

重要排序算法的总结

前置知识：之前讲的所有排序，本节课涉及的所有排序，之前的视频都讲了

稳定性

排序算法的稳定性是指：同样大小的样本在排序之后不会改变原始的相对次序

每个算法都说明一下

稳定性对基础类型对象来说毫无意义；稳定性对非基础类型对象有意义，可以保留之前的相对次序

主要算法时间、空间、稳定性总结

	时间	空间	稳定性
<i>SelectionSort</i>	$O(N^2)$	$O(1)$	无
<i>BubbleSort</i>	$O(N^2)$	$O(1)$	有
<i>InsertionSort</i>	$O(N^2)$	$O(1)$	有
<i>MergeSort</i>	$O(N \cdot \log N)$	$O(N)$	有
<i>QuickSort</i>	$O(N \cdot \log N)$	$O(\log N)$	无
<i>HeapSort</i>	$O(N \cdot \log N)$	$O(1)$	无
<i>CountSort</i>	$O(N)$	$O(M)$	有
<i>RadixSort</i>	$O(N)$	$O(M)$	有

注意：随机快速排序的复杂度一定要按照概率上的期望指标来估计，用最差的复杂度估计无意义，随机快排讲解视频里已经有详细的说明

重要排序算法的总结

基于比较的排序，时间复杂度 $O(n \log n)$ ，空间复杂度低于 $O(n)$ ，还具有稳定性的排序算法目前没有找到
*TimSort*也不行，虽然在实际应用中*TimSort*通常不需要这么多的额外空间，但空间复杂度指标就是 $O(n)$
有兴趣的同学可以研究，但是在算法面试、笔试、比赛中都很少用到*TimSort*算法
同时还有希尔排序(*ShellSort*)也不常用，有兴趣的同学可以研究一下，就是加入步长调整的插入排序

所以，一切看你在排序过程中在乎什么

数据量非常小的情况下可以做到非常迅速：插入排序

性能优异、实现简单且利于改进（面对不同业务可以选择不同划分策略）、不在乎稳定性：随机快排

性能优异、不在乎额外空间占用、具有稳定性：归并排序

性能优异、额外空间占用要求 $O(1)$ 、不在乎稳定性：堆排序