

数据结构第三章

1. 解释术语：

1. 线性表：一个线性表是 $n \geq 0$ 个数据元素 $a_1, a_2, a_3, \dots, a_n$ 的有限序列，序列中除第一及最后一个元素以外，每个元素有且只有一个直接前驱和直接后继。
2. 链表：通过指针联系起来的节点的整体
3. 表头结点：给链表增加的一个附加结点，该结点一般放置于链表的最前面，但该结点不用以存储数据元素
4. 静态链表：以整型变量的值作为存储链接的指针值，从而联系起来的节点的整体

2. 设集合A,B分别用向量来表示，试写出实现集合的差运算 $C = A - B$ 的算法。 (提示:C由所有属于A且不属于B的元素构成)

```
PROC SetDelete(Var A,B,C:SQLIST;var a:0..maxa;b:0..maxb;var
c,m,n:Integer)
BEGIN
  c←0;
  m←0;
  n←0;
  WHILE(m<=a)
    [ WHILE (n<=b AND A[m]≠B[n])
      [ n←n+1; ]
    IF n>b //A有B没有
      [ C[c]=A[m];
        c←c+1;]
    ELSE //AB都有
      [ m←m+1;]
    ]
  ]
```

3.试写出将一个采用向量结构存储的线性表A的数据元素进行逆置的算法

```
PROC ListReverse(VAR A:SQLIST; var n:0..max; var x:Integer; var
tmp:datatype)
BEGIN
  FOR x←0 TO (n-(n%2))/2 DO
    [ tmp ← A[x];
      A[x]←A[n-x];
      A[n-x] ←tmp;]
  END
```

4. 试写出将一个采用单链表存储的线性表A的数据元素进行逆置的算法

```
PROC InsLinkList(VAR A,B,C:Link)
BEGIN
  B↑←A↑.next;
  A↑.next←nil;
  WHILE B↑ ≠ nil DO
    [C↑←B↑.next;
      B↑.next←A↑.next;
      A↑.next←B↑;
      B↑←C↑;]
  END
```

5. 设A和B是两个升序有序的循环链表， P_a 和 P_b 分别指向两个表的表头结点，试写出一个将这两个表归并为一个有序的循环链表的算法

```
PROC CombineList(VAR A,B:link)
BEGIN
    Pa←A↑.next;
    Pb←B↑.next;
    NEW(C);
    Pc←C;
    REPEAT
        [ IF (Pa↑.data>Pb↑.data) THEN [ NEW(X);
                                         X↑.data←Pb↑.data;
                                         Pb←Pb↑.next]
        ELSE [ NEW(X);
              X↑.data←Pa↑.data;
              Pa←Pa↑.next]

        X↑.next←C;
        Pc↑.next←X;
        Pc↑←Pc.next;]
    UNTIL (Pa↑.data<Pc↑.data OR Pb↑.data<Pc↑.data)
END
```