



本节主题：

线性表的逻辑结构与基本运算

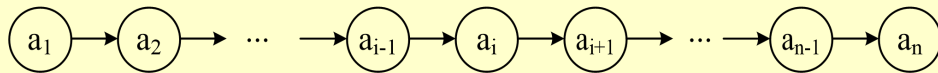
线性表的定义

线性表是具有相同特性的数据元素的一个有限序列。



表示为

$$L = (a_1, a_2, \dots, a_{i-1}, a_i, a_{i+1}, \dots, a_n)$$



线性表的**长度**：序列中所含元素的个数—— n ， $n \geq 0$

空表：当 $n=0$ 时，表示线性表是一个空表，即表中不包含任何元素。

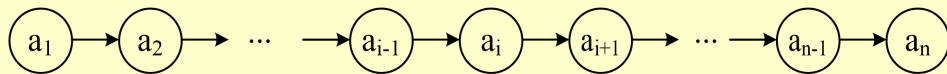
前驱： a_{i-1} 是 a_i 的前驱， $2 \leq i \leq n$

后继： a_{i+1} 是 a_i 的后继， $1 \leq i \leq n-1$

表头元素：表中第一个元素 a_1

表尾元素：最后一个元素 a_n

线性表属于线性结构



□ $L = (D, R)$

□ $D = \{a_i \mid a_i \in \text{ElemType}, i=1,2,\dots,n, n \geq 0\}$ //ElemType为类型标识符

□ $R = \{r\}$

□ $r = \{\langle a_{i-1}, a_i \rangle \mid a_{i-1}, a_i \in D, i=2,3,\dots,n\}$

□ 例

□ 小明大学4年8学期修读的课程数：(6, 7, 7, 9, 7, 6, 4, 1)

□ 一副扑克的点数：(2, 3, 4, ..., J, Q, K, A)

□ 某班同学的基本情况：(('2014414101', '张里户', '男', 06/24/1983),
('2014414102', '王土司', '男', 08/12/1984), ...,
('2014414102', '李麦瑞', '女', 08/12/1984))

ADT

ADT List

{

数据对象：

$D = \{a_i \mid a_i \in \text{ElemType}, i=1,2,\dots,n, n \geq 0\}$ //ElemType为类型标识符

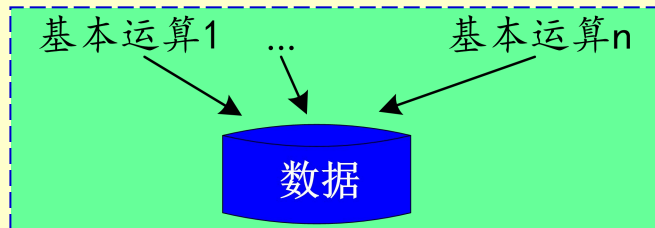
数据关系：

$R = \{\langle a_{i-1}, a_i \rangle \mid a_{i-1}, a_i \in D, i=2,3,\dots,n\}$

数据操作：

- (1) 初始化线性表**InitList(&L)**：构造一个空的线性表L
- (2) 销毁线性表**DestroyList(&L)**：释放线性表L占用的内存空间
- (3) 判线性表是否为空表**ListEmpty(L)**：若L为空表，则返回真，否则返回假
- (4) 求线性表的长度**ListLength(L)**：返回L中元素个数
- (5) 输出线性表**DispList(L)**：当线性表L不为空时，顺序显示L中各节点的值域
- (6) 求线性表L中指定位置的某个数据元素**GetElem(L,i,&e)**：用e返回L中第 i 个元素的值
- (7) 查找元素**LocateElem(L,e)**：返回线性表L中第1个与e相等的序号，找不到返回0
- (8) 插入元素**ListInsert(&L, i, &e)**：在线性表L中的第i个位置插入元素e；
- (9) 删除元素**ListDelete(&L, i, &e)**：在线性表L中删除第i个元素，有e返回删除的值；

}



例

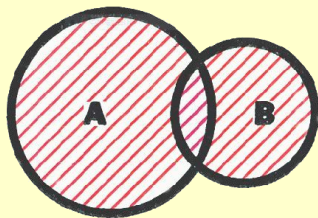
☐ 给定线性表 $L = (1, 3, 1, 8, 2)$

☐ 下列基本运算的结果

操 作	结 果
ListLength(L)	5
ListEmpty(L)	false
GetElem(L,3,&e)	$e=1$
LocateElem(L,1)	1
ListInsert(L, 4, 5)	L 变成 (1, 3, 1, 5, 8, 2)
ListDelete(L,3, e)	L 变成 (1, 3, 5, 8, 2), e为1

例：线性表的合并

问题



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。

编写一个算法求一个新的集合 $C=A \cup B$ ，即将两个集合的并集放在线性表LC中。

解题思路

$LC \leftarrow LA$

$LC \leftarrow LB$ 中不在LA中的元素

算法？

```
void unionList(List LA,List LB,List &LC)
{
    int lena,i;
    ElemType e;
    InitList(LC);
    //将LA的所有元素插入到LC中
    for (i=1; i<=ListLength(LA); i++)
    {
        GetElem(LA,i,e);
        ListInsert(LC,i,e);
    }
    lena=ListLength(LA);
    //将LB的所有元素插入到LC
    for (i=1; i<=ListLength(LB); i++)
    {
        GetElem(LB,i,e);
        if (!LocateElem(LA,e))
            ListInsert(LC,++lena,e);
    }
}
```

时间复杂度： $O(\text{ListLength}(LA) * \text{ListLength}(LB))$

学习方法探讨：用程序实践算法

```
void unionList(List LA, List LB, List &LC)
{
    int lena, i;
    ElemType e;
    InitList(LC);
    //将LA的所有元素插入到LC中
    for (i=1; i<=ListLength(LA); i++)
    {
        GetElem(LA, i, e);
        ListInsert(LC, i, e);
    }
    lena = ListLength(LA);
    //将LB的所有元素插入到LC
    for (i=1; i<=ListLength(LB); i++)
    {
        GetElem(LB, i, e);
        if (!LocateElem(LA, e))
            ListInsert(LC, ++lena, e);
    }
}
```

实践步骤

❏ 定义数据类型

❏ List

❏ ElemType

❏ 实现基本操作

❏ InitList

❏ ListLength

❏ GetElem

❏ ListInsert

❏ 设计测试函数

❏ main

