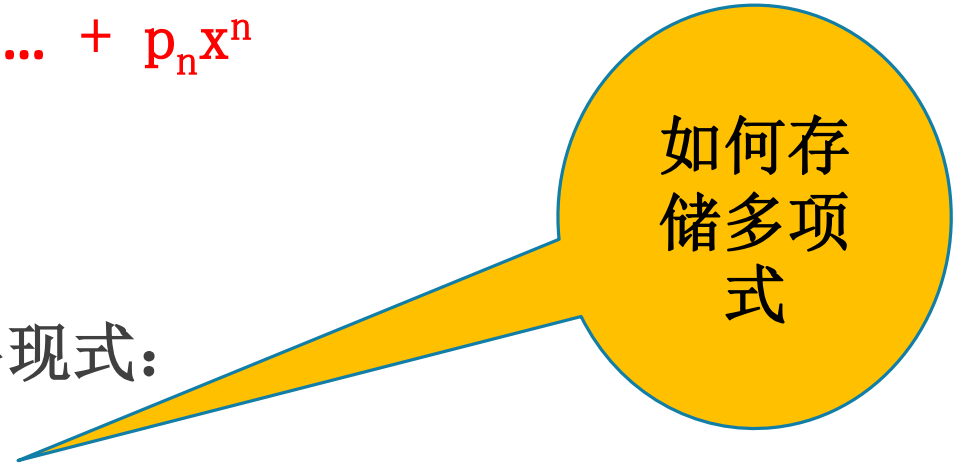
2.11 线性表的应用：一元多项式的加法

在数学上，一个一元多项式$P_n(x)$可按升幂写成：

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

它由 n+1 个系数唯一确定



可用一个线性表 P 来表示多现式：

$$P = (p_0, p_1, p_2, \dots, p_n)$$

每一项的指数 i 隐含在其系数 p_i 的序号里

1、全部系数顺序存储结构

优点：多项式相加的算法定义十分简洁。

缺点：对S型的浪费空间

$$S(x) = 1 + 3x^{10000} + 2x^{20000}$$

就要用一长度为20001的线性表来表示

(表中仅有三个非0元素)

2、非零系数顺序存储结构,必须同时存储相应的指数

$$P_n(x) = p_1x^{e1} + p_2x^{e2} + \ldots + p_nx^{em}$$

p_i 是指数为 e_i 的项的非 0 系数，且满足

$$0 \le e_1 < e_2 < \ldots < e_m = n$$

线性表中的数据元素有两个数据项：系数和指数：

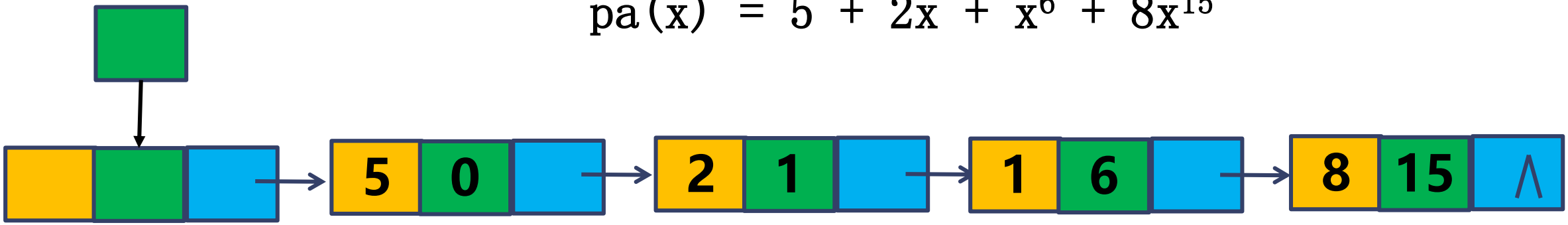
$$((p_1, e_1), (p_2, e_2), \ldots, (p_m, e_m))$$

在**最坏情况**下，n+1 (=m) 个系数都不为 0，则比只存储每项系数的方案要多存储一倍的数据。对于S(x)类的多项式，这种表示将大大节省空间。

3、非零系数单链表存储结构

若只对多项式进行“求值”等不改变多项式的系数和指数的运算，则采用顺序存储结构即可；否则应采用链表表示。

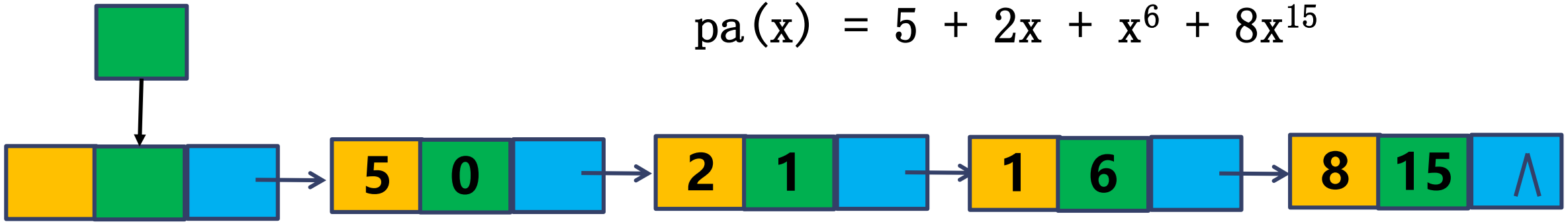
$$pa(x) = 5 + 2x + x^6 + 8x^{15}$$



实现一元多项式的相加运算，显然应采用链式存储结构

2.11 线性表的应用：一元多项式的加法

$$pa(x) = 5 + 2x + x^6 + 8x^{15}$$



$$pb(x) = -2x + 3x^6 + 4x^8$$

2.11 线性表的应用：一元多项式的加法

如何实现用有序链表表示的多项式的加法运算?

根据一元多项式相加的运算规则：

1.对于两个一元多项式中所有指数相同的项，对应系数相加；

2.若其和不为0，则构成和“多项式”中的一项；

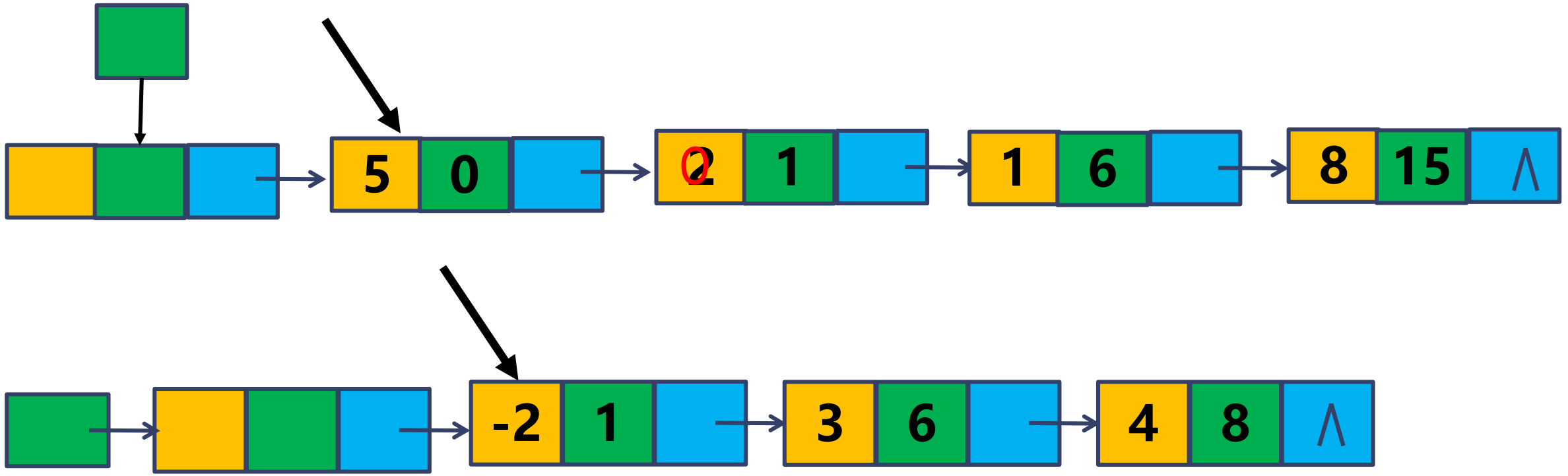
3.对于两个一元多项式中所有指数不同的项，则分别复制到“和多项式”中去；

2.11 线性表的应用：一元多项式的加法

（1）pa->exp< pb->exp：摘去*pa插到“和多项式”链表中

（2）pa->exp= =pb->exp：将pa->coef + pb->coef
若不为0，则修改*pa,释放*pb；若为0，则删除*pa和*pb，并释放*pa和*pb

（3）pa->exp> pb->exp：摘去*pb插到“和多项式”链表中

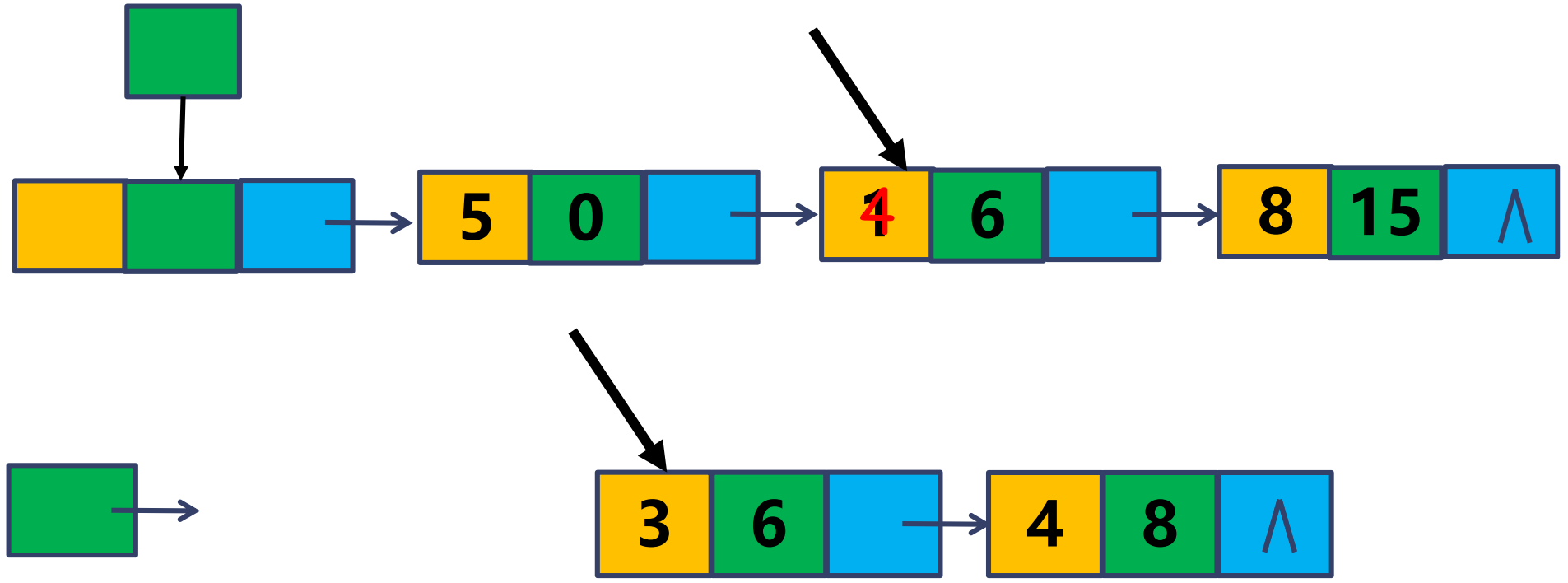


2.11 线性表的应用：一元多项式的加法

（1）pa->exp< pb->exp：摘去*pa插到“和多项式”链表中

（2）pa->exp= =pb->exp：将pa->coef + pb->coef
若不为0，则修改*pa,释放*pb；若为0，则删除*pa和*pb，并释放*pa和*pb

（3）pa->exp> pb->exp：摘去*pb插到“和多项式”链表中

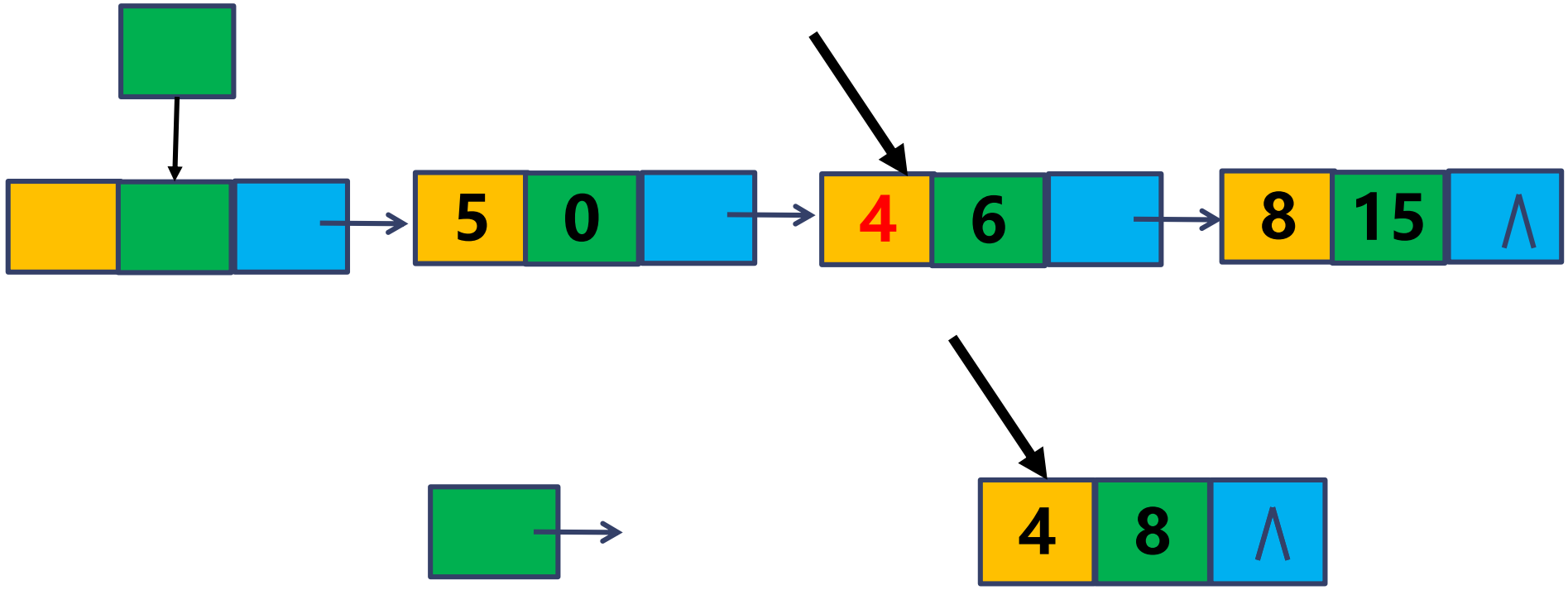


2.11 线性表的应用：一元多项式的加法

（1）pa->exp< pb->exp：摘去*pa插到“和多项式”链表中

（2）pa->exp= =pb->exp：将pa->coef + pb->coef
若不为0，则修改*pa,释放*pb；若为0，则删除*pa和*pb，并释放*pa和*pb

（3）pa->exp> pb->exp：摘去*pb插到“和多项式”链表中

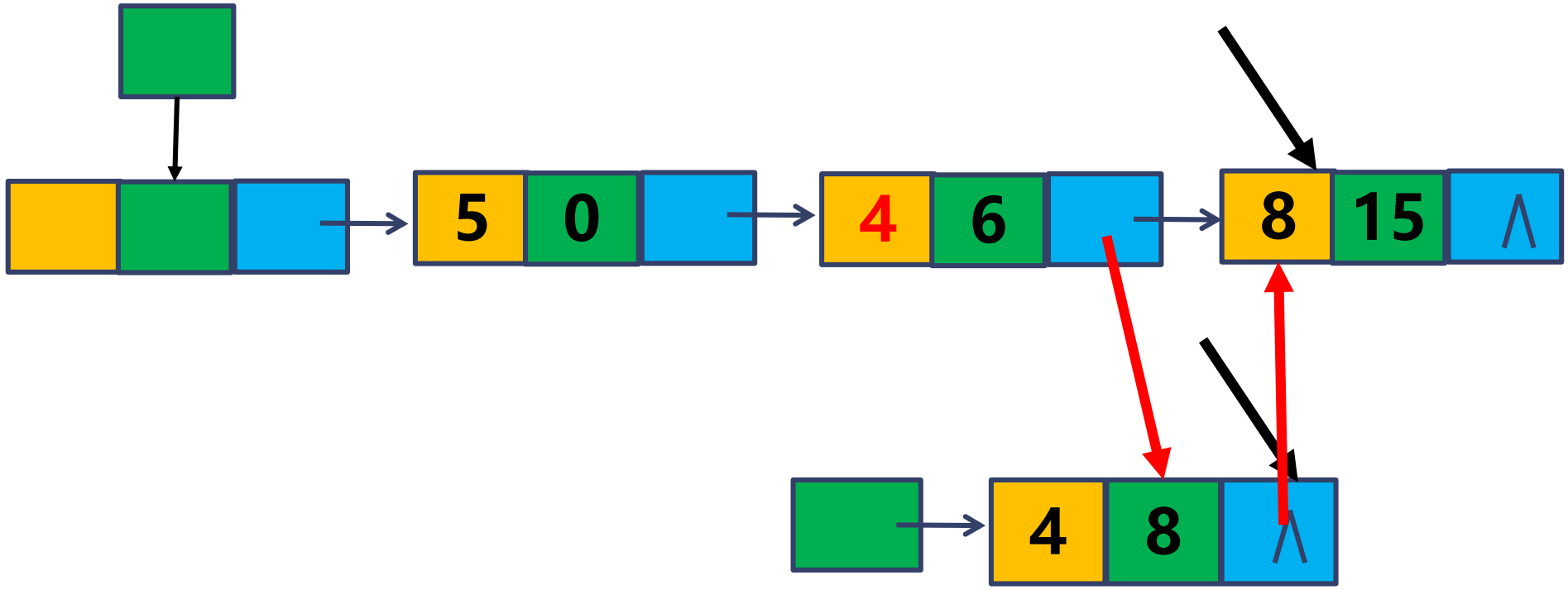


2.11 线性表的应用：一元多项式的加法

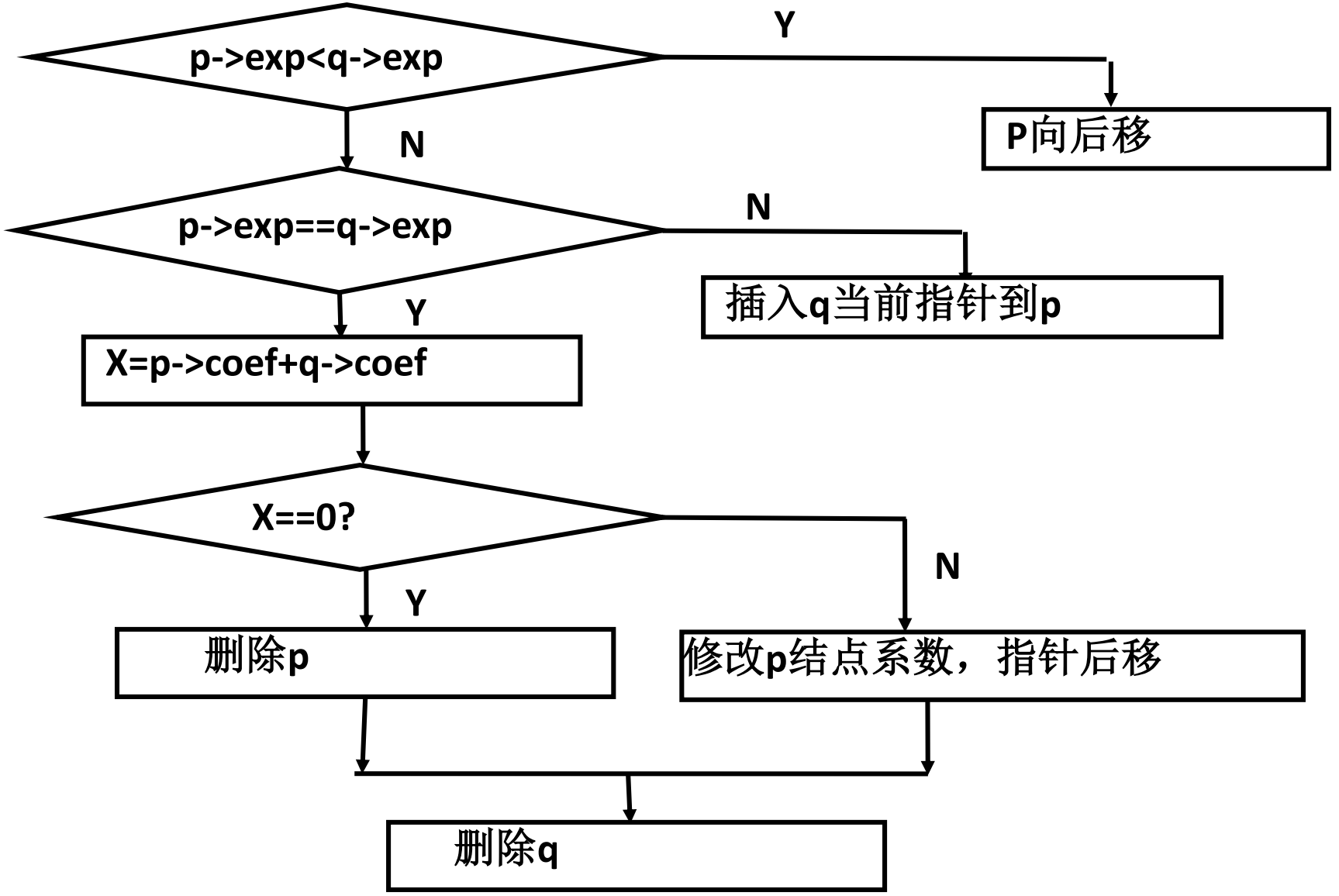
（1）pa->exp< pb->exp：摘去*pa插到“和多项式”链表中

（2）pa->exp= =pb->exp：将pa->coef + pb->coef
若不为0，则修改*pa,释放*pb；若为0，则删除*pa和*pb，并释放*pa和*pb

（3）pa->exp> pb->exp：摘去*pb插到“和多项式”链表中



2.11 线性表的应用：一元多项式的加法



2.11 线性表的应用：一元多项式的加法

```
void AddPolyn(linklist *pa, linklist *pb)
{ p=pa->link;q=pb->link;
while(p!=NULL && q!=NULL))
{ if (p->exp<q->exp) {取p的结点,p后移}
 else {if (p->exp==q->exp) {
        系数相加x= p->coef+q->coef;
        if x==0 删除p当前结点;
         else { 修改p当前结点的系数为x;p后移;}
       删除q当前结点; }
      else 将q当前结点插入到p中;
}
```

```
if(q!=NULL)
{ 将q剩余结点加到和多项式中 }
 else { do nothing? ? ? }
```

多项式加法涉及到的主要算法：

1、建立多项式，考虑加上排序功能

2、多项式的输出

3、多项式的加法