排序技术 1

Tan Yiqing

2025 年 12 月 15 日

1 概念

1.1 定义

给定一组记录的集合 $R = \{r_1, r_2, \ldots, r_n\}$，其相应的关键码分别为 $K = \{k_1, k_2, \ldots, k_n\}$，排序是将这些记录排列成顺序为 $R' = \{r'_1, r'_2, \ldots, r'_n\}$ 的一个序列，使得相应的关键码满足 $k'_1 \leq k'_2 \leq \ldots \leq k'_n$ 或者 $k'_1 \geq k'_2 \geq \ldots \geq k'_n$。

排序是线性结构的一种操作。

排序算法的稳定性：假定在一组记录中，存在两个记录 r_i 和 r_j，它们的关键码相等，即 $k_i = k_j$，如果在排序前 r_i 位于 r_j 之前，经过排序后 r_i 仍然位于 r_j 之前，则称该排序算法是稳定的，否则称为不稳定的。

1.2 分类

根据不同的标准，排序可以分为以下几类：

按照顺序：

1. 升序排列：关键码从小到大排序
2. 降序排列：关键码从大到小排序。

按照依靠的关键码：

1. 单键排序：根据单个关键码进行排序
2. 多键排序：根据多个关键码进行排序。

按照存储方式：

1. 内排序：排序过程中，所有的记录都存放在内存中进行排序。
2. 外排序：排序过程中，部分记录存放在外存中进行排序。

按照排序方法：

1. 基于比较的排序：如插入排序、交换排序、选择排序、归并排序、快速排序、堆排序等。
2. 非基于比较的排序：如分配排序。

定理 1. *任意基于比较的排序算法在最坏情况下的时间复杂度下界为 $O(n \log n)$。*

证明. 设有 n 个待排序的互异元素。基于比较的排序可用一棵二叉决策树表示：每个内部节点为一次二分比较（两个子分支），每个叶子对应一种最终排列。因为元素互异，可能的排列数为 $n!$，故决策树的叶子数 $L = n!$。

在一棵每个内部结点都有两个孩子的二叉树中，度为 2 的内部结点数记为 n_2，叶子数为 n_0，则:

$$n_0 = n_2 + 1,$$

因此树的结点总数为

$$N = n_2 + n_0 = 2n_0 - 1 = 2n! - 1.$$

设决策树的高度为 h（根到叶的最大边数），高度为 h 的二叉树至多有 $2^h - 1$ 个结点，因此：

$$h \geqq \lfloor \log_2(2n! - 1) \rfloor + 1 \geqq \log_2(2n! - 1) \geqq \log_2(2n!)$$

利用 Stirling 公式估计：

$$n! \backsim \sqrt{2\pi n} \left(\frac{n}{e}\right)^n,$$

因此

$$h \backsim \Omega(n \log n).$$

即任意基于比较的排序在最坏情况下需要至少 $\Omega(n \log n)$ 次比较，从而时间下界为 $O(n \log n)$。 □

1.3 排序算法的性能

排序算法的性能通常通过以下几个方面来评估：

1. 基本操作
   (a) 比较次数：排序过程中进行关键码比较的次数。
   (b) 交换或移动次数：排序过程中记录交换或移动的次数。
2. 辅助存储空间：排序算法在执行过程中所需的额外存储空间。
3. 算法本身复杂程度。

2 插入排序 (Insertion Sort)

插入排序的主要操作是插入，其基本思想是每次将一个待排序的记录按其关键码的大小插入到一个已经排好序的有序序列中，直到全部记录排好序为止。

在插入第 i 个元素时，假设前 $i-1$ 个元素已经排序好，将第 i 个元素与前 $i-1$ 个元素依次比较，找到合适的位置插入。

	r[0]	r[1]	r[2]	r[3]	r[4]	r[5]	r[6]
初始		2	6	4	1	6	3
$i=2$	6	2	6	4	1	6	3
$i=3$	4	2	4	6	1	6	3
$i=4$	1	1	2	4	6	6	3
$i=5$	6	1	2	4	6	6	3
$i=6$	3	1	2	3	4	6	6

说明：每次将待插入元素暂存于 r[0]（红色 key），再在已排序区中右移元素并插入。

图 1: 插入排序示例：r[1..6]=[2,6,4,1,6,3]，r[0] 作为暂存位

简要步骤说明：

1. 初始：r[1..6]=[2,6,4,1,6,3]，r[0] 空。
2. 插入第 2 个：key=6 放入 r[0]，已排序区 [2]，6 插入到合适位置（此处保持原位）。
3. 插入第 3 个：key=4 存入 r[0]，6 右移，插入到 2 之后。
4. 插入第 4 个：key=1 存入 r[0]，2/4/6 依次右移，1 插入最前。
5. 依此继续直到处理完 r[6]，得到最终有序结果。

2.1 性能分析

时间复杂度 分析如下：

1. 最好情况：正序
   (a) 比较次数：$n-1$ 次。

(b) 移动次数：$2(n-1)$ 次。

时间复杂度为 $O(n)$。

2. 最坏情况：逆序

(a) 比较次数：$\Sigma_{i=2}^{n} i = \frac{(n+2)(n-1)}{2}$ 次。

(b) 移动次数：$\Sigma_{i=2}^{n}(i+1) = \frac{(n+4)(n-1)}{2}$ 次。

时间复杂度为 $O(n^2)$。

3. 平均情况：

(a) 比较次数：$\Sigma_{i=2}^{n} i/2 = \frac{(n+2)(n-1)}{2}$ 次。

(b) 移动次数：$\Sigma_{i=2}^{n}(i+1)/2 = \frac{(n+4)(n-1)}{2}$ 次。

时间复杂度为 $O(n^2)$。

空间复杂度 分析如下：

1. 辅助存储空间：$O(1)$，只需常数级别的额外空间用于存放 key。
2. 总体空间复杂度：$O(n)$，主要用于存放待排序数组。

稳定性 插入排序是稳定的排序算法。

2.2 改进方法

1. 使用折半查找代替顺序查找：在已排序序列中使用折半查找法定位待插入元素的位置，从而减少比较次数，但移动次数不变。
2. 希尔排序 (shell sort)：将整个待排序记录分割成若干个子序列，在子序列内分别进行直接插入排序，待整个序列中的记录基本有序时，对全体记录进行直接插入排序。

希尔排序 可以理解为，它让序列变得“基本有序”，从而提高效率

设 $r[1..9] = [7,5,8,5,3,4,1,6,2]$，取常用增量序列 $d = 4,2,1$。(除以 2 递降)

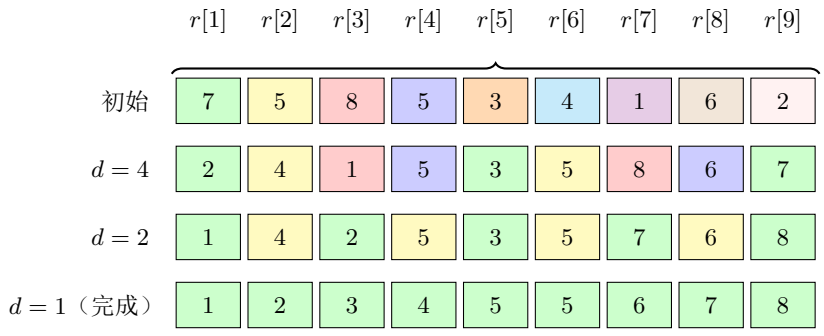


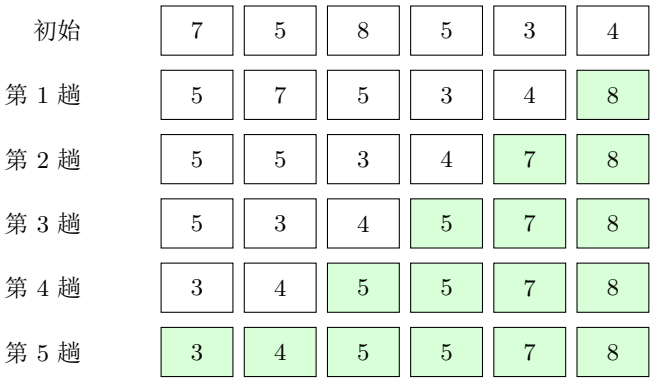
图 2: 希尔排序示例：$r[1..9] = [7,5,8,5,3,4,1,6,2]$，增量 $d = 4,2,1$

说明（以 $d = 4$ 为例）：子序列 $(r_1, r_5, r_9) = (7,3,2)$ 插入排序后为 $(2,3,7)$，回填到原下标 $1,5,9$；其余子序列同理处理。随后减小增量直到 $d = 1$，完成整体排序。

希尔排序的时间复杂度在不同增量序列下有所不同，一般介于 $O(n \log n)$ 和 $O(n^2)$ 之间，在某个特定范围内大致为 $O(n^{1.3})$。

3 交换排序 (Bubble Sort)

交换排序的主要操作是交换，其主要思想是：在待排序列中选两个记录，将它们的关键码相比较，如果反序（即排列顺序与排序后的次序正好相反），则交换它们的存储位置。也叫冒泡排序。



说明：每一趟从左到右比较相邻元素并交换，使当前最大元素逐步“冒泡”到序列末尾（绿色部分）。

图 3: 冒泡排序过程示例（升序）：初始序列 $[7, 5, 8, 5, 3, 4]$

3.1 性能分析

时间复杂度 分析如下：

1. 最好情况：正序
   (a) 比较次数：$n-1$ 次。
   (b) 交换次数：0 次。
   时间复杂度为 $O(n)$。
2. 最坏情况：逆序
   (a) 比较次数：$\Sigma_{i=1}^{n-1}(n-i) = \frac{n(n-1)}{2}$ 次。
   (b) 交换次数：$\Sigma_{i=1}^{n-1}3(n-i) = \frac{3n(n-1)}{2}$ 次。
   时间复杂度为 $O(n^2)$。
3. 平均情况：时间复杂度为 $O(n^2)$。

空间复杂度 分析如下：

1. 辅助存储空间：$O(1)$，只需常数级别的额外空间用于交换元素。
2. 总体空间复杂度：$O(n)$，主要用于存放待排序数组。

稳定性 冒泡排序是稳定的排序算法。

3.2 改进方法

冒泡排序的思想是通过交换相邻元素来实现排序，记录每次交换只能上移或下移一个单元，因而总的比较次数和移动次数较多。我们可以减少总的比较次数和移动次数，从而增大记录的比较和移动距离，把较大记录从前面直接移动到后面，较小记录从后面直接移动到前面。

快速排序 (Quick Sort) 快速排序的基本思想：选取一个枢轴（pivot），通过一次**划分（partition）**把序列分成三段

$$L = \{x \mid x < p\}, \quad E = \{x \mid x = p\}, \quad G = \{x \mid x > p\},$$

再递归地对 L 和 G 排序，最后拼接为 $L + E + G$。

示例（对序列 758534162 进行快速排序） 设初始序列 $[7, 5, 8, 5, 3, 4, 1, 6, 2]$，每次取**子序列第一个元素**为枢轴，采用三路划分（$<, =, >$）：

1. 取 $p = 7$，划分得 $L = [5, 5, 3, 4, 1, 6, 2]$, $E = [7]$, $G = [8]$。
2. 对 L 递归：取 $p = 5$，得 $L = [3, 4, 1, 2]$, $E = [5, 5]$, $G = [6]$。
3. 对 $[3, 4, 1, 2]$ 递归：取 $p = 3$，得 $L = [1, 2]$, $E = [3]$, $G = [4]$。
4. 对 $[1, 2]$ 递归：取 $p = 1$，得 $L = []$, $E = [1]$, $G = [2]$（递归结束）。
5. 自底向上拼接：得到最终有序序列 $[1, 2, 3, 4, 5, 5, 6, 7, 8]$。

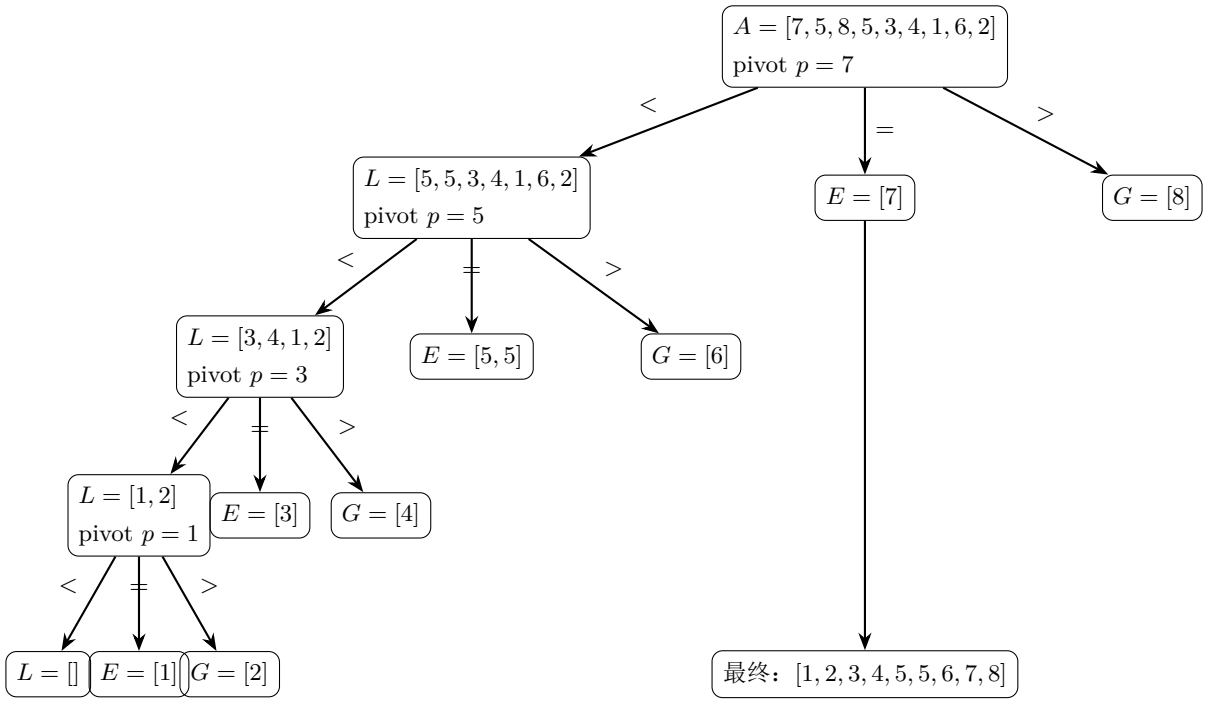


图 4: 快速排序示例（取子序列首元素为枢轴，三路划分 $<, =, >$）

3.3 性能分析

见排序技术 2