线性表是一种最简单的线性结构。**线性结构的特点为：**

在数据元素的非空有限集中，

1．集合中存在唯一的一个“第一元素”；

2．集合中存在唯一的一个 “最后元素” ；

3．除最后元素在外，均有 唯一的后继；

4．除第一元素之外，均有 唯一的前驱。

2.1 线性表的类型定义

抽象数据类型**线性表**的定义如下：

ADT List {

数据对象：

$D = \{ a_i \mid a_i \in ElemSet, i=1,2,...,n, \ n \geq 0 \}$

{称 **n** 为线性表的**表长；n=0** 时的线性表为**空表**。}

数据关系：

$R1 = \{ <a_{i-1}, a_i> \mid a_{i-1}, a_i \in D, \ i=2,...,n \}$

简言之，一个线性表是n个数据元素（a_1，a_2,…，a_i，…，a_n）的有限序列。称 i 为 a_i 在线性表中的**位序**。

基本操作：

结构初始化操作

结构销毁操作

引用型操作

加工型操作

} **ADT** List

初始化操作

InitList(&L)

操作结果：构造一个空的线性表L。

结构销毁操作

DestroyList(&L)

初始条件： 线性表 L 已存在。

操作结果： 销毁线性表 L。

引用型操作:

ListEmpty(L)

ListLength(L)

PriorElem(L, cur_e, &pre_e)

NextElem(L, cur_e, &next_e)

GetElem(L, i, &e)

LocateElem(L, e, compare( ))

ListTraverse(L, visit( ))

ListEmpty(L)（线性表判空）

初始条件：　线性表**L**已存在。

操作结果：　若**L**为空表，则返回TRUE，否则FALSE。

ListLength(L)（求线性表的长度）

初始条件：　线性表**L**已存在。

操作结果：　返回**L**中元素个数。

PriorElem(L, cur_e, &pre_e)（求数据元素的前驱）

初始条件：　线性表L已存在。

操作结果：　若cur_e是L的元素，但不是第一个，则用pre_e 返回它的前驱，否则操作失败，pre_e无定义。

NextElem(L, cur_e, &next_e)（求数据元素的后继）

初始条件：　线性表L已存在。

操作结果：　若cur_e是L的元素，但不是最后一个，则用next_e返回它的后继，否则操作失败，next_e无定义。

GetElem(L, i, &e)（求线性表中某个数据元素）

初始条件：　线性表**L**已存在，且 **1≤i≤LengthList(L)**。

操作结果：　用 e 返回**L**中第 i 个元素的值。

LocateElem(L, e, compare( ))（定位函数）

初始条件：　线性表**L**已存在，e为给定值，compare( )是元素判定函数。

操作结果：　返回**L中第1个**与e**满足**关系compare( )的元素的位序。若这样的元素不存在，则返回值为0。

ListTraverse(L, visit( ))　　（遍历线性表）

初始条件：　线性表**L**已存在，Visit() 为某个访问函数。

操作结果：　**依次**对**L**的每个元素调用函数visit( )。一旦visit( )失败，则操作失败。

加工型操作

ClearList(&L)

PutElem(&L, i, &e)

ListInsert(&L, i, e)

ListDelete(&L, i, &e)

ClearList(&L)　（线性表置空）

初始条件：　线性表L已存在。

操作结果：　将L重置为空表。

PutElem(&L, i, &e)　（改变数据元素的值）

初始条件：　线性表L已存在，且 1≤i≤LengthList(L) 。

操作结果：　L中第i个元素赋值同e的值。

ListInsert(&L, i, e)　（插入数据元素）

初始条件：　线性表**L**已存在， 且 1≤i≤LengthList(L)+1 。

操作结果：　在**L**的第**i**个元素之前**插入**新的元素e，L的长度增1

ListDelete(&L, i, &e)　（删除数据元素）

初始条件：　线性表L已存在且非空，**1≤i≤LengthList(L)** 。

操作结果：　删除L的第i个元素，并用e返回其值，L的长度减1。

例 2-1：假设有两个集合 A 和 B 分别用两个线性表 LA 和 LB 表示即：线性表中的数据元素即为集合中的成员。 现要求一个新的集合A = A∪B。

上述问题可演绎为：

要求对线性表作如下操作：扩大线性表 LA，将**存在于线性表LB 中**而**不存在于线性表 LA 中**的数据元素**插入到线性表 LA 中**去。

操作步骤：

1．从线性表LB中依次察看每个数据元素；

GetElem（LB, i）→e

2．依值在线性表LA中进行查访；

LocateElem（LA, e, equal( )）

3．若不存在，则插入之。

ListInsert(LA, n+1, e)

```
void union(List &La, List Lb) {
    La_len = ListLength(La)；  //求线性表的长度
    Lb_len = ListLength(Lb)；
      for (i = 1;  i <= Lb_len ;   i++) {
        GetElem(Lb, i, e); // 取Lb中第i个数据元素赋给e
        if  ( !LocateElem ( La, e, equal( ) ) )
           ListInsert(La, ++La_len, e)；
         // La中不存在和 e 相同的数据元素，则插入之
      }
 } // union
```

算法：2.1

例2-2：已知线性表LA 和LB中数据元素按值非递减有序排列，现要求将LA 和LB归并为一个新的线性表LC，且LC中的数据元素仍按值非递减有序排列

例如：LA=（3，5，8，11），LB=（2，6，8，9，11，15，20），

则 LC=（2，3，5，6，8，8，9，11，11，15，20）

从问题要求可知，LC中的数据元素或是LA中的数据元素，或是LB中的数据元素，则只要先设LC为空表，然后将LA或LB中的数据元素逐个插入到LC中即可。

可设两指针 i 和 j 分别指向LA和LB中的某个元素。若设 i 当前所指元素为 a， j 当前所指元素为 b，则应插入到LC中的元素 c 为：

$$C=\begin{cases} a & \text{当} a<=b \text{ 时} \\ b & \text{当} a>b \text{ 时} \end{cases}$$

```
void MergeList (List La, List Lb, List &Lc) {
 //已知线性表La 和Lb中数据元素按值非递减有序排列，归并La
//和Lb得到新的线性表Lc，Lc中的数据元素也按值非递减排列。
InitList(Lc);  // 构造空的线性表 Lc
i = j = 1;    k = 0;
La_len = ListLength(La);
Lb_len = ListLength(Lb);
while ( ( i <= La_len ) &&  ( j <= Lb_len ) ){
 // La 和 Lb 均非空，i = j = 1, k = 0
    GetElem ( La，i，ai ) ；
    GetElem ( Lb，j，bj ) ；
    if  ( ai <= bj ) {  // 将 ai 插入到 Lc 中
       ListInsert ( Lc，  ++k，  ai ) ；  ++i ；  }
    else {   // 将 bj 插入到 Lc 中
       ListInsert ( Lc，  ++k，  bj ) ；  ++j ；  }
}                        //接下页//
```

```
    while（i <= La_len）{ // 当La不空时
        GetElem（La，i++，ai）；
        ListInsert（Lc，++k，ai）；
    } // 插入 La 表中剩余元素
    while（j <= Lb_len）{ // 当Lb不空时
        GetElem（Lb，j++，bj）；
        ListInsert（Lc，++k，bj）；
    } // 插入 Lb 表中剩余元素
} // merge_list
```

算法：2.2