

10.3 交换排序

- ♣ 交换排序是一类基于“交换”的排序，系统地交换反序的记录的对，直到不再有这样的偶对为止。
- ♣ 交换排序的基本思路：两两比较待排序记录的关键字，并交换不满足次序要求的那些偶对，直到全部满足为止。
- ♣ 主要的交换排序算法：冒泡排序(Bubble Sort)和快速排序(Quick Sort)。

9.3.1 冒泡排序

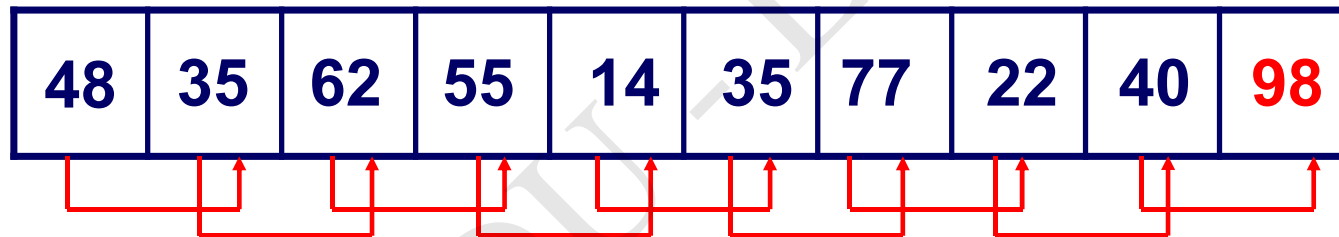
冒泡排序的基本思路：

- 设待排序记录序列中的记录个数为 n 。
- 最多作 $n-1$ 趟冒泡， $i = 1, 2, \dots, n-1$ 。
- 在第 i 趟中从后向前， $j = n-1, n-2, \dots, i$ ，顺次两两比较 $r[j-1].key$ 和 $r[j].key$ 。
- 如果发生逆序，即 $r[j-1].key > r[j].key$ ，则交换 $r[j-1].key$ 和 $r[j].key$ 。

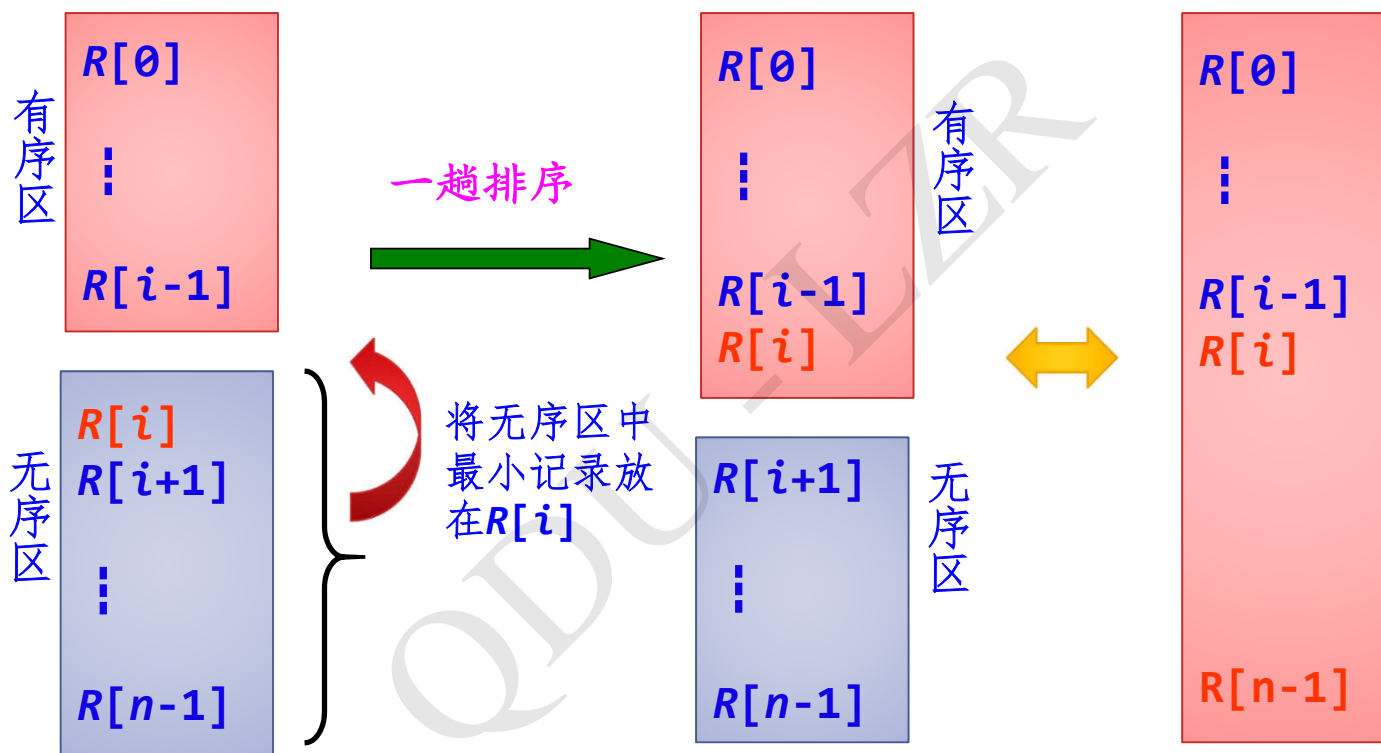
1	2	3	4	5	6	7	8	9	10
48	62	35	77	55	14	35	98	22	40

➤ 1趟冒泡排序演示

9次



9.3.1 冒泡排序



■ 初始有序区为空。

■ $i=0\sim n-2$, 共 $n-1$ 趟使整个数据有序。

有序区总是全局有序的

- $R[j]$ 和 $R[j-1]$ 反序 \Rightarrow 交换。
- 若某一趟比较时不出现任何记录交换，说明所有记录已排好序了，就可以结束本算法。

【示例】 已知有10个待排序的记录，它们的关键字序列为
(75,87,68,92,88,61,77,96, 80,72)，给出用冒泡排序法进行排序的过程。

初始序列	75	87	68	92	88	61	77	96	80	72
i=0	[61]	75	87	68	92	88	72	77	96	80
i=1	61	[68]	75	87	72	92	88	77	80	96
i=2	61	68	[72]	75	87	77	92	88	80	96
i=3	61	68	72	[75]	77	87	80	92	88	96
i=4	61	68	72	75	[77]	80	87	88	92	96
i=5	61	68	72	75	77	[80]	87	88	92	96
最后结果	61	68	72	75	77	80	87	88	92	96

冒泡排序算法如下：

```
void BubbleSort(RedType R[],int n)
{ int i,j,exchange;
  RedType tmp;
  for(i=0;i<n-1;i++)
  {
    exchange=0; //本趟排序前置exchange为0
    for (j=n-1;j>i;j--) //比较,找出最小关键字的记录
      if(R[j].key<R[j-1].key)
      {
        tmp=R[j]; //R[j]与R[j-1]进行交换,
        R[j]=R[j-1]; //将最小关键字记录前移
        R[j-1]=tmp;
        exchange=1; //本趟排序发生交换置exchange为1
      }
    if (exchange==0) //本趟未发生交换时结束算法
      return;
  }
}
```

■ 最好的情况（正序）：只需进行一趟冒泡

“比较”的次数：

$$n-1$$

“移动”的次数：

$$0$$

■ 最坏的情况（反序）：需进行 $n-1$ 趟冒泡

“比较”的次数：

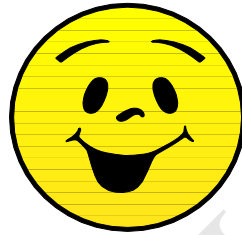
$$\sum_{i=0}^{n-1} (n-i-1) = \frac{n(n-1)}{2}$$

“移动”的次数：

$$\sum_{i=0}^{n-2} 3(n-i-1) = \frac{3n(n-1)}{2}$$

归纳起来，冒泡排序算法的性能如表所示。

时间复杂度			空间复杂度	稳定性
最好情况	最坏情况	平均情况		
$O(n)$	$O(n^2)$	$O(n^2)$	$O(1)$	稳定



— END —