

МИНОБРНАУКИ РОССИИ

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**

Кафедра МО ЭВМ

ОТЧЕТ

по лабораторной работе №3

по дисциплине «Параллельные алгоритмы»

Тема: Параллельное умножение матриц.

.

Студент гр. 9304

Попов Д.С.

Преподаватель

Сергеева Е.И.

Санкт-Петербург

2022

Цель работы.

Реализовать параллельный алгоритм и алгоритм «быстрого» умножения матриц.

Задание.

4.1 Реализовать параллельный алгоритм умножения матриц. Исследовать масштабируемость выполненной реализации.

4.2 Реализовать параллельный алгоритм “быстрого” умножения матриц (Штрассена или его модификации).

- Проверить, что результаты вычислений реализаций 4.1 и 4.2 совпадают.
- Сравнить производительность с реализацией 4.1 на больших размерностях данных (порядка $10^4 - 10^6$).

Выполнение работы.

Класс `Matrix` представляет из себя двойной вектор для хранения данных, размер стороны, а также перегруженные операторы `+`, `-`, `*` и `==`.

Для параллельного алгоритма умножения матриц была реализована функция `matrixMultiplication`, которая принимает два указателя на класс матрицы `Matrix`. Каждому потоку достается определенная область, в которой необходимо провести умножение и отобразить результат на результирующую матрицу.

Алгоритм «быстрого» умножения представлен функцией `matrixStrassenMultiplication`, которому на вход так же подается два указателя матриц. Для реализации алгоритма Штрассена были реализованы лямбды, которые «собирают» и «разбивают» матрицу из/на 4 подматрицы, а также рекурсивная лямбда, задача которой производить разбиение и подсчет до тех пор, пока подматрицы не станут размером 64, ибо дальнейшее разбиение не целесообразно.

Для наглядности также была реализовано однопоточное умножение.

Сравнение производительности параллельного и «быстрого» алгоритмов.

Для параллельного умножения матриц было использовано 8 потоков, а в реализации Штрассена 7.

В таблице 1 представлено время выполнения для каждого алгоритма, при разных размерах матрицы:

Размер матрицы	Однопоточная, мс.	Параллельная, мс.	Штрассен, мс.
10^4	47	41	14
10^5	105	94	26
10^6	5780	2791	2791

Таблица 1 – Зависимость времени выполнения от размера матриц

Выводы.

В ходе выполнения лабораторной работы была реализована программа на языке программирования C++ для умножения матриц. Было произведено сравнение и установлено, что алгоритм Штрассе показывает наилучший результат.