

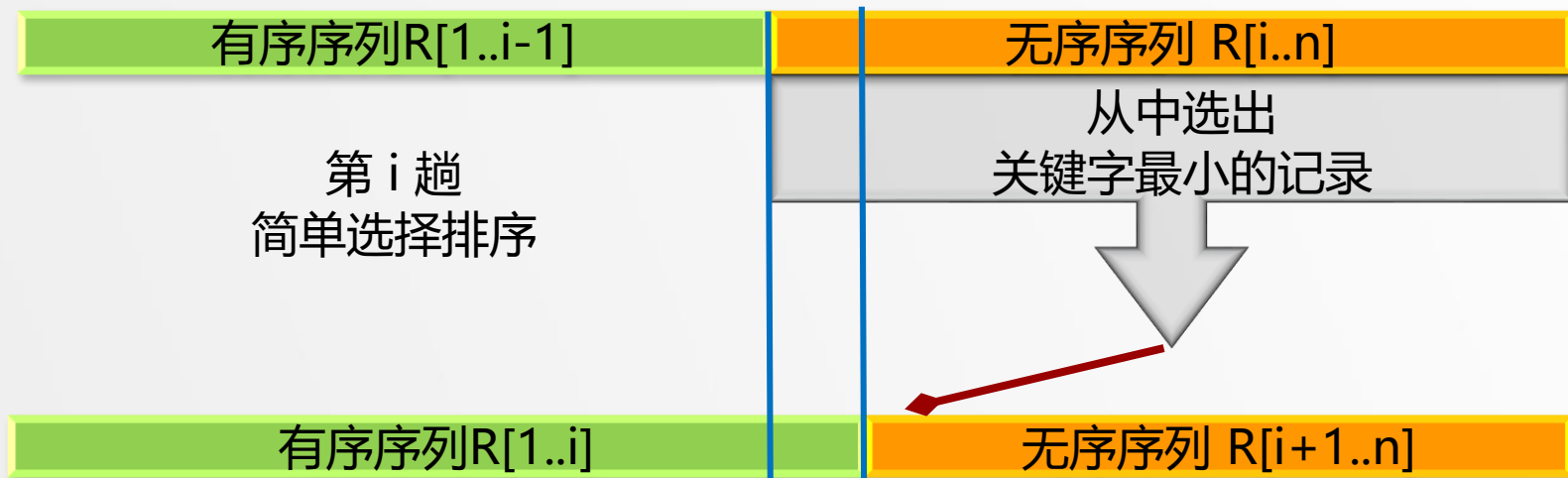


# 简单选择排序

## 简单选择排序

**基本思想：**从无序子序列中“**选择**”关键字**最小或最大**的记录，并将它**加入到有序子序列**中，以此方法增加记录的有序子序列的长度。

假设排序过程中，待排记录序列的状态为：



## 排序过程

- 首先通过 $n-1$ 次关键字比较，从 $n$ 个记录中找出关键字最小的记录，将它与第一个记录交换
- 再通过 $n-2$ 次比较，从剩余的 $n-1$ 个记录中找出关键字次小的记录，将它与第二个记录交换
- 重复上述操作，共进行 $n-1$ 趟排序后，排序结束

例: [ 49    38    65    97    76    13    27 ]

一趟:    13    [38    65    97    76    49    27 ]

二趟:    13    27    [65    97    76    49    38 ]

三趟:    13    27    38    [97    76    49    65 ]

四趟:    13    27    38    49    [76    97    65 ]

五趟:    13    27    38    49    65    [97    76 ]

六趟:    13    27    38    49    65    76    [97 ]

排序结束:    13    27    38    49    65    76    97

## 算法描述

```
void smp_selesort(JD r[], int n) {  
    int i, j, k;  
    JD x;  
    for (i = 1; i < n; i++) {  
        k = i;  
        for (j = i + 1; j <= n; j++)  
            if (r[j].key < r[k].key) k = j;  
        if (i != k) {  
            x = r[i];  
            r[i] = r[k];  
            r[k] = x;  
        }  
    }  
}
```

## ❖ 算法评价

### ⌘ 时间复杂度

#### ❖ 记录移动次数

☆ 最好情况: 0

☆ 最坏情况:  $3(n-1)$

❖ 比较次数:  $\sum_{i=1}^{n-1} (n-i) = \frac{1}{2}(n^2 - n)$

$$T(n) = O(n^2)$$

⌘ 空间复杂度:  $S(n) = O(1)$

## 简单选择排序性能分析

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 为：

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

移动记录的次数，最小值为 0, 最大值为  $3(n-1)$

$$T(n) = O(n^2)$$

$$\text{空间复杂度: } S(n) = O(1)$$

## 稳定性分析

- 初始数据: 3,3,1,4
- 第一趟排序: 1,3,3,4

不稳定!

- 1,3,3,4
- 1,3,3,4