快速排序算法

1.介绍：

　　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 个项目要Ο(n log n)次比较。在最坏状况下则需要Ο(n2)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Ο(n log n) 算法更快，因为它的内部循环(inner loop)可以在大部分的架构上很有效率地被实现出来，且在大部分真实世界的数据，可以决定设计的选择，减少所需时间的二次方项之可能性。

2.步骤：

　　从数列中挑出一个元素，称为 "基准"(pivot)，重新排序数列，所有元素比基准值小的摆放在基准前面，所有元素比基准值大的摆在基准的后面(相同的数可以到任一边)。在这个分区退出之后，该基准就处于数列的中间位置。这个称为分区(partition)操作。

　　递归地(recursive)把小于基准值元素的子数列和大于基准值元素的子数列排序。

3.排序效果：



4.详细过程：

快速排序采用的思想是分治思想。

快速排序是找出一个元素（理论上可以随便找一个）作为基准(pivot),然后对数组进行分区操作,使基准左边元素的值都不大于基准值,基准右边的元素值 都不小于基准值，如此作为基准的元素调整到排序后的正确位置。递归快速排序，将其他n-1个元素也调整到排序后的正确位置。最后每个元素都是在排序后的正 确位置，排序完成。所以快速排序算法的核心算法是分区操作，即如何调整基准的位置以及调整返回基准的最终位置以便分治递归。

举例说明一下吧，这个可能不是太好理解。假设要排序的序列为

2 2 4 9 3 6 7 1 5 首先用2当作基准，使用i j两个指针分别从两边进行扫描，把比2小的元素和比2大的元素分开。首先比较2和5，5比2大，j左移

2 2 4 9 3 6 7 1 5 比较2和1，1小于2，所以把1放在2的位置

2 1 4 9 3 6 7 1 5 比较2和4，4大于2，因此将4移动到后面

2 1 4 9 3 6 7 4 5 比较2和7，2和6，2和3，2和9，全部大于2，满足条件，因此不变

经过第一轮的快速排序，元素变为下面的样子

[1] 2 [4 9 3 6 7 5]

之后，在把2左边的元素进行快排，由于只有一个元素，因此快排结束。右边进行快排，递归进行，最终生成最后的结果。