排序算法复杂度分析

# 快速排序quickSort

快速排序是由东尼·霍尔所发展的一种排序算法。**在平均状况下，排序n个元素要O(nlogn)次比较**。**在最坏状况下则需要O(n^2)次比较**，但这种状况并不常见。事实上，快速排序通常明显比其他**O(nlogn)算法**更快，**因为它的内部循环可以在大部分的架构上很有效率地被实现出来**。

　　快速排序使用**分治策略(Divide and Conquer)**来把一个序列分为两个子序列。

　　快速排序是**不稳定的排序算法**，不稳定发生在基准元素与A[tail+1]交换的时刻。

　　比如序列：{ 1, 3, 4, 2, 8, 9, 8, 7, 5 }，基准元素是5，一次划分操作后5要和第一个8进行交换，从而改变了两个元素8的相对次序。

Java系统提供的Arrays.sort函数。对于基础类型，底层使用**快速排序**。对于非基础类型，底层使用**归并排序**。请问是为什么？

答：这是考虑到排序算法的稳定性。对于基础类型，相同值是无差别的，排序前后相同值的相对位置并不重要，所以选择更为高效的快速排序，尽管它是不稳定的排序算法；而对于非基础类型，排序前后相等实例的相对位置不宜改变，所以选择稳定的归并排序。

# 归并排序