

## DS Homework 2

注：请使用 A4 纸作答，写上姓名学号，并于下一次上课时提交。

1、请从存储空间、运算时间和程序设计语言这三个角度，给出线性表的顺序存储和链式存储结构选择的原则。

存储空间。顺序存储结构要求事先分配其存储空间（即静态分配），所以难以估计存储空间大小，估计过大会浪费存储空间，估计过小会造成存储空间溢出。链式存储结构的存储空间是动态分配的，只要机器中又空闲空间，就不会造成存储空间的益处。另外，还可以从存储密度的角度考虑。一般来说，存储密度越大，存储空间的利用率越高。显然，顺序存储结构的存储密度为 1，而链式存储结构的存储密度小于 1。

运算时间。顺序存储结构是已知的随机存储结构，便于元素的随机存储，即表中每个元素都可以在  $O(1)$  时间复杂度下迅速存取。而链式存储结构中为了访问一个结点，必须从头指针开始顺序查找，时间复杂度为  $O(n)$ ，故对于只进行查找操作而很少进行插入和删除操作的情况，采用顺序存储比较合适。

程序设计语言。在不支持指针的概念和指针类型的程序设计语言中，可以采用静态链表（即利用游标实现）的方法来模拟动态存储结构。对于问题规模不大的问题，采用静态链表来实现更加方便。

2、编写一个函数将一个顺序表 A（有多个元素并且任何元素不为 0）分拆成两个顺序表。使 A 中大于 0 的元素存放在 B 中，小于 0 的 存放在 C 中。

```
void split(Sqlist A, Sqlist &B, Sqlist &C)
```

```
{// 拆分 A，将大于 0 的元素存放在 B 中，小于 0 的元素存放在 C 中
```

```
    int i;
```

```
    B.length = C.length = 0;
```

```
    for (i = 0; i < A.length; i++)
```

```
    {    if (A.elem[i] > 0)
```

```
        {    B.elem[B.length] = A.elem[i];
```

```
            B.length++;
```

```
        }
```

```
    else
```

```
        {    C.elem[C.length] = A.elem[i];
```

```
            C.length++;
```

```
        }
```

```
    }
```

```
}
```

3、已知一个顺序存储的线性表的元素非递减有序排列，请用类 C/C++伪代码，编写删除表中多余的值相同元素的算法（自定义变量类型）。

由题已知，值相同的元素必为相邻的元素，为了提高元素删除效率，保留的元素只能移动一次，为此设置两个指针 i 和 j，i 用来表示前面没有相同元素的最后位置，j 表示当前位置，若 j 与 i 位置元素不同，则指针 i 位置后移，并将指针 j 所指元素赋给位置 i；同时 j 后移一个位置。参考实现如下：

```
void dele(SqList &L)
{
    int i = 0, j = 1;
    while(j < L.length)
    {
        if (L.elem[j] != L.elem[i])
            L.elem[++i] = L.elem[j];
        j++;
    }
    L.length = i + 1;
}
```