第02讲 线性表的顺序表示与实现

六星教育首席架构师：Vico老师
官方助理冰芯老师QQ：1930070991

线性结构的定义：

若结构是非空有限集，则有且仅有一个开始结点和一个终端结点，并且所有结点都最多只有一个直接前趋和一个直接后继。

可表示为：（a_1，a_2，……，a_n）

线性结构表达式：（a_1，　a_2　，……，　　a_n）

线性结构的特点：

① 只有一个首结点和尾结点；
② 除首尾结点外，其他结点只有一个直接前驱和一个直接后继。

简言之，线性结构反映结点间的逻辑关系是 一对一 的

线性结构包括线性表、堆栈、队列、字符串、数组等等，其中，最典型、最常用的是

线性表

教学内容

2.1 线性表的定义和特点

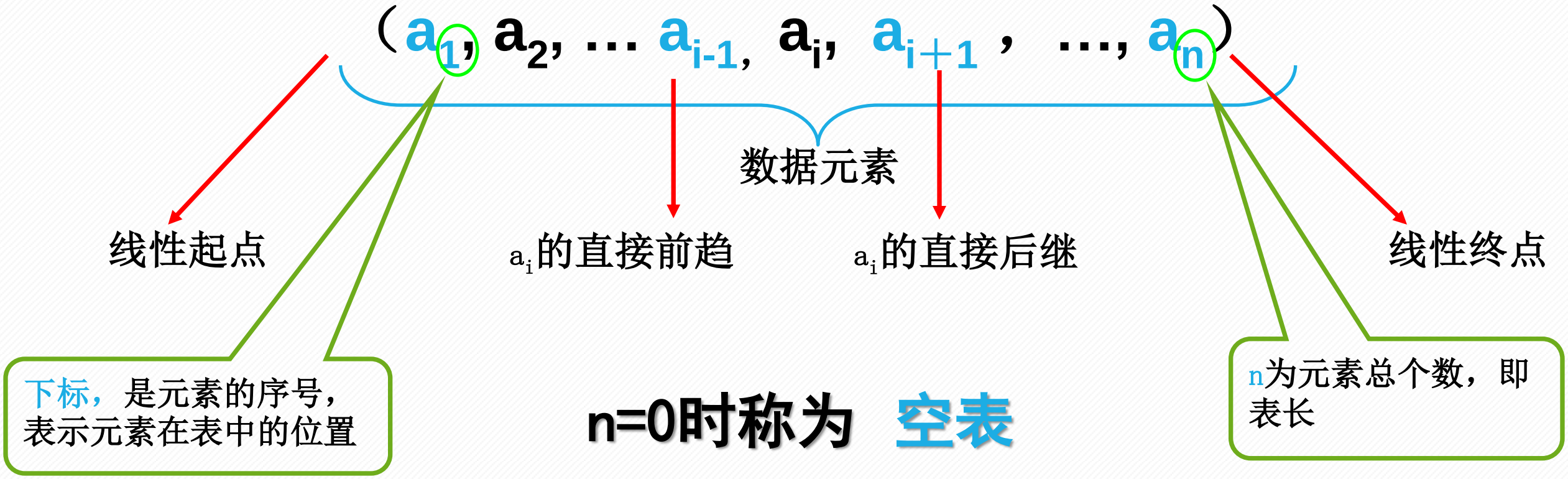
2.2 案例引入

2.3 线性的类型定义

2.4 线性表的顺序表示和实现

2.1 线性表的定义和特点

线性表的定义： 用数据元素的有限序列表示



n=0时称为 空表

例1 分析26 个英文字母组成的英文表

（A，B，C，D，……，Z）

数据元素都是字母；　元素间关系是线性

例2 分析学生情况登记表

学号	姓名	性别	年龄	班级
041810205	于春梅	女	18	04级计算机1班
041810260	何仕鹏	男	20	04级计算机2班
041810284	王　　爽	女	19	04级计算机3班
041810360	王亚武	男	18	04级计算机4班
：	：	：	：	：

数据元素都是记录；元素间关系是线性

同一线性表中的元素必定具有相同特性

2.2 案例引入

案例2.1：一元多项式的运算

$$P_n(x) = p_0 + p_1x + p_2x^2 + \ldots + p_nx^n$$

线性表P = $(p_0,p_1,p_2,\ldots,p_n)$

$$P(x) = 10 + 5x - 4x^2 + 3x^3 + 2x^4$$

数组表示

（每一项的指数i隐含在其系数p_i的序号中）

指数（下标i）	0	1	2	3	4
系数p[i]	**10**	**5**	**-4**	**3**	**2**

案例2.2　稀疏多项式的运算

多项式非零项的数组表示

（a）$A(x) = 7 + 3x + 9x^8 + 5x^{17}$　　（b）$B(x) = 8x + 22x^7 - 9x^8$

下标i	0	1	2	3
系数a[i]	7	3	9	5
指数	0	1	8	17

下标i	0	1	2
系数b[i]	8	22	-9
指数	1	7	8

$$P_n(x) = p_1 x^{e_1} + p_2 x^{e_2} + \ldots + p_m x^{e_m}$$

线性表P =($(p_1, e_1), (p_2, e_2),\ldots,(p_m, e_m)$)

- 创建一个新数组c
- 分别从头遍历比较a和b的每一项
  - ✓指数相同，对应系数相加，若其和不为零，则在c中增加一个新项
  - ✓指数不相同，则将指数较小的项复制到c中
- 一个多项式已遍历完毕时，将另一个剩余项依次复制到c中即可

●顺序存储结构存在问题

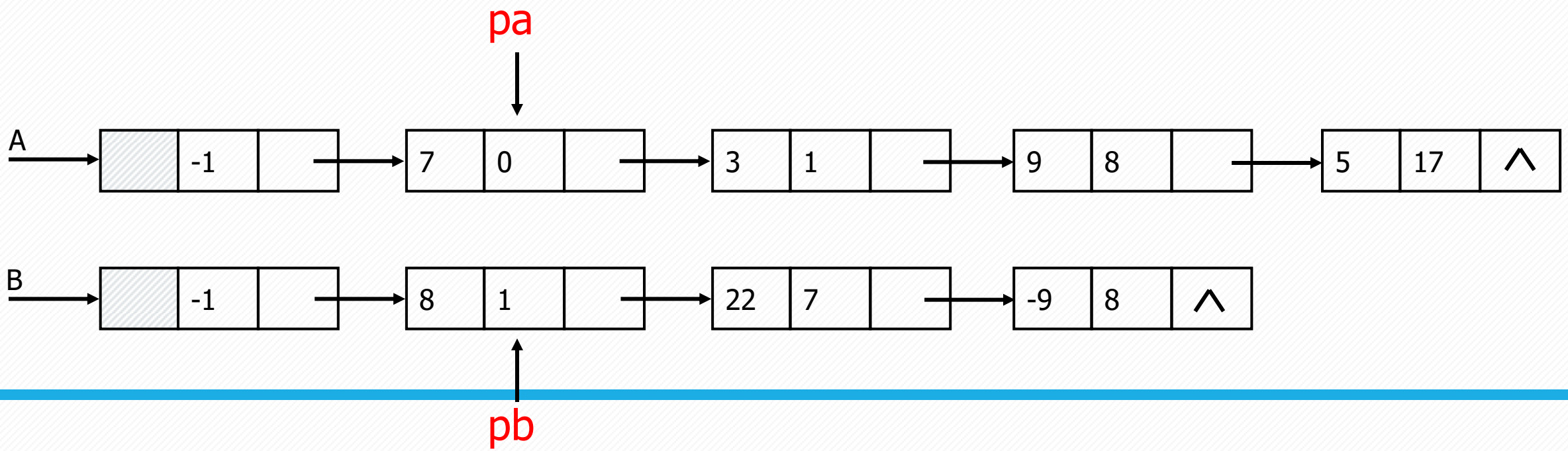
✓存储空间分配不灵活

✓运算的空间复杂度高

链式存储结构

$A_{17}(x)=7+3x+9x^8+5x^{17}$

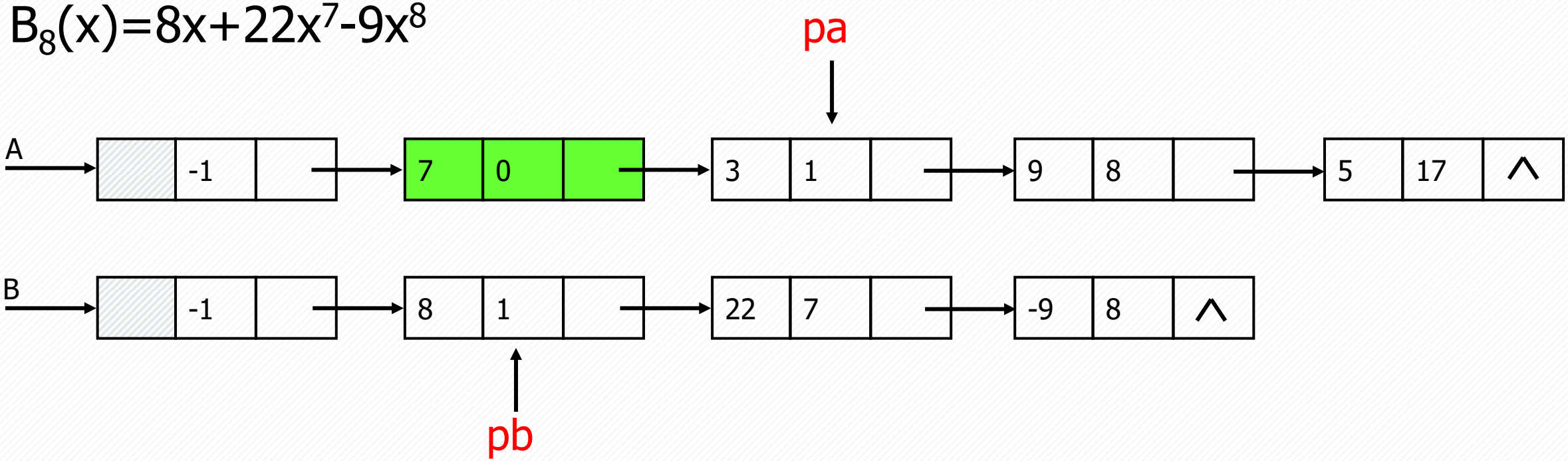
$B_8(x)=8x+22x^7-9x^8$



多项式相加

$A_{17}(x)=7+3x+9x^8+5x^{17}$

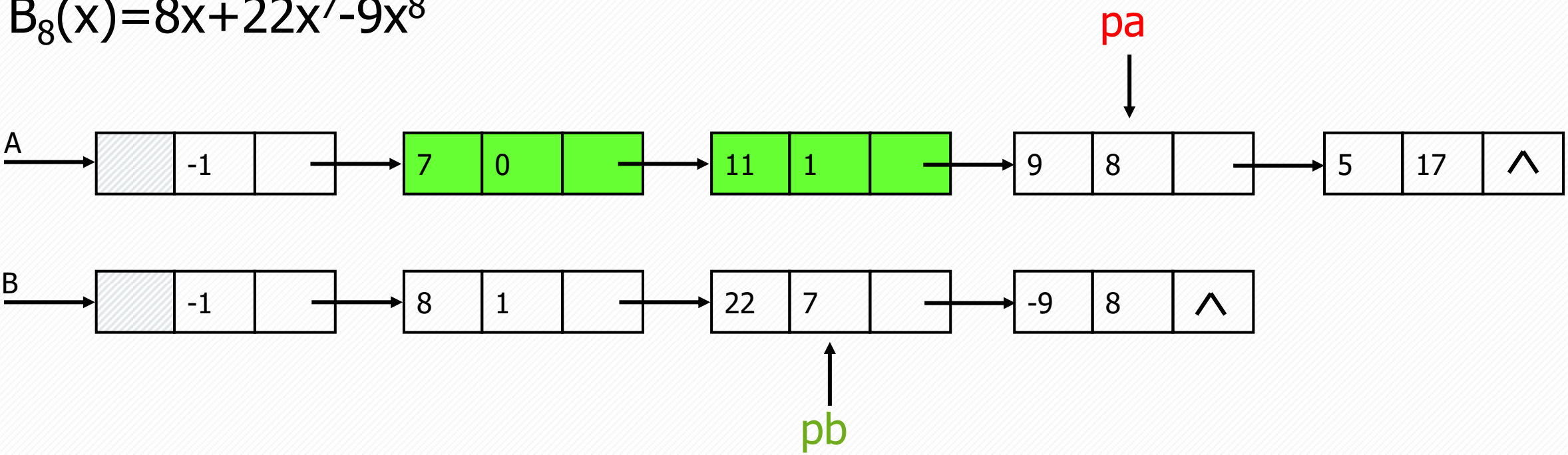
$B_8(x)=8x+22x^7-9x^8$



多项式相加

$A_{17}(x)=7+3x+9x^8+5x^{17}$

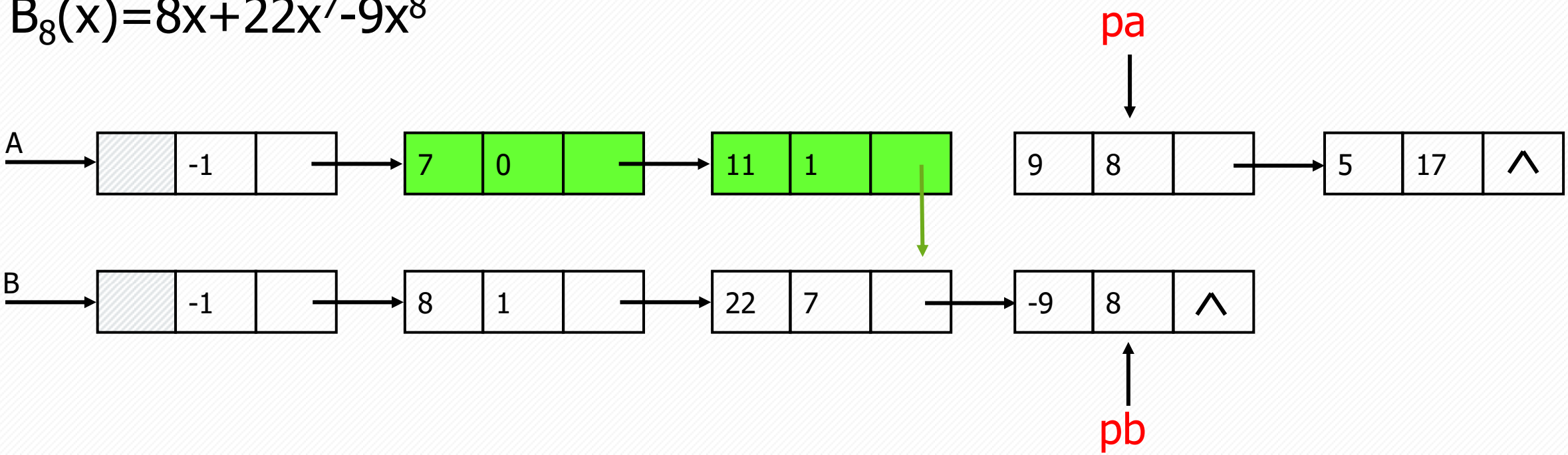
$B_8(x)=8x+22x^7-9x^8$



多项式相加

$A_{17}(x)=7+3x+9x^8+5x^{17}$

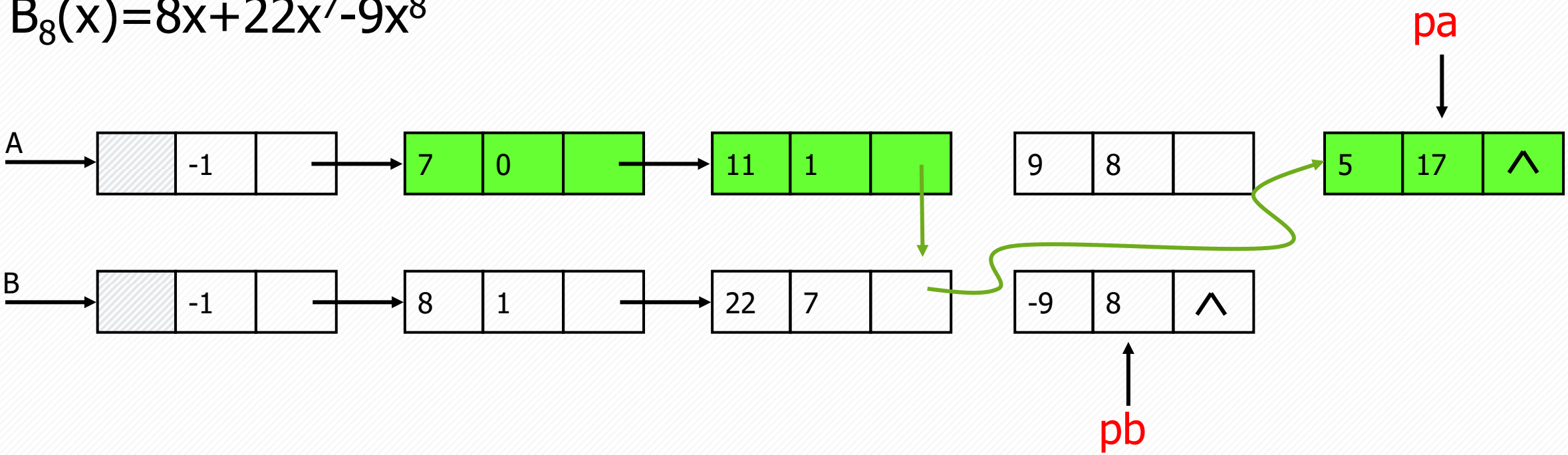
$B_8(x)=8x+22x^7-9x^8$



多项式相加

$A_{17}(x)=7+3x+9x^8+5x^{17}$

$B_8(x)=8x+22x^7-9x^8$



案例2.3　图书信息管理系统


```
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*                                                                         *
*     1. 建立          2. 输入          3. 取值          4. 查找          *
*                                                                         *
*     5. 插入          6. 删除          7. 输出          0. 退出          *
*                                                                         *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
请选择:1
成功建立顺序表

请选择:2
输入 book.txt 信息完毕
```

```
请选择:7
当前图书系统信息（顺序表）读出:
9787302257646    程序设计基础                                    25
9787302219972    单片机技术及应用                                32
9787302203513    编译原理                                        46
9787811234923    汇编语言程序设计教程                            21
9787512100831    计算机操作系统                                  17
9787302265436    计算机导论实验指导                              18
9787302180630    实用数据结构                                    29
9787302225065    数据结构（C语言版）                             38
9787302171676    C#面向对象程序设计                              39
9787302250692    C语言程序设计                                   42
9787302150664    数据库原理                                      35
9787302260806    Java编程与实践                                  56
9787302252887    Java程序设计与应用教程                          39
9787302198505    嵌入式操作系统及编程                            25
9787302169666    软件测试                                        24
9787811231557    Eclipse基础与应用                               35
```

book.txt - 记事本

文件(F)　编辑(E)　格式(O)　查看(V)　帮助(H)

ISBN	书名	定价
9787302257646	程序设计基础	25
9787302219972	单片机技术及应用	32
9787302203513	编译原理	46
9787811234923	汇编语言程序设计教程	21
9787512100831	计算机操作系统	17
9787302265436	计算机导论实验指导	18
9787302180630	实用数据结构	29
9787302225065	数据结构（C语言版）	38
9787302171676	C#面向对象程序设计	39
9787302250692	C语言程序设计	42
9787302150664	数据库原理	35
9787302260806	Java编程与实践	56
9787302252887	Java程序设计与应用教程	39
9787302198505	嵌入式操作系统及编程	25
9787302169666	软件测试	24
9787811231557	Eclipse基础与应用	35

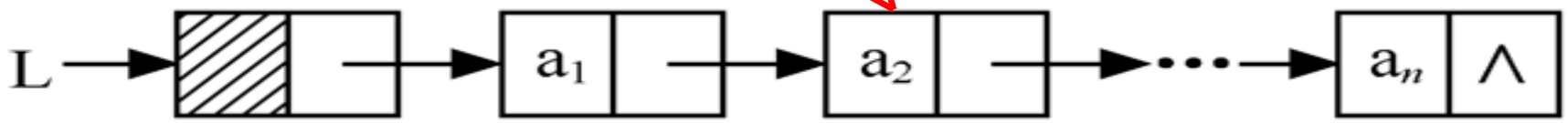
图书顺序表

空闲区

elem[0]	elem[1]	elem[2]	…	elem[length-1]	
a_1	a_2	a_3	…	a_{length}	

ISBN	书名	价格

图书链表



●线性表中数据元素的类型可以为简单类型，也可以为复杂类型。

●许多实际应用问题所涉的基本操作有很大相似性，不应为每个具体应用单独编写一个程序。

●从具体应用中抽象出共性的逻辑结构和基本操作（抽象数据类型），然后实现其存储结构和基本操作。

2. 3 线性表的类型定义

线性表的重要基本操作

1. 初始化
2. 取值
3. 查找
4. 插入
5. 删除

2.4 线性表的顺序表示和实现

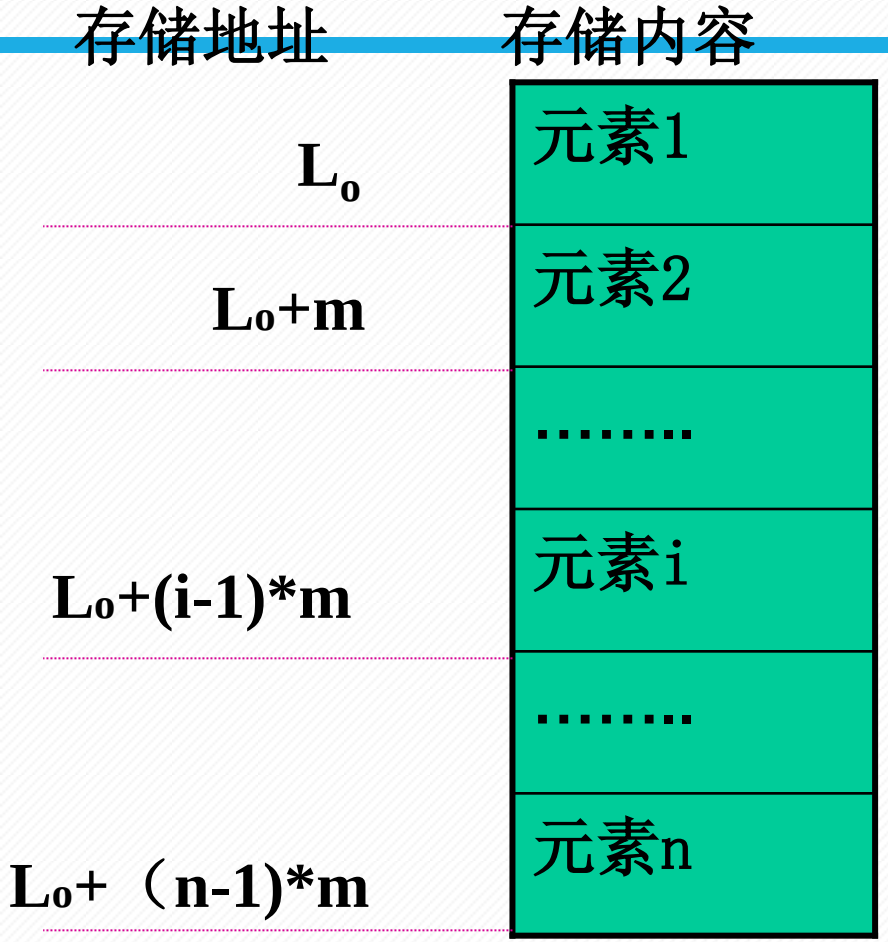
线性表的顺序表示又称为顺序存储结构或顺序映像。

顺序存储定义：把*逻辑上相邻*的数据元素存储在*物理上相邻*的存储单元中的存储结构。

简言之，逻辑上相邻，物理上也相邻

顺序存储方法：用一组地址连续的存储单元依次存储线性表的元素，可通过数组V[n]来实现。

顺序存储



Loc(元素i)=Lo+（i-1)*m

顺序表的类型定义


```
#define  MAXSIZE 100              // 最大长度
typedef  struct {
   ElemType  *elem;               // 指向数据元素的基地址
   int  length;                   // 线性表的当前长度
 }SqList;
```

图书表的顺序存储结构类型定义


```
#define MAXSIZE 10000          // 图书表可能达到的最大长度
typedef struct                 // 图书信息定义
{
   char no[20];                // 图书ISBN
   char name[50];              // 图书名字
   float price;                // 图书价格
}Book;
typedef struct
{
   Book *elem;                 // 存储空间的基地址
   int length;                 // 图书表中当前图书个数
}SqList;                       // 图书表的顺序存储结构类型为SqList
```

线性表的重要基本操作

线性表的重要基本操作

1 初始化

2 取值

3 查找

4 插入

5 删除

重要基本操作的算法实现

1、初始化线性表L （参数用引用）

```
Status InitList_Sq(SqList &L){                  //构造一个空的顺序表L
    L.elem=new ElemType[MAXSIZE];               //为顺序表分配空间
    if(!L.elem) exit(OVERFLOW);                 //存储分配失败
    L.length=0;                                 //空表长度为0
    return OK;
}
```

1、初始化线性表L （参数用指针）

```
Status InitList_Sq(SqList *L){                    //构造一个空的顺序表L
    L-> elem=new ElemType[MAXSIZE];               //为顺序表分配空间
    if(! L-> elem) exit(OVERFLOW);                //存储分配失败
    L-> length=0;                                 //空表长度为0
    return OK;
}
```

2、销毁线性表L

```
void DestroyList(SqList &L)
{
   if (L.elem) delete[]L.elem;          //释放存储空间
}
```

3、清空线性表L

```
void ClearList(SqList &L)
{
   L.length=0;                          //将线性表的长度置为0
}
```

4、求线性表L的长度

```
int GetLength(SqList L)
{   return (L.length);
}
```

5、判断线性表L是否为空

```
int IsEmpty(SqList L)
{  if (L.length==0) return 1;
   else return 0;
}
```

线性表的重要基本操作

线性表的重要基本操作

1 初始化

2 取值

3 查找

4 插入

5 删除

2. 取值
（根据位置i获取相应位置数据元素的内容）

获取线性表L中的某个数据元素的内容

```
int GetElem(SqList L,int i,ElemType &e)
{   if (i<1||i>L.length) return ERROR;
    //判断i值是否合理，若不合理，返回ERROR
    e=L.elem[i-1];    //第i-1的单元存储着第i个数据
    return OK;
}
```

随机存取

3. 查找（根据指定数据获取数据所在的位置）

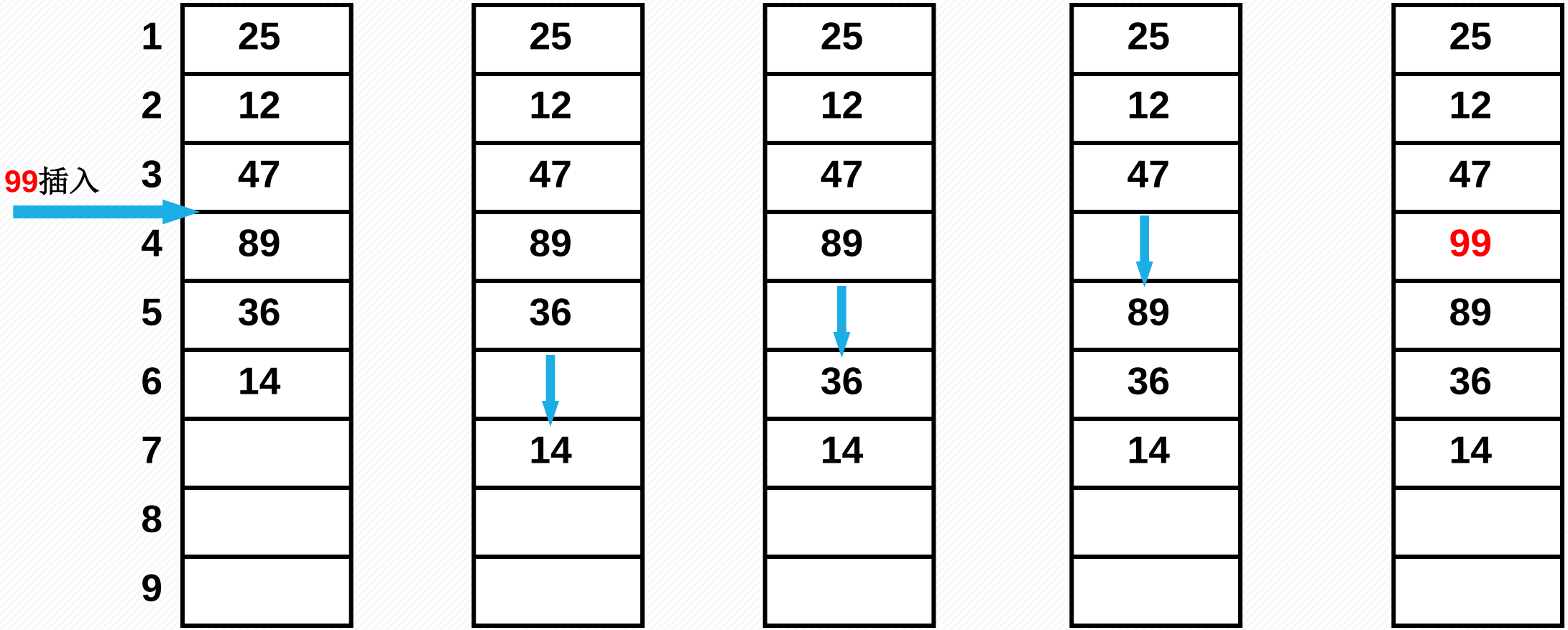
顺序查找图示

3. 查找（根据指定数据获取数据所在的位置）

在线性表L中查找值为e的数据元素

```
int LocateELem(SqList L,ElemType e)
{
  for (i=0;i< L.length;i++)
      if (L.elem[i]==e) return i+1;
  return 0;
}
```

4. 插入（插在第i个结点之前）



插第 4 个结点之前，移动 6－4+1 次

插在第 i 个结点之前，移动 n-i+1 次

【算法步骤】

（1）判断插入位置 i 是否合法。

（2）判断顺序表的存储空间是否已满。

（3）将第n至第i 位的元素依次向后移动一个位置，空出第i个位置。

（4）将要插入的新元素e放入第i个位置。

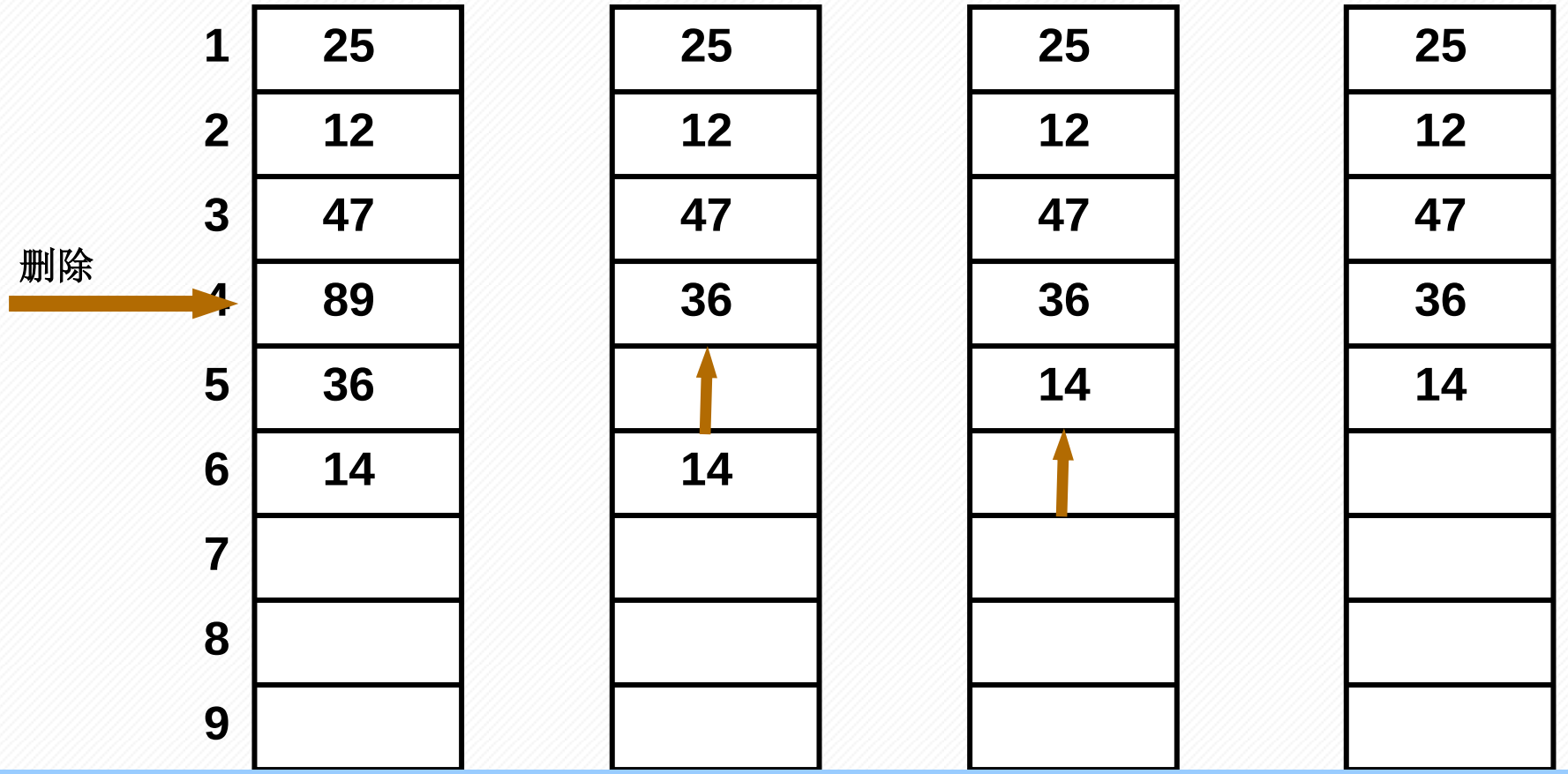
（5）表长加1，插入成功返回OK。

【算法描述】

4、在线性表L中第i个数据元素之前插入数据元素e

```
Status ListInsert_Sq(SqList &L,int i ,ElemType e){
   if(i<1 || i>L.length+1) return ERROR;          //i值不合法
   if(L.length==MAXSIZE) return ERROR;        //当前存储空间已满
   for(j=L.length-1;j>=i-1;j--)
       L.elem[j+1]=L.elem[j];     //插入位置及之后的元素后移
    L.elem[i-1]=e;                    //将新元素e放入第i个位置
    ++L.length;                       //表长增1
  return OK;
}
```

5. 删除（删除第 i 个结点）



删除第 4 个结点，移动 6－4 次

删除第 i 个结点，移动 n-i 次

【算法步骤】

（1）判断删除位置i 是否合法（合法值为1≤i≤n）。

（2）将欲删除的元素保留在e中。

（3）将第i+1至第n 位的元素依次向前移动一个位置。

（4）表长减1，删除成功返回OK。

【算法描述】

5、将线性表L中第i个数据元素删除

```
Status ListDelete_Sq(SqList &L,int i){
   if((i<1)||(i>L.length)) return ERROR;      //i值不合法
   for (j=i;j<=L.length-1;j++)
         L.elem[j-1]=L.elem[j];                    //被删除元素之后的元素前移
   --L.length;                                     //表长减1
  return OK;
}
```

顺序表（顺序存储结构）的特点

（1）利用数据元素的存储位置表示线性表中相邻数据元素之间的前后关系，即线性表的逻辑结构与存储结构一致

（2）在访问线性表时，可以快速地计算出任何一个数据元素的存储地址。因此可以粗略地认为，访问每个元素所花时间相等

这种存取元素的方法被称为**随机存取法**

顺序表的优缺点

优点：

- ✓**存储密度大**（结点本身所占存储量/结点结构所占存储量）
- ✓可以**随机存取**表中任一元素

缺点：

- ✓在插入、删除某一元素时，需要移动大量元素
- ✓浪费存储空间
- ✓属于静态存储形式，数据元素的个数不能自由扩充

为克服这一缺点 → 链表

CimFAX
传真服务器常用
问答

将来的你

冲刺 加油!

让我们全力以赴

一定会感激现在拼命的自己