

10.1 概述

10.2 插入排序

10.3 快速排序

10.4 堆排序

10.5 归并排序

10.6 基数排序

10.7 各种排序方法的综合比较

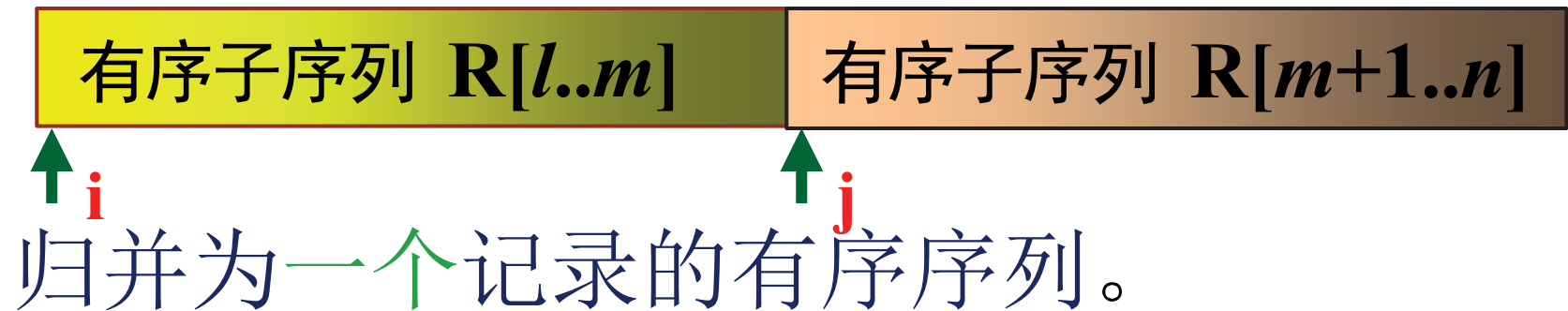
10.8 外部排序

10.5 归并排序

归并排序的过程基于下列
基本思想进行：

将两个或两个以上的有序子
序列“归并”为一个有序序列。

在排序中，通常采用的是**2-路归并**排序。即：将两个位置相邻的记录有序子序列



这个操作对顺序表而言，是轻而易举的。

```
void Merge (RcdType SR[], RcdType &TR[],  
            int i, int m, int n) {  
    // 将有序的记录序列 SR[i..m] 和 SR[m+1..n]  
    // 归并为有序的记录序列 TR[i..n]  
  
    for (j=m+1, k=i; i<=m && j<=n; ++k)  
    {  
        // 将SR中记录由小到大并入TR  
        if (SR[i].key<=SR[j].key) TR[k] = SR[i++];  
        else TR[k] = SR[j++];  
    }  
  
    ... ..  
  
} // Merge
```

if ($i \leq m$) **TR**[$k..n$] = **SR**[$i..m$];

// 将剩余的 **SR**[$i..m$] 复制到 **TR**

if ($j \leq n$) **TR**[$k..n$] = **SR**[$j..n$];

// 将剩余的 **SR**[$j..n$] 复制到 **TR**

归并排序的算法

如果记录无序序列 $R[s..t]$ 的两部分

$R[s..\lfloor (s+t)/2 \rfloor]$ 和 $R[\lfloor (s+t)/2 \rfloor + 1..t]$

分别按关键字有序，

则利用上述归并算法很容易将它们归并成整个记录序列是一个有序序列。

由此，应该先**分别**对这两部分进行**2-路**归并排序。

例如:

52, 23, 80, 36, 68, 14 (s=1, t=6)

[52, 23, 80] [36, 68, 14]

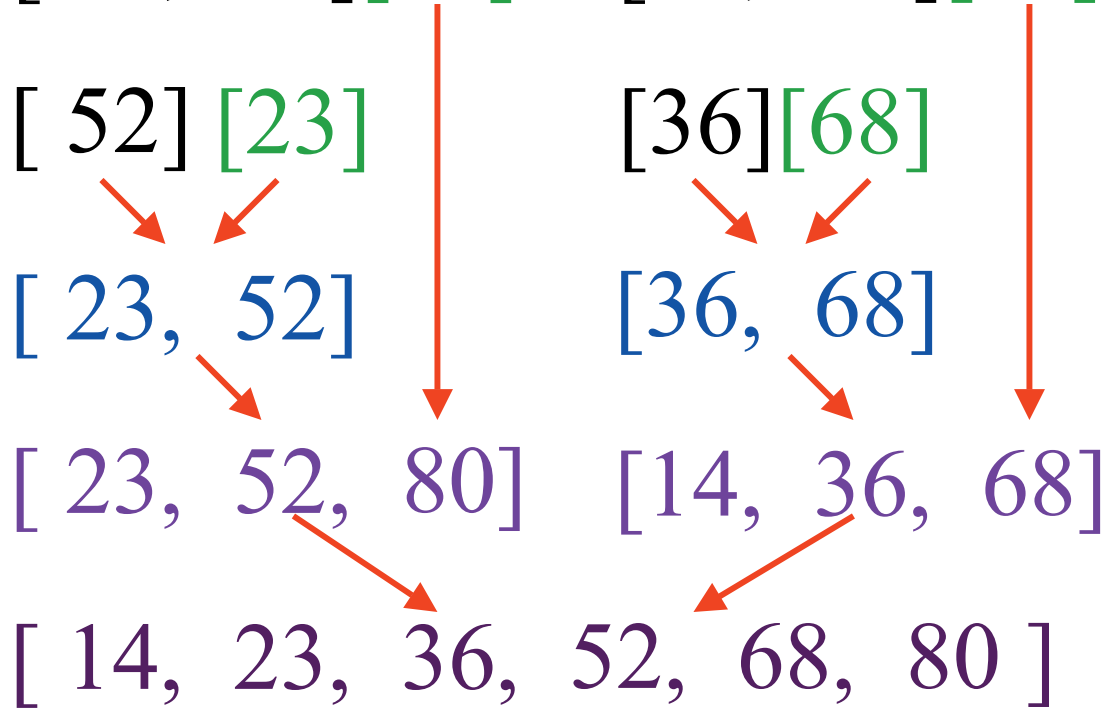
[52, 23][80] [36, 68][14]

[52] [23] [36] [68]

[23, 52] [36, 68]

[23, 52, 80] [14, 36, 68]

[14, 23, 36, 52, 68, 80]



```
void Msort ( RcdType SR[ ],  
             RcdType &TR1[ ], int s, int t ) {  
    // 将SR[s..t] 归并排序为 TR1[s..t]  
    if (s == t) TR1[s]=SR[s];  
    else  
    {  
        ... ..  
    }  
} // Msort
```


以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/095313003242011240>