

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 3 4 X * + 2 Y * 3 / -

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

6. e $2e$

7. 有向无回路

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

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

10. 增加 1

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

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:

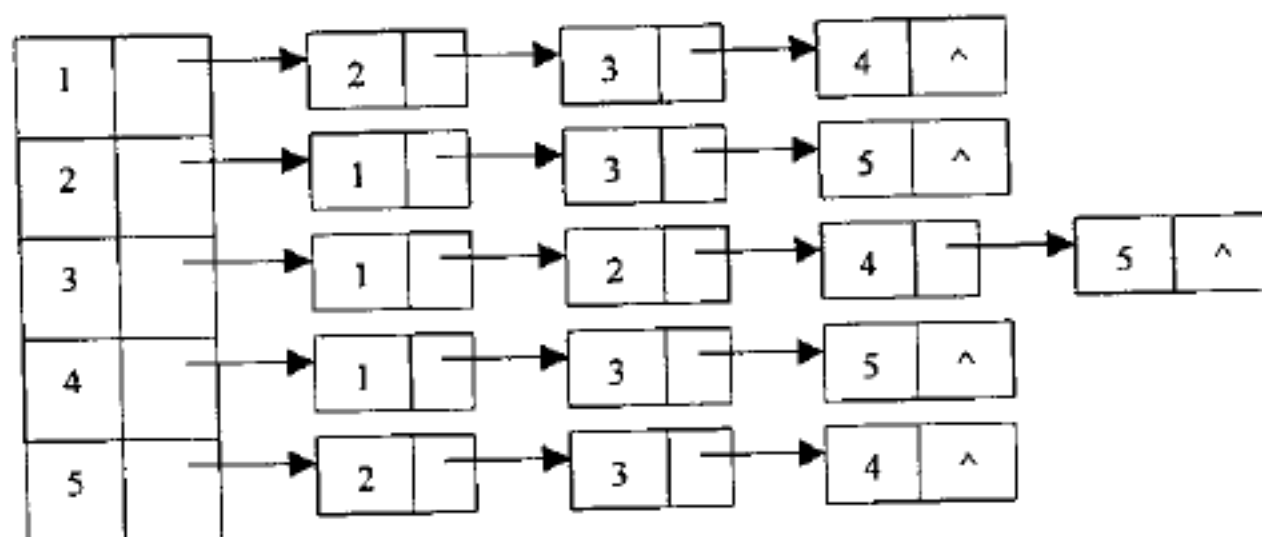
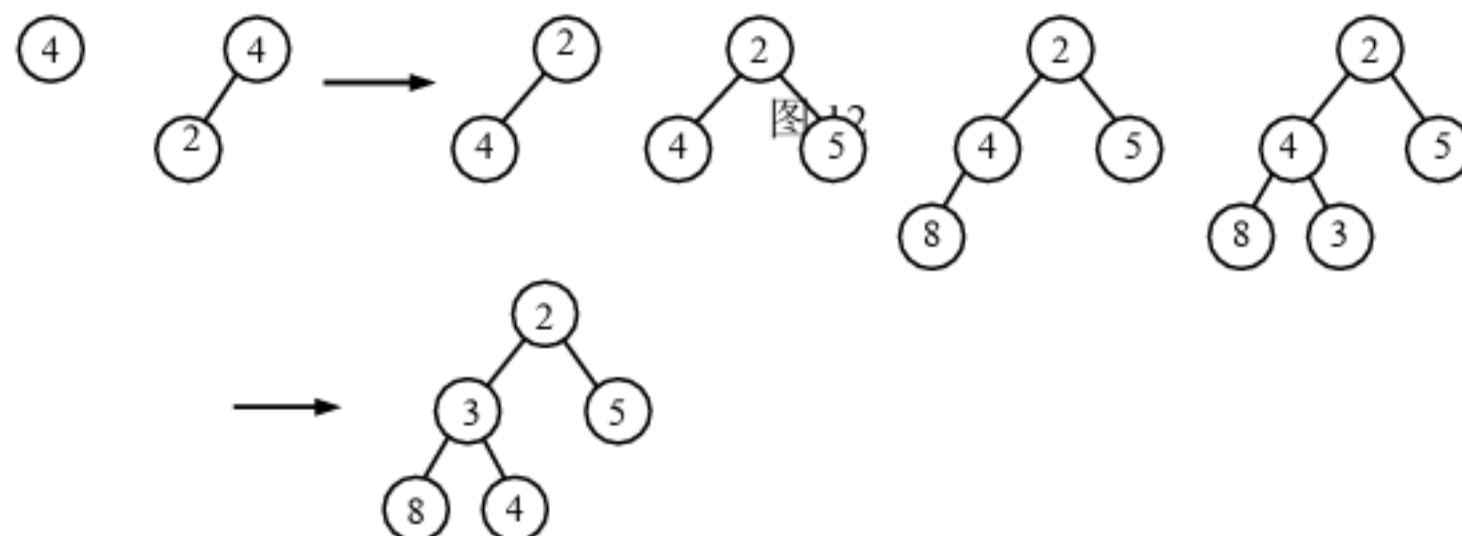


图 11

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. (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
```

数据结构试卷（二）参考答案

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. $q \rightarrow llink = p$; $q \rightarrow rlink = p \rightarrow rlink$; $p \rightarrow rlink \rightarrow llink = q$; $p \rightarrow rlink = q$;
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

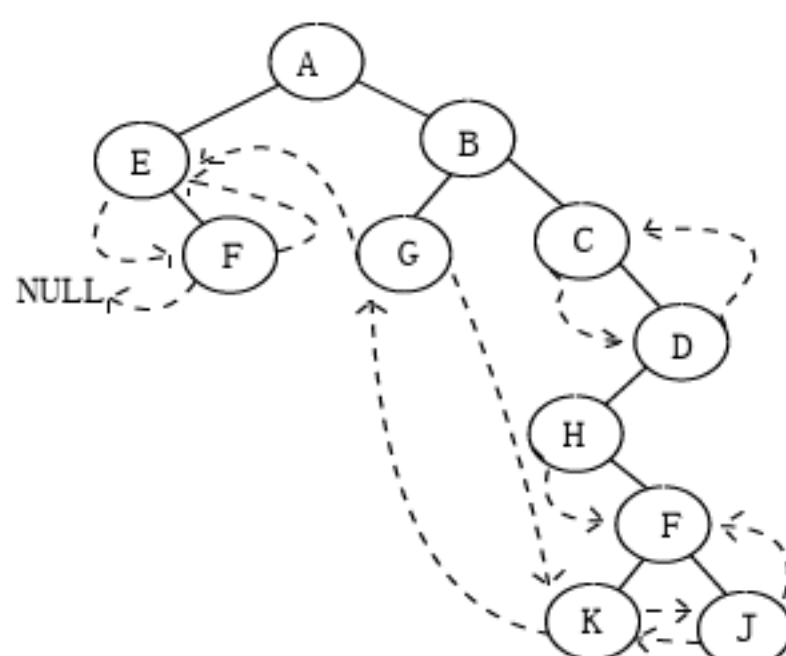
1. 设有一组初始记录关键字序列 (K_1, K_2, \dots, K_n) ，要求设计一个算法能够在 $O(n)$ 的时间复杂度内将线性表划分成两部分，其中左半部分的每个关键字均小于 K_i ，右半部分的每个关键字均大于等于 K_i 。

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,(12,18,18)
(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)
```

```

{
    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 |

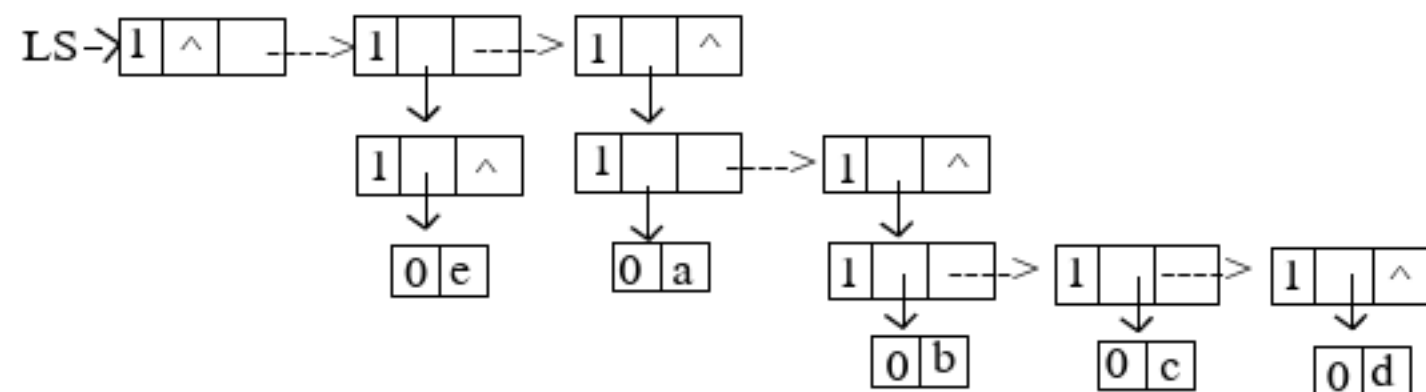
二、填空题

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. $hashtable[i]=0, hashtable[k]=s$

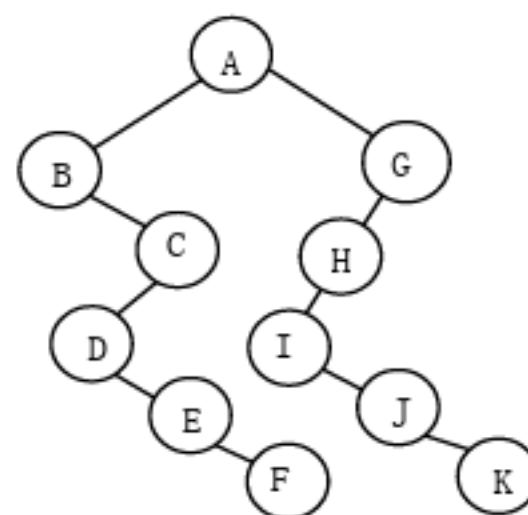
三、计算题

1.

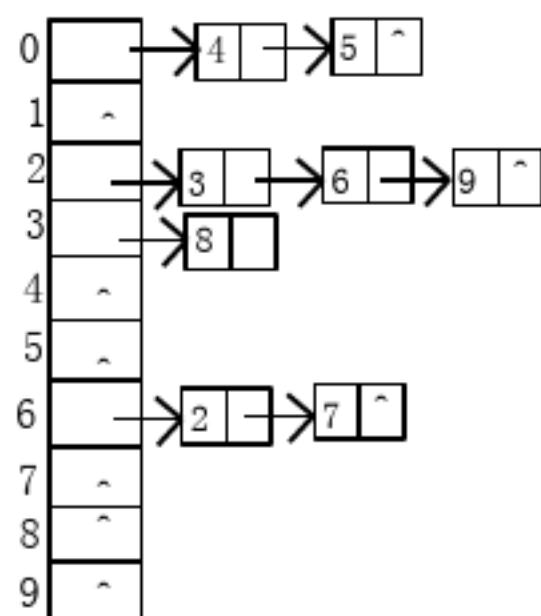


2.

(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)
{
```

```

    lklist *p; ha=0,hb=0,hc=0;
    for(p=head;p!=0;p=head)
    {
        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. $\text{mid}=(\text{low}+\text{high})/2, \ r[\text{mid}].\text{key}>k$

三、应用题

2. DEBCA
3. $E=\{(1,5),(5,2),(5,3),(3,4)\}, W=10$
4. $ASL=(1*1+2*2+3*4)/7=17/7$
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 simpleselectsortlklst(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;
}
```