

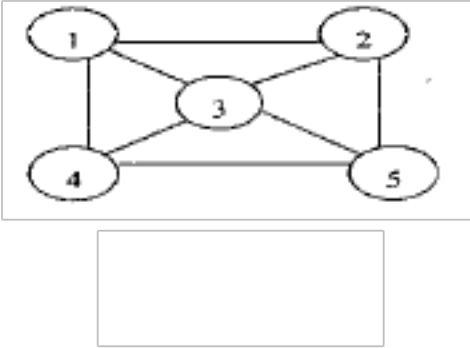
数据构造试卷〔一〕

三、计算题〔每题 6 分，共 24 分〕

1. 在如下数组 A 中链接存储了一个线性表，表头指针为 A [0].next，试写出该线性表。

A	0	1	2	3	4	5	6	7
data		60	50	78	90	34		40
next	3	5	7	2	0	4		1

2. 请画出以下列图的邻接矩阵和邻接表。



3. 一个图的顶点集 V 和边集 E 分别为：V={1, 2, 3, 4, 5, 6, 7}；

E={ (1, 2) 3, (1, 3) 5, (1, 4) 8, (2, 5) 10, (2, 3) 6, (3, 4) 15,

(3, 5) 12, (3, 6) 9, (4, 6) 4, (4, 7) 20, (5, 6) 18, (6, 7) 25}；

用克鲁斯卡尔算法得到最小生成树，试写出在最小生成树中依次得到的各条边。

4. 画出向小根堆中参加数据 4, 2, 5, 8, 3 时，每参加一个数据后堆的变化。

四、阅读算法〔每题 7 分，共 14 分〕

```
1. LinkList mynote(LinkList L)
    { //L 是不带头结点的单链表的头指针
        if(L&&L->next){
            q=L; L=L->next; p=L;
S1:         while(p->next) p=p->next;
S2:         p->next=q; q->next=NULL;
        }
        return L;
    }
```

请答复以下问题：

〔1〕说明语句 S1 的功能；

〔2〕说明语句组 S2 的功能；

〔3〕设链表表示的线性表为 $\{a_1, a_2, \dots, a_n\}$ ，写出算法执行后的返回值所表示的线性表。

```
2. void ABC(BTNode * BT)
    {
        if BT {
            ABC (BT->left);
            ABC (BT->right);
            cout<<BT->data<<' ';
        }
    }
```

该算法的功能是：

五、算法填空〔共 8 分〕

二叉搜索树的查找—递归算法：

```
bool Find(BTreeNode* BST, ElemType& item)
```

```

{
    if (BST==NULL)
        return false; //查找失败
    else {
        if (item==BST->data){
            item=BST->data;//查找成功
            return _____;}
        else if(item<BST->data)
            return Find(_____, item);
        else return Find(_____, item);
    }//if
}

```

六、编写算法（共 8 分）

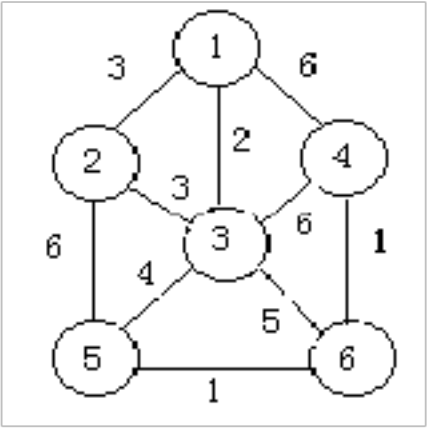
统计出单链表 **HL** 中结点的值等于给定值 **X** 的结点数。

```
int CountX(LNode* HL, ElemType x)
```

数据构造试卷〔二〕

三、应用题(36 分)

1. 设一组初始记录关键字序列为(45, 80, 48, 40, 22, 78), 那么分别给出第 4 趟简单选择排序和第 4 趟直接插入排序后的结果。
2. 设指针变量 p 指向双向链表中结点 A, 指针变量 q 指向被插入结点 B, 要求给出在结点 A 的后面插入结点 B 的操作序列(设双向链表中结点的两个指针域分别为llink 和 rlink)。
3. 设一组有序的记录关键字序列为(13, 18, 24, 35, 47, 50, 62, 83, 90), 查找方法用二分查找, 要求计算出查找关键字 62 时的比较次数并计算出查找成功时的平均查找长度。
4. 设一棵树 T 中边的集合为{(A, B), (A, C), (A, D), (B, E), (C, F), (C, G)}, 要求用孩子兄弟表示法〔二叉链表〕表示出该树的存储构造并将该树转化成对应的二叉树。
5. 设有一组无向图 G, 要求给出用普里姆算法构造最小生成树所走过的边的集合。



6. 设有一组初始记录关键字为(45, 80, 48, 40, 22, 78), 要求构造一棵二叉排序树并给出构造过程。

四、算法设计题(16 分)

1. 设有一组初始记录关键字序列 (K_1, K_2, \dots, K_n) , 要求设计一个算法可以在 $O(n)$ 的时间复杂度内将线性表划分成两部分, 其中左半部分的每个关键字均小于 K_i , 右半部分的每个关键字均大于等于 K_i 。
2. 设有两个集合 A 和集合 B, 要求设计生成集合 $C=A \cap B$ 的算法, 其中集合 A、B 和 C 用链式存储构造表示。

数据构造试卷〔三〕

二. 填空题

1. 以下算法实如今顺序散列表中查找值为 x 的关键字，请在下划线处填上正确的语句。

```
struct record{int key; int others;};
int hashsqsearch(struct record hashtable[ ],int k)
{
    int i,j;   j=i=k % p;
    while (hashtable[j].key!=k&&hashtable[j].flag!=0){j=(____) %m; if (i==j) return(-1);}
    if (_____) return(j); else return(-1);
}
```

2. 以下算法实如今二叉排序树上查找关键值 k，请在下划线处填上正确的语句。

```
typedef struct node{int key; struct node *lchild; struct node *rchild;}bitree;
bitree *bstsearch(bitree *t, int k)
{
    if (t==0 ) return(0);else while (t!=0)
        if (t->key==k)_____; else if (t->key>k) t=t->lchild; else_____;
}
```

三、计算题(每题 10 分，共 30 分)

- 1.二叉树的前序遍历序列是 AEFBGCDHIKJ，中序遍历序列是 EFAGBCHKIJD，画出此二叉树，并画出它的后序线索二叉树。
2. 待散列的线性表为〔36， 15， 40， 63， 22〕，散列用的一维地址空间为[0.6]，假定选用的散列函数是 $H(K) = K \bmod 7$ ，假设发生冲突采用线性探查法处理，试：

〔1〕计算出每一个元素的散列地址并在以下列图中填写出散列表：

、

0	1	2	3	4	5	6

- 〔2〕求出在查找每一个元素概率相等情况下的平均查找长度。
3. 序列〔10， 18， 4， 3， 6， 12， 1， 9， 18， 8〕请用快速排序写出每一趟排序的结果。

四、算法设计题(每题 15 分，共 30 分)

1. 设计在单链表中删除值一样的多余结点的算法。
2. 设计一个求结点 x 在二叉树中的双亲结点算法。

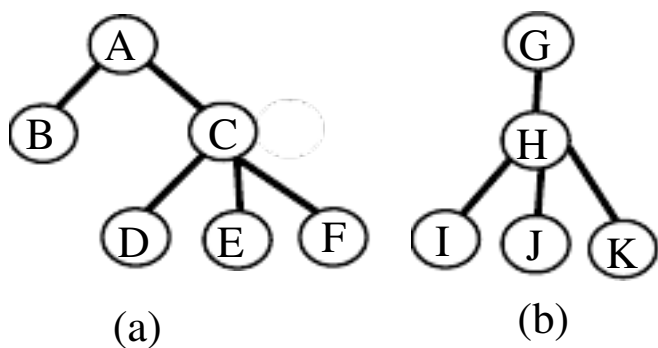
数据构造试卷〔四〕

1. 设一组初始记录关键字序列为(20, 18, 22, 16, 30, 19), 那么以 20 为中轴的一趟快速排序结果为_____。
2. 设一组初始记录关键字序列为(20, 18, 22, 16, 30, 19), 那么根据这些初始关键字序列建成的初始堆为_____。
3. 设某无向图 G 中有 n 个顶点, 用邻接矩阵 A 作为该图的存储构造, 那么顶点 i 和顶点 j 互为邻接点的条件是_____。
4. 设无向图对应的邻接矩阵为 A , 那么 A 中第 i 上非 0 元素的个数_____第 i 列上非 0 元素的个数 (填等于, 大于或小于)。
5. 设前序遍历某二叉树的序列为 ABCD, 中序遍历该二叉树的序列为 BADC, 那么后序遍历该二叉树的序列为_____。
6. 设散列函数 $H(k)=k \bmod p$, 解决冲突的方法为链地址法。要求在以下算法划线处填上正确的语句完成在散列表 hashtable 中查找关键字值等于 k 的结点, 成功时返回指向关键字的指针, 不成功时返回标志 0。

```
typedef struct node {int key; struct node *next;} lklist;
void createlkhash(lklist *hashtable[ ])
{
    int i,k; lklist *s;
    for(i=0;i<m;i++)_____
    for(i=0;i<n;i++)
    {
        s=(lklist *)malloc(sizeof(lklist)); s->key=a[i];
        k=a[i] % p; s->next=hashtable[k];_____
    }
}
```

三、计算题(每题 10 分, 共 30 分)

1. 画出广义表 $LS=((), (e), (a, (b, c, d)))$ 的头尾链表存储构造。
2. 以下列图所示的森林:
 - (1) 求树〔a〕的先根序列和后根序列;
 - (2) 求森林先序序列和中序序列;
 - (3) 将此森林转换为相应的二叉树;



3. 设散列表的地址范围是 $[0..9]$, 散列函数为 $H(key) = (key \times 2 + 2) \bmod 9$, 并采用链表处理冲突, 请画出元素 7、4、5、3、6、2、8、9 依次插入散列表的存储构造。

四、算法设计题(每题 10 分, 共 30 分)

1. 设单链表中有仅三类字符的数据元素(大写字母、数字和其它字符), 要求利用原单链表中结点空间设计出三个单链表的算法, 使每个单链表只包含同类字符。
2. 设计在链式存储构造上交换二叉树中所有结点左右子树的算法。
3. 在链式存储构造上建立一棵二叉排序树。

数据构造试卷〔五〕

1. 下面程序段的功能是实现冒泡排序算法，请在下划线处填上正确的语句。

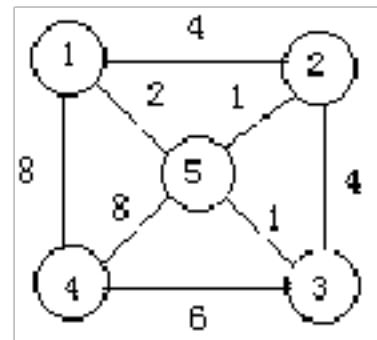
```
void bubble(int r[n])
{
    for(i=1;i<=n-1;i++)
    {
        for(exchange=0,j=0;j<_____;j++)
            if (r[j]>r[j+1]){temp=r[j+1];_____;r[j]=temp;exchange=1;}
        if (exchange==0) return;
    }
}
```

2. 下面程序段的功能是实现二分查找算法，请在下划线处填上正确的语句。

```
struct record{int key; int others;};
int bisearch(struct record r[ ], int k)
{
    int low=0,mid,high=n-1;
    while(low<=high)
    {
        _____;
        if(r[mid].key==k) return(mid+1); else if(_____) high=mid-1;else low=mid+1;
    }
    return(0);
}
```

三、应用题(32 分)

1. 设某棵二叉树的中序遍历序列为 DBEAC，前序遍历序列为 ABDEC，要求给出该二叉树的后的后序遍历序列。
2. 设无向图 G (如右图所示)，给出该图的最小生成树上边的集合并计算最小生成树各边上的权值之和。
3. 设一组初始记录关键字序列为 (15, 17, 18, 22, 35, 51, 60)，要求计算出成功查找时的平均查找长度。
4. 设散列表的长度为 8，散列函数 $H(k)=k \bmod 7$ ，初始记录关键字序列为(25, 31, 8, 27, 13, 68)，要求分别计算出用线性探测法和链地址法作为解决冲突方法的平均查找长度。



四、算法设计题(28 分)

1. 设计判断两个二叉树是否一样的算法。
2. 设计两个有序单链表的合并排序算法。

数据构造试卷〔六〕

四、算法设计题(20 分)

1. 设计在顺序有序表中实现二分查找的算法。
2. 设计判断二叉树是否为二叉排序树的算法。
3. 在链式存储构造上设计直接插入排序算法

数据构造试卷〔七〕

三、填空题(30 分)

1. 下面程序段的功能是实现一趟快速排序，请在下划线处填上正确的语句。

```
struct record {int key;datatype others;};  
void quickpass(struct record r[], int s, int t, int &i)  
{  
    int j=t; struct record x=r[s]; i=s;  
    while(i<j)  
    {  
        while (i<j && r[j].key>x.key) j=j-1;  if (i<j) {r[i]=r[j];i=i+1;}  
        while (_____) i=i+1;  if (i<j) {r[j]=r[i];j=j-1;}  
    }  
    _____;  
}
```

四、算法设计题(20 分)

1. 设计在链式构造上实现简单项选择择排序算法。
2. 设计在顺序存储构造上实现求子串算法。
3. 设计求结点在二叉排序树中层次的算法。

数据构造试卷〔八〕

三、填空题(30 分)

1. 设一组初始记录关键字序列为(49, 38, 65, 97, 76, 13, 27, 50), 那么以 $d=4$ 为增量的一趟希尔排序完毕后的结果为_____。
2. 下面程序段的功能是实如今二叉排序树中插入一个新结点, 请在下划线处填上正确的内容。

```
typedef struct node{int data;struct node *lchild;struct node *rchild;}bitree;  
void bstinsert(bitree *&t,int k)  
{  
    if (t==0) {_____;t->data=k;t->lchild=t->rchild=0;}  
    else if (t->data>k) bstinsert(t->lchild,k);else_____;  
}
```
3. 设指针变量 p 指向单链表中结点 A , 指针变量 s 指向被插入的结点 X , 那么在结点 A 的后面插入结点 X 需要执行的语句序列: $s->next=p->next$; _____;。
4. 设指针变量 $head$ 指向双向链表中的头结点, 指针变量 p 指向双向链表中的第一个结点, 那么指针变量 p 和指针变量 $head$ 之间的关系是 $p=$ _____和 $head=$ _____ (设结点中的两个指针域分别为 $llink$ 和 $rlink$)。
5. 设某棵二叉树的中序遍历序列为 $ABCD$, 后序遍历序列为 $BADC$, 那么其前序遍历序列为_____。
6. 完全二叉树中第 5 层上最少有_____个结点, 最多有_____个结点。
7. 设有向图中不存在有向边 $\langle V_i, V_j \rangle$, 那么其对应的邻接矩阵 A 中的数组元素 $A[i][j]$ 的值等于_____。
8. 设一组初始记录关键字序列为(49, 38, 65, 97, 76, 13, 27, 50), 那么第 4 趟直接选择排序完毕后的结果为_____。
9. 设连通图 G 中有 n 个顶点 e 条边, 那么对应的最小生成树上有_____条边。
10. 设有一组初始记录关键字序列为(50, 16, 23, 68, 94, 70, 73), 那么将它们调整成初始堆只需把 16 与_____互相交换即可。

四、算法设计题(20 分)

1. 设计一个在链式存储构造上统计二叉树中结点个数的算法。
2. 设计一个算法将无向图的邻接矩阵转为对应邻接表的算法。

数据构造试卷〔九〕

五、算法设计题(20 分)

1. 设计计算二叉树中所有结点值之和的算法。
2. 设计将所有奇数移到所有偶数之前的算法。
3. 设计判断单链表中元素是否是递增的算法。

数据构造试卷〔十〕

二、填空题(48 分, 其中最后两小题各 6 分)

1. 设指针变量 p 指向单链表中结点 A, 那么删除结点 A 的语句序列为:
q=p->next; p->data=q->data; p->next=_____ ; free(q);
2. 数据构造从逻辑上划分为三种根本类型: _____、_____和_____。
3. 设无向图 G 中有 n 个顶点 e 条边, 那么用邻接矩阵作为图的存储构造进展深度优先或广度优先遍历时的时间复杂度为_____ ; 用邻接表作为图的存储构造进展深度优先或广度优先遍历的时间复杂度为_____。
4. 设散列表的长度为 8, 散列函数 $H(k)=k \% 7$, 用线性探测法解决冲突, 那么根据一组初始关键字序列 (8, 15, 16, 22, 30, 32) 构造出的散列表的平均查找长度是_____。
5. 设一组初始关键字序列为 (38, 65, 97, 76, 13, 27, 10), 那么第 3 趟冒泡排序完毕后的结果为_____。
6. 设一组初始关键字序列为 (38, 65, 97, 76, 13, 27, 10), 那么第 3 趟简单选择排序后的结果为_____。
7. 设有向图 G 中的有向边的集合 $E=\{ \langle 1, 2 \rangle, \langle 2, 3 \rangle, \langle 1, 4 \rangle, \langle 4, 5 \rangle, \langle 5, 3 \rangle, \langle 4, 6 \rangle, \langle 6, 5 \rangle \}$, 那么该图的一个拓扑序列为_____。
8. 下面程序段的功能是建立二叉树的算法, 请在下划线处填上正确的内容。

```
typedef struct node {int data; struct node *lchild; _____; } bitree;
void createbitree(bitree *&bt)
{
    scanf("%c", &ch);
    if(ch=='#') _____; else
        { bt=(bitree*)malloc(sizeof(bitree)); bt->data=ch; _____; createbitree(bt->rchild); }
}
```
9. 下面程序段的功能是利用从尾部插入的方法建立单链表的算法, 请在下划线处填上正确的内容。

```
typedef struct node {int data; struct node *next;} lklist;
void lklistcreate(_____ *&head)
{
    for (i=1; i<=n; i++)
    {
        p=(lklist *)malloc(sizeof(lklist)); scanf("%d", &(p->data)); p->next=0;
        if(i==1) head=q=p; else { q->next=p; _____; }
    }
}
```

三、算法设计题(22 分)

1. 设计在链式存储构造上合并排序的算法。
2. 设计在二叉排序树上查找结点 X 的算法。
3. 设关键字序列 $(k_1, k_2, \dots, k_{n-1})$ 是堆, 设计算法将关键字序列 $(k_1, k_2, \dots, k_{n-1}, x)$ 调整为堆。

数据构造试卷〔一〕参考答案

三、计算题〔每题 6 分，共 24 分〕

1. 线性表为: {78, 50, 40, 60, 34, 90}

2. 邻接矩阵:
$$\begin{bmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{bmatrix}$$

邻接表如图 11 所示:

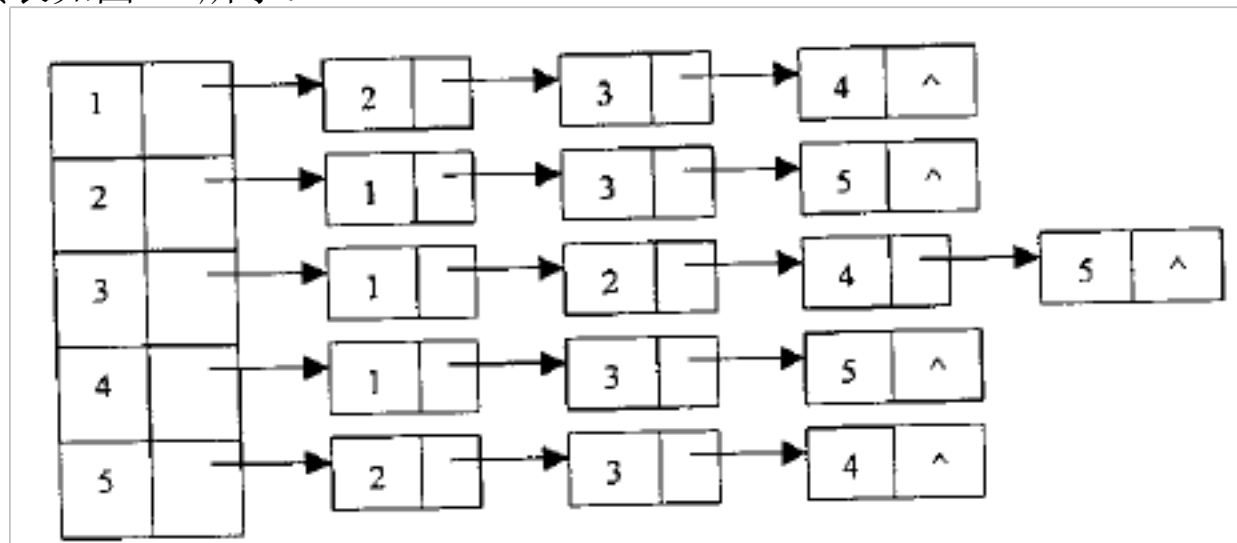
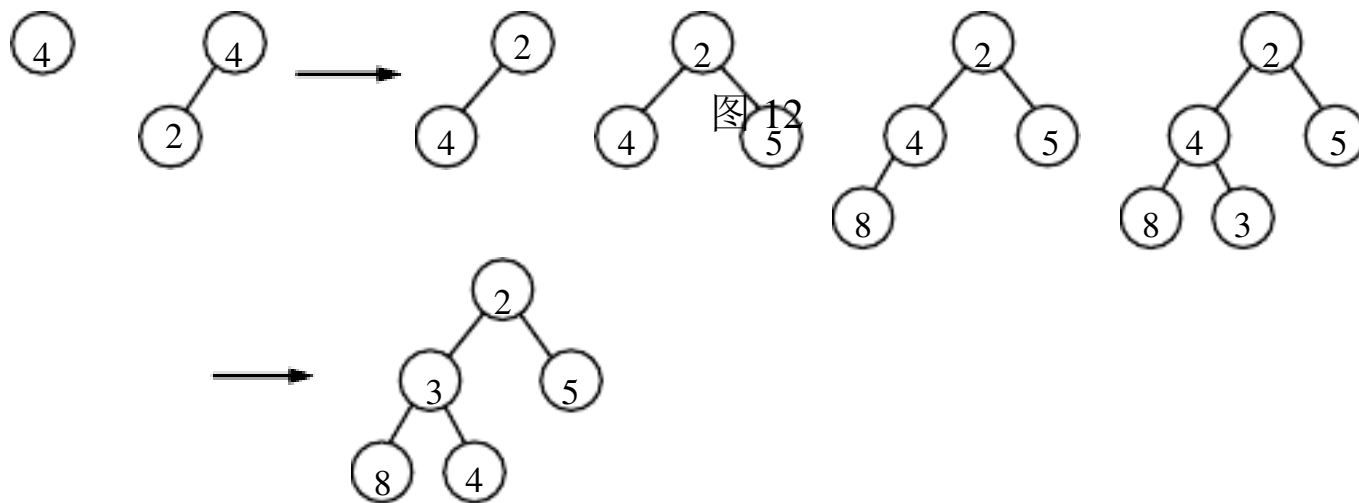


图 11

3. 用克鲁斯卡尔算法得到的最小生成树为:

(1,2)3, (4,6)4, (1,3)5, (1,4)8, (2,5)10, (4,7)20

4. 见图 12



四、读算法〔每题 7 分，共 14 分〕

1. (1) 查询链表的尾结点

(2) 将第一个结点链接到链表的尾部，作为新的尾结点

(3) 返回的线性表为 $\{a_2, a_3, \dots, a_n, a_1\}$

2. 递归地后序遍历链式存储的二叉树。

五、法填空〔每空 2 分，共 8 分〕

true BST->left BST->right

六、编写算法〔8 分〕

```
int CountX(LNode* HL, ElemType x)
{
    int i=0; LNode* p=HL; //i 为计数器
    while(p!=NULL)
    {
        if (P->data==x) i++;
        p=p->next;
    } //while, 出循环时 i 中的值即为 x 结点个数
    return i;
} //CountX
```

数据构造试卷〔二〕参考答案

三、应用题

1. (22, 40, 45, 48, 80, 78), (40, 45, 48, 80, 22, 78)
2. $q \rightarrow llink = p; q \rightarrow rlink = p \rightarrow rlink; p \rightarrow rlink \rightarrow llink = q; p \rightarrow rlink = q;$
3. $2, ASL = 91 \times 1 + 2 \times 2 + 3 \times 4 + 4 \times 2 = 25/9$
4. 树的链式存储构造略, 二叉树略
5. $E = \{(1, 3), (1, 2), (3, 5), (5, 6), (6, 4)\}$
6. 略

四、算法设计题

1. 设有一组初始记录关键字序列 (K_1, K_2, \dots, K_n) , 要求设计一个算法可以在 $O(n)$ 的时间复杂度内将线性表划分成两部分, 其中左半部分的每个关键字均小于 K_i , 右半部分的每个关键字均大于等于 K_i 。

```
void quickpass(int r[], int s, int t)
```

```
{
    int i=s, j=t, x=r[s];
    while(i<j){
        while (i<j && r[j]>x) j=j-1; if (i<j) {r[i]=r[j];i=i+1;}
        while (i<j && r[i]<x) i=i+1; if (i<j) {r[j]=r[i];j=j-1;}
    }
    r[i]=x;
}
```

2. 设有两个集合 A 和集合 B, 要求设计生成集合 $C=A \cap B$ 的算法, 其中集合 A、B 和 C 用链式存储构造表示。

```
typedef struct node {int data; struct node *next;}lklist;
```

```
void intersection(lklist *ha,lklist *hb,lklist *&hc)
```

```
{
    lklist *p,*q,*t;
    for(p=ha,hc=0;p!=0;p=p->next)
    {   for(q=hb;q!=0;q=q->next) if (q->data==p->data) break;
        if(q!=0){ t=(lklist *)malloc(sizeof(lklist)); t->data=p->data;t->next=hc; hc=t;}
    }
}
```

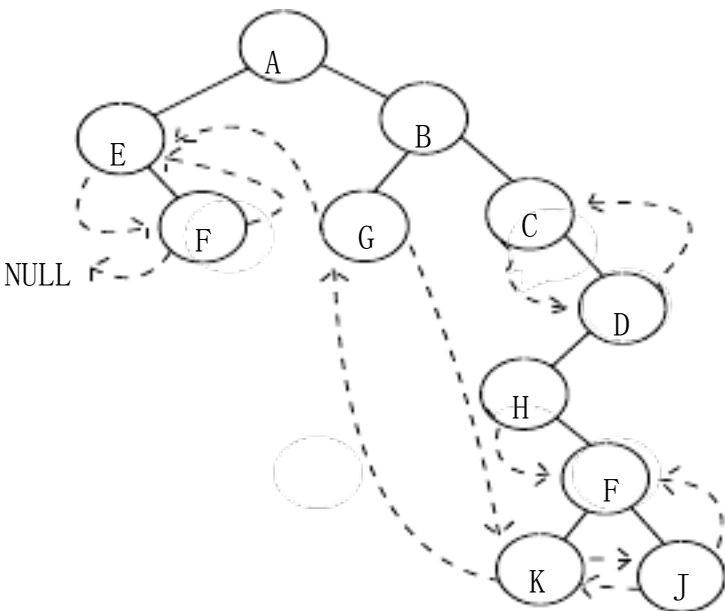
数据构造试卷〔三〕参考答案

二. 填空题

- 13.j+1, hashtable[j].key==k
14. return(t), t=t->rchild

三、计算题

1.



- 2、 $H(36)=36 \bmod 7=1$;
 $H(15)=15 \bmod 7=1$;....冲突
 $H_1(15)=(1+1) \bmod 7=2$;
 $H(40)=40 \bmod 7=5$;
 $H(63)=63 \bmod 7=0$;
 $H(22)=22 \bmod 7=1$;冲突
 $H_1(22)=(1+1) \bmod 7=2$;冲突
 $H_2(22)=(2+1) \bmod 7=3$;

(1)

0	1	2	3	4	5	6
63	36	15	22		40	

(2) $ASL=\frac{1+2+1+1+3}{5}=1.6$

- 3、 (8,9,4,3,6,1),10,(12,18,18)
(1,6,4,3),8,(9),10,12,(18,18)
1,(3,4,6),8,9,10,12,18,(18)
1,3,(4,6),8,9,10,12,18,18
1,3, 4,6,8,9,10,12,18,18

四、算法设计题

1. 设计在单链表中删除值一样的多余结点的算法。
- ```
typedef int datatype;
typedef struct node {datatype data; struct node *next;}lklist;
void delredundant(lklist *&head)
{
 lklist *p,*q,*s;
 for(p=head;p!=0;p=p->next)
 {
```

```

 for(q=p->next,s=q;q!=0;)
 if (q->data==p->data) {s->next=q->next; free(q);q=s->next;}
 else {s=q,q=q->next;}
 }
}
2. 设计一个求结点 x 在二叉树中的双亲结点算法。
typedef struct node {datatype data; struct node *lchild,*rchild;} bitree;
bitree *q[20]; int r=0,f=0,flag=0;
void preorder(bitree *bt, char x)
{
 if (bt!=0 && flag==0)
 if (bt->data==x) { flag=1; return;}
 else {r=(r+1)% 20; q[r]=bt; preorder(bt->lchild,x); preorder(bt->rchild,x); }
}
void parent(bitree *bt,char x)
{
 int i;
 preorder(bt,x);
 for(i=f+1; i<=r; i++) if (q[i]->lchild->data==x || q[i]->rchild->data==x) break;
 if (flag==0) printf("not found x\n");
 else if (i<=r) printf("%c",bt->data); else printf("not parent");}

```

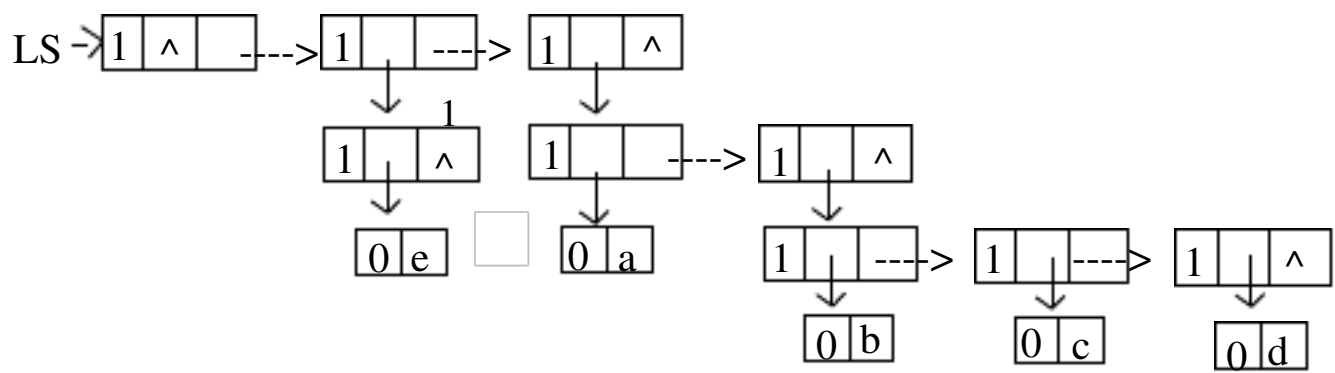
数据构造试卷〔四〕 参考答案

二、填空题

- 1. (19, 18, 16, 20, 30, 22)
- 2. (16, 18, 19, 20, 32, 22)
- 3.  $A[i][j]=1$
- 4. 等于
- 5. BDCA
- 6. hashtable[i]=0, hashtable[k]=s

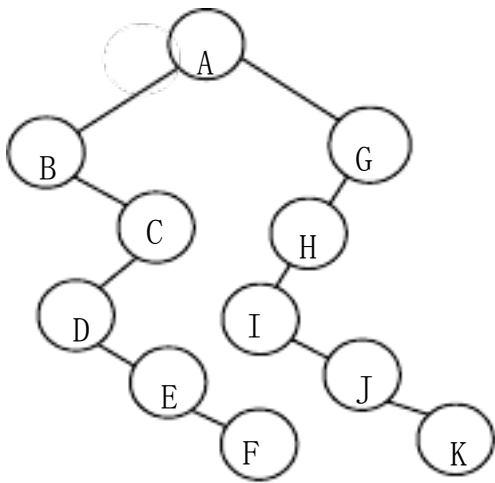
三、计算题

1.

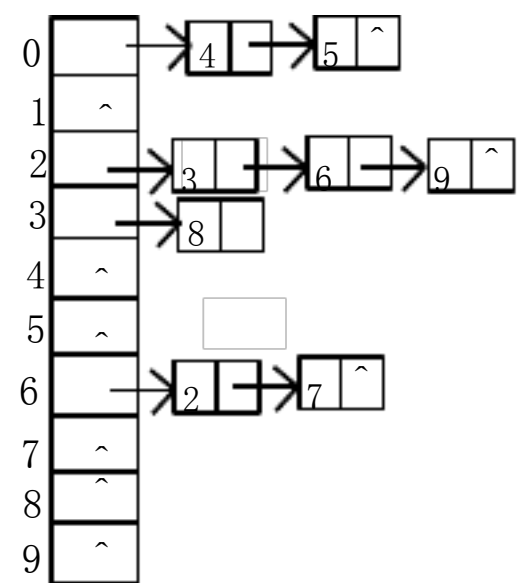


2.

(1) ABCDEF; BDEFCA; (2) ABCDEFGHIJK; BDEFCAIJKHG 林转换为相应的二叉树;



3.  $H(4)=H(5)=0, H(3)=H(6)=H(9)=2, H(8)=3, H(2)=H(7)=6$



四、算法设计题

- 1. 设单链表中有仅三类字符的数据元素(大写字母、数字和其它字符)，要求利用原单链表中结点空间设计出三个单链表的算法，使每个单链表只包含同类字符。

```
typedef char datatype;
typedef struct node {datatype data; struct node *next;}lklist;
```



```

void split(lklist *head,lklist *&ha,lklist *&hb,lklist *&hc)
{
 lklist *p; ha=0,hb=0,hc=0;
 for(p=head;p!=0;p=p->next)
 {
 head=p->next; p->next=0;
 if (p->data>='A' && p->data<='Z') {p->next=ha; ha=p;}
 else if (p->data>='0' && p->data<='9') {p->next=hb; hb=p;} else {p->next=hc; hc=p;}
 }
}

```

2. 设计在链式存储构造上交换二叉树中所有结点左右子树的算法。

```

typedef struct node {int data; struct node *lchild,*rchild;} bitree;
void swapbitree(bitree *bt)
{
 bitree *p;
 if(bt==0) return;
 swapbitree(bt->lchild); swapbitree(bt->rchild);
 p=bt->lchild; bt->lchild=bt->rchild; bt->rchild=p;
}

```

3. 在链式存储构造上建立一棵二叉排序树。

```

#define n 10
typedef struct node{int key; struct node *lchild,*rchild;}bitree;
void bstinsert(bitree *&bt,int key)
{
 if (bt==0){bt=(bitree *)malloc(sizeof(bitree)); bt->key=key;bt->lchild=bt->rchild=0;}
 else if (bt->key>key) bstinsert(bt->lchild,key); else bstinsert(bt->rchild,key);
}
void createbsttree(bitree *&bt)
{
 int i;
 for(i=1;i<=n;i++) bstinsert(bt,random(100));
}

```

## 数据构造试卷〔五〕参考答案

### 二、填空题

1.  $n-i$ ,  $r[j+1]=r[j]$
2.  $\text{mid}=(\text{low}+\text{high})/2$ ,  $r[\text{mid}].\text{key}>k$

### 三、应用题

2. DEBCA
3.  $E=\{(1,5),(5,2),(5,3),(3,4)\}, W=10$
4.  $\text{ASL}=(1*1+2*2+3*4)/7=17/7$
5.  $\text{ASL}_1=7/6$ ,  $\text{ASL}_2=4/3$

### 四、算法设计题

1. 设计判断两个二叉树是否一样的算法。

```
typedef struct node {datatype data; struct node *lchild,*rchild;} bitree;
int judgebitree(bitree *bt1,bitree *bt2)
{
 if (bt1==0 && bt2==0) return(1);
 else if (bt1==0 || bt2==0 || bt1->data!=bt2->data) return(0);
 else return(judgebitree(bt1->lchild,bt2->lchild)*judgebitree(bt1->rchild,bt2->rchild));
}
```

2. 设计两个有序单链表的合并排序算法。

```
void mergelklist(lklist *ha,lklist *hb,lklist *&hc)
{
 lklist *s=hc=0;
 while(ha!=0 && hb!=0)
 if(ha->data<hb->data){if(s==0) hc=s=ha; else {s->next=ha; s=ha;};ha=ha->next;}
 else {if(s==0) hc=s=hb; else {s->next=hb; s=hb;};hb=hb->next;}
 if(ha==0) s->next=hb; else s->next=ha;
}
```

## 数据构造试卷〔六〕 参考答案

### 四、算法设计题

1. 设计在顺序有序表中实现二分查找的算法。

```
struct record {int key; int others;};
int bisearch(struct record r[], int k)
{
 int low=0,mid,high=n-1;
 while(low<=high)
 {
 mid=(low+high)/2;
 if(r[mid].key==k) return(mid+1); else if(r[mid].key>k) high=mid-1; else low=mid+1;
 }
 return(0);
}
```

2. 设计判断二叉树是否为二叉排序树的算法。

```
int minnum=-32768,flag=1;
typedef struct node{int key; struct node *lchild,*rchild;}bitree;
void inorder(bitree *bt)
{
 if (bt!=0) {inorder(bt->lchild); if(minnum>bt->key)flag=0;
 minnum=bt->key;inorder(bt->rchild);}
}
```

3. 在链式存储构造上设计直接插入排序算法

```
void straightinsertsort(lklist *&head)
{
 lklist *s,*p,*q; int t;
 if (head==0 || head->next==0) return;
 else for(q=head,p=head->next;p!=0;p=q->next)
 {
 for(s=head;s!=q->next;s=s->next) if (s->data>p->data) break;
 if(s==q->next)q=p;
 else{q->next=p->next; p->next=s->next; s->next=p;
 t=p->data;p->data=s->data;s->data=t;}
 }
}
```

## 数据构造试卷〔七〕参考答案

### 三. 填空题

1.  $i < j \ \&\& \ r[i].key < x.key, \ r[i] = x$

### 四、算法设计题

1. 设计在链式构造上实现简单项选择排序算法。

```
void simpleselectsortlklist(lklist *&head)
{
 lklist *p,*q,*s; int min,t;
 if(head==0 ||head->next==0) return;
 for(q=head; q!=0;q=q->next)
 {
 min=q->data; s=q;
 for(p=q->next; p!=0;p=p->next) if(min>p->data){ min=p->data; s=p;}
 if(s!=q){ t=s->data; s->data=q->data; q->data=t;}
 }
}
```

2. 设计在顺序存储构造上实现求子串算法。

```
void substring(char s[], long start, long count, char t[])
{
 long i,j,length=strlen(s);
 if (start<1 || start>length) printf("The copy position is wrong");
 else if (start+count-1>length) printf("Too characters to be copied");
 else { for(i=start-1,j=0; i<start+count-1;i++,j++) t[j]=s[i]; t[j]= '\0';}
}
```

3. 设计求结点在二叉排序树中层次的算法。

```
int lev=0;
typedef struct node{int key; struct node *lchild,*rchild;}bitree;
void level(bitree *bt,int x)
{
 if (bt!=0)
 {lev++; if (bt->key==x) return; else if (bt->key>x) level(bt->lchild,x); else
level(bt->rchild,x);}
}
```

## 数据构造试卷〔八〕参考答案

### 三、填空题

1. (49, 13, 27, 50, 76, 38, 65, 97)
2. `t=(bitree *)malloc(sizeof(bitree)), bstinsert(t->rchild,k)`
3. `p->next=s`
4. `head->rlink, p->llink`
5. CABD
6. 1, 16
7. 0
8. (13, 27, 38, 50, 76, 49, 65, 97)
9. `n-1`
10. 50

### 四、算法设计题

1. 设计一个在链式存储构造上统计二叉树中结点个数的算法。

```
void countnode(bitree *bt,int &count)
{
 if(bt!=0)
 {count++; countnode(bt->lchild,count); countnode(bt->rchild,count);}
}
```

2. 设计一个算法将无向图的邻接矩阵转为对应邻接表的算法。

```
typedef struct {int vertex[m]; int edge[m][m];}gadjmatrix;
typedef struct node1 {int info;int adjvertex; struct node1 *nextarc;}glinklistnode;
typedef struct node2 {int vertexinfo;glinklistnode *firstarc;}glinkheadnode;
void adjmatrixtoadjlist(gadjmatrix g1[],glinkheadnode g2[])
{
 int i,j; glinklistnode *p;
 for(i=0;i<=n-1;i++) g2[i].firstarc=0;
 for(i=0;i<=n-1;i++) for(j=0;j<=n-1;j++)
 if (g1.edge[i][j]==1)
 {
 p=(glinklistnode *)malloc(sizeof(glinklistnode));p->adjvertex=j;
 p->nextarc=g[i].firstarc; g[i].firstarc=p;
 p=(glinklistnode *)malloc(sizeof(glinklistnode));p->adjvertex=i;
 p->nextarc=g[j].firstarc; g[j].firstarc=p;}
}
```

## 数据构造试卷〔九〕参考答案

### 四、算法设计题

1. 设计计算二叉树中所有结点值之和的算法。

```
void sum(bitree *bt,int &s)
{
 if(bt!=0) {s=s+bt->data; sum(bt->lchild,s); sum(bt->rchild,s);}
}
```

2. 设计将所有奇数移到所有偶数之前的算法。

```
void quickpass(int r[], int s, int t)
{
 int i=s,j=t,x=r[s];
 while(i<j)
 {
 while (i<j && r[j]%2==0) j=j-1; if (i<j) {r[i]=r[j];i=i+1;}
 while (i<j && r[i]%2==1) i=i+1; if (i<j) {r[j]=r[i];j=j-1;}
 }
 r[i]=x;
}
```

3. 设计判断单链表中元素是否是递增的算法。

```
int isriselk(lklist *head)
{
 if(head==0||head->next==0) return(1);else
 for(q=head,p=head->next; p!=0; q=p,p=p->next)if(q->data>p->data) return(0);
 return(1);}

```

## 数据构造试卷〔十〕 参考答案

### 二、填空题

1. q->next
2. 线性构造, 树型构造, 图型构造
3.  $O(n^2)$ ,  $O(n+e)$
4.  $8/3$
5. (38, 13, 27, 10, 65, 76, 97)
6. (10, 13, 27, 76, 65, 97, 38)
7. 124653
8. struct node \*rchild, bt=0, createbitree(bt->lchild)
9. lklist, q=p

### 三、算法设计题

1. 设计在链式存储构造上合并排序的算法。

```
void mergelklist(lklist *ha, lklist *hb, lklist *&hc)
{
 lklist *s=hc=0;
 while(ha!=0 && hb!=0)
 if(ha->data<hb->data){if(s==0) hc=s=ha; else {s->next=ha; s=ha;};ha=ha->next;}
 else {if(s==0) hc=s=hb; else {s->next=hb; s=hb;};hb=hb->next;}
 if(ha==0) s->next=hb; else s->next=ha;
}
```

2. 设计在二叉排序树上查找结点 X 的算法。

```
bitree *bstsearch1(bitree *t, int key)
{
 bitree *p=t;
 while(p!=0) if (p->key==key) return(p);else if (p->key>key)p=p->lchild; else p=p->rchild;
 return(0);
}
```

3. 设关键字序列  $(k_1, k_2, \dots, k_{n-1})$  是堆, 设计算法将关键字序列  $(k_1, k_2, \dots, k_{n-1}, x)$  调整为堆。

```
void adjustheap(int r[],int n)
{
 int j=n,i=j/2,temp=r[j-1];
 while (i>=1) if (temp>=r[i-1])break; else{r[j-1]=r[i-1]; j=i; i=i/2;}
 r[j-1]=temp;
}
```