阅读：973　　作者：解学武

内部排序算法的优势分析

本章介绍了以下几种常见的排序算法：

- 插入排序：直接插入排序、折半插入排序、2-路插入排序、表插入排序和希尔排序；
- 起泡排序（冒泡排序）；
- 快速排序（快排）；
- 选择排序：简单选择排序、树形选择排序和堆排序；
- 归并排序；
- 基数排序；

时间性能上的分析

排序方法	平均时间	最坏情况	辅助存储空间
简单排序	$O(n^2)$	$O(n^2)$	O(1)
快速排序	O(nlogn)	$O(n^2)$	O(logn)
堆排序	O(nlogn)	O(nlogn)	O(1)
归并排序	O(nlogn)	O(nlogn)	O(n)
基数排序	O(d*n)	O(d*n)	O(d*n)

> 上表中的简单排序包含出希尔排序之外的所有插入排序，起泡排序和简单选择排序。同时表格中的 n 表示无序表中记录的数量；基数排序中的 d 表示进行分配和收集的次数。

在上表表示的所有“简单排序算法”中，以直接插入排序算法最为简单，当无序表中的记录数量 n 较小时，选择该算法为最佳排序方法。

所有的排序算法中单就平均时间性能上分析，快速排序算法最佳，其运行所需的时间最短，但其在最坏的情况下的时间性能不如堆排序和归并排序；堆排序和归并排序相比较，当无序表中记录的数量 n 较大时，归并排序所需时间比堆排序短，但是在运行过程中所需的辅助存储空间更多（以空间换时间）。

从基数排序的时间复杂度上分析，该算法最适用于对 n 值很大但是关键字较小的序列进行排序。

在所有基于“比较”实现的排序算法中（以上排序算法中除了基数排序，都是基于“比较”实现），其在最坏情况下能达到的最好的时间复杂度为 O(nlogn)。

算法稳定性

本章所介绍的所有排序算法中，选择排序、快速排序和希尔排序都不是稳定的排序算法；而冒泡排序、插入排序、归并排序和基数排序都是稳定的排序算法。

算法实现的存储结构

本章所介绍的大多数算法都是在顺序存储结构的基础上实现的，基于顺序存储结构的局限性，排序算法在排序过程都需要进行大量记录的移动，影响算法本身的效率。

当无序表中记录的数量很大时，就需要采用静态链表替换顺序存储结构，例如：表插入排序、链式基数排序算法，是以修改指针代替大量移动记录的方式提高算法效率。

本章小结

通过比较所有的排序算法，没有哪一种是绝对最优的，在使用时需要根据不同的实际情况适当选择合适的排序算法，甚至可以考虑将多种排序算法结合起来使用。