

分析各种排序算法的运行过程

Target: 通过实际的示例，理解和分析四种常见的排序算法——冒泡排序、插入排序、归并排序、和快速排序——在处理整数序列时的执行过程。

示例

冒泡算法

顾名思义，小V大往上浮，不断迭代有序队列。

时间复杂度 $O(n^2)$

示例模拟：

- 比较1和2，不需要交换。
- 比较2和3，不需要交换。
- 比较3和100，不需要交换。
- 比较100和20，交换它们，序列变为：1, 2, 3, 20, 100, 1, 19, 23, 50。
- 比较100和1，交换它们，序列变为：1, 2, 3, 20, 1, 100, 19, 23, 50。
- 比较100和19，交换它们，序列变为：1, 2, 3, 20, 1, 19, 100, 23, 50。
- 比较100和23，交换它们，序列变为：1, 2, 3, 20, 1, 19, 23, 100, 50。
- 比较100和50，交换它们，序列变为：1, 2, 3, 20, 1, 19, 23, 50, 100。

第二轮：

- 比较1和2，不需要交换。
- 比较2和3，不需要交换。
- 比较3和20，不需要交换。
- 比较20和1，交换它们，序列变为：1, 2, 3, 1, 20, 19, 23, 50, 100。
- 比较20和19，交换它们，序列变为：1, 2, 3, 1, 19, 20, 23, 50, 100。
- 比较20和23，不需要交换。
- 比较23和50，不需要交换。
- 比较50和100，不需要交换。

第三轮：

- 比较1和2，不需要交换。
- 比较2和3，不需要交换。
- 比较3和1，交换它们，序列变为：1, 2, 1, 3, 19, 20, 23, 50, 100。
- 比较3和19，不需要交换。
- 比较19和20，不需要交换。
- 比较20和23，不需要交换。

- 比较23和50，不需要交换。
- 比较50和100，不需要交换。

第四轮：

- 比较1和2，交换它们，序列变为：1, 1, 2, 3, 19, 20, 23, 50, 100。
- 比较2和3，不需要交换。
- 比较3和19，不需要交换。
- 比较19和20，不需要交换。
- 比较20和23，不需要交换。
- 比较23和50，不需要交换。
- 比较50和100，不需要交换。

经过四轮冒泡排序，整数序列已经排好序：1, 1, 2, 3, 19, 20, 23, 50, 100

插入排序

基本原理：插入排序是一种逐步构建有序序列的排序算法。它从第一个元素开始，将后续元素逐一插入到已排序的序列中，直到整个序列排好序为止。

对于给定的整数序列：1, 2, 3, 100, 20, 1, 19, 23, 50，插入排序的过程如下：

初始状态：有序部分：1，无序部分：2, 3, 100, 20, 1, 19, 23, 50。

第一步：将2插入到有序部分。

- 比较2和1，交换它们，序列变为：1, 2, 3, 100, 20, 1, 19, 23, 50。

第二步：将3插入到有序部分。

- 比较3和2，交换它们，序列变为：1, 2, 3, 100, 20, 1, 19, 23, 50。

... 依此类推，逐步插入剩余的元素。

最终排序完成，整数序列为：1, 1, 2, 3, 19, 20, 23, 50, 100。

归并排序

基本原理：归并排序是一种分治算法，它将序列分成小的子序列，然后将这些子序列合并成一个有序序列。合并的过程中，两个有序子序列会逐一比较元素，将较小的元素放入新的有序序列中。

对于给定的整数序列：1, 2, 3, 100, 20, 1, 19, 23, 50，归并排序的过程如下：

初始状态：将整个序列分成单个元素的子序列。

第一步：合并子序列。

- 合并(1)和(2)得到有序序列：1, 2。
- 合并(3)和(100)得到有序序列：3, 100。
- 合并(1)和(2)得到有序序列：1, 2。
- 合并(3, 100)得到有序序列：3, 100。

- 合并(1)和(2)得到有序序列：1, 2。
- 合并(19)和(23)得到有序序列：19, 23。
- 合并(50)和(1)得到有序序列：1, 50。
- 最后一次合并(2, 3, 100)和(1, 19, 23, 50)得到有序序列：1, 2, 3, 19, 23, 50, 100。

整个序列已经排好序：1, 2, 3, 19, 23, 50, 100。

快速排序

基本原理：快速排序是一种分治算法，它选择一个元素作为基准，将序列分成两部分，左边的部分包含小于基准的元素，右边的部分包含大于基准的元素。然后递归地对左右两部分进行快速排序。

对于给定的整数序列：1, 2, 3, 100, 20, 1, 19, 23, 50，快速排序的过程如下：

选择基准元素（通常选择序列的第一个元素，这里选择1）。

分区操作：

- 将序列分成两部分，小于1的放在左边，大于1的放在右边。
- 左边部分：1, 1，右边部分：2, 3, 100, 20, 19, 23, 50。
- 此时1已经在正确的位置上，左边部分和右边部分分别是有序的。

对左边部分（1, 1）和右边部分（2, 3, 100, 20, 19, 23, 50）分别进行快速排序。

左边部分已经有序，不需要再排序。

对右边部分进行分区操作：

- 选择基准元素（这里选择2）。
- 分区后，左边部分：2, 3, 20, 19, 1，右边部分：23, 50, 100。

继续对左边部分（2, 3, 20, 19, 1）和右边部分（23, 50, 100）进行快速排序。

左边部分分区操作：

- 选择基准元素（这里选择2）。
- 分区后，左边部分：2, 1，右边部分：3, 20, 19。

左边部分再次进行分区操作：

- 选择基准元素（这里选择2）。
- 分区后，左边部分：1, 2，右边部分：3, 20, 19。

左边部分已经有序，不需要再排序。

对右边部分（3, 20, 19）进行分区操作：

- 选择基准元素（这里选择3）。
- 分区后，左边部分：3，右边部分：20, 19。

左边部分已经有序，不需要再排序。

右边部分（20, 19）分区操作：

- 选择基准元素（这里选择20）。
- 分区后，左边部分：19，右边部分：20。

整个序列已经排好序：1, 1, 2, 3, 19, 20, 23, 50, 100。