

数据结构试卷 (一)	1	数据结构试卷 (一) 参考答案	27
数据结构试卷 (二)	5	数据结构试卷 (二) 参考答案	28
数据结构试卷 (三)	8	数据结构试卷 (三) 参考答案	30
数据结构试卷 (四)	11	数据结构试卷 (四) 参考答案	32
数据结构试卷 (五)	14	数据结构试卷 (五) 参考答案	34
数据结构试卷 (六)	16	数据结构试卷 (六) 参考答案	35
数据结构试卷 (七)	19	数据结构试卷 (七) 参考答案	37
数据结构试卷 (八)	21	数据结构试卷 (八) 参考答案	38
数据结构试卷 (九)	23	数据结构试卷 (九) 参考答案	39
数据结构试卷 (十)	25	数据结构试卷 (十) 参考答案	40

数据结构试卷 (一)

一、单选题 (每题 2 分, 共 20 分)

栈和队列的共同特点是(**A**)。

A. 只允许在端点处插入和删除元素

B. 都是先进后出

C. 都是先进先出

D. 没有共同点

1. 用链接方式存储的队列, 在进行插入运算时(**D**)。

A. 仅修改头指针

B. 头、尾指针都要修改

C. 仅修改尾指针

D. 头、尾指针可能都要修改

2. 以下数据结构中哪一个是非线性结构? (**D**)

A. 队列

B. 栈

C. 线性表

D. 二叉树

3. 设有一个二维数组 $A[m][n]$, 假设 $A[0][0]$ 存放位置在 $644_{(10)}$, $A[2][2]$ 存放位置在 $676_{(10)}$, 每个元素占一个空间, 问 $A[3][3]_{(10)}$ 存放在什么位置? 脚注 $_{(10)}$ 表示用 10 进制表示 (**C**)。

A. 688

B. 678

C. 692

D. 696

4. 树最适合用来表示(**C**)。

A. 有序数据元素

B. 无序数据元素

C. 元素之间具有分支层次关系的数据

D. 元素之间无联系的数据

5. 二叉树的第 k 层的结点数最多为(**D**)。

A. $2^k - 1$

B. $2K + 1$

C. $2K - 1$

D. 2^{k-1}

6. 若有 18 个元素的有序表存放在一维数组 $A[19]$ 中, 第一个元素放 $A[1]$ 中, 现进行二分查找, 则查找 $A[3]$ 的比较序列的下标依次为(**D**)

A. 1, 2, 3

B. 9, 5, 2, 3

C. 9, 5, 3

D. 9, 4, 2, 3

解析: 计算的时候 $low=0$; $high=18$;

7. 对 n 个记录的文件进行快速排序, 所需要的辅助存储空间大致为 (**C**)

A. $O(1)$

B. $O(n)$

C. $O(\log_2 n)$

D. $O(n^2)$

解析: 不断递归调用

8. 对于线性表 (7, 34, 55, 25, 64, 46, 20, 10) 进行散列存储时, 若选用 $H(K) = K \% 9$ 作为散列函数, 则散列地址为 1 的元素有 (**D**) 个,

A. 1

B. 2

C. 3

D. 4

9. 设有 6 个结点的无向图, 该图至少应有(**A**)条边才能确保是一个连通图。

A. 5

B. 6

C. 7

D. 8

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。

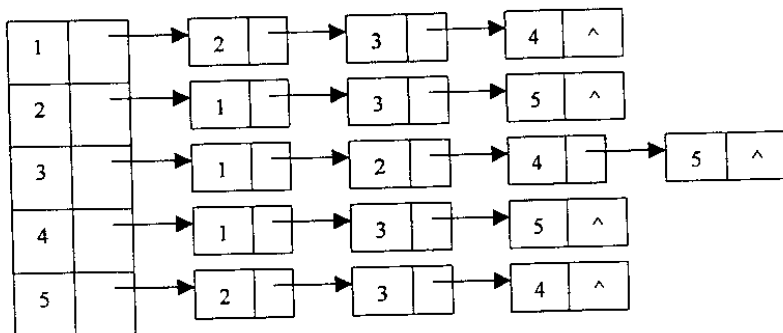
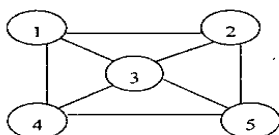
A	0	1	2	3	4	5	6	7
data		60	50	78	90	34		40

next	3	5	7	2	0	4		1
------	---	---	---	---	---	---	--	---

$$\begin{bmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{bmatrix}$$

线性表为: (78, 50, 40, 60, 34, 90)

2. 请画出下图的邻接矩阵和邻接表。



3. 已知一个图的顶点集 V 和边集 E 分别为: $V=\{1, 2, 3, 4, 5, 6, 7\}$;

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

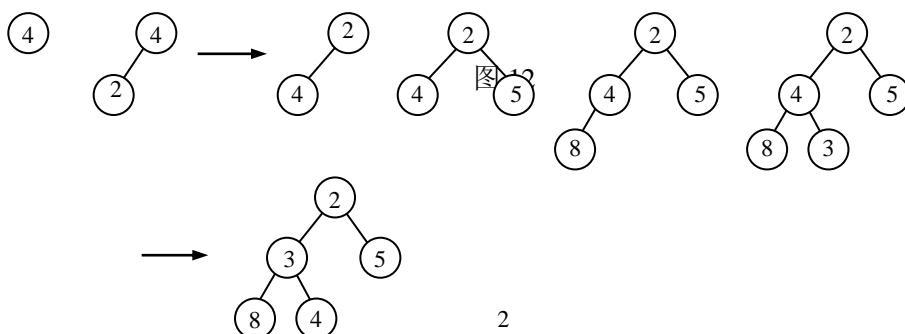
$(3, 5)12, (3, 6)9, (4, 6)4, (4, 7)20, (5, 6)18, (6, 7)25\}$;

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。

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. 画出向小根堆中加入数据 4, 2, 5, 8, 3 时, 每加入一个数据后堆的变化。见图 12



4.

图 11

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;
    }
```

请回答下列问题：

(1) 说明语句 S1 的功能：

查询链表的尾结点

(2) 说明语句组 S2 的功能：

将第一个结点链接到链表的尾部，作为新的尾结点

(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。

返回的线性表为 $(a_2, a_3, \dots, a_n, a_1)$

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

该算法的功能是：

递归地后序遍历链式存储的二叉树

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 true;
        }
        else if (item<BST->data)
            return Find(BST->left, item);
        else return Find(BST->right, 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)
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
```

数据结构试卷（二）

一、选择题(24 分)

- 下面关于线性表的叙述错误的是（ ）。
(A) 线性表采用顺序存储必须占用一片连续的存储空间
(B) 线性表采用链式存储不必占用一片连续的存储空间
(C) 线性表采用链式存储便于插入和删除操作的实现
(D) **线性表采用顺序存储便于插入和删除操作的实现**
 - 设哈夫曼树中的叶子结点总数为 m ，若用二叉链表作为存储结构，则该哈夫曼树中总共有（ ）个空指针域。
(A) $2m-1$ (B) **$2m$** (C) $2m+1$ (D) $4m$
 - 设顺序循环队列 $Q[0:M-1]$ 的头指针和尾指针分别为 F 和 R ，头指针 F 总是指向队头元素的前一位置，尾指针 R 总是指向队尾元素的当前位置，则该循环队列中的元素个数为（ ）。
(A) $R-F$ (B) $F-R$ (C) **$(R-F+M) \% M$** (D) $(F-R+M) \% M$
 - 设某棵二叉树的中序遍历序列为 $ABCD$ ，前序遍历序列为 $CABD$ ，则后序遍历该二叉树得到序列为（ ）。
(A) **$BADC$** (B) $BCDA$ (C) $CDAB$ (D) $CBDA$
 - 设某完全无向图中有 n 个顶点，则该完全无向图中有（ ）条边。
(A) **$n(n-1)/2$** (B) $n(n-1)$ (C) n^2 (D) n^2-1
 - 设某棵二叉树中有 2000 个结点，则该二叉树的最小高度为（ ）。
(A) 9 (B) 10 (C) **11** (D) 12
- 解析：区别：树的高度的定义，只有一个节点的树的高度为 0（则该题选择 B）
- 设某有向图中有 n 个顶点，则该有向图对应的邻接表中有（ ）个表头结点。
(A) $n-1$ (B) **n** (C) $n+1$ (D) $2n-1$
 - 设一组初始记录关键字序列 $(5, 2, 6, 3, 8)$ ，以第一个记录关键字 5 为基准进行一趟快速排序的结果为（ ）。
(A) $2, 3, 5, 8, 6$ (B) $3, 2, 5, 8, 6$
(C) **$3, 2, 5, 6, 8$** (D) $2, 3, 6, 5, 8$

解析：pivotpos==i 时则不交换，i++

三、应用题(36 分)

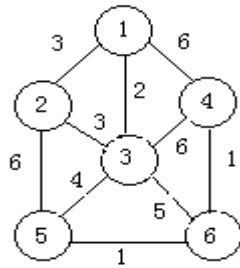
- 设一组初始记录关键字序列为 $(45, 80, 48, 40, 22, 78)$ ，则分别给出第 4 趟简单选择排序和第 4 趟直接插入排序后的结果。
 $(22, 40, 45, 48, 80, 78)$, $(22, 40, 45, 48, 80, 78)$
- 设指针变量 p 指向双向链表中结点 A，指针变量 q 指向被插入结点 B，要求给出在结点 A 的后面插入结点 B 的操作序列（设双向链表中结点的两个指针域分别为 $llink$ 和 $rlink$ ）。
 $q->llink=p; q->rlink=p->rlink; p->rlink=q; p->rlink=q;$
- 设一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。

$2, ASL = (1*1 + 2*2 + 3*4 + 4*2) = 25/9$

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)表示出该树的存储结构并将该树转化成对应的二叉树。

树的链式存储结构略, 二叉树略

5. 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。



$E=\{(1, 3), (1, 2), (3, 5), (5, 6), (6, 4)\}$

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) , 要求设计一个算法能够在 $O(n)$ 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 。

设有一组初始记录关键字序列 (K_1, K_2, \dots, K_n) , 要求设计一个算法能够在 $O(n)$ 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 用链式存储结构表示。

```
typedef struct node {int data; struct node *next;}lklst;
void intersection(lklst *ha,lklst *hb,lklst *&hc)
{
    lklst *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;}
    }
}
```

数据结构试卷（三）

一、选择题(每题 1 分, 共 20 分)

1. 设某数据结构的二元组形式表示为 $A=(D, R)$, $D=\{01, 02, 03, 04, 05, 06, 07, 08, 09\}$, $R=\{r\}$, $r=\{<01, 02>, <01, 03>, <01, 04>, <02, 05>, <02, 06>, <03, 07>, <03, 08>, <03, 09>\}$, 则数据结构 A 是 ()。

(A) 线性结构 (B) 树型结构 (C) 物理结构 (D) 图型结构

2. 下面程序的时间复杂为 ()

```
for (i=1, s=0; i<=n; i++)
{
    t=1;
    for(j=1; j<=i; j++)
        t=t*j;
    s=s+t; }
```

(A) $O(n)$ (B) $O(n^2)$ (C) $O(n^3)$ (D) $O(n^4)$

3. 设指针变量 p 指向单链表中结点 A, 若删除单链表中结点 A, 则需要修改指针的操作序列为 ()。

(A) $q=p \rightarrow next; p \rightarrow data=q \rightarrow data; p \rightarrow next=q \rightarrow next; free(q);$

(B) $q=p \rightarrow next; q \rightarrow data=p \rightarrow data; p \rightarrow next=q \rightarrow next; free(q);$

(C) $q=p \rightarrow next; p \rightarrow next=q \rightarrow next; free(q);$

(D) $q=p \rightarrow next; p \rightarrow data=q \rightarrow data; free(q);$

4. 设有 n 个待排序的记录关键字, 则在堆排序中需要 () 个辅助记录单元。

(A) 1 (B) n (C) $n \log_2 n$ (D) n^2

解析: 每次进行一次交换, 只需要一个 temp 的位置即可

5. 设一组初始关键字记录关键字为 (20, 15, 14, 18, 21, 36, 40, 10), 则以 20 为基准记录的一趟快速排序结束后的结果为 ()。

(A) 10, 15, 14, 18, 20, 36, 40, 21

(B) 10, 15, 14, 18, 20, 40, 36, 21

(C) 10, 15, 14, 20, 18, 40, 36, 21

(D) 15, 10, 14, 18, 20, 36, 40, 21

注意: 10 的时候 pivotpos++

6. 设二叉排序树中有 n 个结点, 则在二叉排序树的平均平均查找长度为 ()。

(A) $O(1)$ (B) $O(\log_2 n)$ (C) (D) $O(n^2)$

7. 设无向图 G 中有 n 个顶点 e 条边, 则其对应的邻接表中的表头结点和表结点的个数分别为 ()。

(A) n, e (B) e, n (C) $2n, e$ (D) n, $2e$

8. 设某强连通图中有 n 个顶点, 则该强连通图中至少有 () 条边。

(A) $n(n-1)$ (B) $n+1$ (C) n (D) $n(n+1)$

注意: 存在 $vi \rightarrow vj$, 必然有 $vj \rightarrow vi$, 但是到达的路径长度没有限制

9. 设有 5000 个待排序的记录关键字, 如果需要用最快的方法选出其中最小的 10 个记录关键字, 则用下列 () 方法可以达到此目的。

(A) 快速排序 (B) 堆排序 (C) 归并排序 (D) 插入排序

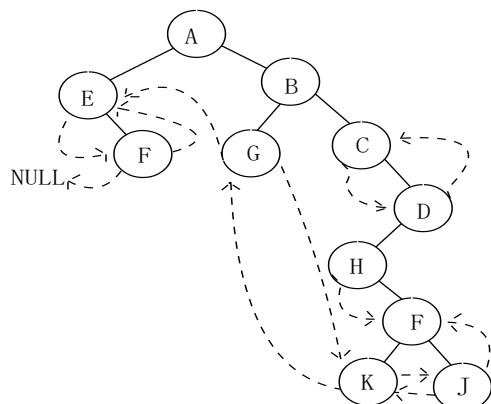
10. 下列四种排序中 () 的空间复杂度最大。

(A) 插入排序 (B) 冒泡排序 (C) 堆排序 (D) 归并排序

快速排序 $O(\log n)$

三、计算题(每题 10 分, 共 30 分)

1. 已知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. 已知待散列的线性表为 (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 $H(K) = K \bmod 7$, 若发生冲突采用线性探查法处理, 试:

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

(1) 计算出每一个元素的散列地址并在下图中填写出散列表:

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

(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。

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

3. 已知序列 (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。

$(8, 4, 3, 6, 1, 9), 10, (12, \underline{18}, 18)$
 $(1, 4, 3, 6), 8, (9), 10, 12, (\underline{18}, 18)$
 $1, (4, 3, 6), 8, 9, 10, 12, \underline{18}, (18)$
 $1, 3, (4, 6), 8, 9, 10, 12, \underline{18}, 18$
 $1, 3, 4, 6, 8, 9, 10, 12, \underline{18}, 18$

四、算法设计题(每题 15 分, 共 30 分)

1. 设计在单链表中删除值相同的多余结点的算法。

设计在单链表中删除值相同的多余结点的算法。

```

typedef int datatype;
typedef struct node {datatype data; struct node *next;}lklst;
void delredundant(lklst *&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;}
    }
}

```

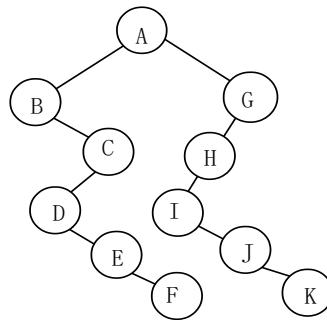
2. 设计一个求结点 x 在二叉树中的双亲结点算法。

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");
}

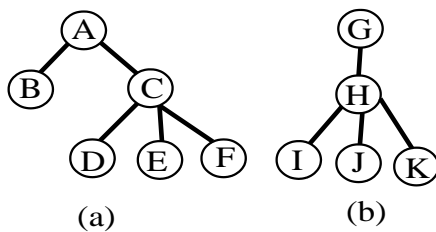
```

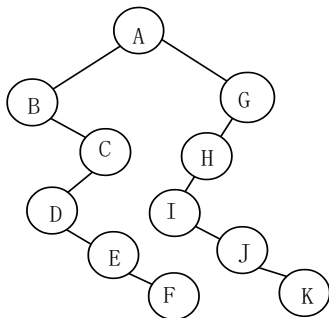
(2) 求森林先序序列和中序序列；

ABCDEF; BDEFCA;

(3) 将此森林转换为相应的二叉树；

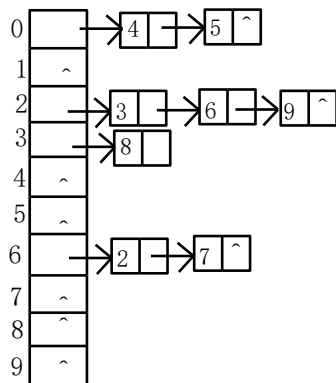


(2) ABCDEFGHIJK; BDEFCAIJKHG 林转换为相应的二叉树；



3、设散列表的地址范围是[0..9]，散列函数为 $H(\text{key}) = (\text{key}^2 + 2) \text{MOD } 9$ ，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结构。

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



四、算法设计题(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。

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;} llist;
void split(llist *head,    llist *&ha,    llist *&hb,    llist *&hc)
{
    llist *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. 设计在链式存储结构上交换二叉树中所有结点左右子树的算法。

设计在链式存储结构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. 在链式存储结构上建立一棵二叉排序树。

在链式存储结构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));
}
```

数据结构试卷（五）

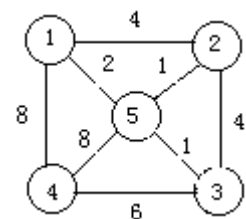
一、选择题(20 分)

- 数据的最小单位是 ()。
(A) **数据项** (B) 数据类型 (C) 数据元素 (D) 数据变量
- 设一组初始记录关键字序列为(50, 40, 95, 20, 15, 70, 60, 45)，则以增量 $d=4$ 的一趟希尔排序结束后前 4 条记录关键字为 ()。
(A) 40, 50, 20, 95 (B) **15, 40, 60, 20**
(C) 15, 20, 40, 45 (D) 45, 40, 15, 20
- 设一组初始记录关键字序列为(25, 50, 15, 35, 80, 85, 20, 40, 36, 70)，其中含有 5 个长度为 2 的有序子表，则用归并排序的方法对该记录关键字序列进行一趟归并后的结果为 ()。
(A) **15, 25, 35, 50, 20, 40, 80, 85, 36, 70**
(B) 15, 25, 35, 50, 80, 20, 85, 40, 70, 36
(C) 15, 25, 35, 50, 80, 85, 20, 36, 40, 70
(D) 15, 25, 35, 50, 80, 20, 36, 40, 70, 85
- 函数 `substr("DATASTRUCTURE", 5, 9)` 的返回值为 ()。
(A) **"STRUCTURE"** (B) "DATA"
(C) "ASTRUCTUR" (D) "DATASTRUCTURE"
- 设一个有序的单链表中 n 个结点，现要求插入一个新结点后使得单链表仍然保持有序，则该操作的时间复杂度为 ()。
(A) $O(\log_2 n)$ (B) $O(1)$ (C) $O(n^2)$ (D) **$O(n)$**
- 设一棵 m 叉树中度数为 0 的结点数为 N_0 ，度数为 1 的结点数为 N_1 ，……，度数为 m 的结点数为 N_m ，则 $N_0 =$ ()。
(A) $N_1 + N_2 + \dots + N_m$ (B) **$1 + N_2 + 2N_3 + 3N_4 + \dots + (m-1)N_m$**
(C) $N_2 + 2N_3 + 3N_4 + \dots + (m-1)N_m$ (D) $2N_1 + 3N_2 + \dots + (m+1)N_m$
- 设有序表中有 1000 个元素，则用二分查找查找元素 x 最多需要比较 () 次。
(A) 25 (B) **10** (C) 7 (D) 1
- 设连通图 G 中的边集 $E = \{(a, b), (a, e), (a, c), (b, e), (e, d), (d, f), (f, c)\}$ ，则从顶点 a 出发可以得到一种深度优先遍历的顶点序列为 ()。
(A) $abedfc$ (B) **$acfebd$** (C) $aebdfc$ (D) $aedfcb$
- 设输入序列是 1、2、3、……、 n ，经过栈的作用后输出序列的第一个元素是 n ，则输出序列中第 i 个输出元素是 ()。
(A) $n-i$ (B) $n-1-i$ (C) **$n+1-i$** (D) 不能确定
- 设一组初始记录关键字序列为(45, 80, 55, 40, 42, 85)，则以第一个记录关键字 45 为基准而得到一趟快速排序的结果是 ()。
(A) 40, 42, 45, 55, 80, 83 (B) 42, 40, 45, 80, 85, 88
(C) **42, 40, 45, 55, 80, 85** (D) 42, 40, 45, 85, 55, 80

三、应用题(32 分)

- 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序遍历序列。

DEBCA



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。

$E=\{(1,5),(5,2),(5,3),(3,4)\}, W=10$

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。

$ASL=(1*1+2*2+3*4)/7=17/7$

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。

$ASL1=7/6, ASL2=4/3$

四、算法设计题(28 分)

1. 设计判断两个二叉树是否相同的算法。

设计判断两个二叉树是否相同的算法。

```
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。

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;
}
```

数据结构试卷（六）

一、选择题(30 分)

1. 设一组权值集合 $W=\{2, 3, 4, 5, 6\}$ ，则由该权值集合构造的哈夫曼树中带权路径长度之和为（ ）。
(A) 20 (B) 30 (C) 40 (D) 45
2. 执行一趟快速排序能够得到的序列是（ ）。
(A) [41, 12, 34, 45, 27] 55 [72, 63]
(B) [45, 34, 12, 41] 55 [72, 63, 27]
(C) [63, 12, 34, 45, 27] 55 [41, 72]
(D) [12, 27, 45, 41] 55 [34, 63, 72]
3. 设一条单链表的头指针变量为 head 且该链表没有头结点，则其判空条件是（ ）。
(A) head==0 (B) head->next==0
(C) head->next==head (D) head!=0
4. 时间复杂度不受数据初始状态影响而恒为 $O(n\log_2 n)$ 的是（ ）。
(A) 堆排序 (B) 冒泡排序 (C) 希尔排序 (D) 快速排序
5. 设二叉树的先序遍历序列和后序遍历序列正好相反，则该二叉树满足的条件是（ ）。
(A) 空或只有一个结点 (B) 高度等于其结点数
(C) 任一结点无左孩子 (D) 任一结点无右孩子
6. 一趟排序结束后不一定能够选出一个元素放在其最终位置上的是（ ）。
(A) 堆排序 (B) 冒泡排序 (C) 快速排序 (D) 希尔排序
7. 设某棵二叉树中有 40 个结点，则该二叉树的最小高度为（ ）。
(A) 3 (B) 4 (C) 5 (D) 6
8. 顺序查找不论在顺序线性表中还是在链式线性表中的时间复杂度为（ ）。
(A) $O(n)$ (B) $O(n^2)$ (C) $O(n^{1/2})$ (D) $O(\log_2 n)$
9. 二路归并排序的时间复杂度为（ ）。
(A) $O(n)$ (B) $O(n^2)$ (C) $O(n\log_2 n)$ (D) $O(\log_2 n)$
10. 深度为 k 的完全二叉树中最少有（ ）个结点。
(A) $2^{k-1}-1$ (B) 2^{k-1} (C) $2^{k-1}+1$ (D) 2^k-1
11. 设指针变量 front 表示链式队列的队头指针，指针变量 rear 表示链式队列的队尾指针，指针变量 s 指向将要入队列的结点 X，则入队列的操作序列为（ ）。
(A) front->next=s; front=s; (B) s->next=rear; rear=s;
(C) rear->next=s; rear=s; (D) s->next=front; front=s;
12. 设某无向图中有 n 个顶点 e 条边，则建立该图邻接表的时间复杂度为（ ）。
(A) $O(n+e)$ (B) $O(n^2)$ (C) $O(ne)$ (D) $O(n^3)$
13. 设某哈夫曼树中有 199 个结点，则该哈夫曼树中有（ ）个叶子结点。
(A) 99 (B) 100 (C) 101 (D) 102
14. 设二叉排序树上有 n 个结点，则在二叉排序树上查找结点的平均时间复杂度为（ ）。
(A) $O(n)$ (B) $O(n^2)$ (C) $O(n\log_2 n)$ (D) $O(\log_2 n)$
15. 设用邻接矩阵 A 表示有向图 G 的存储结构，则有向图 G 中顶点 i 的入度为（ ）。
(A) 第 i 行非 0 元素的个数之和 (B) 第 i 列非 0 元素的个数之和
(C) 第 i 行 0 元素的个数之和 (D) 第 i 列 0 元素的个数之和

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。

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 biseach(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。

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. 在链式存储结构上设计直接插入排序算法

在链式存储结构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;}
    }
}
```

数据结构试卷（七）

一、选择题(30 分)

1. 设某无向图有 n 个顶点，则该无向图的邻接表中有 () 个表头结点。
(A) $2n$ (B) n (C) $n/2$ (D) $n(n-1)$
2. 设无向图 G 中有 n 个顶点，则该无向图的最小生成树上有 () 条边。
(A) n (B) $n-1$ (C) $2n$ (D) $2n-1$
3. 设一组初始记录关键字序列为 (60, 80, 55, 40, 42, 85)，则以第一个关键字 45 为基准而得到的一趟快速排序结果是 ()。
(A) 40, 42, 60, 55, 80, 85 (B) 42, 45, 55, 60, 85, 80
(C) 42, 40, 55, 60, 80, 85 (D) 42, 40, 60, 85, 55, 80
4. () 二叉排序树可以得到一个从小到大的有序序列。
(A) 先序遍历 (B) 中序遍历 (C) 后序遍历 (D) 层次遍历
5. 设按照从上到下、从左到右的顺序从 1 开始对完全二叉树进行顺序编号，则编号为 i 结点的左孩子结点的编号为 ()。
(A) $2i+1$ (B) $2i$ (C) $i/2$ (D) $2i-1$
6. 程序段 $s=i=0$; do { $i=i+1$; $s=s+i$; }while($i \leq n$); 的时间复杂度为 ()。
(A) $O(n)$ (B) $O(n \log_2 n)$ (C) $O(n^2)$ (D) $O(n^3/2)$
7. 设带有头结点的单向循环链表的头指针变量为 head，则其判空条件是 ()。
(A) $head==0$ (B) $head \rightarrow next==0$
(C) $head \rightarrow next==head$ (D) $head!=0$
8. 设某棵二叉树的高度为 10，则该二叉树上叶子结点最多有 ()。
(A) 20 (B) 256 (C) 512 (D) 1024
9. 设一组初始记录关键字序列为 (13, 18, 24, 35, 47, 50, 62, 83, 90, 115, 134)，则利用二分法查找关键字 90 需要比较的关键字个数为 ()。
(A) 1 (B) 2 (C) 3 (D) 4
10. 设指针变量 top 指向当前链式栈的栈顶，则删除栈顶元素的操作序列为 ()。
(A) $top=top+1$; (B) $top=top-1$;
(C) $top \rightarrow next=top$; (D) $top=top \rightarrow next$;

四、算法设计题(20 分)

1. 设计在链式结构上实现简单选择排序算法。

设计在链式结构上实现简单选择排序算法。

```
void simpleselectsort(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. 设计在顺序存储结构上实现求子串算法。

设计在顺序存储结构上实现求子串算法。

```
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。

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);}
}
```

数据结构试卷（八）

一、选择题(30 分)

1. 字符串的长度是指（ ）。
(A) 串中不同字符的个数 (B) 串中不同字母的个数
(C) 串中所含字符的个数 (D) 串中不同数字的个数
2. 建立一个长度为 n 的有序单链表的时间复杂度为（ ）。
(A) $O(n)$ (B) $O(1)$ (C) $O(n^2)$ (D) $O(\log n)$
3. 两个字符串相等的充要条件是（ ）。
(A) 两个字符串的长度相等 (B) 两个字符串中对应位置上的字符相等
(C) 同时具备(A)和(B)两个条件 (D) 以上答案都不对
4. 设某散列表的长度为 100，散列函数 $H(k)=k \% P$ ，则 P 通常情况下最好选择（ ）。
(A) 99 (B) 97 (C) 91 (D) 93
5. 在二叉排序树中插入一个关键字值的平均时间复杂度为（ ）。
(A) $O(n)$ (B) $O(\log n)$ (C) $O(n \log n)$ (D) $O(n^2)$
6. 设一个顺序有序表 $A[1:14]$ 中有 14 个元素，则采用二分法查找元素 $A[4]$ 的过程中比较元素的顺序为（ ）。
(A) $A[1], A[2], A[3], A[4]$ (B) $A[1], A[14], A[7], A[4]$
(C) $A[7], A[3], A[5], A[4]$ (D) $A[7], A[5], A[3], A[4]$
7. 设一棵完全二叉树中有 65 个结点，则该完全二叉树的深度为（ ）。
(A) 8 (B) 7 (C) 6 (D) 5
8. 设一棵三叉树中有 2 个度数为 1 的结点，2 个度数为 2 的结点，2 个度数为 3 的结点，则该三叉链权中有（ ）个度数为 0 的结点。
(A) 5 (B) 6 (C) 7 (D) 8
9. 设无向图 G 中的边的集合 $E=\{(a, b), (a, e), (a, c), (b, e), (e, d), (d, f), (f, c)\}$ ，则从顶点 a 出发进行深度优先遍历可以得到的一种顶点序列为（ ）。
(A) aedfcb (B) acfebd (C) aebcfd (D) aedfbc
10. 队列是一种（ ）的线性表。
(A) 先进先出 (B) 先进后出 (C) 只能插入 (D) 只能删除

四、算法设计题(20 分)

1. 设计一个在链式存储结构上统计二叉树中结点个数的算法。

设计一个在链式存储结构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。

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;
}
}

```

数据结构试卷（九）

一、选择题(30 分)

1. 下列程序段的时间复杂度为 ()。
for(i=0; i<m; i++) for(j=0; j<t; j++) c[i][j]=0;
for(i=0; i<m; i++) for(j=0; j<t; j++) for(k=0; k<n; k++) c[i][j]=c[i][j]+a[i][k]*b[k][j];
(A) $O(m*n*t)$ (B) $O(m+n+t)$ (C) $O(m+n*t)$ (D) $O(m*t+n)$
2. 设顺序线性表中有 n 个数据元素, 则删除表中第 i 个元素需要移动 () 个元素。
(A) $n-i$ (B) $n+1-i$ (C) $n-1-i$ (D) i
3. 设 F 是由 T_1 、 T_2 和 T_3 三棵树组成的森林, 与 F 对应的二叉树为 B , T_1 、 T_2 和 T_3 的结点数分别为 N_1 、 N_2 和 N_3 , 则二叉树 B 的根结点的左子树的结点数为 ()。
(A) N_1-1 (B) N_2-1 (C) N_2+N_3 (D) N_1+N_3
4. 利用直接插入排序的思想建立一个有序线性表的时间复杂度为 ()。
(A) $O(n)$ (B) $O(n\log_2 n)$ (C) $O(n^2)$ (D) $O(\log_2 n)$
5. 设指针变量 p 指向双向链表中结点 A , 指针变量 s 指向被插入的结点 X , 则在结点 A 的后面插入结点 X 的操作序列为 ()。
(A) $p \rightarrow \text{right} = s$; $s \rightarrow \text{left} = p$; $p \rightarrow \text{right} \rightarrow \text{left} = s$; $s \rightarrow \text{right} = p \rightarrow \text{right}$;
(B) $s \rightarrow \text{left} = p$; $s \rightarrow \text{right} = p \rightarrow \text{right}$; $p \rightarrow \text{right} = s$; $p \rightarrow \text{right} \rightarrow \text{left} = s$;
(C) $p \rightarrow \text{right} = s$; $p \rightarrow \text{right} \rightarrow \text{left} = s$; $s \rightarrow \text{left} = p$; $s \rightarrow \text{right} = p \rightarrow \text{right}$;
(D) $s \rightarrow \text{left} = p$; $s \rightarrow \text{right} = p \rightarrow \text{right}$; $p \rightarrow \text{right} \rightarrow \text{left} = s$; $p \rightarrow \text{right} = s$;
6. 下列各种排序算法中平均时间复杂度为 $O(n^2)$ 是 ()。
(A) 快速排序 (B) 堆排序 (C) 归并排序 (D) 冒泡排序
7. 设输入序列 1、2、3、...、 n 经过栈作用后, 输出序列中的第一个元素是 n , 则输出序列中的第 i 个输出元素是 ()。
(A) $n-i$ (B) $n-1-i$ (C) $n+1-i$ (D) 不能确定
8. 设散列表中有 m 个存储单元, 散列函数 $H(\text{key}) = \text{key} \% p$, 则 p 最好选择 ()。
(A) 小于等于 m 的最大奇数 (B) 小于等于 m 的最大素数
(C) 小于等于 m 的最大偶数 (D) 小于等于 m 的最大合数
9. 设在一棵度数为 3 的树中, 度数为 3 的结点数有 2 个, 度数为 2 的结点数有 1 个, 度数为 1 的结点数有 2 个, 那么度数为 0 的结点数有 () 个。
(A) 4 (B) 5 (C) 6 (D) 7
10. 设完全无向图中有 n 个顶点, 则该完全无向图中有 () 条边。
(A) $n(n-1)/2$ (B) $n(n-1)$ (C) $n(n+1)/2$ (D) $(n-1)/2$
11. 设顺序表的长度为 n , 则顺序查找的平均比较次数为 ()。
(A) n (B) $n/2$ (C) $(n+1)/2$ (D) $(n-1)/2$
12. 设有序表中的元素为 (13, 18, 24, 35, 47, 50, 62), 则在其中利用二分法查找值为 24 的元素需要经过 () 次比较。
(A) 1 (B) 2 (C) 3 (D) 4
13. 设顺序线性表的长度为 30, 分成 5 块, 每块 6 个元素, 如果采用分块查找, 则其平均查找长度为 ()。
(A) 6 (B) 11 (C) 5 (D) 6.5
14. 设有向无环图 G 中的有向边集合 $E = \{ \langle 1, 2 \rangle, \langle 2, 3 \rangle, \langle 3, 4 \rangle, \langle 1, 4 \rangle \}$, 则下列属于该有向图 G 的一种拓扑排序序列的是 ()。
(A) 1, 2, 3, 4 (B) 2, 3, 4, 1 (C) 1, 4, 2, 3 (D) 1, 2, 4, 3

15. 设有一组初始记录关键字序列为(34, 76, 45, 18, 26, 54, 92), 则由这组记录关键字生成的二叉排序树的深度为 ()。

(A) 4 (B) 5 (C) 6 (D) 7

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。

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。

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。

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);
}
```


数据结构试卷（十）

一、选择题(24 分)

1. 下列程序段的时间复杂度为 ()。
 $i=0, s=0; \text{ while } (s<n) \{s=s+i; i++; \}$
 (A) $O(n^{1/2})$ (B) $O(n^{1/3})$ (C) $O(n)$ (D) $O(n^2)$
2. 设某链表中最常用的操作是在链表的尾部插入或删除元素, 则选用下列 () 存储方式最节省运算时间。
 (A) 单向链表 (B) 单向循环链表
 (C) 双向链表 (D) **双向循环链表**
3. 设指针 q 指向单链表中结点 A, 指针 p 指向单链表中结点 A 的后继结点 B, 指针 s 指向被插入的结点 X, 则在结点 A 和结点 B 插入结点 X 的操作序列为 ()。
 (A) $s \rightarrow next = p \rightarrow next; p \rightarrow next = s;$ (B) **$q \rightarrow next = s; s \rightarrow next = p;$**
 (C) $p \rightarrow next = s \rightarrow next; s \rightarrow next = p;$ (D) $p \rightarrow next = s; s \rightarrow next = q;$
4. 设输入序列为 1、2、3、4、5、6, 则通过栈的作用后可以得到的输出序列为 ()。
 (A) 5, 3, 4, 6, 1, 2 (B) **3, 2, 5, 6, 4, 1**
 (C) 3, 1, 2, 5, 4, 6 (D) 1, 5, 4, 6, 2, 3
5. 设有一个 10 阶的下三角矩阵 A (包括对角线), 按照从上到下、从左到右的顺序存储到连续的 55 个存储单元中, 每个数组元素占 1 个字节的存储空间, 则 $A[5][4]$ 地址与 $A[0][0]$ 的地址之差为 ()。
 (A) 10 (B) **19** (C) 28 (D) 55
6. 设一棵 m 叉树中有 N_1 个度数为 1 的结点, N_2 个度数为 2 的结点, \dots , N_m 个度数为 m 的结点, 则该树中共有 () 个叶子结点。
 (A) $\sum_{i=1}^m (i-1)N_i$ (B) $\sum_{i=1}^m N_i$ (C) $\sum_{i=2}^m N_i$ (D) **$1 + \sum_{i=2}^m (i-1)N_i$**
7. 二叉排序树中左子树上所有结点的值均 () 根结点的值。
 (A) **<** (B) > (C) = (D) !=
8. 设一组权值集合 $W=(15, 3, 14, 2, 6, 9, 16, 17)$, 要求根据这些权值集合构造一棵哈夫曼树, 则这棵哈夫曼树的带权路径长度为 ()。
 (A) 129 (B) 219 (C) 189 (D) **229**
9. 设有 n 个关键字具有相同的 Hash 函数值, 则用线性探测法把这 n 个关键字映射到 HASH 表中需要做 () 次线性探测。
 (A) n^2 (B) $n(n+1)$ (C) $n(n+1)/2$ (D) **$n(n-1)/2$**
10. 设某棵二叉树中只有度数为 0 和度数为 2 的结点且度数为 0 的结点数为 n, 则这棵二叉树中共有 () 个结点。
 (A) 2n (B) n+1 (C) **2n-1** (D) 2n+1
11. 设一组初始记录关键字的长度为 8, 则最多经过 () 趟插入排序可以得到有序序列。
 (A) 6 (B) **7** (C) 8 (D) 9
12. 设一组初始记录关键字序列为(Q, H, C, Y, P, A, M, S, R, D, F, X), 则按字母升序的第一趟冒泡排序结束后的结果是 ()。
 (A) F, H, C, D, P, A, M, Q, R, S, Y, X
 (B) P, A, C, S, Q, D, F, X, R, H, M, Y
 (C) A, D, C, R, Q, M, S, Y, P, H, X
 (D) **H, C, Q, P, A, M, S, R, D, F, X, Y**

三、算法设计题(22 分)

1. 设计在链式存储结构上合并排序的算法。

设计在链式存储结构上合并排序的算法。

```
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。

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。

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;
}
```

数据结构试卷（一）参考答案

一、选择题（每题 2 分，共 20 分）

1.A 2.D 3.D 4.C 5.C 6.D 7.D 8.C 9.D 10.A

二、填空题（每空 1 分，共 26 分）

1. 正确性 易读性 强壮性 高效率

2. $O(n)$

3. 9 3 3

4. $-1 \quad 3 \quad 4 \quad X * + \quad 2 \quad Y * \quad 3 \quad / \quad -$

5. $2n \quad n-1 \quad n+1$

6. $e \quad 2e$

7. 有向无回路

8. $n(n-1)/2 \quad n(n-1)$

9. (12, 40) () (74) (23, 55, 63)

10. 增加 1

11. $O(\log_2 n) \quad O(n \log_2 n)$

12. 归并

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:

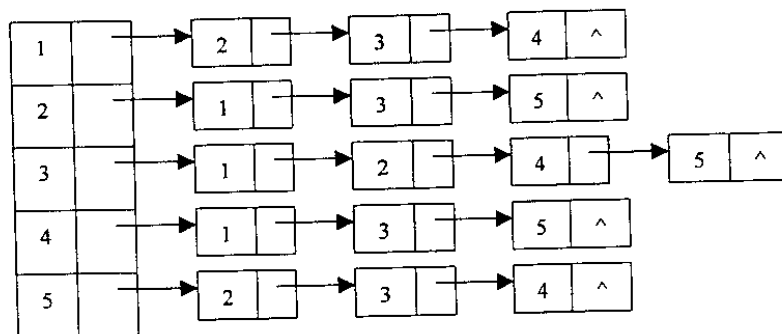
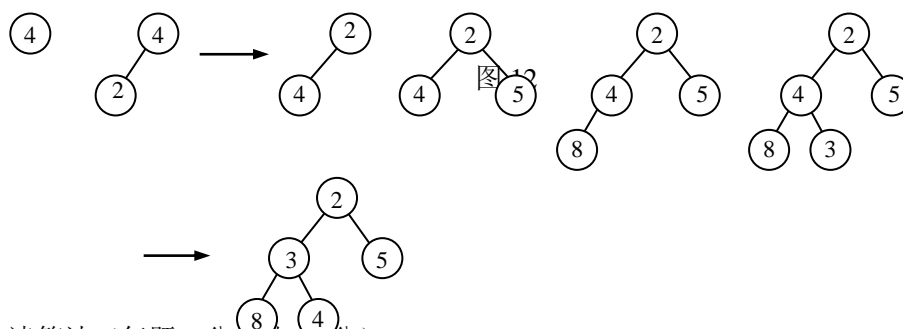


图 11

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) 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 $(a_2, a_3, \dots, a_n, a_1)$
- 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

```

一、选择题

1. D 2. B 3. C 4. A 5. A 6. C 7. B 8. C

二、填空题

1. 构造一个好的 HASH 函数，确定解决冲突的方法
2. `stack.top++`, `stack.s[stack.top]=x`
3. 有序
4. $O(n^2)$, $O(n\log_2 n)$
5. N_0-1 , $2N_0+N_1$
6. $d/2$
7. (31, 38, 54, 56, 75, 80, 55, 63)
8. (1, 3, 4, 5, 2), (1, 3, 2, 4, 5)

三、应用题

1. (22, 40, 45, 48, 80, 78), (40, 45, 48, 80, 22, 78)
2. `q->llink=p; q->rlink=p->rlink; p->rlink->llink=q; p->rlink=q;`
3. 2, $ASL=91*1+2*2+3*4+4*2=25/9$
4. 树的链式存储结构略，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) ，要求设计一个算法能够在 $O(n)$ 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 用链式存储结构表示。

```
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;}
    }
}
```

}

数据结构试卷（三）参考答案

一、选择题

1. B 2. B 3. A 4. A 5. A
6. B 7. D 8. C 9. B 10. D

第 3 小题分析：首先用指针变量 q 指向结点 A 的后继结点 B，然后将结点 B 的值复制到结点 A 中，最后删除结点 B。

第 9 小题分析：9 快速排序、归并排序和插入排序必须等到整个排序结束后才能够求出最小的 10 个数，而堆排序只需要在初始堆的基础上再进行 10 次筛选即可，每次筛选的时间复杂度为 $O(\log_2 n)$ 。

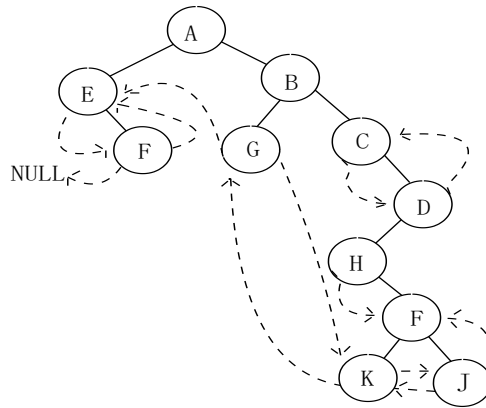
二、填空题

1. 顺序存储结构、链式存储结构
2. 9, 501
3. 5
4. 出度，入度
5. 0
6. $e=d$
7. 中序
8. 7
9. $O(1)$
10. $i/2, 2i+1$
11. (5, 16, 71, 23, 72, 94, 73)
12. (1, 4, 3, 2)
13. $j+1, \text{hashtable}[j].\text{key}==k$
14. $\text{return}(t), t=t \rightarrow \text{rchild}$

第 8 小题分析：二分查找的过程可以用一棵二叉树来描述，该二叉树称为二叉判定树。在有序表上进行二分查找时的查找长度不超过二叉判定树的高度 $1+\log_2 n$ 。

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、 $(8,9,4,3,6,1), 10, (12, \underline{18}, 18)$

$(1,6,4,3), 8, (9), 10, 12, (\underline{18}, 18)$

$1, (3,4,6), 8, 9, 10, 12, \underline{18}, (18)$

$1, 3, (4,6), 8, 9, 10, 12, \underline{18}, 18$

$1, 3, 4, 6, 8, 9, 10, 12, \underline{18}, 18$

四、算法设计题

1. 设计在单链表中删除值相同的多余结点的算法。

```
typedef int datatype;
typedef struct node {datatype data; struct node *next;} llist;
void delredundant(llist *&head)
{
    llist *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) break;
    if (flag==0) printf("not found x\n");
    else if (i<=r) printf("%c",bt->data); else printf("not parent");
}

```

数据结构试卷（四）参考答案

一、选择题

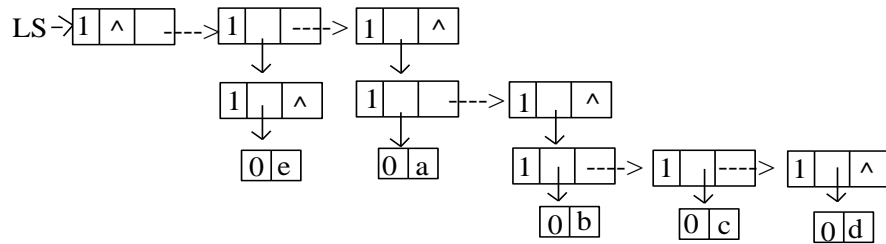
- | | | | | |
|------|------|------|------|-------|
| 1. C | 2. D | 3. D | 4. B | 5. C |
| 6. A | 7. B | 8. A | 9. C | 10. A |

二、填空题

1. $O(n^2)$, $O(n \log_2 n)$
2. $p \rightarrow llink \rightarrow rlink = p \rightarrow rlink$; $p \rightarrow rlink \rightarrow llink = p \rightarrow rlink$
3. 3
4. 2^{k-1}
5. $n/2$
6. 50, 51
7. $m-1, (R-F+M)\%M$
8. $n+1-i, n-i$
9. (19, 18, 16, 20, 30, 22)
10. (16, 18, 19, 20, 32, 22)
11. $A[i][j]=1$
12. 等于
13. BDCA
14. $hashtable[i]=0, hashtable[k]=s$

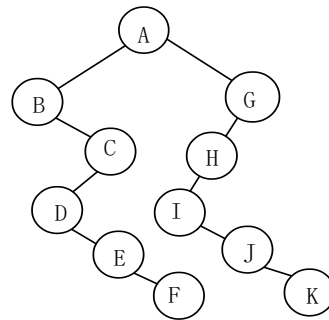
三、计算题

- 1.

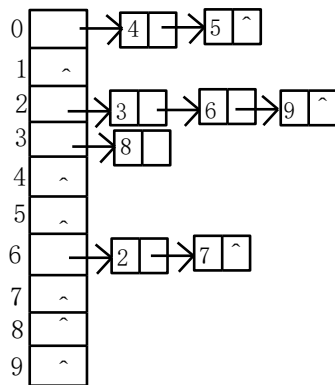


2.

(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; } llist;
void split(llist *head, llist *&ha, llist *&hb, llist *&hc)
{
    llist *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. 设计在链式存储结构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. 在链式存储结构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));
}

```

数据结构试卷（五）参考答案

一、选择题

- | | | | | |
|------|------|------|------|-------|
| 1. A | 2. B | 3. A | 4. A | 5. D |
| 6. B | 7. B | 8. B | 9. C | 10. C |

二、填空题

1. $top1+1=top2$
2. 可以随机访问到任一个顶点的简单链表
3. $i(i+1)/2+j-1$
4. FILO, FIFO
5. ABDECF, DBEAFC, DEBFCA
6. 8, 64
7. 出度, 入度
8. $k_i \leq k_{2i} \ \&\& \ k_i \leq k_{2i+1}$
9. $n-i, \ r[j+1]=r[j]$
10. $mid=(low+high)/2, \ r[mid].key>k$

三、应用题

2. DEBCA
3. $E=\{(1,5),(5,2),(5,3),(3,4)\}, W=10$

4. $ASL = (1 \times 1 + 2 \times 2 + 3 \times 4) / 7 = 17/7$

5. $ASL1 = 7/6$, $ASL2 = 4/3$

四、算法设计题

1. 设计判断两个二叉树是否相同的算法。

```
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;
}
```

数据结构试卷（六）参考答案

一、选择题

1. D 2. A 3. A 4. A 5. D
 6. D 7. B 8. A 9. C 10. B
 11. C 12. A 13. B 14. D 15. B

二、判断题

1. 错 2. 对 3. 对 4. 对 5. 错
 6. 错 7. 对 8. 错 9. 对 10. 对

三、填空题

1. $O(n)$
 2. $s \rightarrow next = p \rightarrow next; p \rightarrow next = s$
 3. (1, 3, 2, 4, 5)
 4. $n-1$
 5. 129
 6. $F == R$
 7. $p \rightarrow lchild == 0 \&\& p \rightarrow rchild == 0$
 8. $O(n^2)$
 9. $O(n \log_2 n)$, $O(n)$

10. 开放定址法，链地址法

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. 在链式存储结构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;}
    }
}
```

数据结构试卷（七）参考答案

一、选择题

- | | | | | |
|------|------|------|------|-------|
| 1. B | 2. B | 3. C | 4. B | 5. B |
| 6. A | 7. C | 8. C | 9. B | 10. D |

二、判断题

- | | | | | |
|------|------|------|------|-------|
| 1. 对 | 2. 对 | 3. 对 | 4. 对 | 5. 对 |
| 6. 对 | 7. 对 | 8. 错 | 9. 错 | 10. 错 |

三、填空题

1. $s \rightarrow \text{left} = p, p \rightarrow \text{right}$
2. $n(n-1), n(n-1)/2$
3. $n/2$
4. 开放定址法, 链地址法
5. 14
6. $2^{h-1}, 2^h-1$
7. (12, 24, 35, 27, 18, 26)
8. (12, 18, 24, 27, 35, 26)
9. 5
10. $i < j \ \&\& \ r[i].\text{key} < x.\text{key}, \ r[i] = x$

四、算法设计题

1. 设计在链式结构上实现简单选择排序算法。

```
void simpleselectsorlklst(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. 设计在顺序存储结构上实现求子串算法。

```
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);}
    }

```

数据结构试卷（八）参考答案

一、选择题

1. C 2. C 3. C 4. B 5. B
6. C 7. B 8. C 9. A 10. A

二、判断题

1. 对 2. 错 3. 对 4. 错 5. 错
6. 对 7. 对 8. 对 9. 对 10. 对

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结构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;
        }
}

```

数据结构试卷（九）参考答案

一、选择题

1. A 2. A 3. A 4. C 5. D
 6. D 7. C 8. B 9. C 10. A
 11. C 12. C 13. D 14. A 15. A

二、填空题

1. p->next, s->data
2. 50
3. m-1
4. 6, 8
5. 快速, 堆
6. 19/7
7. CBDA
8. 6
9. (24, 65, 33, 80, 70, 56, 48)
10. 8

三、判断题

1. 错 2. 对 3. 对 4. 对 5. 错
 6. 错 7. 对 8. 对 9. 错 10. 对

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);
}

```

数据结构试卷（十）参考答案

一、选择题

1. A 2. D 3. B 4. B 5. B 6. D
 7. A 8. D 9. D 10. C 11. B 12. D

二、填空题

1. 4, 10
2. $O(n\log_2 n)$, $O(n^2)$
3. n
4. 1, 2
5. $n(m-1)+1$
6. q->next
7. 线性结构, 树型结构, 图型结构
8. $O(n^2)$, $O(n+e)$
9. 8/3
10. (38, 13, 27, 10, 65, 76, 97)
11. (10, 13, 27, 76, 65, 97, 38)
12. 124653
13. struct node *rchild, bt=0, createbitree(bt->lchild)
14. lklist, q=p

三、算法设计题

1. 设计在链式存储结构上合并排序的算法。

```
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;
}
```