

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**  
**Кафедра МО ЭВМ**

**ОТЧЕТ**  
**по лабораторной работе №4**  
**по дисциплине «Параллельные алгоритмы»**  
**Тема: Параллельное умножение матриц**

Студентка гр. 0304

Говорющенко А.В.

Преподаватель

Сергеева Е.И.

Санкт-Петербург

2023

## **Цель работы.**

Изучить умножения матриц с помощью алгоритма Штрассена с разбиением на потоки и сравнить производительность разных алгоритмов.

## **Задание.**

4.1 Реализовать параллельный алгоритм умножения матриц с масштабируемым разбиением по потокам.

Исследовать масштабируемость выполненной реализации с реализацией из работы 1.

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  $10^4 - 10^6$ )

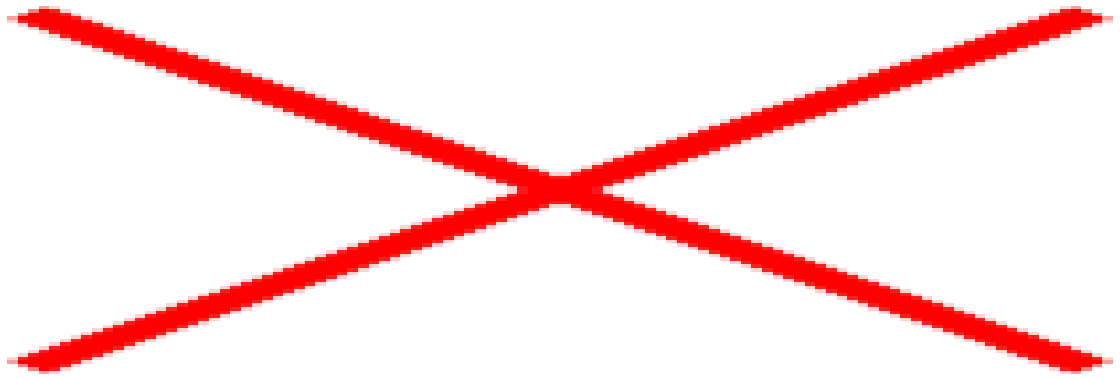
## **Выполнение работы.**

За основу данной работы был взят код из первой лабораторной с классом Matrix, который был расширен методами для реализации алгоритма Штрассена.

Алгоритм Штрассена заключается в перемножении матриц, размерность  $n$  которых является степенью двойки, в случае если размерность будет не соответствовать данным условиям работа алгоритма не будет начата. Вычисление проводится рекурсивно за счет того, что в самом перемножении используются подматрицы размера  $n/2$ . Таким образом получаем подматрицы исходной матрицы:



Теперь необходимо вычислить 7 промежуточных матриц размерности  $n/2$ , из которых будет вычисляться результирующая матрица. Вычисления выполняются в соответствии с формулами представленными ниже.



Матрицы перемножаются рекурсивно с помощью того же алгоритма. Так как на матрицах малых размеров простое умножение работает эффективнее, то на большой глубине рекурсии вместо алгоритма Штрассена применяется оно.

Для получения доступа к подматрице был создан метод *getSubmatrix* и метод для записи в участок матрицы *setSubmatrix*.

Были проведены измерения времени от размеров вычисляемых матриц, результаты приведены в таблице 1.

Таблица 1 — Зависимость времени от размера матриц.

Размер	Простой алгоритм,	Параллельный	Алгоритм
	мс	алгоритм, мс	Штрассена, мс
16x16	0.216748	0.754479	0.637789
32x32	1.88019	1.26431	1.27589
64x64	8.65574	7.38064	5.17074
256x256	314	187	103
1024x1024	20987	12592	4300
2048x2048	553211	219707	29606

Из полученных результатов видно, что для матриц небольших размеров (16x16) наиболее эффективным оказался простой алгоритм, а уже для больших размеров — алгоритм Штрассена. Начиная с матриц размера 1024 отличии во времени вычисления уже идет на порядок.

### **Выводы.**

В ходе работы был исследован и реализован алгоритм Штрассена для быстрого умножения матриц. По итогам сравнения данный алгоритм оказался самым быстрым, следующим за ним идет параллельный алгоритм и затем простое умножение матриц.