如何实现一个通用的高性能的排序函数?

一、如何选择合适的排序算法?

1.排序算法一览表

	时间复杂度	是稳定排序?	是原地排序?
冒泡排序	O(n^2)	是	是
插入排序	O(n^2)	是	是
选择排序	O(n^2)	否	是
快速排序	O(nlogn)	否	是
归并排序	O(nlogn)	是	否
桶排序	O(n)	是	否
计数排序	O(n+k)，k是数据范围	是	否
基数排序	O(dn)，d是纬度	是	否

2.为什选择快速排序?

1）线性排序时间复杂度很低但使用场景特殊，如果要写一个通用排序函数，不能选择线性排序。

2）为了兼顾任意规模数据的排序，一般会首选时间复杂度为O(nlogn)的排序算法来实现排序函数。

3）同为O(nlogn)的快排和归并排序相比，归并排序不是原地排序算法，所以最优的选择是快排。

二、如何优化快速排序?

导致快排时间复杂度降为O(n)的原因是分区点选择不合理，最理想的分区点是：被分区点分开的两个分区中，数据的数量差不多。如何优化分区点的选择？有2种常用方法，如下：

1.三数取中法

①从区间的首、中、尾分别取一个数，然后比较大小，取中间值作为分区点。

②如果要排序的数组比较大，那“三数取中”可能就不够用了，可能要“5数取中”或者“10数取中”。

2.随机法：每次从要排序的区间中，随机选择一个元素作为分区点。

3.警惕快排的递归发生堆栈溢出，有2中解决方法，如下：

①限制递归深度，一旦递归超过了设置的阈值就停止递归。

②在堆上模拟实现一个函数调用栈，手动模拟递归压栈、出栈过程，这样就没有系统栈大小的限制。

三、通用排序函数实现技巧

1.数据量不大时，可以采取用时间换空间的思路

2.数据量大时，优化快排分区点的选择

3.防止堆栈溢出，可以选择在堆上手动模拟调用栈解决

4.在排序区间中，当元素个数小于某个常数是，可以考虑使用O(n^2)级别的插入排序

5.用哨兵简化代码，每次排序都减少一次判断，尽可能把性能优化到极致

四、思考

1.Java中的排序函数都是用什么排序算法实现的？有有哪些技巧?