# 排序

## 选择排序

选择排序的思想：是每一趟在剩余待排序的元素（包括本位置）中选择最大或者最小的元素作为当前位置的元素。

选择排序的时间复杂度和数据的输入无关，任何情况下都是o（n2），选择排序的交换次数最大仅仅为n-1，这是选择排序相对于其它排序算法的一个优点。选择排序不稳定，比如5,8,5,2,9.两个5的位置会发生改变。比较次数始终为n（n-1）/2。

## 冒泡排序

冒泡排序的基本思想：从前往后或者从后往前两两比较元素，若是逆序则交换元素，每一趟将一个元素放在正确的位置，下次不参与冒泡。最多冒泡n-1次可以将元素排序好。在冒泡时，如果在过程中不发生交换，则说明有序，停止此过程。

时间复杂度最好的情况为o（n），平均和最差为o（n2）。冒泡排序是一个稳定的排序算法。

以上两种算法属于蛮力思想。

## 插入排序

直接插入排序：

直接插入排序的思想：每次将一个元素插入到已排序好的序列中，和选择排序不同，这个元素的位置不一定是最终的位置。减治的思想。

时间复杂度最好的情况为o（n），平均和最差为o（n2）。插入排序冒泡排序是一个稳定的排序算法。

优化:可以使用二分查找找到要插入的位置。

希尔排序：

取一个小于n的步长d1,将元素分为d1个组，所有距离为d1倍数的元素放在同一个组中，然后对每个组进行直接插入排序；然后取第二个步长d2<d1,重复以上步骤，直到步长为1。所有的元素放在一个组中，再直接进行插入排序，由于已经具有较好的局部性，故可以很快的得到最终的结果。希尔提出的一个较好的增量序列为d1 = n/2, di+1 = di/2,最后的增量为1。

希尔排序不稳定

## 归并排序

归并排序初始将每一个元素视为一个子表，然后两两归并，得到n/2个长度为1 或者为2的有序表，然后再两两归并，直到合成一个有序表为止。两两归并时先将两个表复制到辅助的数组中，然后每次从两个表中选出一个元素比较，较小的放入原数组，如果其中一个表检测完，则直接将辅助数组剩余元素复制到原数组中。

时间复杂度为o（nlogn）。空间复杂度为o（n）。快速排序是一个稳定的排序算法。

## 快速排序

快速排序在所有的元素pivot中选取一个数作为基准，将其它元素划分为两部分，其中一部分元素都小于piovt，另外一部分元素都大于piovt。基准元素piovt放在了正确的位置。然后对两部分递归的重复上面的过程，直到每部分只有一个元素或者为空为止。

时间复杂度平均和最好的情况为o（nlogn），最差为o（n2）。空间复杂度最好为o（logn），最坏为o（n）。快速排序是一个不稳定的排序算法。

快速排序是所有内部排序算法中平均性能最优的排序算法。

以上两个算法都运用了分而治之的思想。