

排序方法	最好时间	平均时间	最坏时间	辅助空间	稳定性	备注
直接插入排序	$O(n)$	$O(n^2)$	$O(n^2)$	$O(1)$	稳定	基本思路：顺序地把待排序的数据元素按大小插入到已经排序的数据元素的子集中，排序过程如下：
希尔排序（缩小增量直接插入艾许）		$O(n^{1.3})$		$O(1)$	不稳定	shell 排序基本原理是把待排序的数据，分成若干小组，然后对同一小组使用直接插入排序，这样，小组的个数逐次减少，当所有元素在同一小组时完成排序过程。
直接选择排序	$O(n^2)$	$O(n^2)$	$O(n^2)$	$O(1)$	不稳定	直接选择排序：每次从待排序中选择最小的放到最前面，如下图
堆排序	$O(n\log_2 n)$	$O(n\log_2 n)$	$O(n\log_2 n)$	$O(1)$	不稳定	在直接选择排序中，待排序的数据元素集合构成一个线性表结构，要从 n 个数据元素的线性表中选择一个最小的元素需要 $n-1$ 次，如果能把待排序的数据集合构成一个完全二叉树结构，则每次选择最大或者最小的元素只需要 $\log_2 N$ ，所以排序算法的时间复杂度就会变为 $O(n\log_2 n)$ 。这就是堆排序的思想。
冒泡排序	$O(n)$	$O(n^2)$	$O(n^2)$	$O(1)$	稳定	冒泡排序的基本思想是：第一次时，设数组 a 中存放了 n 和数据元素，循环进行 $n-1$ 次比较相邻两个元素大小，若前一个大于后一个则交换，这样作为 $n-1$ 次后，最大的元素就放置在 $a[n-1]$ 中，第二次同理最大元素放置到 $a[n-2]$ 中，以此类推，当 $n-1$ 次排序后，数组能够得到排序。 冒泡法的一个优化，通过设计一个 flag 变量，flag 常量用于标记本次过程是否有交换动作，若没有交换动作则证明已经排序完成，无需后续操作。
快速排序	$O(n\log_2 n)$	$O(n\log_2 n)$	$O(n^2)$	$O(n\log_2 n)$	不稳定	设数组 a 中存放了 n 个数据元素， low 为数组的低端下标， $high$ 为数组的高端下标，从数组 a 中去一个元素通常取 $a[low]$ 作为标注，调整数组 a 中各个元素的位置，使排在标准元素前面的关键字均小于标准元素的关键字，排在标准元素后面的关键字均大于后者等于标准元素的关键字，这样一次排序过程后，一方面将标准元素放在了未来排好序的位置，另一方面，将数组中以该标准元素分为两个子数组，左边的均小于标准值，右边的均大于标准值，然后对这两个子数组中的元素分别进行相同的递归快速排序，递归的出口条件是 $high > low$ 。
归并排序	$O(n\log_2 n)$	$O(n\log_2 n)$	$O(n\log_2 n)$	$O(n)$	稳定	归并排序主要是二路归并排序。二路归并排序的基本思想是：设数组 a 中存放了 n 个数据元素，初始时把他们看成是 n 个长度为 1 的有序子数组，然后从第一个子数组开始，把相邻的子数组亮亮合并，得到 $n/2$ 的整数上界个长度为 2 的新的有序子数组，当 n 为奇数时，最后一个有序子数组的长度为 1 ；对这些新的有序的子数组再两两归并；如此重复，直到得到一个长度为 n 的有序数组为止。
基数排序（链式队列）	$O(mn)$	$O(mn)$	$O(mn)$	$O(n)$	稳定	基本思想如下：设待排序的数据元素关键字是 m 位 d 进制的整数，不足 m 位的高位补零，设置 d 个桶，令其编号分别为 $0, 1, 2, \dots, d-1$ 。首先按关键字最低位的数值依次把各数据元素放到对应的桶中，这样按照桶号从小到大和进入桶中的先后顺序来一次收集分配在桶中的元素，这就是一次基数排序，同理进行接下来较高一位的排序，一直到第 m 次排序完成，那么就可以得到排好序的数据元素序列。