

2.4 线性表应用

一元多项式及相加的数学表示

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

可表示为: $P = (p_0, p_1, p_2, \dots, p_n)$

则另对于 $Q_m(x) = q_0 + q_1x + q_2x^2 + \dots + q_mx^m$ 亦有

$Q = (q_0, q_1, q_2, \dots, q_m)$,

不妨设 $m < n$, 则 $R_n(x) = P_n(x) + Q_m(x)$ 可表示为:

$R = (p_0 + q_0, p_1 + q_1, p_2 + q_2, \dots, p_m + q_m, p_{m+1}, \dots, p_n)$

尽管这种数学模型可以抽象为以系数为元素的线性表, 但无论采用顺序表还是链表, 都存在着空间浪费严重的情况。

2.4 线性表应用（续）

一元多项式数据结构抽象

实际数据结构选取方案：

$$P_n(x) = p_1x^{e_1} + p_2x^{e_2} + \dots + p_mx^{e_m}$$

P_i 是指数为 e_i 的项的非零系数，且满足：

$$0 \leq e_1 < e_2 < \dots < e_i < e_m = n$$

则可抽象的线性表为（每个元素是一个二元组）：

$$((p_1, e_1), (p_2, e_2), \dots, (p_m, e_m))$$

2.4 线性表应用（续）

一元多项式的抽象数据结构

ADT Polynomial {

数据对象: $D = \{a_i \mid a_i \text{ 属于 TermSet}, i=1,2,\dots,m, m \geq 0, \text{TermSet 中的每个元素是一个由实数与整数组成的二元组}\}$

数据关系: $R1 = \{ \langle a_{i-1}, a_i \rangle \mid a_{i-1}, a_i \text{ 属于 } D, \text{ 且 } a_{i-1} \text{ 中的指数值} < a_i \text{ 中的指数值}, i=1,2,\dots,n \}$

基本操作:

CreatePolyn(&P,m);

AddPolyn(&Pa,&Pb) // $Pa = Pa + Pb$

MutiptyPolyn(&Pa,&Pb) // $Pa = Pa * Pb$

.....

} ADT Polynomial

2.4 线性表应用（续）

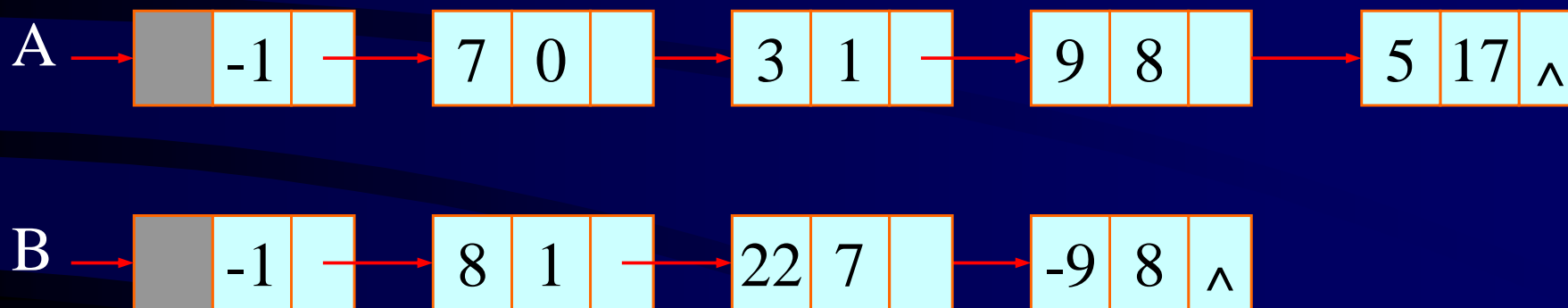
一元多项式的链式存储结构

```
//-----用结构指针描述-----  
  
typedef struct Term{  
    float    coef    //系数数据域  
    int      expn     //指数数据域  
    struct Term *next //指针域  
} Term, *LinkList
```

2.4 线性表应用（续）

一元多项式的链式表示

设 $A_{17}(x) = 7 + 3x + 9x^8 + 5x^{17}$, $B_8(x) = 8x + 22x^7 - 9x^8$

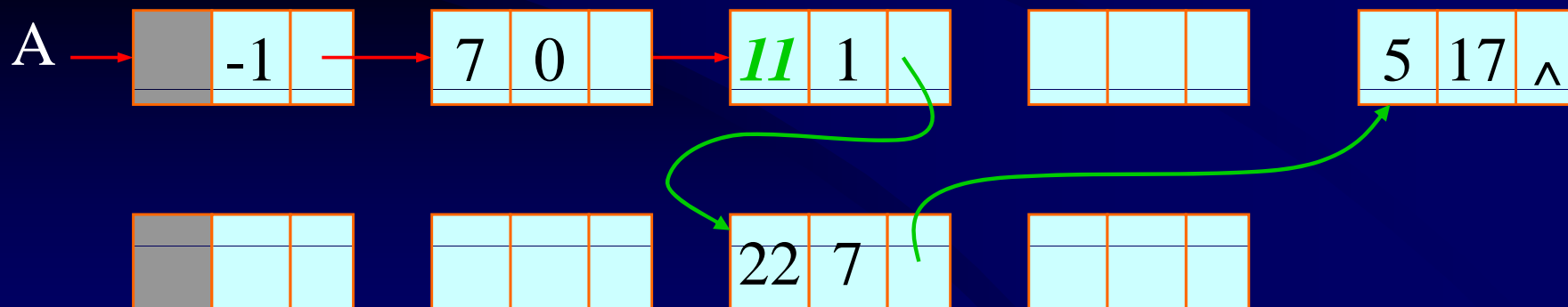


2.4 线性表应用（续）

一元多项式的相加

从逻辑上讲，指数相同的项对应相加，若相加后系数为0，则删除该结点；对于指数不同的项，复抄即可，但须保证和多项式仍然以系数递增排列。从物理实现看，本质上就是结点的插入和删除。

$$A_{17}(x) + B_8(x) = 7 + 11x + 22x^7 + 5x^{17}$$



2.4 线性表应用（续）

一元多项式链式存储的类PASCAL表示

```
//-----用结构指针描述-----  
TYPE  linktp= ^nodetp;  
      nodetp=RECORD  
          coef:  real;    //系数数据域  
          exp:   integer; //指数数据域  
          next:  linktp   //指针域  
      END;  
polytp=linktp;
```

2.4 线性表应用（续）

一元多项式相加的运算规则

“和多项式”中结点无需另生成，可看成是将多项式B加到多项式A上。

运算规则：设 p 和 q 分别指向多项式A和B中某一结点，比较结点中的指数项

- 若 $p^{\wedge}.exp < q^{\wedge}.exp$ ，则 p 结点应是“和多项式”中的一项，令 p 指针向后移；
- 若 $p^{\wedge}.exp > q^{\wedge}.exp$ ，则 q 结点应是“和多项式”中的一项，令 q 结点插入在 p 结点之前，且 q 指针在原来的链表上后移；
- 若 $p^{\wedge}.exp = q^{\wedge}.exp$ ，则将两个结点中的系数相加，当和不为零时修改 p 结点中的系数域，释放 q 结点；反之，“和多项式”中没有此项，从A表中删去 p 结点，同时释放 p 和 q 结点。

PROC add_poly(VAR pa:polytp; pb:polytp);

p:=pa^.next; q:=pb^.next;

pre:=pa; pc:=pa;

WHILE (p!=NIL)AND (q!=NIL) DO

CASE

p^.exp<q^.exp:

[pre:=p; p:=p^.next];

p^.exp=q^.exp:

[x:=p^.coef+q^.coef;

IF x!=0 THEN [p^.coef:=x; pre:=p]

ELSE [pre^.next:=p^.next;dispose(p)];

p:=pre^.next; u:=q; q:=q^.next; dispose(u)]

p^.exp>q^.exp:

[u:=q^.next; q^.next:=p; pre^.next:=q; pre:=q; q:=u]

IF q!= NIL THEN pre^.next:= q;

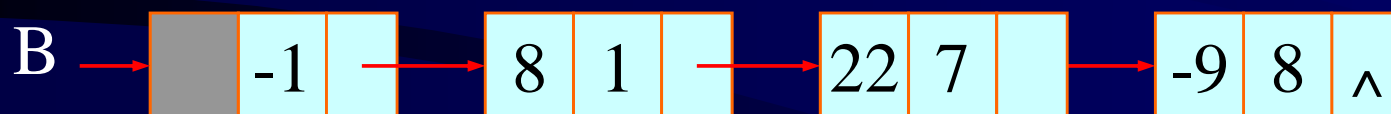
dispose(pb)

ENDP; {add_poly}

2.4 线性表应用（续）

一元多项式的相加

设 $A_{17}(x) = 7 + 3x + 9x^8 + 5x^{17}$, $B_8(x) = 8x + 22x^7 - 9x^8$



$A_{17}(x) + B_8(x) = 7 + 11x + 22x^7 + 5x^{17}$

