

# 选择排序介绍\*

软件学院实训基地编辑

## 1 基本思想

选择排序的算法思想是逐个找出第*i*小的记录，并将其放在数组的第*i*个位置。选择排序的关键在于如何从剩余的未排序记录中找出最小（或最大的那个记录）。本文将简单介绍直接选择排序算法。

## 2 直接选择排序

### 2.1 算法实现

下面介绍一种直接选择排序，如算法1所示。图1给出了一个直接选择排序的例子。

#### 算法 1 直接选择排序

```
1: template <class Record>
2: void SelectSort(Record Array[], int n){
3:   int i, j, Smallest;
4:   for (i = 0; i < n - 1; i++) do
5:     Smallest=i;
6:     for (j = i + 1; j < n; j++) do
7:       if (Array[j]<Array[Smallest]) then
8:         Smallest=j;
9:       end if
10:    swap(Array, i, Smallest);
11:   end for
12: end for
13: }
```

### 2.2 算法复杂度分析

直接选择排序外循环迭代 $n - 1$ 次，第*i*次循环内循环迭代 $(n - 1 - i)$ 次，需要进行 $n - 1 - i$ 次比较，1次交换（3次移动）。因此总的比较次数如公式1所示。因此直接选择排序总的时间代价为 $\Theta(n^2)$ 。算法不依赖于原始数据的输入顺序，因此最大、最小、平均时间代价均为 $\Theta(n^2)$ 。

由于算法中使用了交换操作，需要用到一个临时记录，因此空间代价为 $\Theta(1)$ 。

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

\*本文内容全部来自于《数据结构与算法》，张铭等，高等教育出版社



图 1: 选择排序案例（图片来自百度）

### 3 常用排序算法总结

本节我们总结一下常用的排序算法，表1列出了常用排序算法的平均时间复杂度，空间复杂度和稳定性。

表 1: 常用排序算法比较

排序方法	平均时间复杂度	空间复杂度	稳定性
直接插入排序	$O(n^2)$	$O(1)$	稳定
Shell排序	$O(n^{1.3})$	$O(1)$	不稳定
直接选择排序	$O(n^2)$	$O(1)$	不稳定
堆排序	$O(n \log_2 n)$	$O(1)$	不稳定
冒泡排序	$O(n^2)$	$O(1)$	稳定
快速排序	$O(n \log_2 n)$	$O(n \log_2 n)$	不稳定
归并排序	$O(n \log_2 n)$	$O(1)$	稳定
基数排序	$O(d(r + n))$	$O(rd + n)$	稳定