

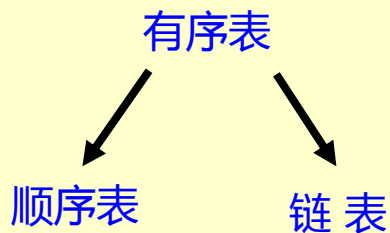
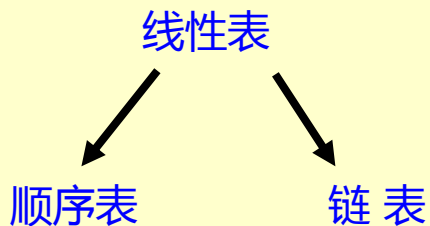


本节主题:

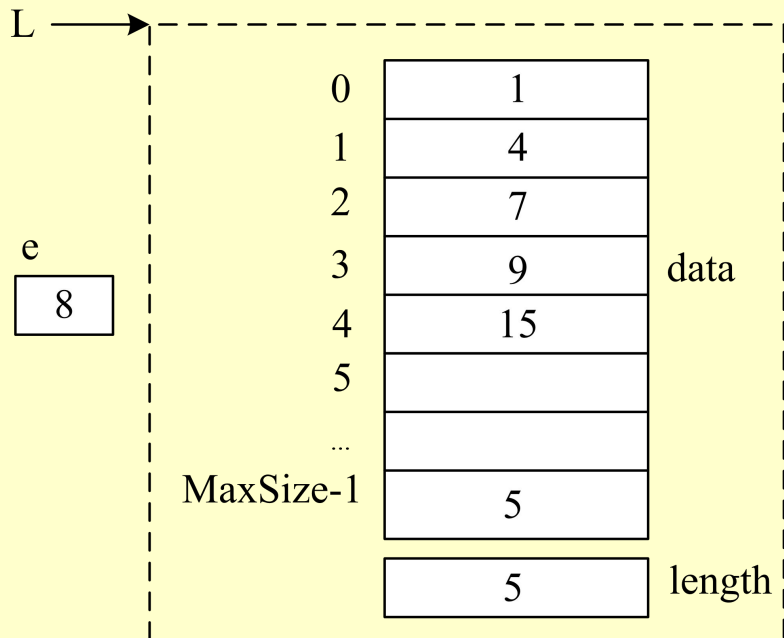
有序表

# 有序表概念

- 有序表：其中所有元素以递增或递减方式有序排列的线性表。
- 有序表 vs. 线性表
  - 逻辑结构相同
  - 可采用的存储结构相同
  - 有序的性质，使运算的实现有所不同

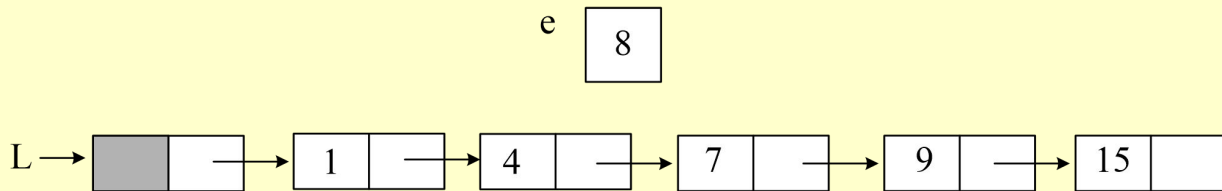


# 有序顺序表的ListInsert()算法



```
void ListInsert(SqList *&L, ElemType e)
{
    int i=0, j;
    while (i<L->length && L->data[i]<e)
        i++;
    for (j=L->length; j>i; j--)
        L->data[j]=L->data[j-1];
    L->data[i]=e;
    L->length++;
}
```

# 有序单链表的ListInsert()的算法



```
void ListInsert(LinkList *&L, ElemType e)
{
    LinkList *pre=L, *p;
    while (pre->next!=NULL && pre->next->data < e)
        pre=pre->next;
    p=(LinkList *)malloc(sizeof(LinkList));
    p->data=e;
    p->next=pre->next;
    pre->next=p;
}
```

# 有序表应用：合并有序表

LA: 1 3 5

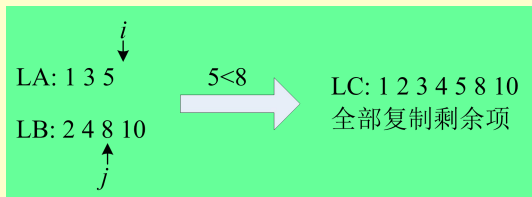
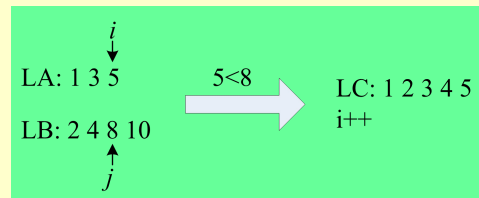
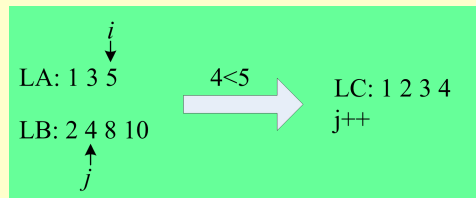
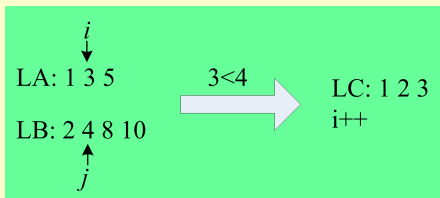
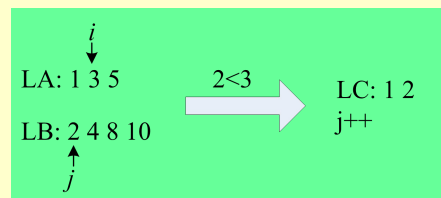
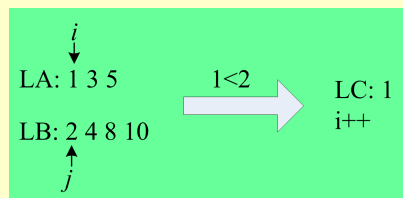
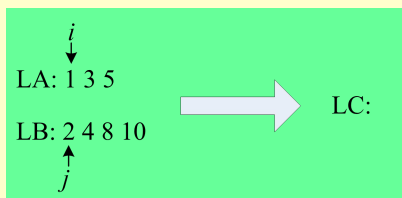
→ LC: 1 2 3 4 5 8 10

LB: 2 4 8 10

## 问题

假设两个有序表LA和LB，将它们合并成一个有序表LC

要求不破坏原有表LA和LB



# 采用顺序表存放有序表时的归并算法

```
void UnionList(SqList *LA,SqList *LB,SqList *&LC)
```

```
{
```

```
    int i=0,j=0,k=0;
```

```
    LC=(SqList *)malloc(sizeof(SqList));
```

```
    //LA和LB均未到达末尾时，择其小加入LC
```

```
    //LA尚未扫描完,将其余元素插入LC中
```

```
    //LB尚未扫描完,将其余元素插入LC中
```

```
    LC->length=k;
```

```
}
```

```
    while (j<LB->length)
```

```
    {
```

```
        LC->data[k]=LB->data[j];
```

```
        j++;
```

```
        k++;
```

```
    }
```

```
while (i<LA->length && j<LB->length)
```

```
{
```

```
    if (LA->data[i]<LB->data[j])
```

```
    {
```

```
        LC->data[k]=LA->data[i];
```

```
        i++;
```

```
    }
```

```
    else //LA->data[i]>LB->data[j]
```

```
    {
```

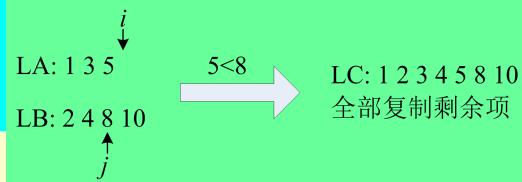
```
        LC->data[k]=LB->data[j];
```

```
        j++;
```

```
    }
```

```
    k++;
```

```
}
```



```
while (i<LA->length)
```

```
{
```

```
    LC->data[k]=LA->data[i];
```

```
    i++;
```

```
    k++;
```

```
}
```

时间复杂度为 $O(m+n)$

空间复杂度为 $O(m+n)$

# 采用单链表存放有序表时的归并算法

```
void UnionList1(LinkList *LA, LinkList *LB, LinkList *&LC)
```

```
{
```

```
    LinkList *pa=LA->next,*pb=LB->next,*r,*s;
```

```
    LC=(LinkList *)malloc(sizeof(LinkList));
```

```
    r=LC;
```

```
    //LA和LB均未到达末尾时，择其小优先尾插
```

```
    //LA未到末尾，复制LA中所有结点
```

```
    //LB未到末尾，复制LB中所有结点
```

```
    r->next=NULL;
```

```
}
```

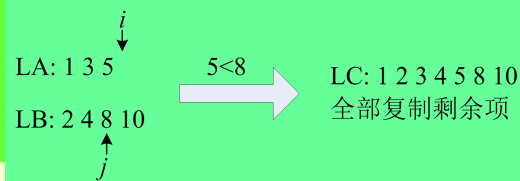
```
    while (pb!=NULL)
```

```
    {
```

```
        .....
```

```
    }
```

```
while (pa!=NULL && pb!=NULL)
{
    s=(LinkList *)malloc(sizeof(LinkList));
    if (pa->data<pb->data)
    {
        s->data=pa->data;
        pa=pa->next;
    }
    else
    {
        s->data=pb->data;
        pb=pb->next;
    }
    r->next=s;
    r=s;
}
```



```
while (pa!=NULL)
```

```
{
```

```
    s=(LinkList *)malloc(sizeof(LinkList));
```

```
    s->data=pa->data;
```

```
    r->next=s;
```

```
    r=s;
```

```
    pa=pa->next;
```

```
}
```

时间复杂度为 $O(m+n)$

空间复杂度为 $O(m+n)$

# 思考题

- ☐ 有序表和线性表有什么异同？
- ☐ 有序表和顺序表有什么不同？