**2024秋-《算法设计与分析》**

**分治法分析实验报告**

|  |
| --- |
| 2024年11月9日 |

**目 录**

[一、 实验要求 1](#_Toc182171546)

[二、 实验报告 1](#_Toc182171547)

[1.1 实验设计 1](#_Toc182171548)

[1.2 数据分析 2](#_Toc182171549)

# 实验要求

对几个经典的排序算法进行分析，理解算法在不同输入时的表现，深入剖析 对几种经典的排序算法进行分析，理解算法在不同输入时的表现，深入剖析算法优缺点及其根源。具体要求如下：

1.实现常见排序算法，至少要实现插入排序、冒泡排序、快速排序、归并排序、shell排序算法；

2.在排序算法中插桩，记录关键操作次数（如比较次数、移动次数等）；

3.以待排序文件的行数n为输入规模，固定n随机产生多组测试样本，统计算法的平均运行时间和关键操作次数，改变n的规模重复多次实验，并对结果进行统计；

4.改变数组规模，对不同规模问题下各算法的结果进行统计并绘制图表，与理论值进行对照分析；

5.优化快速排序的中枢点选取，对优化前后的性能进行分析；

6.对快速排序的三种实现进行性能比较。

# 实验报告

## 实验设计

**1.1.1五种常见排序算法的实现与分析**

实现了插入排序、冒泡排序、快速排序（三种实现）、归并排序和shell排序，实现了算法基本的功能，同时在算法中记录了关键操作（数据比较和数据移动）的次数，并且在主函数中记录了算法运行的时间，用于分析算法的性能。

为了更好、更准确的分析算法性能，本部分的实验设计了多种数据规模，并且每种数据规模进行了5次重复实验，最后分析算法使用的数据是5次重复实验测量的平均值，数据规模有\*\*中，分别是50000、60000、70000、80000\*\*\*\*。单次测试实验中，随机生成指定规模的数组后，将该数组赋值多份用于多个算法排序，即保证单次测试中个算法输入的数据完全相同。

**1.1.2 快速排序优化**

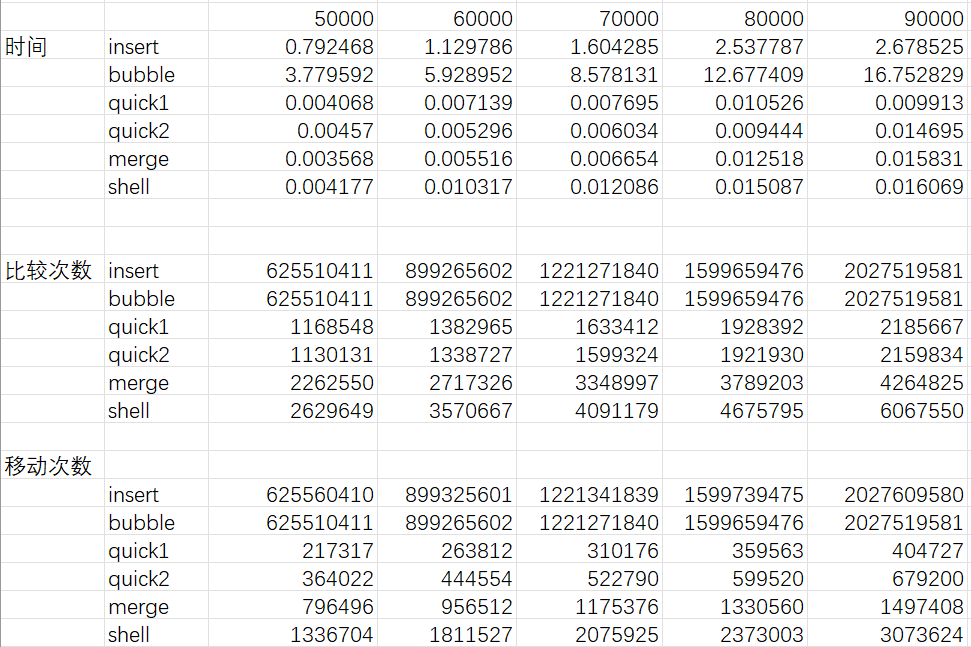
优化快速排序算法有很多途径，本次实验选取了优化基元的方法。优化基元的方法具体采用了三数取中法。三数曲中法的思路是：选择数组的第一个元素、中间元素和最后一个元素中的中位数作为枢轴，这样可以减少最坏情况发生的可能性。本部分实验与前面类似，运行算法，记录运行时间和关键操作次数，与经典实现的快速排序算法及其他算法进行比较。

**1.1.3 比较各种排序算法**

本部分实验通过采集的数据对各种算法进行分析比较。

## 数据分析

实验用于分析的数据如下：（三种经典算法性能类似，分析时使用挖坑法的数据，即quick1，quick2是改进的快排算法）



多种排序算法的比较：

可以看出在比较次数上，冒泡排序最差，其次为插入排序，而shell排序、快速排序、归并排序表现较为良好。在交换次数上，插入排序明显劣于其他四种排序算法。

可以看出，log多项式级别的算法（shell排序，快速排序，归并排序）在数据规模较大时性能显然优于O（n2）的算法（插入排序，冒泡排序）。

快排的改进：

可以看出比较次数方便两者并无太大差距，而交换次数方面原始的快速排序略差于改进后的快速排序。

快速排序的平均时间复杂度为O(n log n)，这一特性在处理大数据集时尤为关键。优化基元的快速排序通过改进基准元素的选择、分区策略等，进一步减少了不必要的比较和交换操作，从而提高了整体排序效率。

普通快速排序在最坏情况下（如输入数组已经有序或完全逆序）的时间复杂度会退化到O(n^2)。而优化基元的快速排序通过采用随机化基准选择、三数取中等策略，有效避免了这种情况的发生，使得算法在最坏情况下也能保持较高的效率。

三种快排比较：

比较次数方面三路快排算法优于其余两者，而交换次数方面三路快排交换次数最多，随机选择基元的快排略差于原始快排。

三路快排的时间复杂度单次递归对应的是log(3/2)n。而随着数据规模的增大，logn级的优势越发明显，性能优于普通的快速排序。