**Київський національний університет імені Тараса Шевченка**

**Факультет комп’ютерних наук та кібернетики**

**Кафедра інформаційних систем**

**Алгоритми та складність**

**Лабораторний проект № 5**

**Алгоритм Штрассена для множення матриць**

**Звіт**

**Виконав:**

Студент групи К-28

Швець Олександр

**Київ-2018**

**Алгоритм Штрассена для множення матриць**

1. **Умова завдання.**

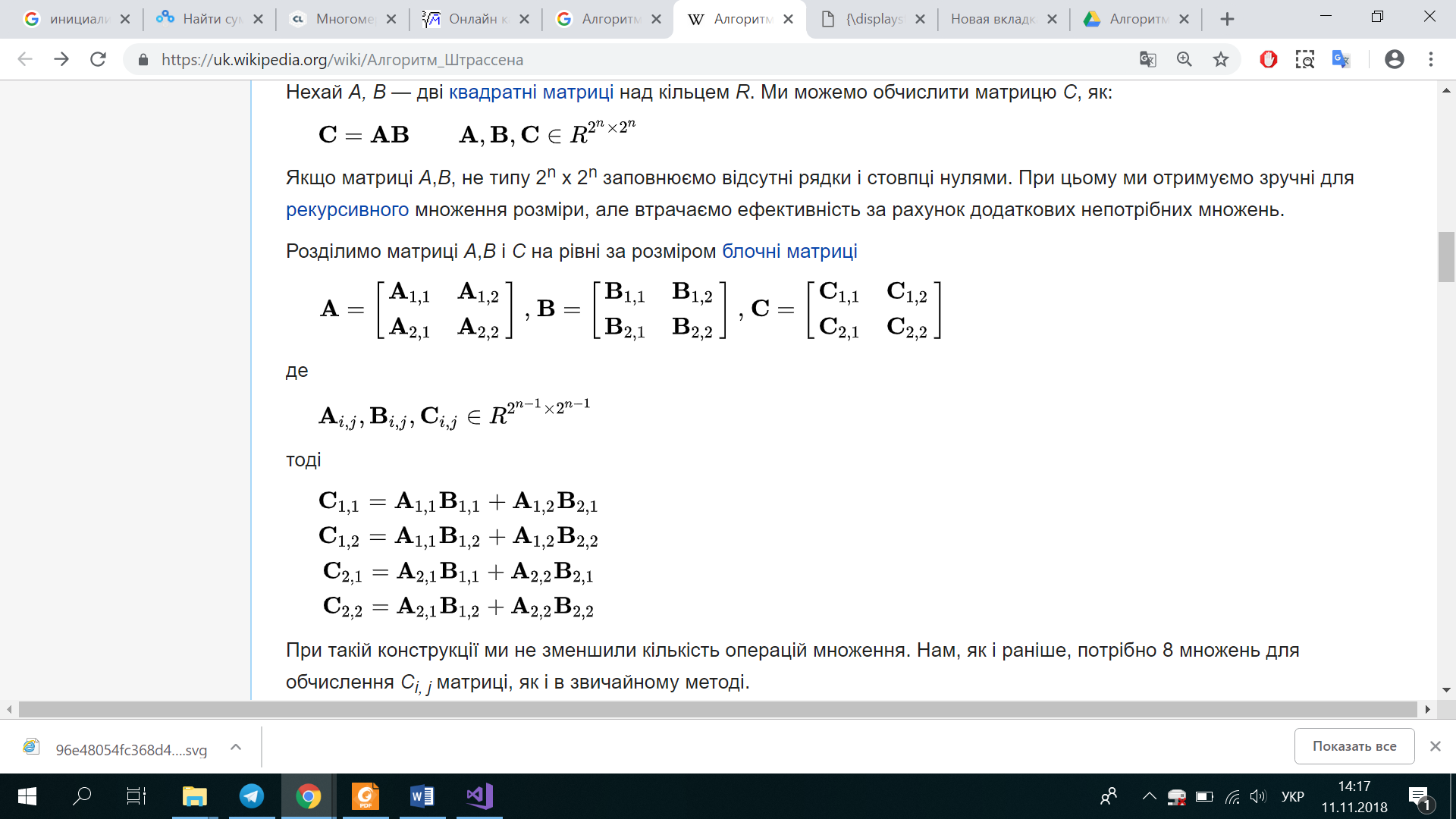
Реалізуйте алгоритм Штрассена для множення матриць. На практиці алгоритм починає застосовуватися для матриць такого розміру, коли з'являється виграш порівняно з класичним способом на основі означення, який використовується для матриць меншого розміру.

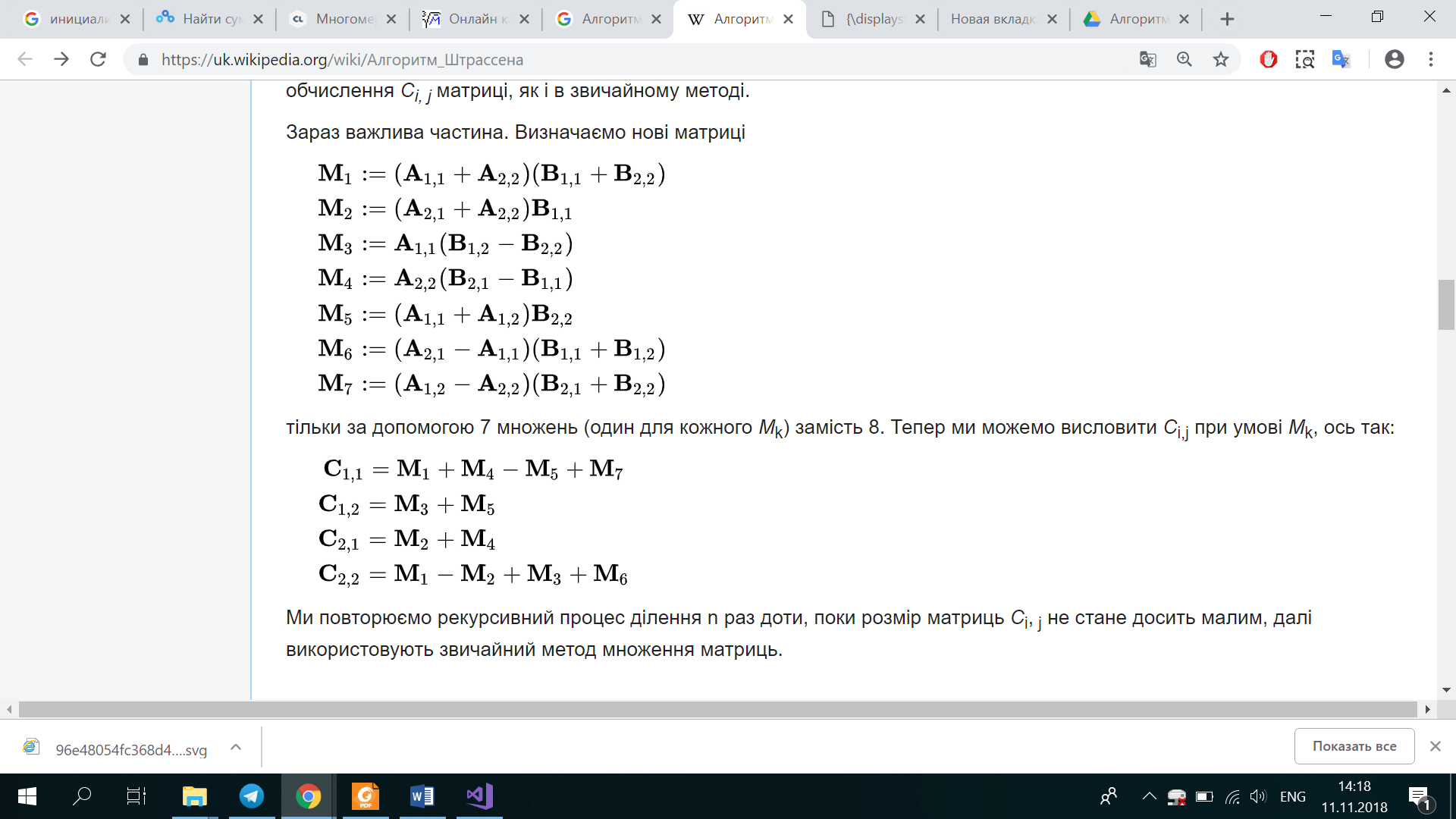
Спробуйте експериментально визначити цю "точку перетину" для свого комп'ютера.

1. **Основні поняття.**

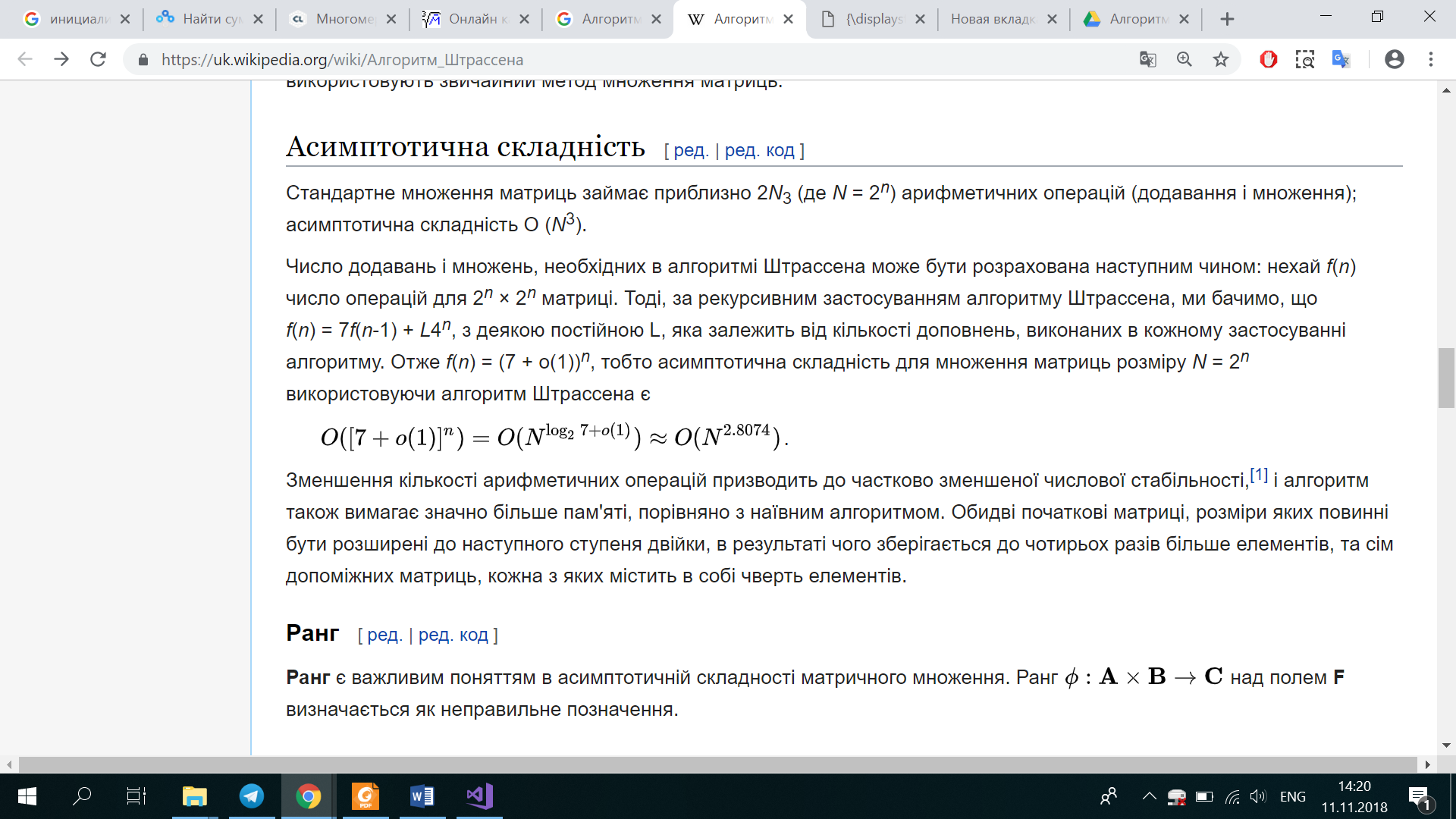
*Алгоритм Штрассена призначений для швидкого множення матриць. Цей алгоритм дозволяє швидше за стандартний спосіб множити матриці.*

1. **Опис алгоритму.**





1. **Аналіз алгоритму.**

****

1. **Реалізація алгоритму.**

Реалізовано на С++

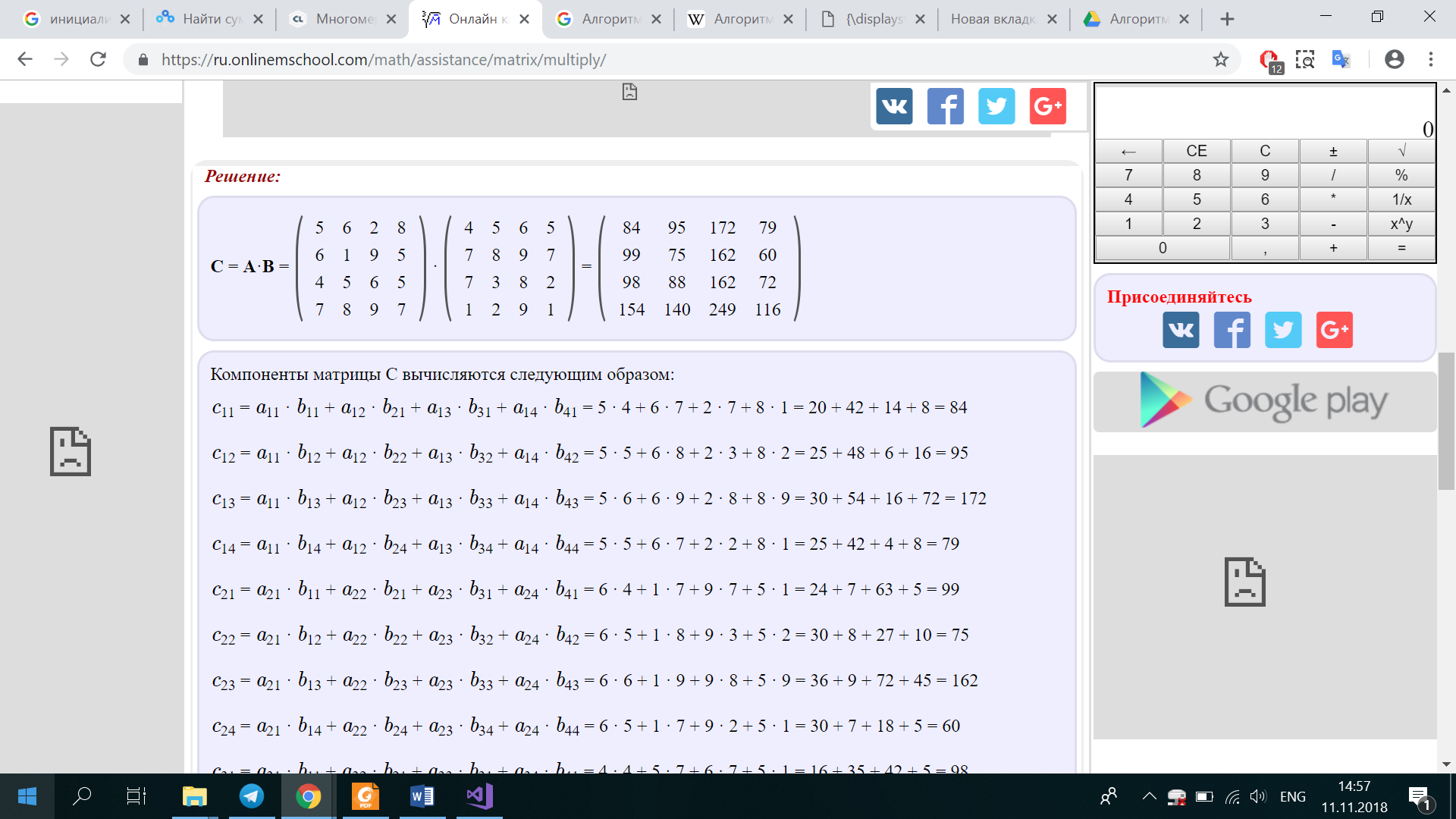
1. **Інтерфейс програми.**

Дані (тести) прописані в файлі срр, а результат виводиться в консоль.

1. **Використані структури даних.**

Масиви типу A[N][N].

1. **Тестовий приклад.**



1. **Основні модулі програми.**

ofstream deb("debug.txt"); зчитка файлу

void division\_matrix(matrix & a11, matrix & a12, matrix & a21, matrix & a22); поділ матриці на 4-и малих

void collect\_matrix(matrix & a11, matrix & a12, matrix & a21, matrix & a22); сбір матриці в одну

matrix::matrix(int n, int m) конструктор матриці

matrix matrix::strassen\_mtrx(matrix a, int privacy) алгоритм штрасена

void matrix::read() читання с файла

matrix matrix::sum(matrix a) додавання матриць

matrix matrix::subtruct(matrix a) віднімання матриць

matrix matrix::multiplication(matrix & a){//множення матриць

void matrix::division\_matrix(matrix & a11, matrix & a12, matrix & a21, matrix & a22) поділ матриць

void matrix::collect\_matrix(matrix & a11, matrix & a12, matrix & a21, matrix & a22) 4 в 1

matrix matrix::strassen\_mtrx(matrix & a){//задає новий розмір

1. **Висновки.**

Таким чином, обмінявши одне матричне множення на фіксовану кількість додавань матриць вдалося зменшити число рекурсивних викликів функції множення.

Замість 8 множень і 4 додавань базового прямолінійного алгоритму отримали 7 множень і 18 додавань/віднімань.

Алгоритм дає виграш на великих щільних матрицях.

На практиці при досягненні «точки перетину» (в сенсі розмірності матриць; залежить від конкретної системи) відбувається перехід від методу Штрассена до стандартного множення.

1. **Використані джерела.**

1. Лекція 8 (за 05.11.2018)

2. *https://uk.wikipedia.org/wiki/ Алгоритм\_Штрассена*