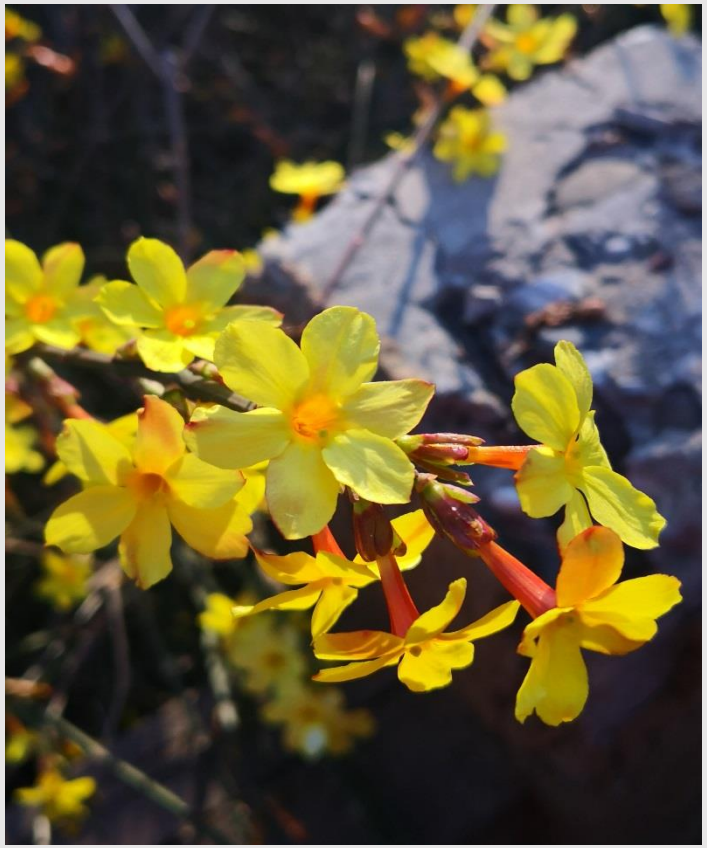
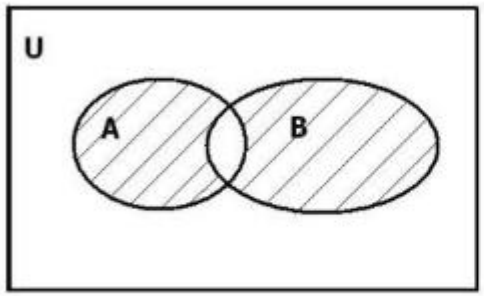
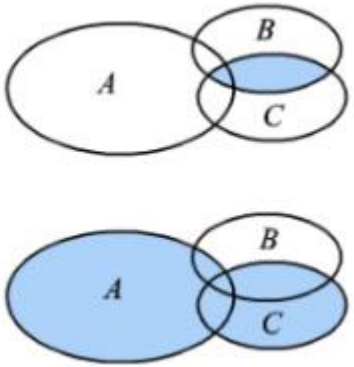


这世上从没有白费的努力，也没有碰巧的成功

3. 线性表的应用

- 集合的表示与运算
- 有序表的归并
- 多项式链表





$(s+1)$
$s^2+1.414s+1$
$(s+1)(s^2+s+1)$
$(s^2+0.7654s+1)(s^2+1.8478s+1)$
$(s+1)(s^2+0.6180s+1)(s^2+1.6180s+1)$

3.1有序表的归并

问题：已知线性表LA和LB中的数据元素按值非递减有序排列，将LA、LB归并成一个新的线性表LC，且LC中的数据元素按值非递减有序排列。

LA=(3，5，8，11)

LB=(2，6，8，9，11，15，20)

LC=(2，3，5，6，8，8，9，11，11，15，20)

顺序表实现

单链表实现

算法思想

while（LA和LB都未处理完）

{比较两个元素，写入小的}

while（LA为处理完）

{写入LA->LC}

while（LB未处理完）

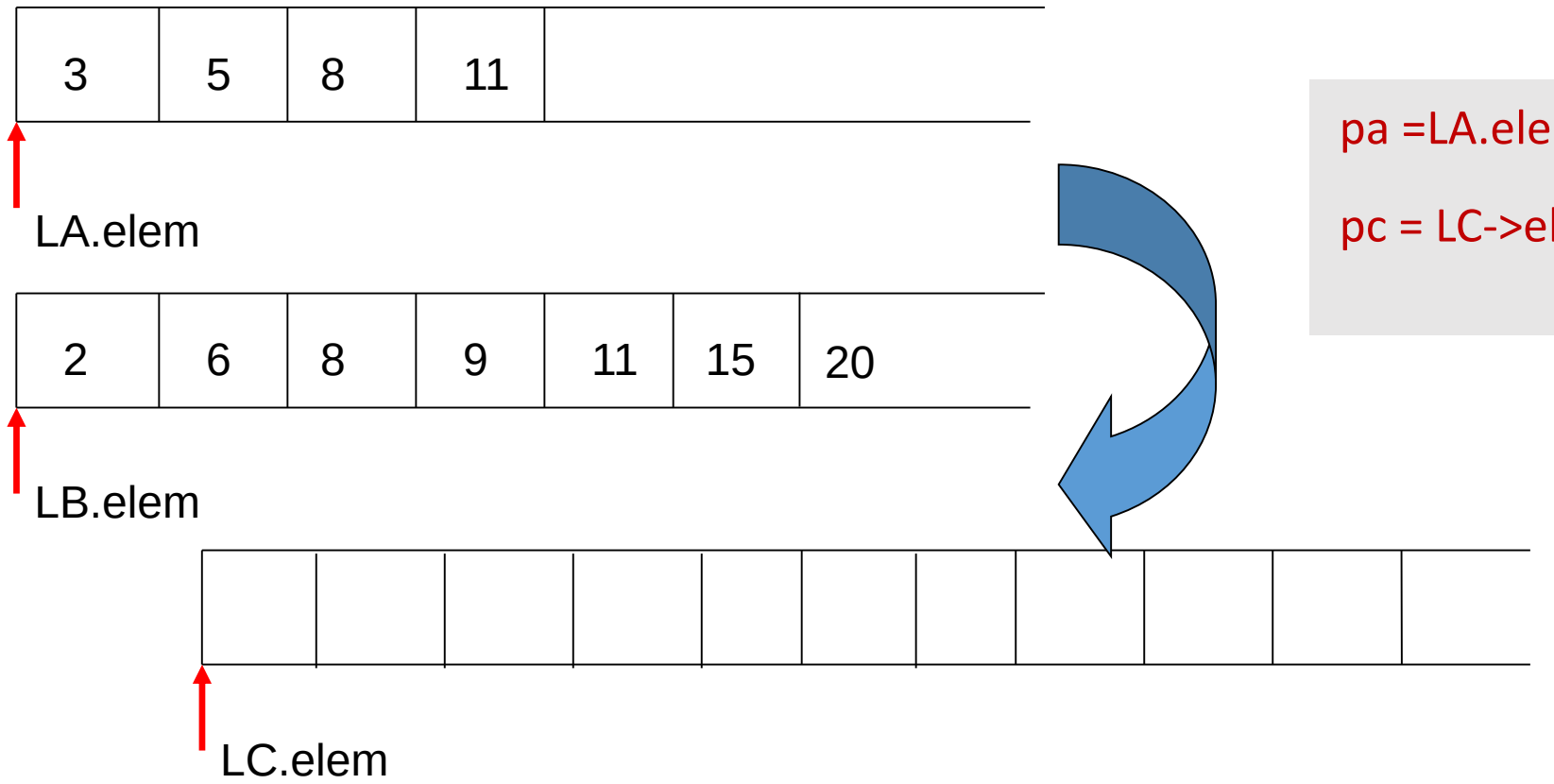
{写入LB->LC}

(1) 顺序表的归并

顺序表数据结构定义

```
# define  MAXLEN
typedef  struct {
ElemType  *elem;     // 存储空间基址
int      length;     // 当前长度
int      listsize;   // 当前分配的存储容量
} SqList;
```

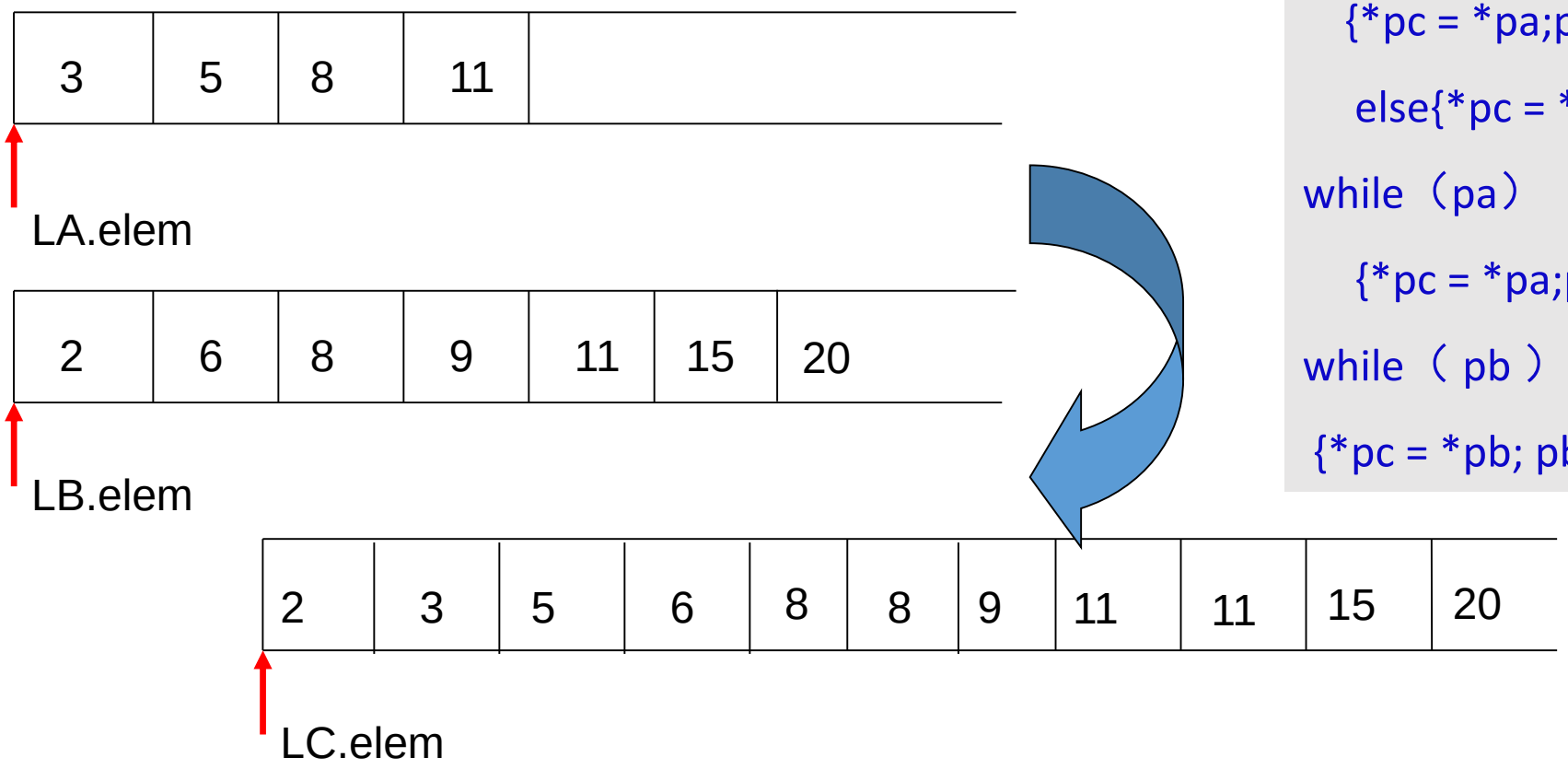
顺序存储结构



pa =LA.elem; pb=LB.elem;

pc = LC->elem;

顺序存储结构




```
while（pa&&pb）
    {if(*pa<=*pb)
   {*pc = *pa;pa++;pc++;  }
    else{*pc = *pb; pb++;  pc++;  }}
while（pa）
    {*pc = *pa;pa++;pc++;  }
while（ pb ）
 {*pc = *pb; pb++;  pc++;  }
```

算法实现

```
void Merge_sq (SqList LA, SqList LB, SqList *LC))
{
    pa =LA.elem;  pb=LB.elem;
    pc = LC->elem;
    pa_last = LA.elem+LA.length-1;
    pb_last = LB.elem+LB.length-1;
    LC->len = LA.length+LB.length;
```

```
 while(pa<=pa_last && pb <= pb_last)
      *pa<*pb? *pc++=*pa++ :  *pc++=*pb++;
 while(pa<=pa_last)
      *pc++=*pa++;
 while(pb<=pb_last)
      *pc++=*pb++;
}//merge_sq
```

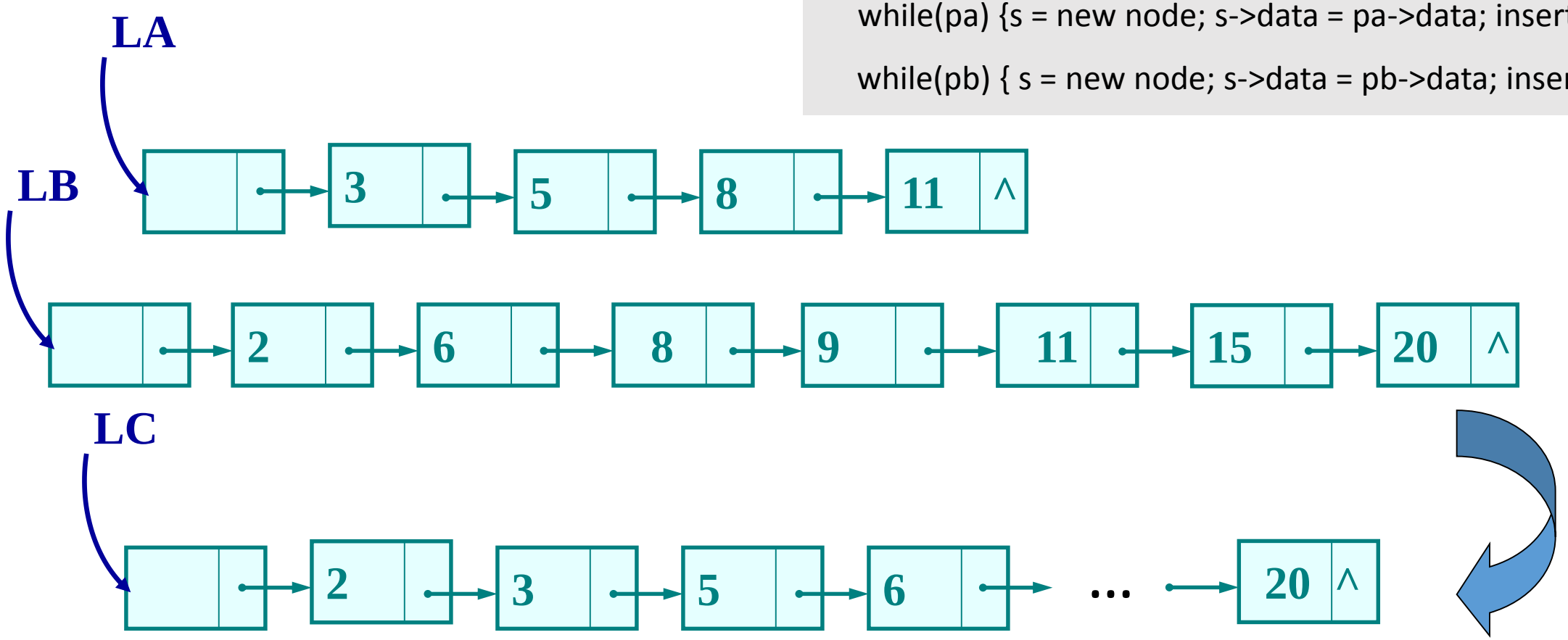
思考：如何修改程序，使合并后的顺序表不包含重复的元素？

修改：

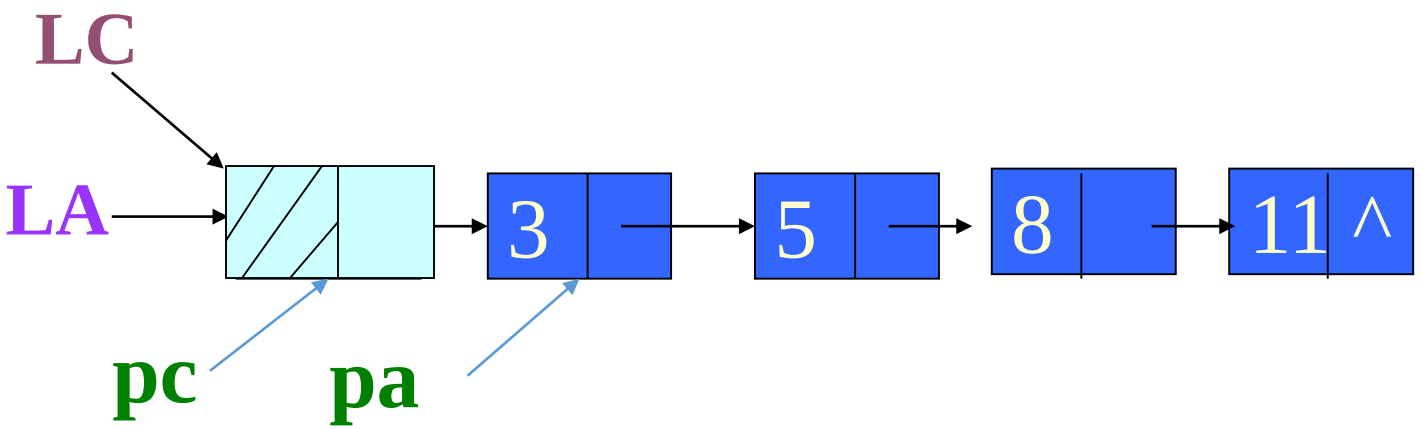
```
while(pa<=pa_last && pb <= pb_last)
  {
      if(*pa<*pb) *pc++=*pa++;
      else if(*pa>*pb) *pc++=*pb++;
      else {*pc++ = *pa; pa++;pb++; LC->length--;}
}
```

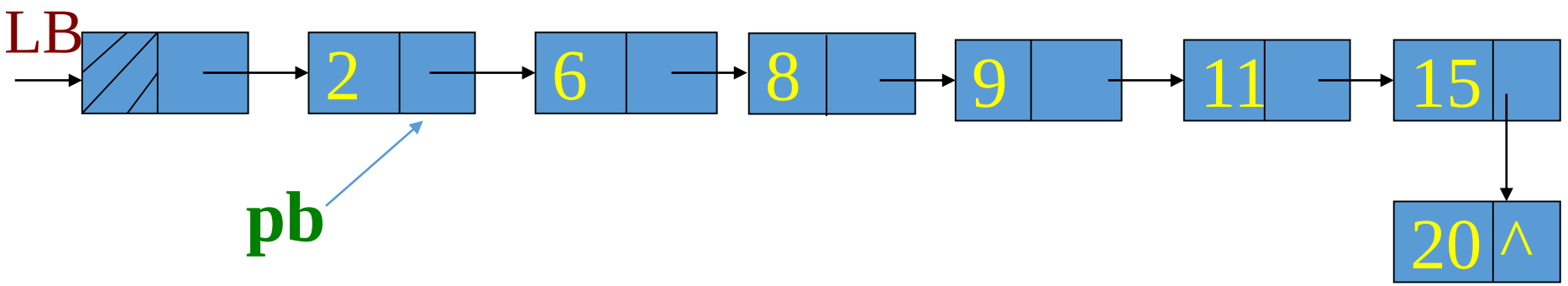
（2）单链表的归并

```
while(pa&&pb)
{ if(pa->data<pb->data)
    { s = new node; s->data = pa->data; insertTail(LC,s)};
 else {s= new node; s->data = pb->data; insertTail(LC,S)} }
while(pa) {s = new node; s->data = pa->data; insertTail(LC,s)};
while(pb) { s = new node; s->data = pb->data; insertTail(LC,s)};
```

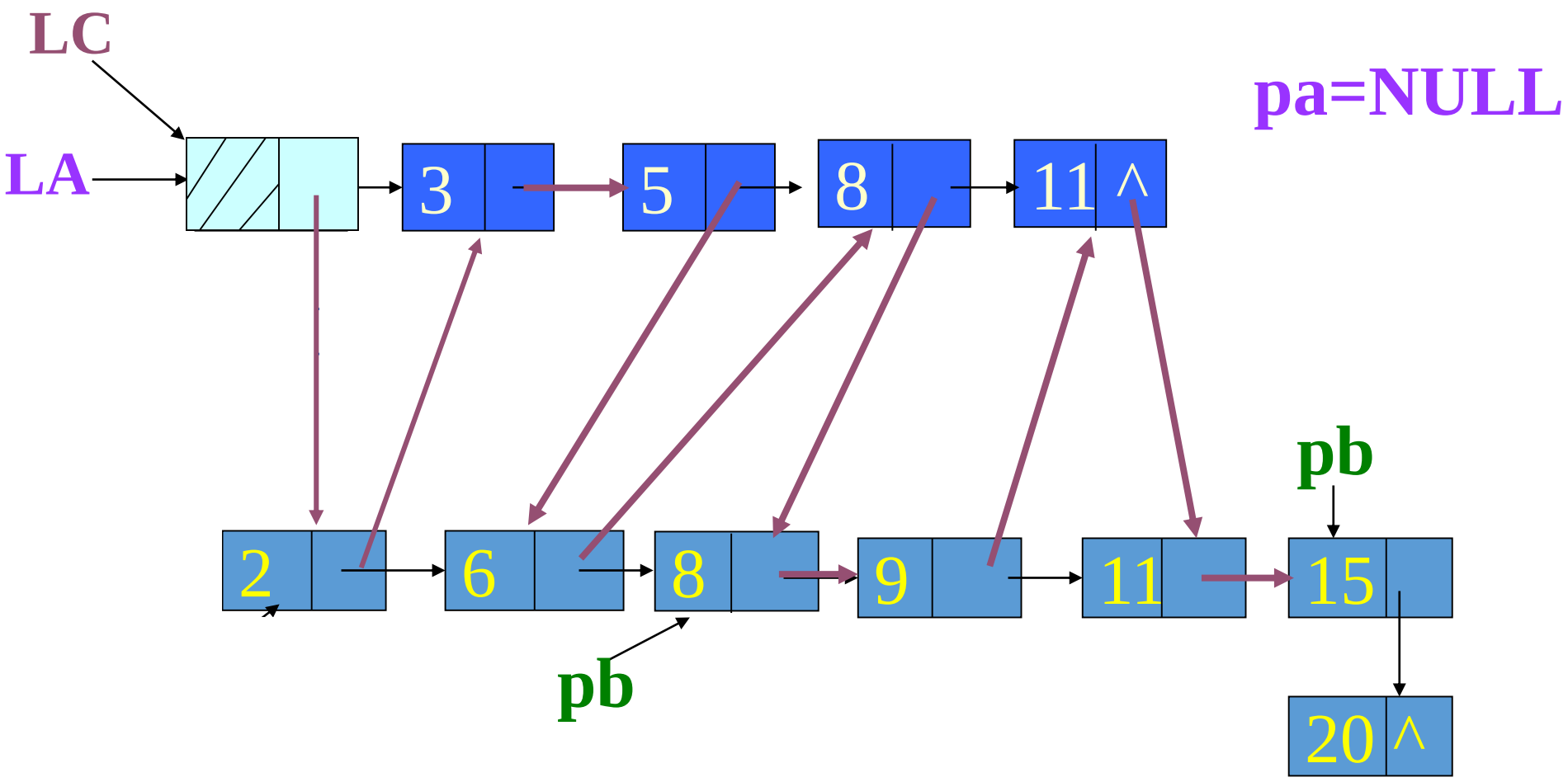


算法思想（不分配新结点）





动画演示



算法实现

```
void mergeLink(LinkList La, LinkList Lb,LinkList   *Lc)
{
   pa = La->next;   pb =Lb->next;  *Lc= pc= La;
   while (pa&&pb){
     if( pa->data <= pb->data)
       { pc->next= pa;  pc= pa; pa=pa->next;}
     else
       {pc->next= pb;  pc= pb; pb=pb->next;}
    }
    if( pa) pc->next=pa;
    else  pc->next= pb;
 }// mergeLink
```

思考：如何修改算法，使合并后的表不包含重复的元素？

```
while (pa&&pb){
    if( pa->data < pb->data)
      { pc->next= pa;  pc= pa; pa=pa->next;}
    else if(pa->data > pb->data)
     {pc->next= pb;  pc= pb; pb=pb->next;}
    else
     {  pc->next = pa; pc = pa; pa = pa->next;
        pu = pb;   pb = pb->next; free(pu);
} }
```

这有问题吗？

3.2 一元多项式的表示

如何用计算机表示一个一元多项式

$$p_n(x) = p_0 + p_1 x + p_2 x^2 + \ldots + p_n x^n$$

可以用一个线性表来表示　P = (p0, p1, …,pn)

那么，形如 $S(x) = 1 + 3x^{10000} - 2x^{20000}$ 的多项式如何表示？

一般的表示形式

一般情况下的**一元稀疏多项式**可写成

$$P_n(x) = p_1x^{e1} + p_2x^{e2} + \text{---} + p_m x^{em}$$

其中：pi 是指数为ei 的项的非零系数

$0 \leqslant e1 < e2 < \cdots < em = n$

可以下列线性表表示：**((p1, e1), (p2, e2), …, (pm, em))**

抽象数据类型一元多项式的定义

ADT Polynomial {

数据对象：

D＝{ a_i | a_i ∈TermSet, i=1,2,...,m, m≥0

TermSet 中的每个元素包含一个表示系数的实数和表示指数的整数 }

数据关系：

R1＝{ <a_{i-1} ,a_i >|a_{i-1} ,a_i∈D, i=2,...,n且a_{i-1}中的指数值＜a_i中的指数值 }

基本操作：

CreatPolyn (*P, m)

DestroyPolyn (*P)

PrintPolyn (P)

PolynLength(P)

AddPolyn (Pa,Pb,*Pc)

SubtractPolyn (Pa, Pb,*Pc)

MultiPolyn(Pa,Pb,*Pc)

} ADT Polynomial

采用链表存储多项式

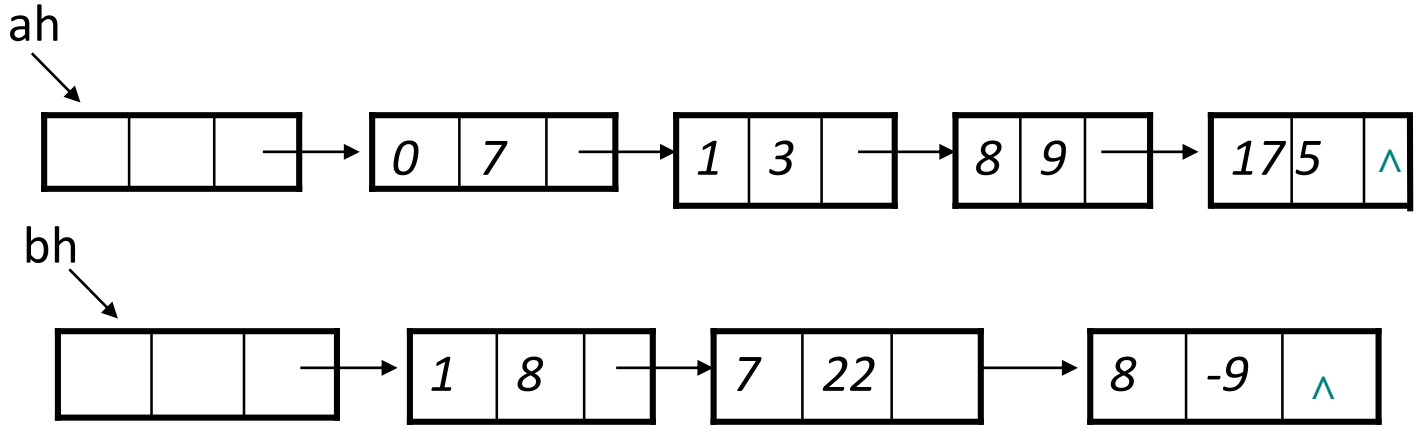
结点结构

exp	coef	next

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

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

```
typedef struct node {
    int exp;
    double coef;
    struct node *next;
}node,*LinkList;
typedef LinkList Polynomial;
```



多项式链表的生成

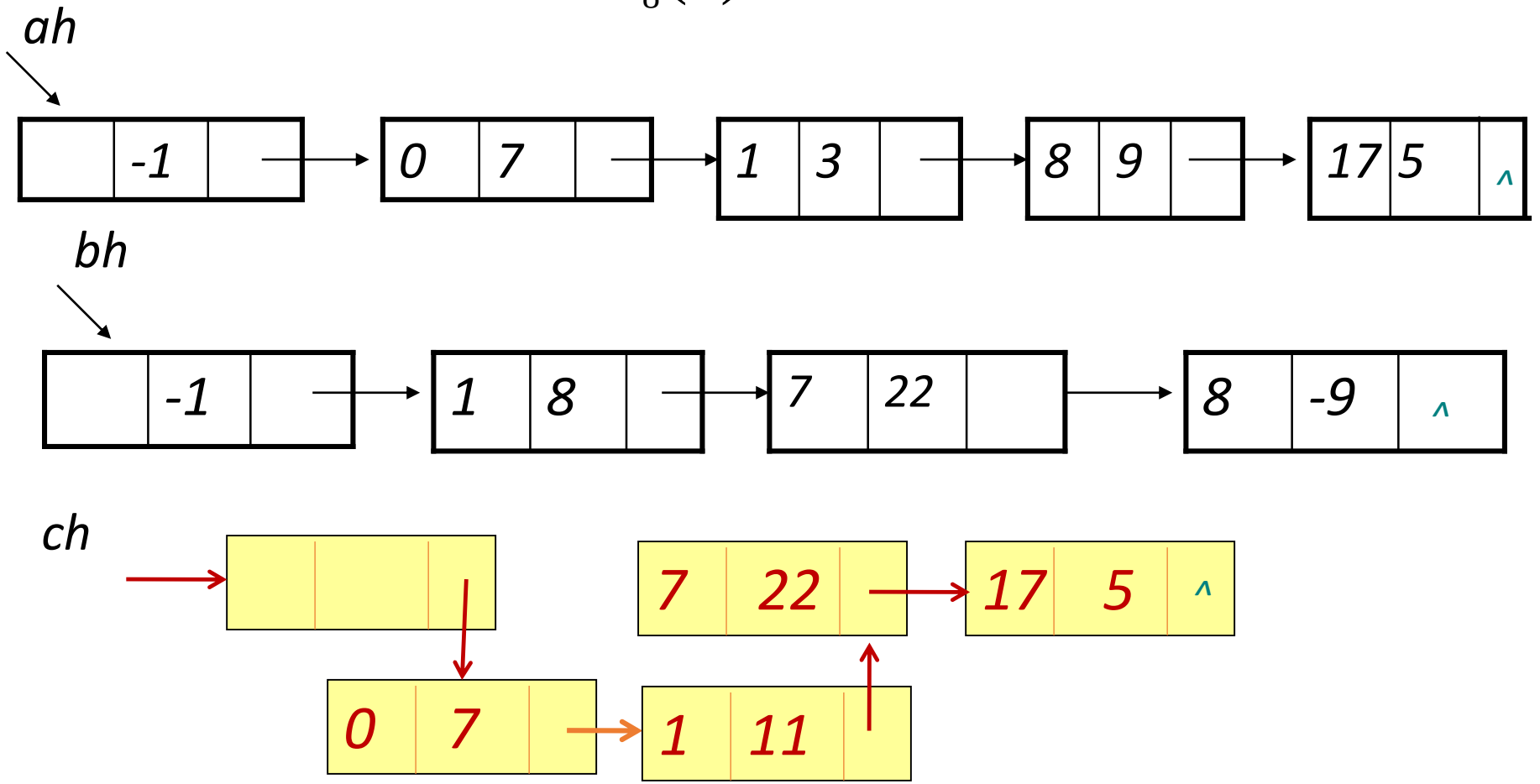
```
Polynomial CreatPolyn (int m)
{  node  * phead, * s, *k;
   //头指针、指向新结点、尾指针
   int e;
   double a;
   s = ( node *)malloc(sizeof(node));
   s->next = NULL; //生成头结点
   phead = s;  //表头指针
   k = s;    //表尾指针
```

```
for(i=1;i<=m;++i)  //循环输入多项式每一项
   {  scanf("%d%1f", &e, &a);
      s = (node *)malloc(sizeof(node));
      s->exp = e; s->coef = a;
      s->next = NULL;
      k->next = s; k = s;
   }
   return phead;
}
```

多项式求和

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

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



```
Polynomial AddPoly(Polynomial ah,  Polynomial  bh)
{ node *s, *pa, *pb; //新结点指针、指向ah当前结点、指向bh当前结点
  node *ch, *pc;  // 结构多项式链表ch表头指针，尾指针
  int e; double d; //指数、系数
  s = (node *)malloc(sizeof(struct node));
  s->next = NULL; ch = s; pc = s;
  pa = ah->next; pb = bh->next;
```

```
while (pa && pb)
{   if (pa->exp == pb->exp)
    { d = pa->coef  + pb->coef;
      e = pa->exp;
      pa = pa->next; pb = pb->next;}
    else if (pa->exp < pb->exp)
    { d  = pa->coef; e = pa->exp;
     pa = pa->next;}
    else{
     d  = pb->coef; e = pb->exp;
     pb = pb->next;}
```

```
if (d!=0)
{s = (node *)malloc(sizeof(node));
 s->exp = e;      s->coef =d;
 s->next = NULL;   pc->next = s;
 pc = s;
}
}//while
```

```
//将ah的剩余结点依次复制到ch中
while(pa)
  {  s=(node*)malloc(sizeof(node));
     s->coef=pa->coef;
     s->exp = pa->exp;
     s->next=NULL;
     pc->next=s; pc=s;
     pa=pa->next;
  }
```

```
    //将bh的剩余结点依次复制到ch中
    while(pb)
     {  s=(node*)malloc(sizeof(node));
        s->coef=pb->coef;
        s->exp = pb->exp;
        s->next=NULL;
        pc->next=s;pc=s;
        pb=pb->next;
     }
return ch;
  }
```

自学部分：参考教材，请同学们自己学习

多项式乘法，有问题及时答疑~

思考

1. 在顺序表的表尾插入或删除一个元素的操作？
2. 在单链表的表头插入或删除一个结点的操作？
3. 在顺序表的表尾插入，表头删除一个元素的操作？
4. 在单链表（带头，不带头）的表尾插入，表头删除一个元素的操作？

练习

1. 定义整型单链表结点结构
2. 初始化创建一个不带头结点的单链表
3. 在表头插入一个结点
4. 删除第一个结点

延伸学习

1. 约瑟夫环问题

2. 字符串的表示与模式匹配

3. 广义表的表示

子曰：温故而知新，可以为师矣