

# 排序算法总结

---

## 一、排序算法的介绍

排序也称排序算法 (Sort Algorithm)，排序是将一组数据，依指定的顺序进行排列的过程。

## 二、排序的分类：

1) 内部排序：

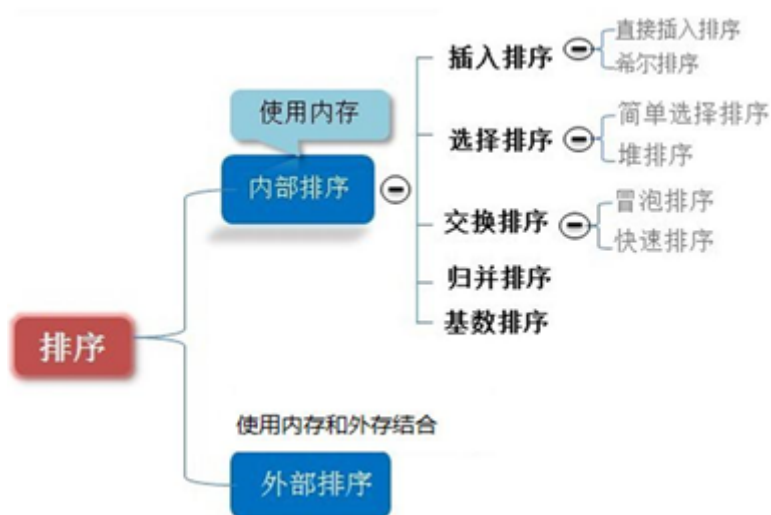
指将需要处理的所有数据都加载到内部存储器(内存)中进行排序。

2) 外部排序法：

数据量过大，无法全部加载到内存中，需要借助外部存储(文件等)进行排序。

3) 常见的排序算法分类(见右图)：

**折半插入排序比插入排序强**



### 三、排序算法对比

类别	排序方法	时间复杂度			空间复杂度	稳定性
		平均情况	最好情况	最坏情况	辅助存储	
插入排序	直接插入	$O(n^2)$	$O(n)$	$O(n^2)$	$O(1)$	稳定
	Shell排序	$O(n^{1.3})$	$O(n)$	$O(n^2)$	$O(1)$	不稳定
选择排序	直接选择	$O(n^2)$	$O(n^2)$	$O(n^2)$	$O(1)$	不稳定
	堆排序	$O(n\log_2 n)$	$O(n\log_2 n)$	$O(n\log_2 n)$	$O(1)$	不稳定
交换排序	冒泡排序	$O(n^2)$	$O(n)$	$O(n^2)$	$O(1)$	稳定
	快速排序	$O(n\log_2 n)$	$O(n\log_2 n)$	$O(n^2)$	$O(\log_2 n)$	不稳定
归并排序		$O(n\log_2 n)$	$O(n\log_2 n)$	$O(n\log_2 n)$	$O(n)$	稳定
基数排序		$O(d(r+n))$	$O(d(n+rd))$	$O(d(r+n))$	$O(rd+n)$	稳定

注：基数排序的复杂度中， $r$ 代表关键字的基数， $d$ 代表长度， $n$ 代表关键字的个数。

排序算法	平均时间复杂度	最好情况	最坏情况	空间复杂度	排序方式	稳定性
冒泡排序	$O(n^2)$	$O(n)$	$O(n^2)$	$O(1)$	In-place	稳定
选择排序	$O(n^2)$	$O(n^2)$	$O(n^2)$	$O(1)$	In-place	不稳定
插入排序	$O(n^2)$	$O(n)$	$O(n^2)$	$O(1)$	In-place	稳定
希尔排序	$O(n \log n)$	$O(n \log^2 n)$	$O(n \log^2 n)$	$O(1)$	In-place	不稳定
归并排序	$O(n \log n)$	$O(n \log n)$	$O(n \log n)$	$O(n)$	Out-place	稳定
快速排序	$O(n \log n)$	$O(n \log n)$	$O(n^2)$	$O(\log n)$	In-place	不稳定
堆排序	$O(n \log n)$	$O(n \log n)$	$O(n \log n)$	$O(1)$	In-place	不稳定
计数排序	$O(n + k)$	$O(n + k)$	$O(n + k)$	$O(k)$	Out-place	稳定
桶排序	$O(n + k)$	$O(n + k)$	$O(n^2)$	$O(n + k)$	Out-place	稳定
基数排序	$O(n \times k)$	$O(n \times k)$	$O(n \times k)$	$O(n + k)$	Out-place	稳定

### 相关术语解释：

1) 稳定：如果 a 原本在 b 前面，而  $a=b$ ，排序之后 a 仍然在 b 的前面；

冒泡排序、插入排序、归并排序和基数排序

2) 不稳定：如果 a 原本在 b 的前面，而  $a=b$ ，排序之后 a 可能会出现在 b 的后面；

选择排序、快速排序、希尔排序、堆排序

3) 内排序：所有排序操作都在内存中完成；

4) 外排序：由于数据太大，因此把数据放在磁盘中，而排序通过磁盘和内存的数据传输才能进行；

5) 时间复杂度：一个算法执行所耗费的时间。

6) 空间复杂度：运行完一个程序所需内存的大小。

7) n：数据规模

8) k：“桶”的个数

9) In-place: 不占用额外内存

10) Out-place: 占用额外内存

## 四、应用场景

(1) 若 $n$ 较小(如 $n \leq 50$ )，可采用直接插入或直接选择排序。

当记录规模较小时，直接插入排序较好；否则因为直接选择移动的记录数少于直接插入，应选直接选择排序为宜。

(2) 若文件初始状态基本有序(指正序)，则应选用直接插入、冒泡或随机的快速排序为宜；

(3) 若 $n$ 较大，则应采用时间复杂度为 $O(n \lg n)$ 的排序方法：快速排序、堆排序或归并排序。

- 快速排序是目前基于比较的内部排序中被认为是最好的方法，当待排序的关键字是随机分布时，快速排序的平均时间最短；
- 堆排序所需的辅助空间少于快速排序，并且不会出现快速排序可能出现的最坏情况。这两种排序都是不稳定的
- 若要求排序稳定，则可选用归并排序。但前面介绍的从单个记录起进行两两归并的排序算法并不值得提倡，通常可以将它和直接插入排序结合在一起使用。先利用直接插入排序求得较长的有序子序列，然后再两两归并之。因为直接插入排序是稳定的，所以改进后的归并排序仍是稳定的