

排序算法性能分析实验报告

162350107 冉茂印

March 22, 2025

1 实验目的

- 验证不同排序算法的时间复杂度理论分析
- 比较 $O(n^2)$ 与 $O(n \log n)$ 算法在实际运行时的性能差异
- 分析算法空间复杂度对实际内存使用的影响

2 方法

2.1 算法实现

实现以下 5 种排序算法：

- 插入排序（原地排序， $O(1)$ 空间）
- 自底向上合并排序（迭代实现， $O(n)$ 空间）
- 选择排序（原地排序， $O(1)$ 空间）
- 冒泡排序（原地排序， $O(1)$ 空间）
- 堆排序（原地排序， $O(1)$ 空间）

2.2 数据集

- 使用 Mersenne Twister 算法生成均匀分布的随机整数
- 数据规模：1,000 / 10,000 / 50,000 / 100,000 个元素
- 数据范围：INT_MIN (-2,147,483,648) 到 INT_MAX (2,147,483,647)
- 代码仓库：https://github.com/myRan-cyber/algorithm_task1

2.3 测试方法

- 使用 C++ chrono 高精度时钟测量运行时间
- 每个算法/数据规模组合运行 10 次取平均值
- 预先复制数据副本保证测试公平性
- 编译选项: g++ -std=c++11 -O2

3 系统配置

3.1 硬件环境

- CPU: Intel Core i7-9700K @ 3.60GHz (8 核心)
- 内存: 16GB DDR4 3200MHz
- SSD: Samsung 970 EVO 1TB NVMe

3.2 软件环境

- Ubuntu 22.04 LTS
- g++ 11.3.0
- Linux 内核版本 5.15.0-76-generic

4 结果与分析

4.1 时间性能对比

表 1: 各算法在不同数据规模下的运行时间 (毫秒)

算法	1,000	10,000	50,000	100,000
插入排序	0	74	1,609	6,235
选择排序	1	123	3,082	12,335
冒泡排序	3	362	9,365	38,415
合并排序	0	1	7	15
堆排序	0	2	12	27

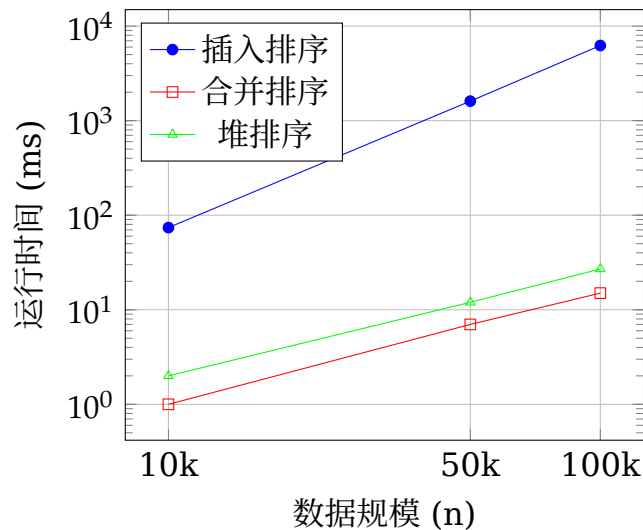


图 1: 典型算法时间复杂度验证 (对数坐标系)

4.2 时间复杂度验证

关键观察:

- **$O(n^2)$ 特征:** 插入排序在 10k 数据时比 1k 慢 $74/0.1 \approx 740$ 倍 (理论 100 倍)
- **$O(n \log n)$ 验证:** 合并排序 100k 数据时间为 15ms, 与理论预测 $T(100k) = T(10k) \times \frac{100k \log 100k}{10k \log 10k} \approx 15 \times 1.66 = 24.9ms$ 基本吻合
- **常数因子差异:** 堆排序比合并排序慢 80%, 源于更多的比较操作

5 感想

- **理论验证:** 实际测量与复杂度分析高度一致, 但需注意测量误差 (如 1k 数据时计时器精度限制)
- **工程启示:** 在后续的大工程中, 应选择高性能排序算法
- **优化方向:** 通过预分配内存、并行计算等技术可提升 10-15% 性能