数据结构

（C语言版）（第2版）

线性表

线性表的应用

主讲教师：汪红松

教 学 内 容 Contents

1 线性表基本概念及顺序存储表示

2 顺序表基本操作

3 线性表的单链表表示与实现

4 线性表的循环链表表示与实现

5 线性表的应用

线性表的应用

线性表的合并

有序表的合并

一、线性表的合并

问题描述：

假设利用两个线性表La和Lb分别表示两个集合A和B,现要求一个新的集合

A=A+B

La=(7, 5, 3, 11)

Lb=(2, 6, 3)

La=(7, 5, 3, 11, 2, 6)

【算法步骤】

依次取出Lb 中的每个元素，执行以下操作：

1.在La中查找该元素；

2.如果找不到，则将其插入La的最后。

【算法描述】

```
void union(List &La, List Lb){
  La_len=ListLength(La);
  Lb_len=ListLength(Lb);
  for(i=1;i<=Lb_len;i++){
      GetElem(Lb,i,e);
      if(!LocateElem(La,e))
          ListInsert(&La,++La_len,e);
  }
}
```

$$O(ListLength(LA) \times ListLength(LB))$$

二、有序表的合并

问题描述：

已知线性表La 和Lb中的数据元素按值非递减有序排列,现要求将La和Lb归并为一个新的线性表Lc,且Lc中的数据元素仍按值非递减有序排列。

La=(1 ,7, 8)

Lb=(2, 4, 6, 8, 10, 11)

Lc=(1, 2, 4, 6, 7 , 8, 8, 10, 11)

【算法步骤】 - 有序的顺序表合并

01

创建一个空表Lc；

02

依次从 La 或 Lb 中**“摘取”元素值较小的结点**插入到 Lc 表的最后，直至其中一个表变空为止；

03

继续将 La 或 Lb 其中一个表的剩余结点插入在 Lc 表的最后。

【算法描述】 - 有序的顺序表合并

```
void MergeList_Sq(SqList LA,SqList LB,SqList &LC){
    pa=LA.elem;  pb=LB.elem;     //指针pa和pb的初值分别指向两个表的第一个元素
    LC.length=LA.length+LB.length;          //新表长度为待合并两表的长度之和
    LC.elem=new ElemType[LC.length];        //为合并后的新表分配一个数组空间
    pc=LC.elem;                             //指针pc指向新表的第一个元素
    pa_last=LA.elem+LA.length-1;  //指针pa_last指向LA表的最后一个元素
    pb_last=LB.elem+LB.length-1;  //指针pb_last指向LB表的最后一个元素
    while(pa<=pa_last && pb<=pb_last){      //两个表都非空
     if(*pa<=*pb)    *pc++=*pa++;           //依次“摘取”两表中值较小的结点
     else            *pc++=*pb++;      }
    while(pa<=pa_last)   *pc++ = *pb++;      //LA表已到达表尾
    while(pb<=pb_last)  *pc++ = *pa++;       //LB表已到达表尾
}//MergeList_Sq
```

$$\mathbf{T(n)=}\ O(ListLength(LA)+ListLength(LB))$$

S(n)= O(n)

二、有序表的合并

1.有序链表合并

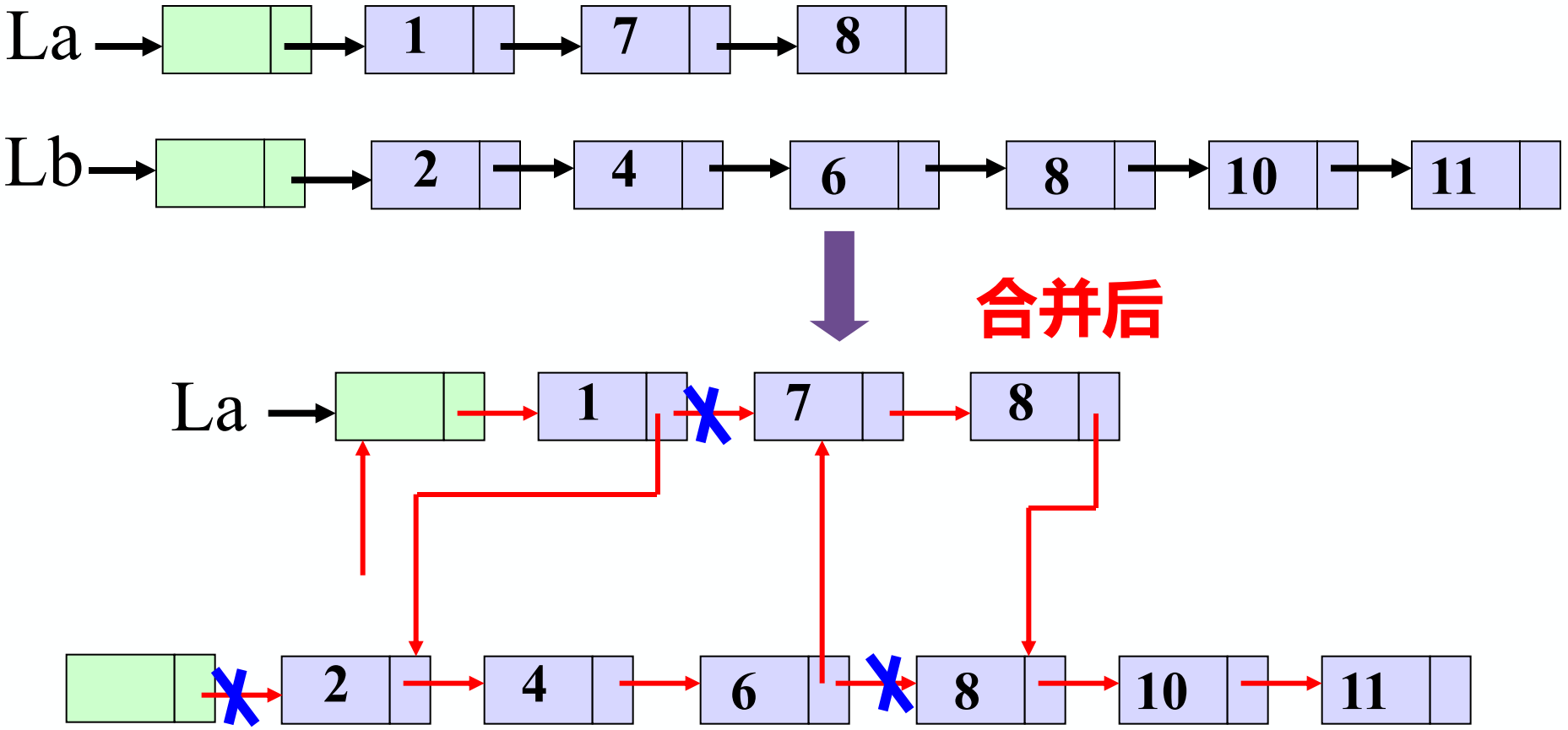
将这两个有序链表合并成一个有序的单链表。

表中允许有重复的数据。

要求结果链表仍使用原来两个链表的存储空间, 不另外占用其它的存储空间。

二、有序表的合并

1.有序链表合并



【算法步骤】 - 有序的链表合并

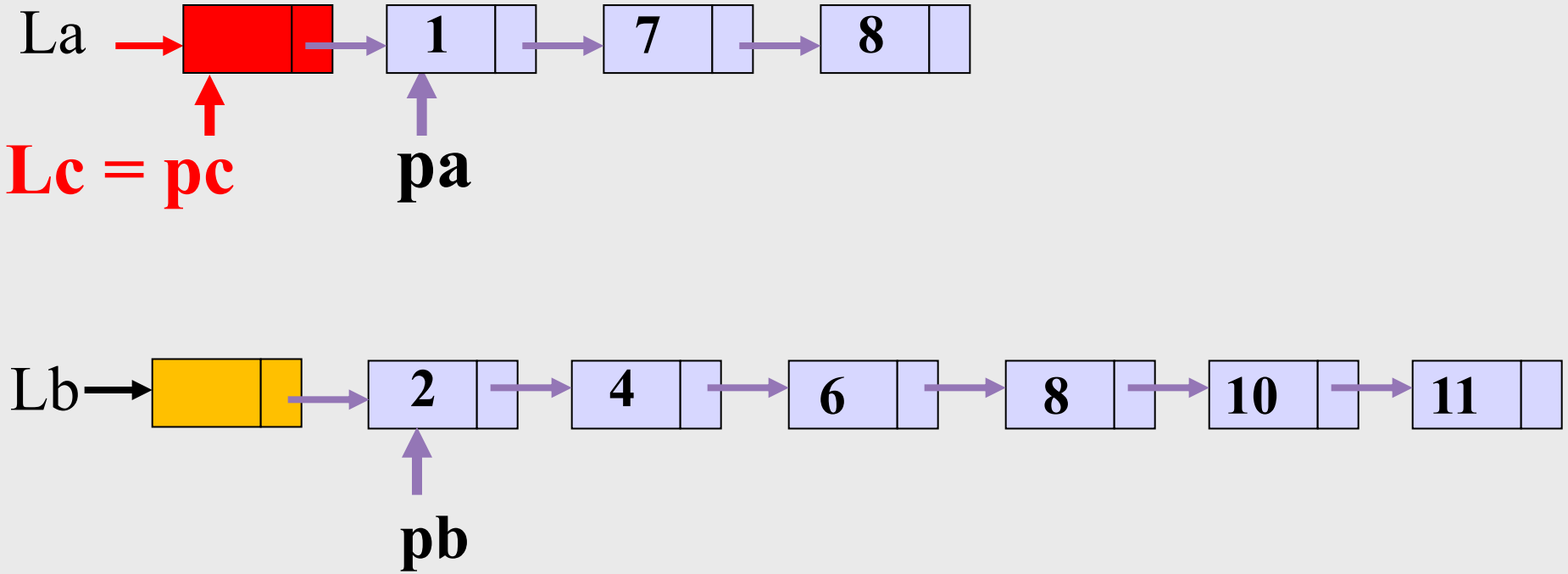
1. Lc指向La；

2. 依次从 La 或 Lb 中“摘取”元素值较小的结点插入到 Lc 表的最后，直至其中一个表变空为止；

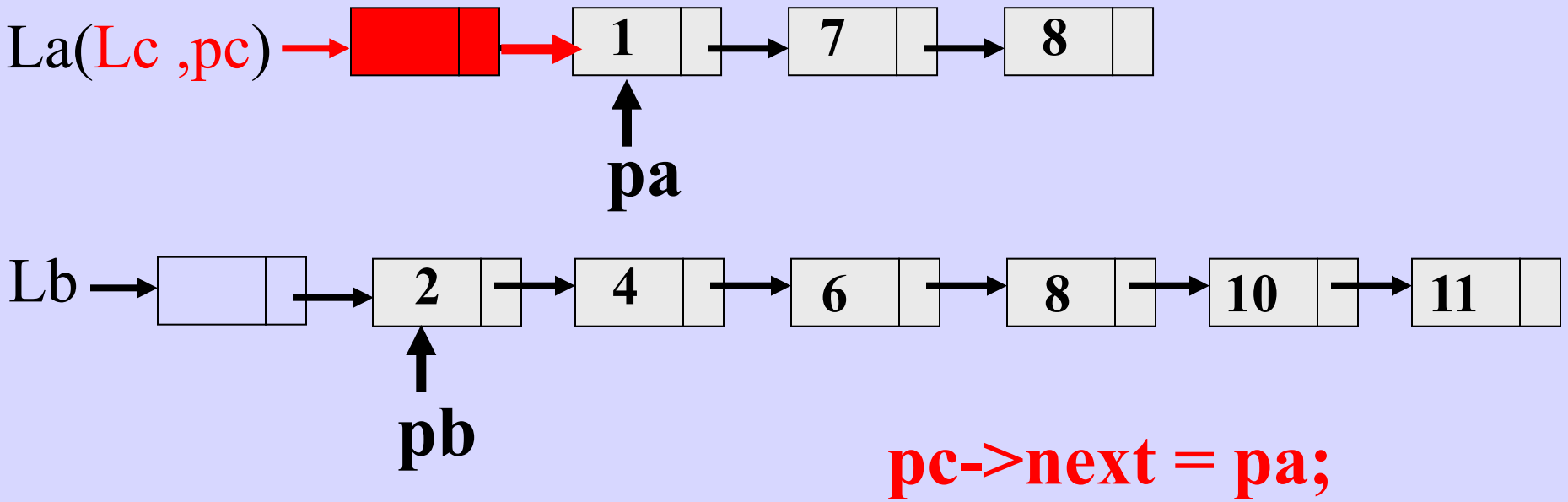
3. 继续将 La 或 Lb 其中一个表的剩余结点插入在 Lc 表的最后；

4. 释放 Lb 表的表头结点。

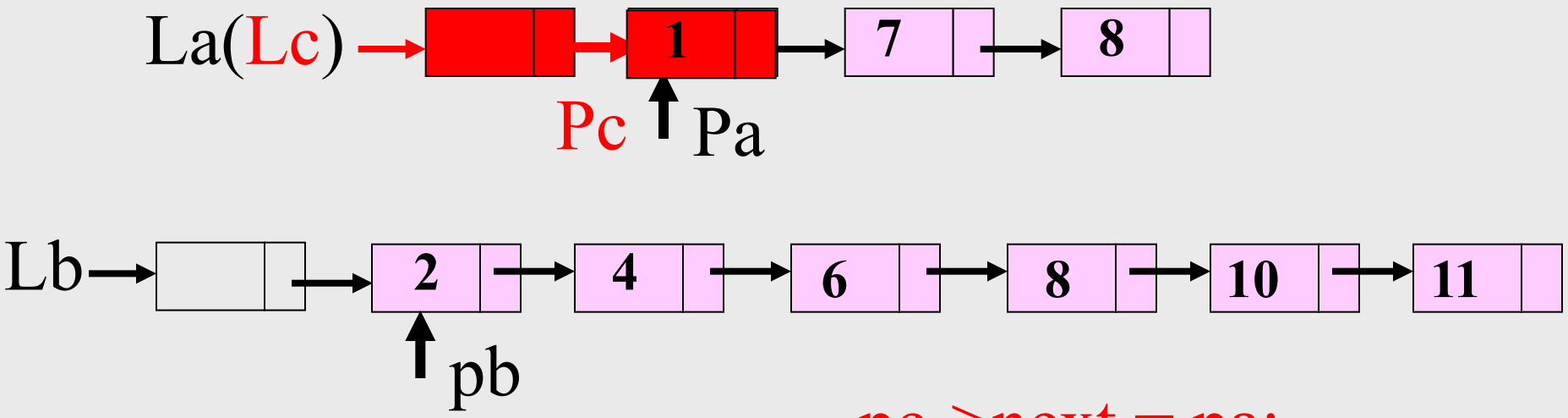
有序链表合并（初始化）



有序链表合并(pa->data < = pb->data)



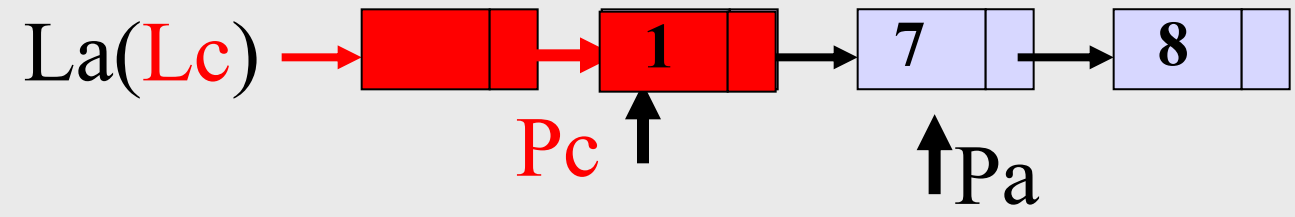
有序链表合并(pa->data < = pb->data)

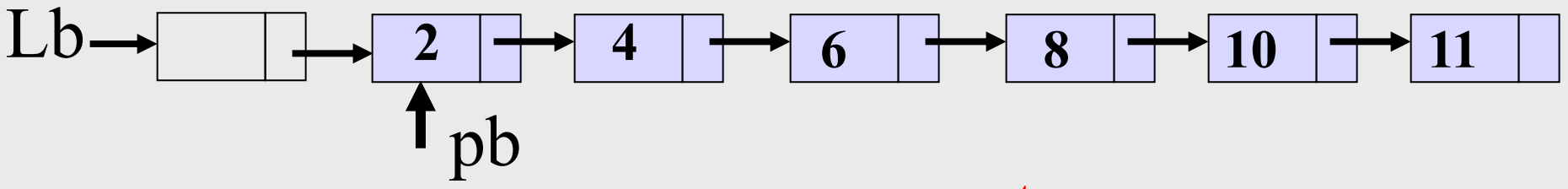


pc->next = pa;

pc= pa;

有序链表合并(pa->data < = pb->data)



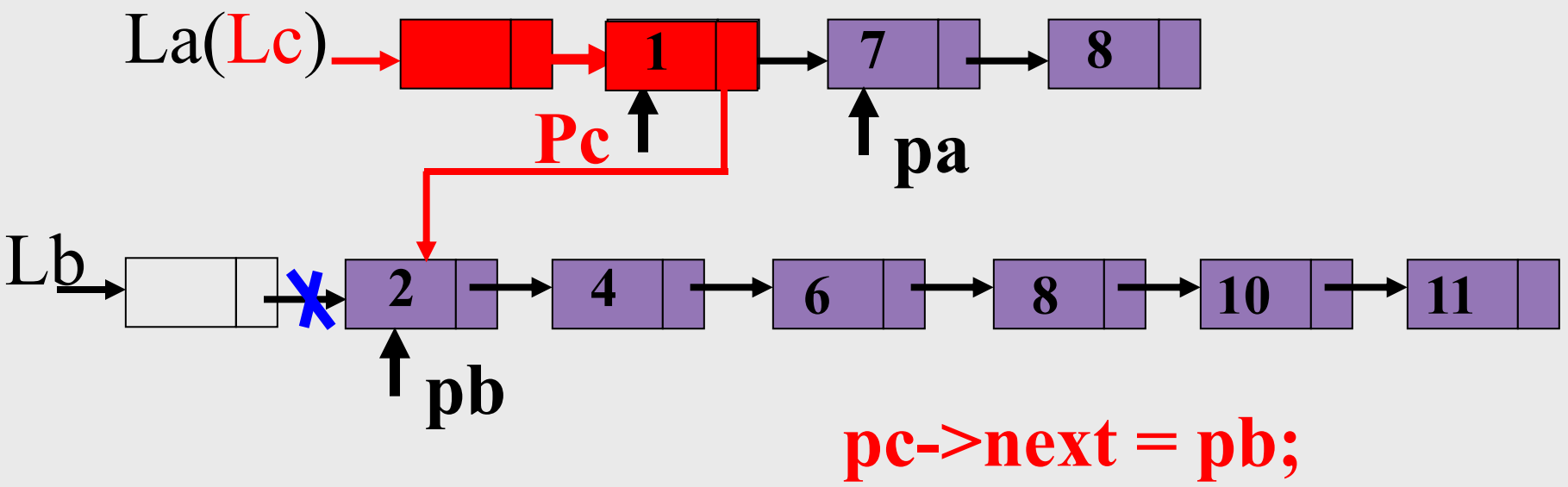


pc->next = pa;

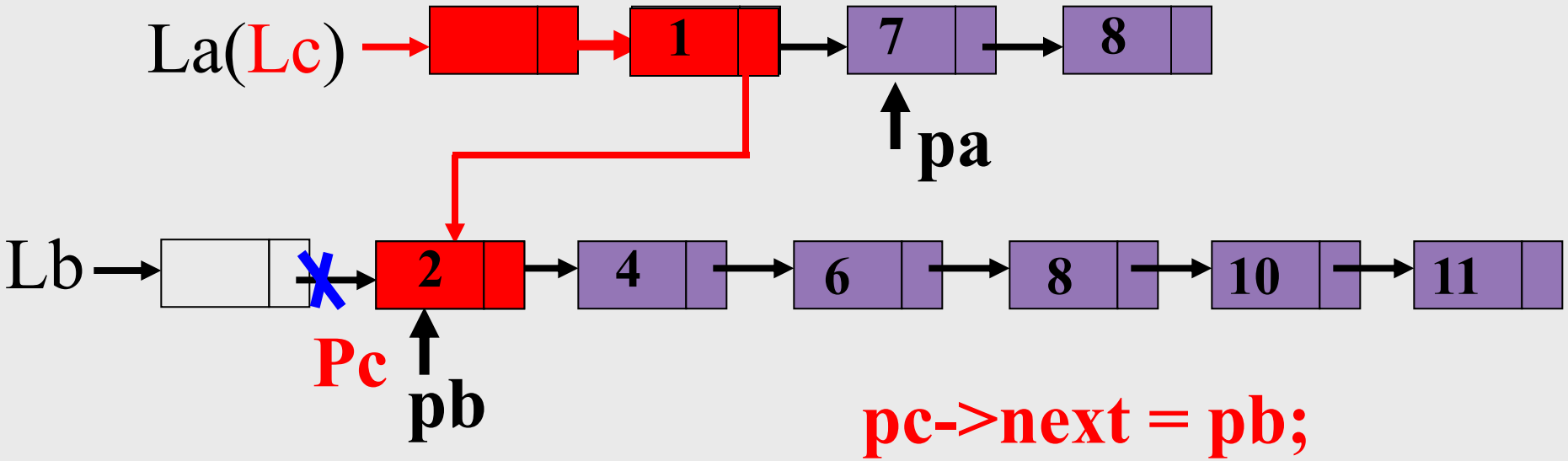
pc= pa;

pa = pa->next;

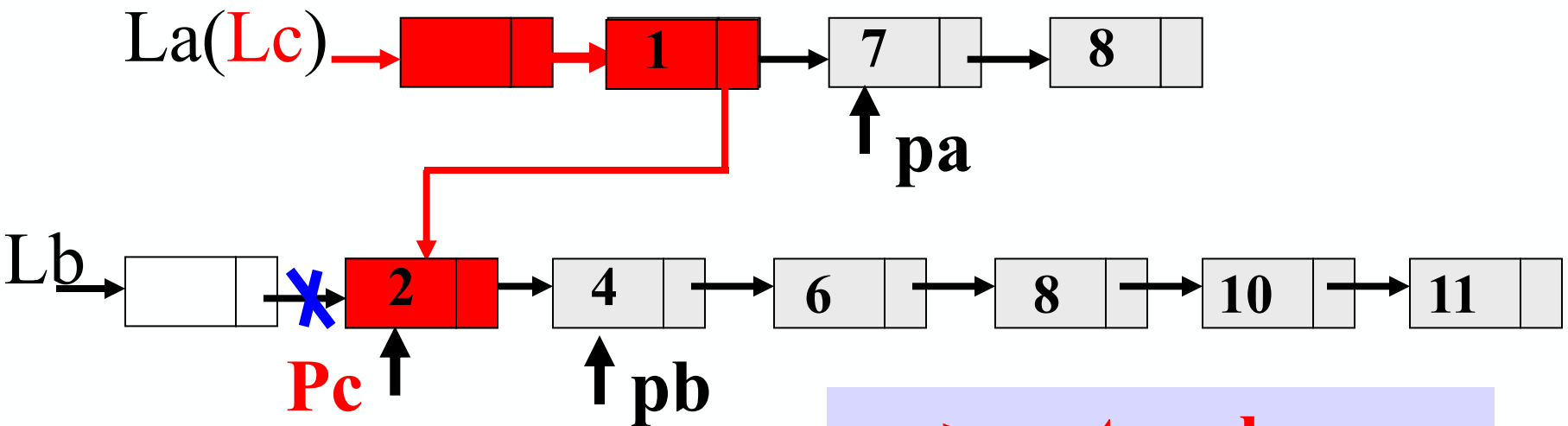
有序链表合并(pa->data >pb->data)



有序链表合并(pa->data >pb->data)



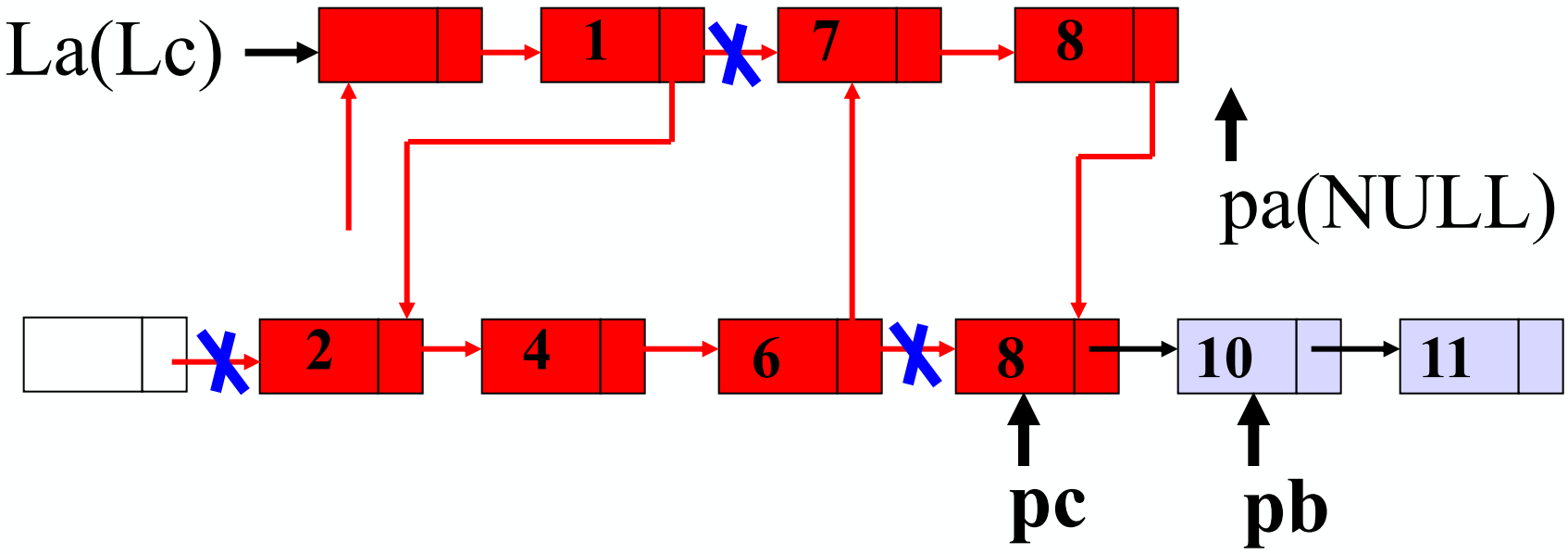
有序链表合并(pa->data >pb->data)




```
pc->next = pb;
pc= pb;
pb =pb->next;
```

二、有序表的合并

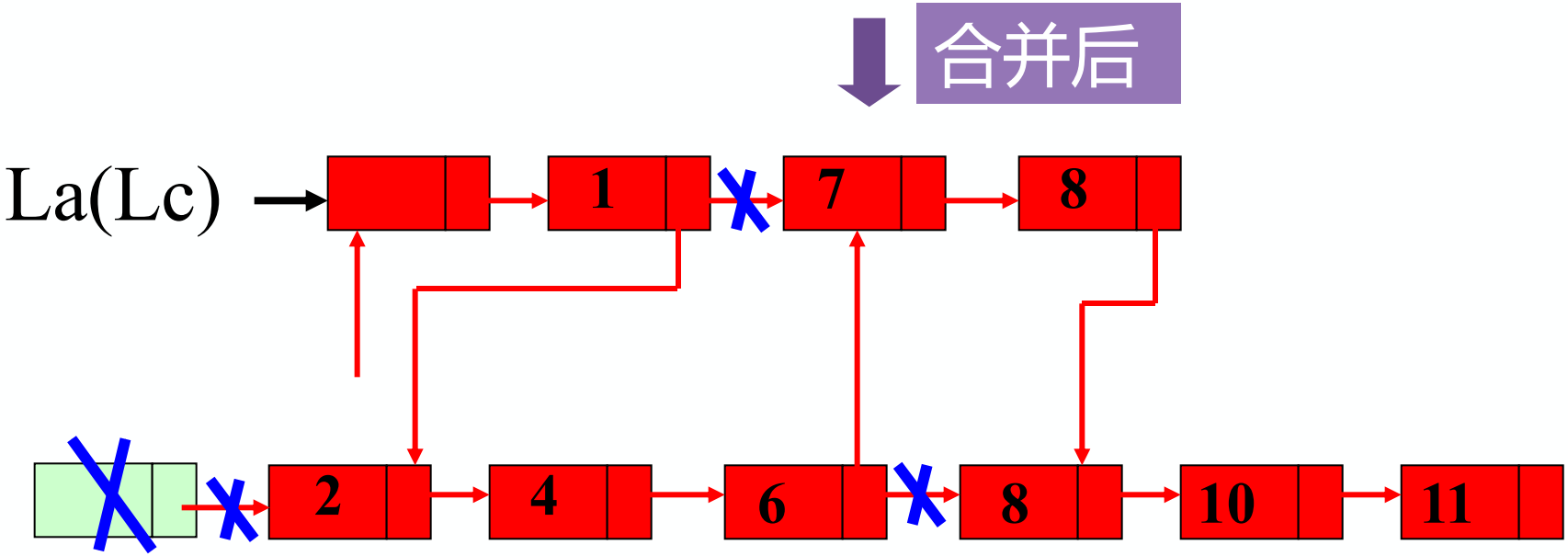
1.有序链表合并



pc-> next=pa?pa:pb;

二、有序表的合并

1.有序链表合并



delete Lb;

【算法描述】 - 有序的链表合并

```
void MergeList_L(LinkList &La,LinkList &Lb,LinkList &Lc){
  pa=La->next;  pb=Lb->next;
  pc=Lc=La;                    //①用La的头结点作为Lc的头结点
  while(pa && pb){             //②”摘取”元素值较小的结点
    if(pa->data<=pb->data){ pc->next=pa;pc=pa;pa=pa->next;}
    else                        {pc->next=pb; pc=pb; pb=pb->next;}
  pc->next=pa?pa:pb;           //③插入剩余段
  delete Lb;                   //④释放Lb的头结点}
```

T(n)= $O(ListLength(LA)+ListLength(LB))$

S(n)= O(1)