# **[常见排序算法思想](https://www.cnblogs.com/tornadomeet/archive/2013/04/18/3027591.html)**

1. 一般约定我们是按照从小到大的顺序排序。
2. 按照向量的形式来参考排序中数组元素的相对位置，行向量中最左边为数组的第一个元素，且称为前面。列向量中最上面的元素为数组的第一个元素，也是称为前边。
3. 对常见的排序算法的思想要有个认识，比如说：冒泡排序，选择排序，插入排序，快速排序，shell排序，堆排序，归并排序，基排序。
4. 算法的稳定性是看原先数据中相同的元素值的顺序在排序后是否改变了？如果依旧保持不变，则说明该算法是稳定的。比如说，选择排序、快速排序、希尔排序、堆排序不是稳定的排序算法，而冒泡排序、插入排序、归并排序和基数排序是稳定的排序算法。
5. 选择排序是每次找到数据中最大的或者最小的数，然后和第一个数交换，每次循环比较多次，但是只交换一次。而交换排序是选定一个元素后，数组中该元素后面的元素依次与选定的元素进行比较，如果后面的元素比该元素小，则交换，直到扫描到数组尾部，所以交换排序每一趟扫描可能会交换多次。选择排序和交换排序都是非稳定的排序方法。
6. 一般来说，插入排序都采用in-place在数组上实现。具体算法描述如下：
   1. 从第一个元素开始，该元素可以认为已经被排序
   2. 取出下一个元素，在已经排序的元素序列中从后向前扫描
   3. 如果该元素（已排序）大于新元素，将该元素移到下一位置
   4. 重复步骤3，直到找到已排序的元素小于或者等于新元素的位置
   5. 将新元素插入到该位置后
   6. 重复步骤2~5。
   7. 如果比较操作的代价比交换操作大的话，可以采用二分查找法来减少比较操作的数目。该算法可以认为是插入排序的一个变种，称为二分查找排序。（插入排序参考的是维基百科[插入排序- 维基百科](https://www.google.com.tw/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=0CCkQFjAA&url=https://zh.wikipedia.org/zh-tw/%E6%8F%92%E5%85%A5%E6%8E%92%E5%BA%8F&ei=Q3GgUffMGYTL4AS72YGIBw&usg=AFQjCNHFs7S1DllhwvrCJv9Z8HG-u6uTYg&sig2=KCxVKfFCSTuJS60a7fhvTw&bvm=bv.47008514,d.bGE&cad=rjt" \t "/home/ydwu/文档\\x/_blank)）
7. 快速排序的思想是在数据中选出一个数据x，以该数据为基点，将小于它的数据放左边，大于它的数据放右边（方法是：首先选取一个pivot，将其放在数组的最后。然后从数据的最后一个元素（此时不包括pivot那个数）向前一直到x，找出第一个比x小的元素，同时从数据序列中第一个位置开始到新的x位置找出第一个比x大的元素，交换这两者（一个大于pivot的数和一个小于pivot的数）的位置，依次交替进行，直到所有的数字已经比较过），当将数据分好为2部分后，分别在各自的部分采用相同的方法进行排列，最终直到分出的组元素个数为0或者为1时结束排序。参考[常见排序算法的实现(五)-快速排序](http://c.chinaitlab.com/c/basic/200905/785203.html)。
8. Shell排序是将原始数据按照一定的增量分为几个小组，然后在每个小组内部使用插入排序方法，待每个小组内都排序完后重新按照前面的位置组合这些数据，接着采用更大的增量值将数据分为更少的组，同样的在组内用插入排序方法进行排序，依次类推，直到最后增量最大即只分为一个组时，排序完后就结束。可以参考维基百科中[希尔排序](http://zh.wikipedia.org/wiki/%E5%B8%8C%E5%B0%94%E6%8E%92%E5%BA%8F)的例子，一目了然。
9. 如果要对数据按照从小到大的堆排序，则应该将这些数据初始化为大堆，反之亦然。堆排序是按照将数据先排列成堆（此时采用的方法是每次从最后一个非叶子节点开始，将以该节点为根节点的调树整成对应的堆形式，然后依次递减找到倒数第2，第3，…等一直到根节点），然后每次将排列好的堆的第一个数据与最后一个叶子节点交换，交换完后重新对原先的堆（少了一个节点）调整为大堆。具体可以参考前面的博文：[算法设计和数据结构学习\_1(一道堆排序作业题)](http://www.cnblogs.com/tornadomeet/archive/2012/11/07/2759007.html)。
10. 归并排序和快速排序一样，也是采用的“分治”思想。方法是：先将数据分成n组（一般每组为2个元素），每组内从小到大排序，然后申请一个2个组大小的空间，并分别设置指针指向那2个组的初始元素，比较两个指针所指向的元素，选择相对小的元素放入到合并空间，并移动指针到下一位置，重复这一步骤直到某一指针达到序列尾。将另一序列剩下的所有元素直接复制到合并序列尾。继续用同样的方法进行归并。可以参考维基百科的[归并排序](http://zh.wikipedia.org/wiki/%E5%BD%92%E5%B9%B6%E6%8E%92%E5%BA%8F)。
11. 基数排序是将数据集中一次每个数据的个位，十位，千位等放入标有序列号的桶中，最后将桶中的数据重新组合成最终的排序序列，该算法是稳定的。可以参考[基数排序](http://baike.baidu.com/view/1170573.htm)里面的例子。

常见的排序算法复杂度比较：　　