**高级算法设计与分析 课程**

**实验报告**

**实验一：实验题目**

姓名：？？？

学号：？？？

班级：？？？

评分表：（由老师填写）

|  |  |
| --- | --- |
| 最终得分： | |
| 对实验题目的理解是否透彻： | |
| 实验步骤是否完整、可信 ： | |
| 代码质量 ： | |
| 实验报告是否规范 ： | |
| 趣味性、难度加分 ： | |
| 特 色： | 1 |
| 2 |
| 3 |

**一、实验题目概述**

优化一个简单的快速排序算法，如下图：

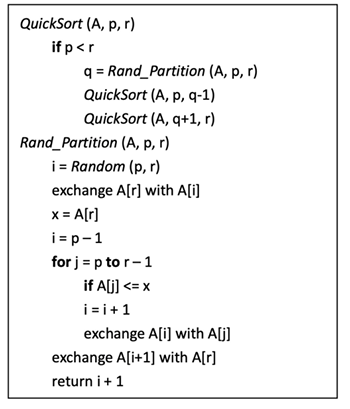


图 1 算法示意图

**二、对实验步骤的详细阐述**

主要讲述优化的思想：

1. 三路排序。将快速排序分为p,q，其中p,q是完全一样的数字。
2. 限制递归深度，如果递归太多，直接使用堆排序来排序剩下的元素。
3. 在区间较小的时候使用常数很小的插入排序。

上述其实是C++的sort实现的一些优化策略，能够显著降低算法在一些极端的输入情况下的时间复杂度，尽量比较快速排序被卡成的时间复杂度。

**三、实验数据**

**1. 实验设置**

**实验环境**：

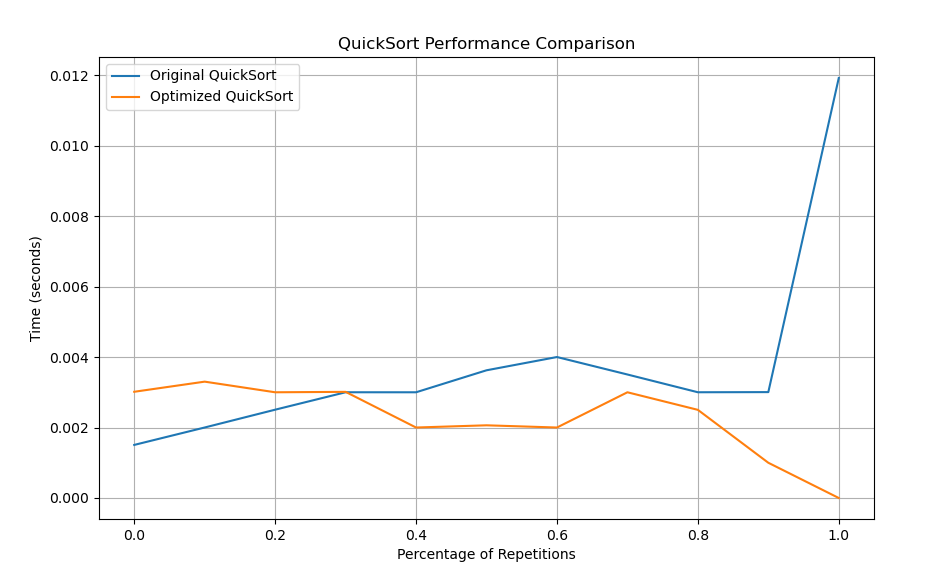
Windows 11，vscode，python

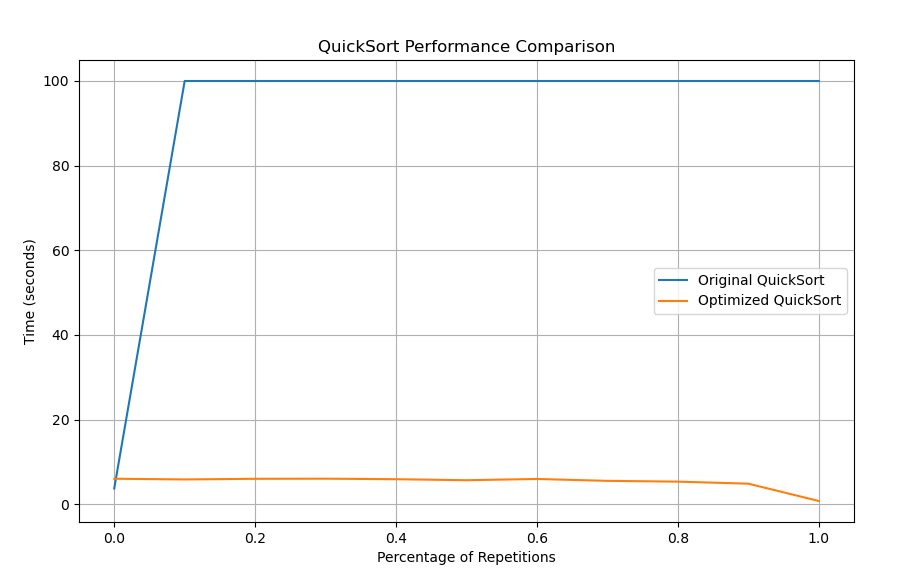
**数据**：

生成 11 个大小为10^6的整数数据集，第𝑖个子集中存在元素重复10^6×

10 ×𝑖%，𝑖 = 0,1,2,⋯,10，

**2. 实验结果**

****

****

**四、对实验结果的理解和分析**

1.优化之前的快速排序代码随着重复元素不断增加，速度不断下降，在达到80%之后，递归深度和栈内存超过了程序的最大内存，出现爆栈。这是重复元素过多，导致原始的quick\_sort每次分割的区间都非常不均衡导致的。

2.修改之后的快速排序方式能够有效地避免这个问题，并且在重复元素增多的情况下，三路排序反而能够排除给多的选项，从而提高算法的效率。

**五、实验过程中最值得说起的几个方面**

1. 在设计随机算法的时候也需要考虑算法的极端情况，从而可以设计策略来处理极端输入，使得算法更具有兼容性和稳定性。

注：对每个大项，可自主根据需要添加小标题