; Q->next=Q;}
    else
        {s->next=Q->next; Q->next=s; Q=s;}
    return OK;
}

```

```

Status DeQue(Que &Q, ElemType &e) {
    // 出队列算法，由参数e返回出队元素
    if(Q==NULL) return ERROR;
    q=Q->next;
    if(Q==q)
        Q=NULL;

```

```
    else
        Q->next=q->next;
    free(q);
    return OK;
}
```

2. 编写递归算法，将二叉树中所有结点的左右子树相互交换。

```
typedef struct node{
    ElemType data;
    struct node *lchild, *rchild;
}Node, *BiTree;
```

```
void swap(BiTree &T){
    if(T)
    {
        swap(T->lchild);
        swap(T->rchild);
        p=T->lchild;
        T->lchild=T->rchild;
        T->rchild=p;
    }
}
```