

苏州大学计算机考研——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：

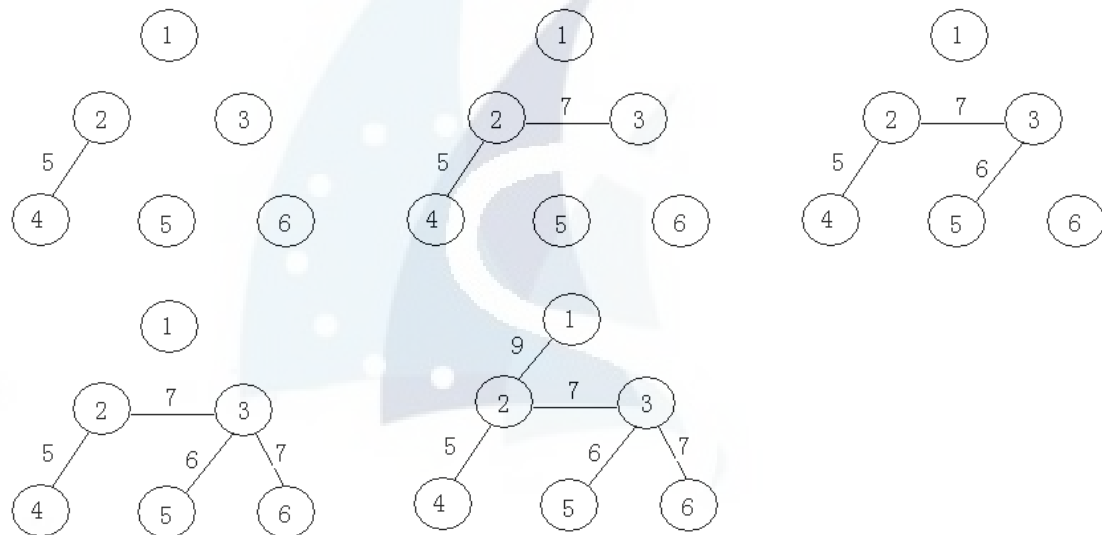
```
typedef struct ChainNode{
    int key;
    struct ChainNode*next;
}ChainNode,*Chain;
Chain ChainHash [ CHAINNUM ] ;//将哈希链表设置为全局变量。
Void DeleteHash(int key){
//使用链地址法解决哈希表元素删除
```

```

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 $K_i = K_j$ ($1 \leq i \leq n, 1 \leq j \leq n$) 且排序前序列为 R_i 领先于 R_j , 若在排序后的序列中 R_i 仍领先于 R_j , 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,

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

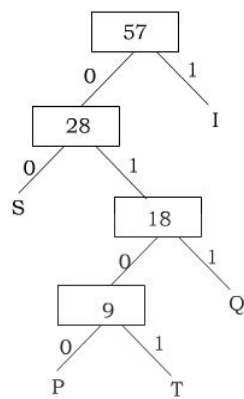
$$\therefore k = 2(I-1) + j;$$

$$\therefore \text{Loc}(A[I, j]) = \text{Loc}(B[1]) + 2(I-1) + j - 1$$

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 $H(k) = (3k) \text{MOD} 11$,

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

由于 41 和 30 冲突则散列 $d1 = H(k) = 2 \quad \therefore (H(30) + d1) \text{mod} 11 = 4$

13 和 46 冲突 13 地址为: $(H(13) + d1) \text{mod} 11 = 1$

01 和 22 冲突, 01 地址: $(H(1) + d1) \text{mod} 11 = 6$

又与 46 冲突, $d2 = (d1 + (7k) \text{mod} 10 + 1) = 11$

01 地址: $(H(1) + d2) \text{mod} 11 = 3$ 又 冲突

再次散列: $d3 = (d2 + (7k) \text{mod} 10 + 1) = 19$

01 地址: $(H(1) + d3) \text{mod} 11 = 0$

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--;
```

```
    }
```

```
}
```

```
}
```

王道论坛

www.cskaoyan.com