

## 10.4 选择排序

♣ 选择排序的基本思想是：

- 每一趟（例如第  $i$  趟,  $i = 1, \dots, n-1$ ）在后面  $n-i+1$  个待排序记录序列中选出关键字值最小的记录，作为有序记录序列的第  $i$  个记录。待到第  $n-1$  趟作完，待排序记录只剩下1个，就不用再选了。
- 选择排序的每一趟可把有序区扩大，直到  $n-1$  趟后即可把有序区扩大到整个待排序序列。

♣ 主要的选择排序算法：简单选择排序和堆排序。

### 10.4.1 简单选择排序

♣ 简单选择排序的基本步骤是：

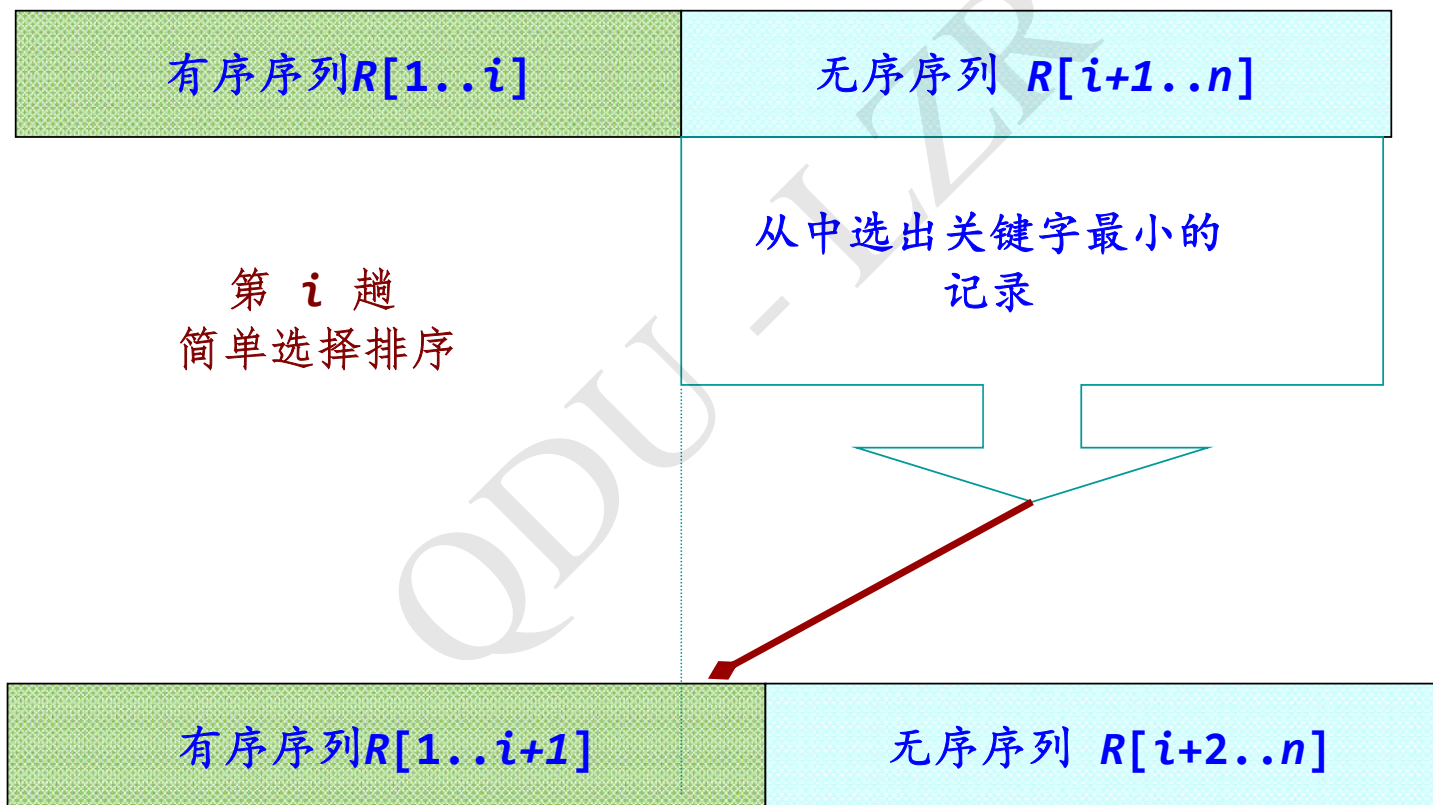
- 在一组记录  $r[i] \sim r[n]$  中选择具有最小关键字的记录；
- 若它不是这组记录中的第一个记录，则将它与这组记录中的第一个记录对调；
- 在这组记录中剔除这个具有最小关键字值的记录。在剩下的记录  $r[i+1] \sim r[n]$  中重复执行第①、②步，直到剩余记录只有一个为止。

● 简单选择排序的演示

第 8 趟

14	35	35	48	55	62	77	98
----	----	----	----	----	----	----	----

### 10.4.1 简单选择排序



【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)，给出用直接选择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	87	68	92	88	75	77	96	80	72
i=1	61	68	87	92	88	75	77	96	80	72
i=2	61	68	72	92	88	75	77	96	80	87
i=3	61	68	72	75	88	92	77	96	80	87
i=4	61	68	72	75	77	92	88	96	80	87
i=5	61	68	72	75	77	80	88	96	92	87
i=6	61	68	72	75	77	80	87	96	92	88
i=7	61	68	72	75	77	80	87	88	92	96
i=8	61	68	72	75	77	80	87	88	92	96
最后结果	61	68	72	75	77	80	87	88	92	96

简单选择排序算法如下：

```
int SelectMinKey(SqList L, int i)
{
    // 返回在L.r[i..L.length]中key最小的记录的序号
    KeyType min;
    int j, k;
    k = i; // 设第i个为最小
    min = L.r[i].key;
    for(j = i + 1; j <= L.length; j++)
        if(L.r[j].key < min) { // 找到更小的
            k = j;
            min = L.r[j].key;
        }
    return k;
}
```

简单选择排序算法如下：

```
void SelectSort(Sqlist &L)
{
    // 对顺序表L作简单选择排序。算法10.9
    int i, j;
    RedType t;
    for(i = 1; i < L.length; ++i) { //选择第i小的记录，并交换到位
        // 在L.r[i..L.length]中选择key最小的记录
        j = SelectMinKey(L, i);
        if(i != j) { //与第i个记录交换
            t = L.r[i];
            L.r[i] = L.r[j];
            L.r[j] = t;
        }
    }
}
```

- 对  $n$  个记录进行简单选择排序，所需进行的关键字间的比较次数总计为：

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

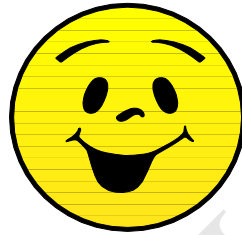
- 移动记录的次数，最小值为 0，最大值为  $3(n-1)$ 。



➤ 归纳起来，简单选择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		
$O(n^2)$	$O(n^2)$	$O(n^2)$	$O(1)$	不稳定

➤ Keys={3, 2, 3, 1} ?



— END —