**Министерство образования Российской Федерации**

**Пензенский государственный университет**

**Кафедра „Вычислительная техника“**

**ОТЧЕТ**

по лабораторной работе №2

по курсу „**Логика и основы алгоритмизации в инженерных задачах**“

на тему „ **Оценка времени выполнения программ**“

Выполнил студент группы 23ВВВ1 :

Усов А.С.

Приняли:

д.т.н. доцент Митрохин М.А.

к.т.н. доцент Юрова О.В.

Пенза 2024

**Цель работы.**

Изучить работу с библиотекой time.h в языке Си и закрепить на практике.

**Задание.**

**Задание 1**:

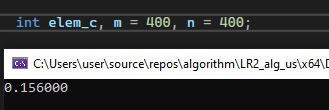
1. Вычислить порядок сложности программы (*О*-символику).
2. Оценить время выполнения программы и кода, выполняющего перемножение матриц, используя функции библиотеки time.h для матриц размерами от 100, 200, 400, 1000, 2000, 4000, 10000.
3. Построить график зависимости времени выполнения программы от размера матриц и сравнить полученный результат с теоретической оценкой.

**Вычисление О-символики**

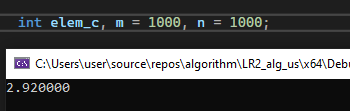
Инициализация каждой матрицы происходит с использованием двух вложенных циклов. Для каждой матрицы операция инициализации имеет временную сложность

O(n^3), что соответствует стандартной сложности алгоритма перемножения матриц.

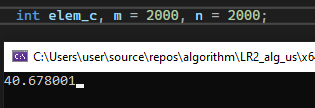
400х400 = 0.156000



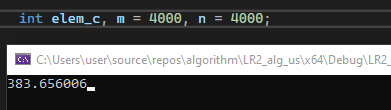
1000х1000 = 2.920000



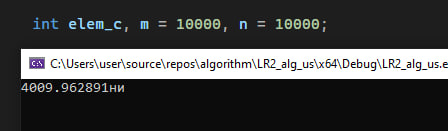
2000х2000 = 40.678001

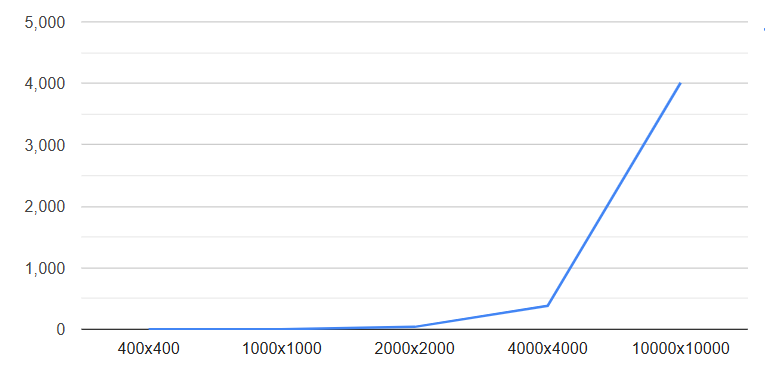


4000х4000 = 383.656006



10000x10000 = 4009.962891





**Сравнение практического результата с теоретической оцуенкой:**

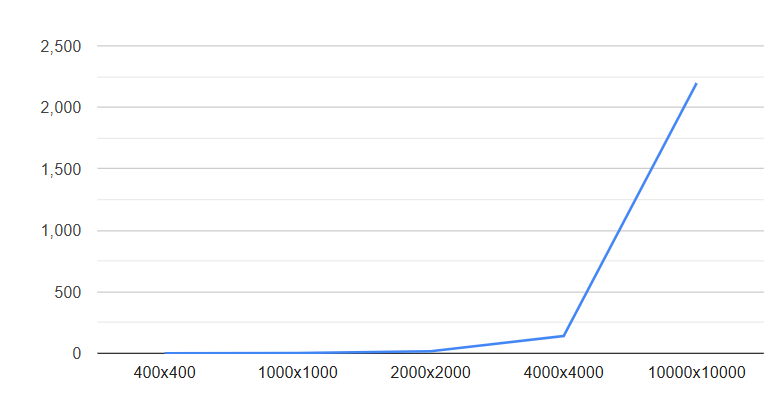
(0.0013 \* 400)^3 = 0.140608

(0.0013 \* 1000)^3 = 2.197

(0.0013 \* 2000)^3 = 17.576

(0.0013 \* 4000)^3 = 140.608

(0.0013 \* 10000)^3 = 2197

****

Теоретическая оценка работы времени программы показала приблизительно идентичные результаты, расхождения получились только в вычислении матриц 10000 на 10000, я считаю что такое произошло из за того, что мощность вычислительной машины была не полностью направлена на данную задачу, а занята другими процессами параллельно.

**Задание 2**:

1. Оценить время работы каждого из реализованных алгоритмов на случайном наборе значений массива.
2. Оценить время работы каждого из реализованных алгоритмов на массиве, представляющем собой возрастающую последовательность чисел.
3. Оценить время работы каждого из реализованных алгоритмов на массиве, представляющем собой убывающую последовательность чисел.
4. Оценить время работы каждого из реализованных алгоритмов на массиве, одна половина которого представляет собой возрастающую последовательность чисел, а вторая, – убывающую.
5. Оценить время работы стандартной функции qsort, реализующей алгоритм быстрой сортировки на выше указанных наборах данных.

Количество чисел в массиве: 10000. Время работы алгоритма измеряется в мс

|  |  |  |  |
| --- | --- | --- | --- |
|  | Сортировка Шелла | Быстрая сортировка | Ф-я б. сорт |
| Сл.набор.данных | 479 | 5 | 7 |
| Возраст.послед. | 1 | 2 | 7 |
| Убыв.посл. | 907 | 2 | 7 |
| Убыв-возраст. | 454 | 1559 | 0 |

Количество чисел в массиве: 20000. Время работы алгоритма измеряется в мс

|  |  |  |  |
| --- | --- | --- | --- |
|  | Сортировка Шелла | Быстрая сортировка | Ф-я б. сорт |
| Сл.набор.данных | 1822 | 10 | 12 |
| Возраст.послед. | 2 | 3 | 15 |
| Убыв.посл. | 3619 | 4 | 15 |
| Убыв-возраст. | 1812 | 6232 | 0 |

**Листинг:**

[rezuqq/LR2\_alg\_us (github.com)](https://github.com/rezuqq/LR2_alg_us)

**Вывод:**

В ходе лабораторной работы изучил работу с библиотекой time.h в языке Си и закрепил на практике, также сравнили время работы стандартной библиотеки quick sort c написанной вручную функцией quick sort и shell, получил следующие результаты стандартная библиотека быстрее написанных вручную функций.