参考答案

【1999年--数据结构】

一、填空

1、 存储密度＝$\frac{\text{串值所占的存储位}}{\text{实际分配的存储位}}$

2、 单个元素，广义表。

3、稠密，非稠密。

二、

算法如下：

```
void DeleteSame (Linklist &L){
//删除链表中值相同的多余结点，L 为表头结点。
      p=q=T;
      while(p->next!=Null){
      while(q->next!=Null){
          if(q->next->data==p->data){
                  r=q->next;//被删结点位置
                  q->next=r->next;//将 q 指向被删的结点的后继结点。
                  free ( r );//删除多余结点。
          }
          else q=q->next;
      }//while
      p=p->next;
      q=p;
      }//while
}
```

三、

算法如下：

```
int get_depth (Bitree * T ){
   if(!T) return 0;
   else{
      m=get_depth(T->lchild);
      n=get_depth(T->rchild);
      return (m>n?m:n)+1;
}
```

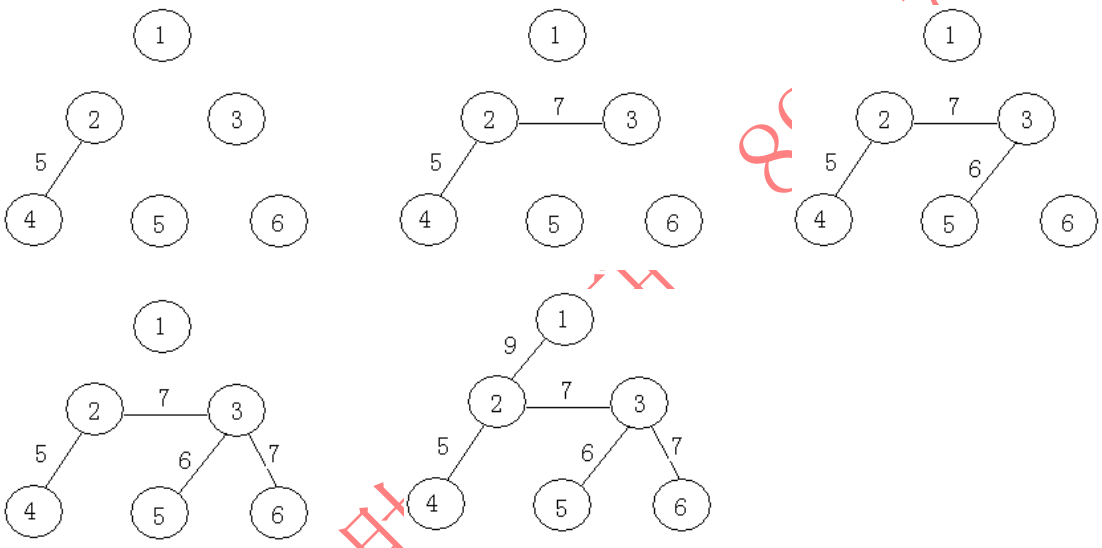
四、

算法如下：

```
typedef struct ChainNode{
     int key;
     struct ChainNode*next;
}ChainNode,*Chain;
Chain ChainHash［CHAINNUM］;//将哈希链表设置为全局变量。
Void DeleteHash(int key){
//使用链地址法解决哈希表元素删除
         int I=Hash(key);//Hash 函数返回关键字 key 所对应的链表。
```

```
        p=ChainHash [ i ];
        while(p->next!=Null&&p->next->key!=key){
            p=p->next;//p 指向所要删除元素的前驱。
        }//while
        if(p->next==Null) return FALSE;
        else{
              q=p->next;
              p->next=q->next;
              free(q);
              return TRUE;
        }
}
```

五、



六、

算法如下：

```
#define N 100
int a[N][N];
main(){
    int i,j,n;
    printf("please input row:\n");
    scanf("%d",&n);
    a[0][n]=1;
    for(i=1;i<n;i++){
      a[i][n-i]=1;
      a[i][n+i]=1;
      for(j=n-i+2;j<n+i;j+=2)
        a[i][j]=a[i-1][j-1]+a[i-1][j+1];
    }
    for(i=0;i<n;i++){
        for(j=0;j<n-i;j++)printf(" ");
```

```
        printf("%d",a[i][j]);
        if(i!=0){
          for(j=j+2;j<n+i;j+=2){
            printf(" ");
            printf("%d",a[i][j]);
          }
          printf(" ");
          printf("%d",a[i][j]);
        }
        printf("\n");
    }
}
```

七、

在最好情况下要做关键字的比较次数为：n－1

在最坏情况下要做关键字的比较次数为：(n+2)(n+1)/2

八、

(1) 设 Ki＝Kj（1<=I<=n,1<=j<=n）且排序前序列为 Ri 领先于 Rj，若在排序后的序列中 Ri 仍领先于 Rj，则称这种排序方法是稳定的。

(2) 直接插入排序和归并排序是稳定的。

希尔排序和快速排序是不稳定的。

(3) 略；

【2000 年--数据结构】

一、 填空

1、 FEGHDCB

2、 $2^K - L$

3、 索引顺序存取；虚拟存储存取。

二、

由题意，

$$k = \begin{cases} 3(i-1) & j = i-1 \\ 3(i-1)+1 & j = i \\ 3(i-1)+2 & j = i+1 \end{cases}$$

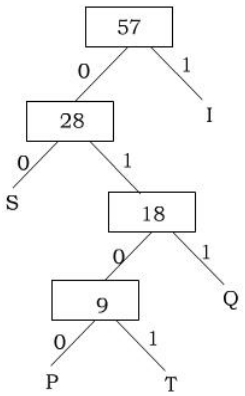
∴ k=2(I-1)+j;

∴Loc(A[I, j])=Loc(B[1])+2(I-1)+j-1

三、

1. 该短文的总长度最短为 112；

哈夫曼编码：



S:00 I：1 P:0100 Q:011 T:0101

2、SPQTS

四、参考严蔚敏的《数据结构》第二版

六、算法如下：

```
#include<stdio.h>
main(){
int m,n,j,i,k,a;
printf(“please input N:”);
scanf(&n);
for(m=0,a=1,k=1;m<=n-1;m++){
      if(a>0){
      I=0;j=m;
      While(I<=m){B[I][j]=k;I++;j--;k++;}
      }
      else{
      I=m;j=0;
      While(j<=m){B[I][j]=k;I--;j++;k++;}
      }
      a=-a;
}
}
```

七、

由题意 H(k)=(3k)MOD11，

H（22）＝3，H（41）＝2，H（53）＝5，H（46）＝6，H（30）＝2，H（13）＝6，H（01）＝3，H（67）＝3，

由于 41 和 30 冲突则散列 d1=H(k)=2　∴(H(30)+d1)mod11=4

13 和 46 冲突 13 地址为：(H(13)+d1)mod11=1

01 和 22 冲突，01 地址：(H(1)+d1)mod11=6

又与 46 冲突，d2=(d1+(7k)mod10+1)=11

01 地址：(H(1)+d2)mod11=3 又 冲突

再次散列：d3=(d2+(7k)mod10+1)=19

01 地址：(H(1)+d3)mod11=0

同理，67 地址：10

0	1	2	3	4	5	6	7	8	9	10
01	13	41	22	30	53	46				67

则查找成功的平均查找长度：ASL=1/8(1+1+1+1+2+2+4+9)=21/8

查找不成功的平均查找长度：uASL=1/11(1+1+1+8+7+6+5+4+3+2+9)=47/11

八、算法如下：

```
void DNF (int & a[],int n)
{//整型数组 a 存放待分类的条块，不使用 a[0]单元，参数 n 为条块个数;
  int Red,Yel,temp;
  Red=1;Yel=n;
  While(Red<=Yel){
     If(a[Red]==RED){Red++;}
    Else{
    temp=a[Red];
    a[Red]=a[Yel];
    a[Yel]=temp;
    Yel--;
    }
  }
}
```

【2001年--数据结构】

一、参考严蔚敏的《数据结构》第二版

二、

算法如下：

```
void ConnectList (Linklist La,Linklist Lb,Linklist Lc){
      pa=La;
      pb=Lb;
      Lc=Null;
      While(pa!=Null&&pb!=Null){
        If(pa->data<pb->data){
           q=pa->next;
           pa->next=Lc;
           Lc=pa;
           pa=q;
         }
         else{
           q=pb->next;
           pb->next=Lc;
           Lc=pb;
           pb=q;
       }
```

```
    }
    if(pb==Null)pb=pa;
    while(pb!=Null){
        q=pb->next;
        pb->next=Lc;
        Lc=pb;
        pb=q;
    }
}
```

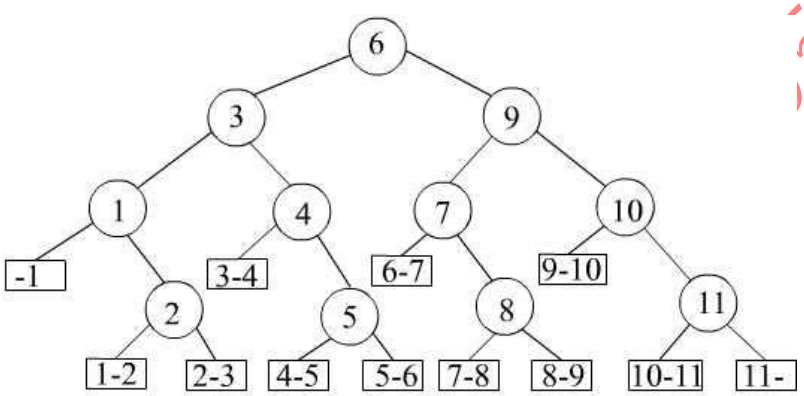
三、1999年第八题。

四、参考严蔚敏的《数据结构》第二版

五、

1、折半查找过程的判别树如下：

图中：1-11对应：15，23，……，102



2、在查找成功时，关键字比较的次数：　　1+2*2+4*3+4*4=33

查找成功的平均查找长度：　　ASL=33/11=3

在查找不成功时，关键字比较的次数　　4*3+8*4=44

查找不成功的平均查找长度：　　UASL=44/12≈3.67

八、

算法如下：

```
#include<stdio.h>
#include<string.h>
#define N 1000
char a[N][N/10];
void main(){
    int i,j,n,l,len,slen=1;
    char b[N][N/10],s1;
    i=j=l=0;n=1;
    printf("please input aggregate element with '#' end :\n");
    while(1){
        scanf("%s",b[i]);
        s1=b[i][0];
        if(s1=='#')break;
        i++;
```

```
    }
    len=i;//集合长度;
    for(i=0;i<len;i++)slen=2*slen;//求幂集大小;
    a[0][0]='$';//空集$是幂集;
    while(n<slen){
        strcpy(a[n],b[l]);
        n++;
        for(i=1;i<=j;i++){
            strcpy(a[n],b[l]);
            strcat(a[n],",");
            strcat(a[n],a[i]);//从1～j，新的集合元素与前结合;
            n++;
        }
        j=n-1;
        l++;
    }
    printf("the power aggregate as followed:\n");
    for(i=0;i<slen;i++){
        printf("{%s},",a[i]);
    }
    printf("\n");
}
```

运行结果：

```
please input aggregate element with '#' end :
zhang
wang
li
zhao
#
the power aggregate as followed:
{$},{zhang},{wang},{wang,zhang},{li},{li,zhang},{li,wang},{li,wang,z
hang},{zhao},{zhao,zhang},{zhao,wang},{zhao,wang,zhang},{zhao,li},{z
hao,li,zhang},{zhao,li,wang},{zhao,li,wang,zhang},
Press any key to continue
```


【2002年--数据结构】

一、填空

1、线性表，入栈，出栈。

2、01123456789；01010101019

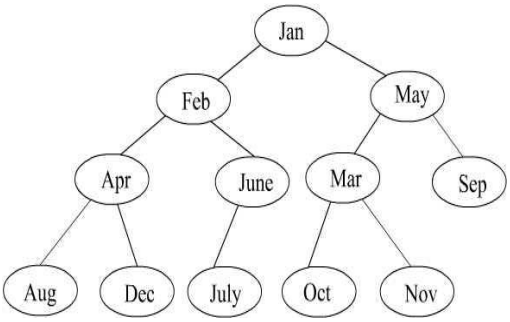
3、存储密度＝$\frac{\text{串值所占的存储位}}{\text{实际分配的存储位}}$，

4、num[col]表示矩阵 M 中第 col 列中非零元的个数；cpot[col]指示 M 中第 col 列的第一个非零元在 b.data 中的恰当位置。

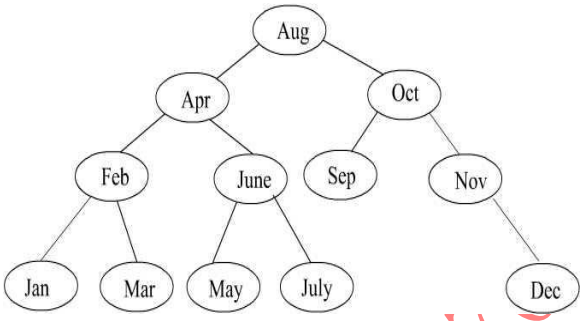
5、520

二、2n(m-2)

三



按字母顺序作权值



按月份顺序作权值

四、（1）可以，先让 1，2，3 依次入栈，再让 3，2 出栈，然后让 4，5 入栈，再让 5 出栈，然后 6 进栈，最后 6，4，1 依次出栈。

（2）不能，因为根据栈的“后进先出”的原则，当 5 入栈后，2，3，4 必在栈中，且出栈相对顺序为 4，3，2,而题中顺序不对。

五、参考严蔚敏的《数据结构》第二版

六、如下表：

		B	c	d	e	f	g	h	vi	s
第一步	I=1	12 (a,b)	15 (a,c)	∞	12 (a,b)	∞	∞	∞	e	{a,e}
第二步	I=2	7 (a,e,b)	15 (a,c)	10 (a,e,d)		6 (a,e,f)	∞	5 (a,e,h)	H	{a,e,h}
第三步	I=3	7 (a,e,b)	15 (a,c)	10 (a,e,d)		6 (a,e,f)	7 (a,e,h,g)		F	{a,e,h,f}
第四步	I=4	7 (a,e,b)	15 (a,c)	10 (a,e,d)			7 (a,e,h,g)		B	{a,e,h,f,b}
第五步	I=5		15 (a,c)	10 (a,e,d)					G	{a,e,h,f,b,g}
第六步	I=6		15 (a,c)	10 (a,e,d)					D	{a,e,h,f,b,g,d}
第七步	I=7		15 (a,c)						c	{a,e,h,f,b,g,d,c}

七、算法如下：

```
void get_next( sstring T,int &next[]){
        I=1;j=0;
        Next[1]=0;
        While(I<m){
            If(j==0||T[I]=T[j]){++I;++j;next[I]=j;}
            Else j=next[j];
        }
}
```

八、算法如下：

```
#include<stdio.h>
#define Max 200
int p[Max],n, m;       /*全局变量*/
void jose();
void writedat();
void main(){
    printf("input n,m:");
    scanf("%d,%d,%d",&n,&m);
    jose();
    writedat();
}
void jose(){
    int p1[Max],i,j,k=0,s1;
    for(i=1;i<=n;i++)p1[i-1]=i;     /*置初始编号*/
    s1=m-1;                         /*找 m 对应的下标 s1*/
    p[k++]=p1[s1];
    p1[s1]=0;
    for(i=1;i<n;i++){               /*I 表示出圈的人数*/
        j=0;
        while(j<m){                 /*找下一个出圈人的编号 s1*/
            s1=(s1+1)%n;
            if(p1[s1]!=0)j++;
        }
        p[k++]=p1[s1]; p1[s1]=0; /*p1[s1]出圈并置为 0*/
    }
}
void writedat(){
    int i;
    char ch;
    FILE *fp;
    fp=fopen("jose.dat","w+");
    for(i=0;i<n;i++){
        fprintf(fp,"%4d",p[i]);
```

```
        if((i+1)%10==0){            /*每 10 人换一行*/
            fprintf(fp,"\n");
        }
    }
    rewind(fp);                     /*文件位置指针置在文件开始处*/
    printf("output as followed:\n");
    ch=fgetc(fp);
    while(ch!=EOF){
        printf("%c",ch);
        ch=fgetc(fp);
    }
    fclose(fp);
}
```