**深 圳 大 学 实 验 报 告**

**课程名称： 算法设计与分析**

**实验名称： 排序算法性能分析**

**学院：计算机与软件学院 专业： 计算机科学与技术（创新班）**

**报告人： 何泽锋 学号： 2022150221 班级： 高性能特色班**

**指导教师： 杨烜**

**实验时间： 2024年3月22日**

**实验报告提交时间： 2024年3月31日**

**教务处制**

**一．实验目的**

1. 掌握选择排序、冒泡排序、插入排序、合并排序、快速排序算法原理

2. 掌握不同排序算法时间效率的经验分析方法，验证理论分析与经验分析的一致性。

3. 求解TOP K问题，并分析比较不同算法效率。

**二. 问题描述**

1. 实现选择排序、冒泡排序、插入排序、合并排序、快速排序算法

2. 分析在不同数据规模下的效率，比较理论效率和实际效率之间的关系

3. 比较不同排序算法的效率，分析曲线关系

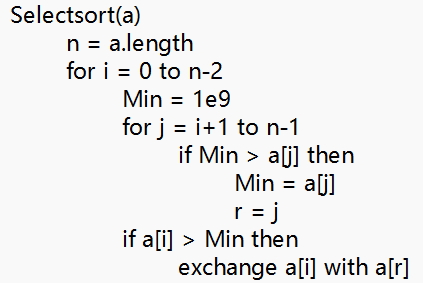
4. 设计能在大规模数据下快速挑选最大的十个数的算法，并在小规模数据上验证算法的正确性

**三．算法原理**

**1.选择排序：**

①排序原理（升序）：遍历数组，从未排序的部分中找到最小的元素，将找到的最小元素与未排序部分的第一个元素交换位置。重复以上步骤，直到所有元素都被排序完成。

②核心伪代码：

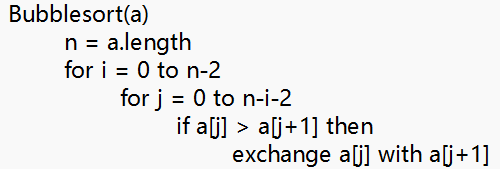


③细节解释：Min需要每轮重新赋值为最大以便寻找最小，同上该参数需要确保随机数最大值小于1e9，可以优化为每轮的首个数即Min=a[i+1],同时让r=i+1

**2.冒泡排序**

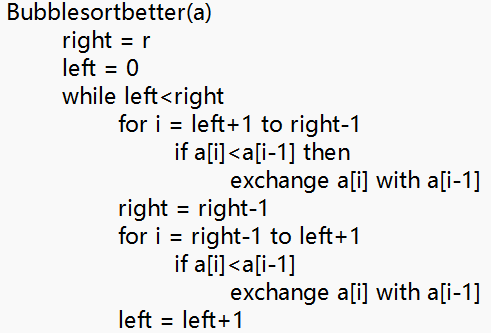
①排序原理（升序）：遍历数组，依次比较相邻的数，将大的数排在后面

②核心伪代码：



③细节解释：需要注意每一轮排序的结束下标，避免j+1溢出

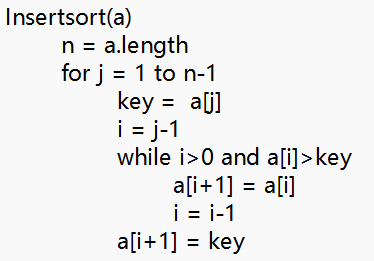
④优化：同时从数组的两端进行冒泡操作，在每一轮排序中同时找到最大值和最小值，并将它们分别放置在正确的位置上，从而减少了排序的比较次数和交换次数



**3.插入排序**

①排序原理（升序）：对于未排序数据，在已排序序列中从后向前比较，找到相应位置并插入。

②核心伪代码：

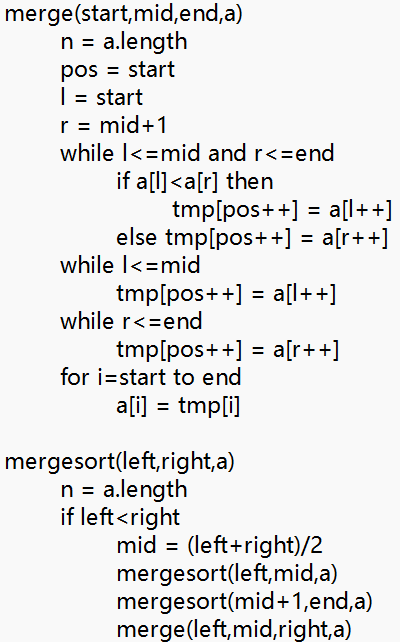


③细节解释：控制i>0避免溢出

**4.归并排序：**

①排序原理：将待排序的数组不断地拆分成长度大致相等的两个子数组，直到每个子数组只有一个元素为止。将已经有序的子数组两两合并，合并过程中按照大小顺序将元素逐个放入临时数组中，直到所有子数组合并为一个完整的有序数组为止

②核心伪代码：

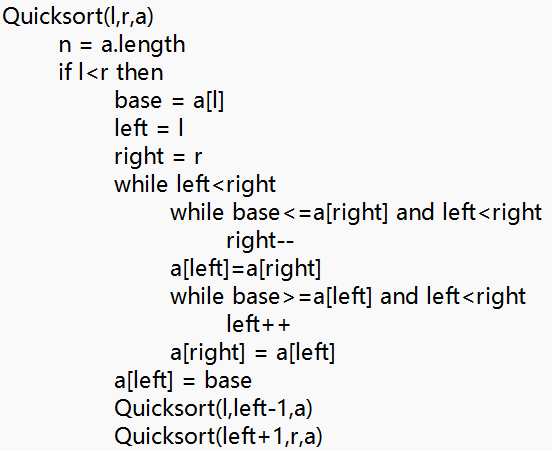


③细节解释：包含两个函数，第一个函数用于合并，第二个函数用于递归分割数组。对于合并函数，需要注意将tmp临时数组开在函数外面，如果动态开辟在函数内可能会影响对排序总时间的计算，导致数据误差较大。

**5.快速排序：**

①排序原理：从数组中选择一个基准元素，从数组最左侧选择第一个比基准大的数，从最右侧选择第一个比基准小的数，比基准值小的元素放在左边，比基准值大的元素放在右边，直到左右指针相遇。然后将基准值放置在最终位置上，对基准值左右两部分分别进行递归排序，直到排序完成。

②核心伪代码：

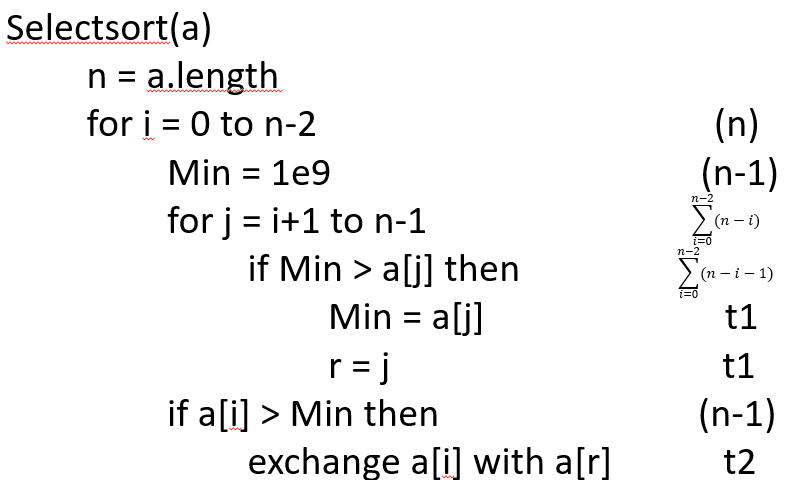


③细节解释：需要注意递归的边界

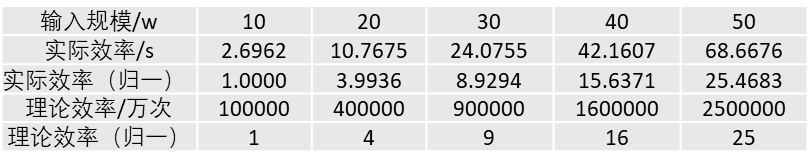
**四．算法效率分析**

**1.排序算法的理论效率和实际效率：**

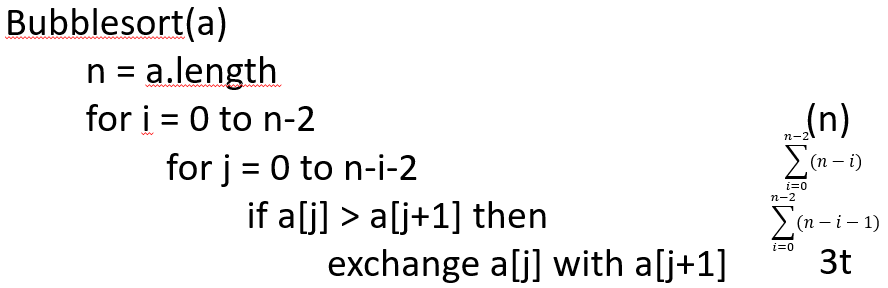
①选择排序：



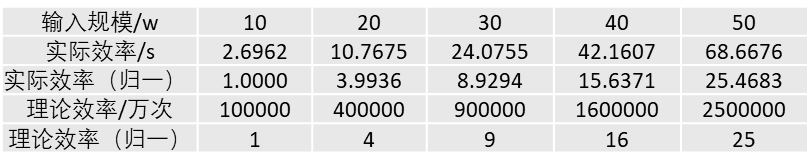
计算得到基本操作次数(理论效率)：



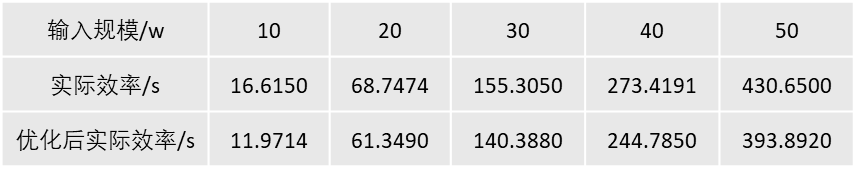
②冒泡排序：



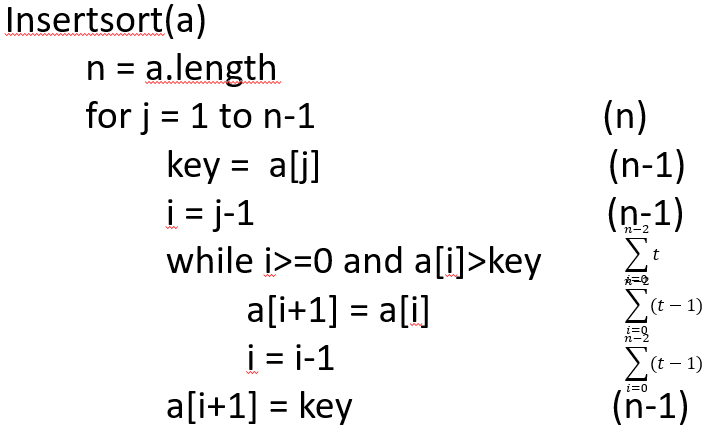
计算得到基本操作次数(理论效率)：



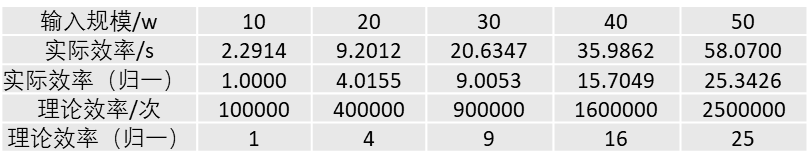
优化后效率比较：



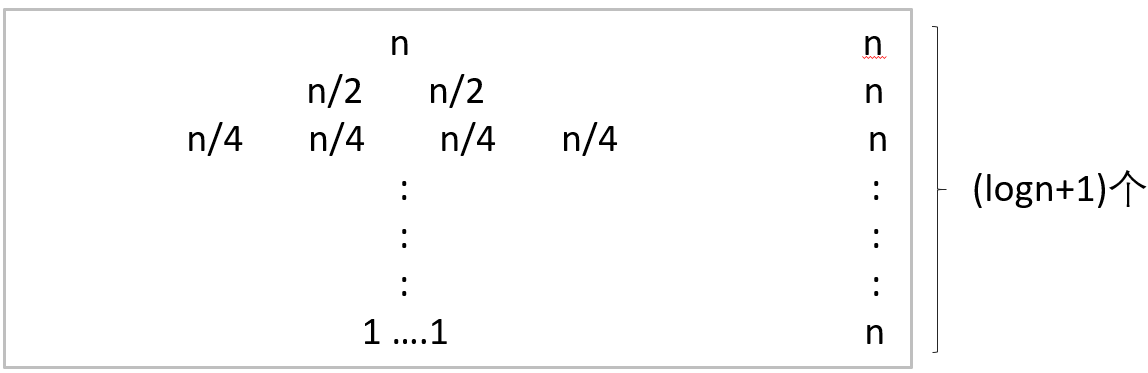
③插入排序：



计算得到基本操作次数(理论效率)：4n-3+ +2



④归并排序：



基本操作次数：T(n)=2T(n/2)+n 由递归树可得n(logn+1)



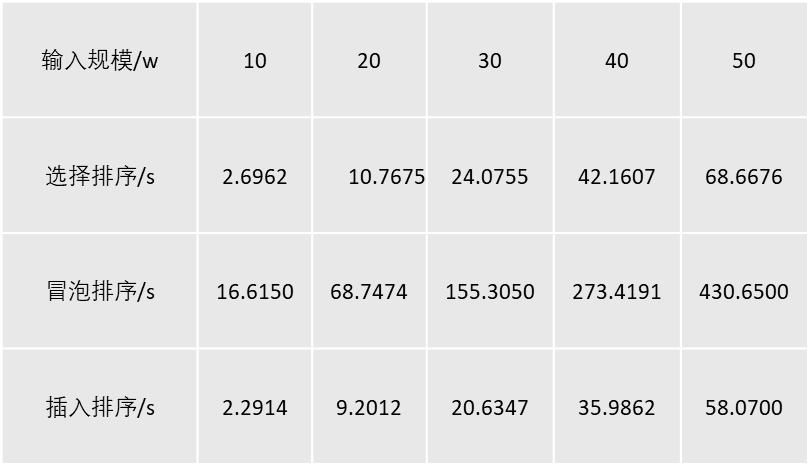
⑤快速排序：

基本操作次数：n+nlogn nlogn



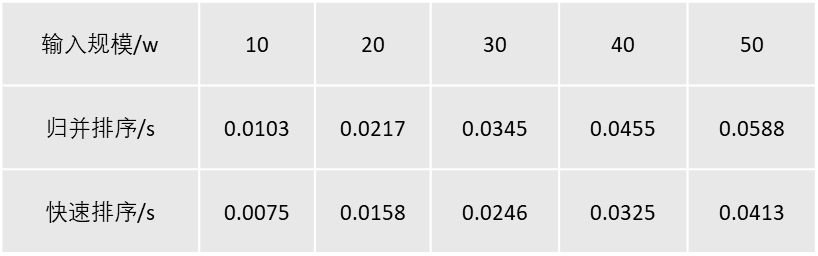
**2. 时间复杂度为的排序算法比较：**

分析表格中的数据可知，在不同规模大小的数据下，插入排序用时<冒泡排序用时<选择排序用时，其中冒泡排序用时明显大于其他两种排序，而插入排序与选择排序差异较小。该比较结果也可同过理论操作次数计算得到，结合前面所得到的理论效率同样可以得到图表中的结论



3. **时间复杂度为的排序算法比较**

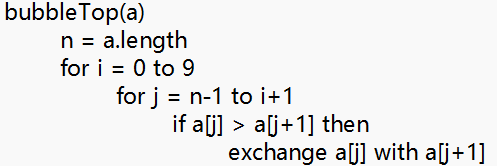
分析表格中的数据可知，归并排序所用时间>快速排序所用时间，但二者的曲线都接近直线，这是因为在较大数据量时logn的值变化不大，主要受到n的影响。同样四nlogn，但快排比归并效率更高，这是因为快排是在原始数组进行操作，无需额外引用地址空间，而归并需要使用临时数组排序并重新整合，这是导致时间差异的主要因素

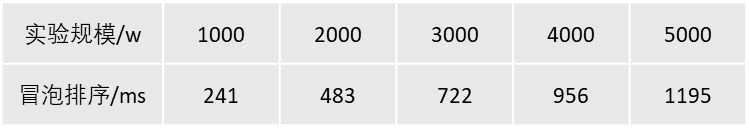


**4. 在10亿的数据（每个数据四个字节）中快速挑选出最大的十个数(TOP K问题)，并在小规模数据上验证算法的正确性**

(1)三个挑选数据的原理

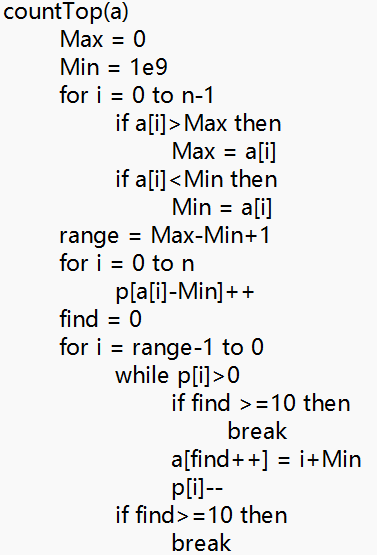
①冒泡排序：利用冒泡排序的原理将，最大的10个数排好后即可结束排序





细节解释：在K的个数较小的时候，冒泡排序效率相对可以接受，T(n)∊O(kn),但k较大时，排序所用时间将接近，因此算法效率较低。

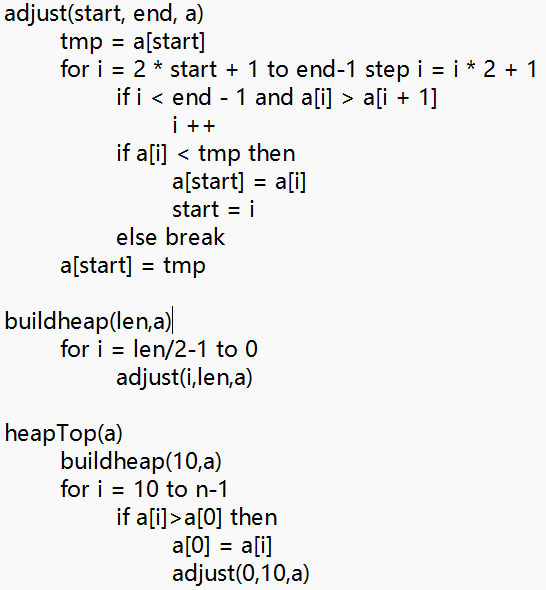
②计数排序：通过对每个元素进行计数，根据计数的结果来对元素进行排序

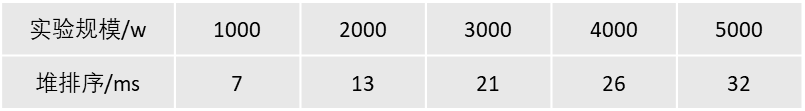




细节解释：计数排序是非计较排序，因此具有较优的时间效率，T(n)∊O(n)，但实际使用时具有较多的限制，首先数据范围不能过大，并且需要都为整数（或者确定小数位数的浮点数，可以通过扩大相同倍数转为整数），通过牺牲空间，开辟大范围的数组来确保每一个数都能放到对应位置的数组中

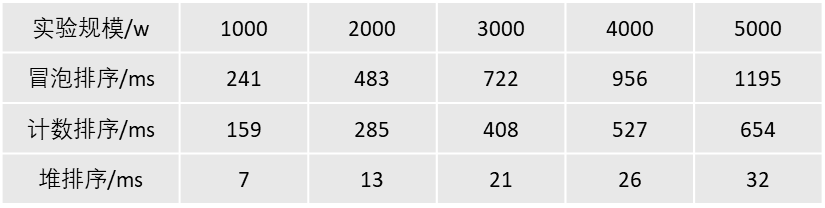
③堆排序：搭建10（k）个数的小根堆，通过与堆顶比较放入大的数并重新维护堆





细节解释：此处采用小根堆，需要先创建一个10个数的堆，然后再往后遍历比较并重新维护堆

（2）比较三种方式的效率



分析以上图表可知，堆排序用时<计数排序用时<冒泡排序用时，当数据量较大时堆排序具有最好的时间效率，效果最好，在空间上，堆排序无需额外开辟空间，相对计数排序空间复杂度也较低，因此在时间和空间上堆排序都较优。

**五．实验总结**

1.排序算法总结：实验测试了选择、冒泡、插入、归并和快速排序五种，其中选择、冒泡、插入时间复杂度为O（n^2）,而归并、快排时间复杂度为O(nlogn)，在实验选取的数据范围中，复杂度为O(nlogn)下，快排所用时间最短。在O（n^2）的复杂度下，冒泡用时最长，选择次之，插入较快。

2.Top K问题总结：实验选取了冒泡、计数、堆排序三种方式。所用时间：堆排序<计数<冒泡，冒泡排序需要依次比较相邻的数，操作次数较多，计数排序需要额外开大量空间；堆排序的时间和空间复杂度都较低，为较优解

3.思考总结：实验运用了多种排序，相同时间复杂度的排序之间也存在差异，不仅仅需要关注时间复杂度，还需要依靠操作次数进一步比较。实验数据规模相对较大，注意需要调整编译器栈堆的分配，防止溢出。对于随机数据来说，也要范围较大，避免数据较多的重复，影响排序时间。对于部分需要额外空间的排序，要在外部提前开辟数组，在函数内开动态数组并赋初值会占用较多时间，影响对实际排序效率的分析

|  |
| --- |
| 指导教师批阅意见：    成绩评定：  指导教师签字：  年 月 日 |
| 备注： |

注：1、报告内的项目或内容设置，可根据实际情况加以调整和补充。

2、教师批改学生实验报告时间应在学生提交实验报告时间后10日内。