第 4 章习题

4.1 设计算法以判断两个顺序串是否相等，若相等，返回 1，否则返回 0。

【解】算法思想：用两个顺序表存储两个串。用指针 i 从 0 开始比较两个串，如果两个串同时结束，且对应字符相等，则两个串相等，返回 1；其它情况皆属不想等情况，返回 0。

【算法描述】

```
bool seqStringCompare(seqList & S1, seqList & S2)
{
 //比较 2 个顺序串相等，使用顺序表表示
 if(S1.listLen!=S2.listLen)
      return 0;   //长度不同，肯定不相等
 for(int i=0; i<S1.listLen; i++)
 {
      if(S1.data[i]!=S2.data[i])
           return 0;   //对应字符不同，返回 0
 }
 return 1;   //程序执行到这里，则 2 个串相等
}
```

4.2 设计算法以判断链串 S1 是否是链串 S2 的子串，若是子串，返回 1，否则返回 0。

【解】算法思想：用一根指针 ps2 保存母串当前开始比较的字符。用两根指针 p1 和 p2 分别指向 S1，S2 当前比较的字符。如果 p1 和 p2 当前指示的字符不同，则 ps2 后移一个字符，p2=ps2，p1 则回到 S1 的第一个字符。以上操作在 p1 和 p2 都不为空时循环执行。如果 p1==NULL 同时 ps2!=NULL，则 S1 是 S2 的子串，否则不是。

【算法描述】

```
//子串比较，S2 为母串，S1 为子串
int SubString(node *S2, node *S1)
{
 node *p1,*p2;
 node *p2s;   //S2 的搜索起点
 p1=S1->next;
 p2=S2->next;
 p2s=S2->next;
 while(p2 && p1)
 {
      if(p1->data==p2->data)
      {
```

```
            p1=p1->next;
            p2=p2->next;
        }
        else
        {
            p2s=p2s->next;   //S2 搜索起点后移一个字符
            p2=p2s;          //S2 当前指针回到搜索起点
            p1=S1->next;     //S1 回到第一个字符
        }
    }
    if(p2s)
        return 1;      //S1 是 S2 的子串
    else
        return 0;      //S1 不是 S2 的子串
}
```

4.3 设计算法以比较链串 S1 和链串 S2 的大小，若 S1<S2，返回-1；若 S1＝S2，返回 0；否则返回 1。

【解】算法思想：用两根指针 p1 和 p2 分别指向两个串中的字符。从第一个字符开始比较，字符相等两个指针同时后移一个字符，如果两串同时结束，则相等。如果出现对应字符不同，根据字符的 ASCII 码值，返回 1 或-1。当一个串结束，另一个串未结束情况，未结束的串大。

【算法思想】

```
//链串比较，S1>S2 返回 1；S1==S2 返回 0；S1<S2 返回-1
int linkedStrComp(node *S1, node *S2)
{
    node *p1,*p2;
    p1=S1->next;
    p2=S2->next;
    while(p2 && p1)
    {
        if(p1->data==p2->data)
        {
            p1=p1->next;
            p2=p2->next;
        }
        else if(p1->data>p2->data)
            return 1;
        else
            return -1;
    }
```

```
 if(p1==NULL && p2==NULL)            //同时结束，相等，返回 0
       return 0;
 else if(p1!=NULL && p2==NULL)     //S1 未结束，S2 结束
       return 1;
 else
       return -1;
}
```

4.4 已知数组 A[n,n]是对称的，完成下列任务：

① 设计算法将 A[n,n]中的下三角中的各元素按行优先次序存储到一维数组 B 中。

② 对任意输入的 A 数组中的元素的下标 i，j，求解出该元素在 B 中的存储位置。

【解】

① 对称矩阵 A[][]特点 $a_{ij}=a_{ji}$，行优先存储下三角元素 a_{ij} 在一维数组 B[]中的下标为 1+2+3+...+...+i+j=i(i+1)/2+j，数组下标从 0 开始。

【算法描述】

```
void rowPriToB(elementType A[][,elementType B[],int n)
{                    //n 为二维数组的大小
       int i,j;
       for(i=0;i<n;i++)
       {
            for(j=0;j<n;j++)
            {
                 if(i>=j)            //存储下三角元素到一维数组
                      B[i*(i+1)/2+j]=A[i][j];
            }
       }
}
```

② 给出二维数组元素下标 i，j 从一位数组 B[]中取出 A[i][j]，如果 i>=j，取出 B[i*(i+1)+j]即可。如果 j>i，交换 i 和 j 的值，变为取下三角对称的元素 A[j][i]。

【算法描述】

```
elementType getElement(elementType B[],int i,int j)
{
       //i,j 二维矩阵中元素的下标
 int k;
 if(j>i)     //对称矩阵上三角元素，改为取下三角对称元素
 {
       k=j;
       j=i;
       i=k;
 }
```

```
 k=i*(i+1)/2+j;
 return B[k];
}
```

4.5 已知数组 A[n,n]的上三角部分的各元素均为同一个值 v0，，完成下列任务：

① 设计算法将 A[n,n]中的下三角中的各元素按行优先次序存储到一维数组 B 中，并将 v0 存放到其后面。

② 对任意输入的 A 数组中的元素的下标 i，j，求解出该元素在 B 中的存储值。

【解】

① 下三角存放方式同上题，在 B[n*(n+1)/2]存放上三角元素 v0，即 B[]最后放 v0 元素。数组下标从 0 开始。

【算法描述】

```
void rowPriToB(elementType A[][],elementType B[MAXLEN*MAXLEN],int n)
{
    int i,j;
    for(i=0;i<n;i++)
    {
        for(j=0;j<n;j++)
        {
            if(i>=j)
                B[i*(i+1)/2+j]=A[i][j];
        }
     }
    B[n*(n+1)/2]=A[0][1];    //任取一个上三角元素 v0 存到 B[]数组最后
}
```

② 当 i>=j 时，按 i*(i+1)/2+j 正常取值。否则 j>i 时，取 B[]最后元素 v0，即取 B[n*(n+1)/2]。

【算法描述】

```
elementType getElement(elementType B[],int i,int j,int n)
{
    //i,j 二维矩阵中元素的下标。n 为二维数组的大小
 if(i>=j)    //正常取下三角元素
 {
    return B[i*(i+1)/2+j];
 }
 else
    return B[n*(n+1)/2];    //上三角元素皆为 B[]数组最后的元素 v0
}
```

4.6 对两个以三元组形式存储的同阶稀疏矩阵 A,B，设计算法求 C=A+B。

4.7 已知广义表 L=(a,(b,c,d),c),运用 head 和 tail 运算组合，取出原子 d 的运算是什么？

【解】

head(tail(tail(head(tail(L)))))

4.8 广义表(a,(a,b),d,e,((i,j),k))的长度是多少？

【解】6

4.9 广义表与线性表的主要区别是什么？

【答】线性表的元素都是相同类型的原子；广义表的元素既可以是原子，也可以是子表。广义表是线性表的扩展。

4.10 求下列广义表操作的结果：

① tail[head[((a,b),(c,d))]];

② tail[head[tail[((a,b),(c,d))]]]

【解】

① tail[head[((a,b),(c,d))]]=(b);

② tail[head[tail[((a,b),(c,d))]]]=(d)

4.11 利用广义表的 head 和 tail 操作写出如上题的函数表达式,把原子 banana 分别从下列广义表中分离出来.

① L =(((apple)),((pear)),(banana),orange);

② L =(apple,(pear,(banana),orange));

【解】

① head(head(tail(tail(L))));

② head(head(tail(head(tail(L)))));