Київський національний університет імені Тараса Шевченка

Факультет комп`ютерних наук та кібернетики

Кафедра інтелектуальних інформаційних систем

Алгоритми та складність

Завдання №5

**Алгоритм Штрассена**

Виконала студентка 2-го курсу

ГрупиК-28

Панченко Тетяна Андріївна

**2020**

**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.

**Теорія**

Алгоритм Штрассена призначений для швидкого множення матриць. На практиці це відчутно на великих матрицях.

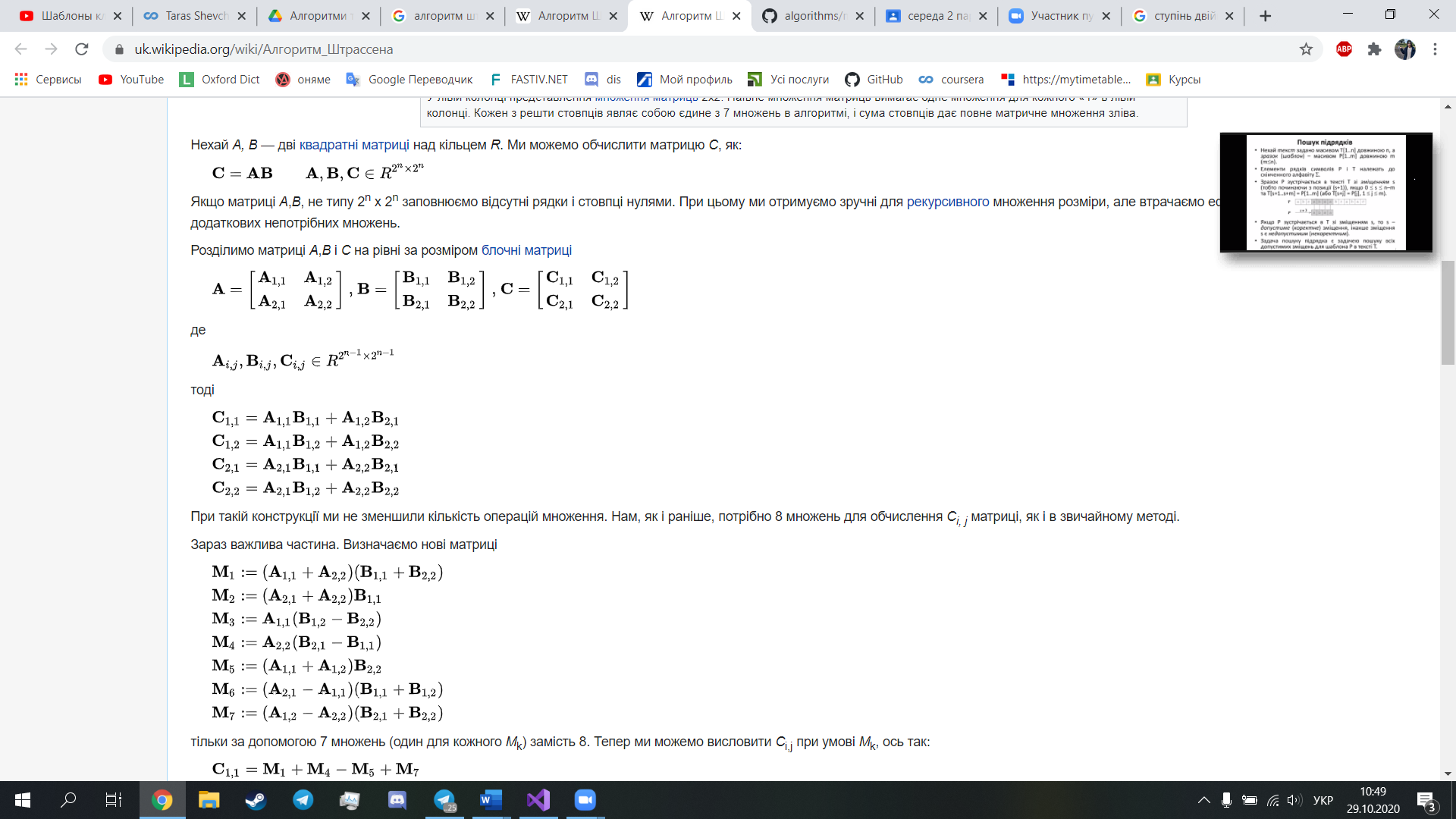
Не зважаючи на те, що алгоритм Штрассена є асимптотично не самим швидким з існуючих алгоритмів швидкого перемноження матриць, він легше програмується і є ефективнішим при відносно малих розмірах матриць, тому він частіше використовується на практиці.

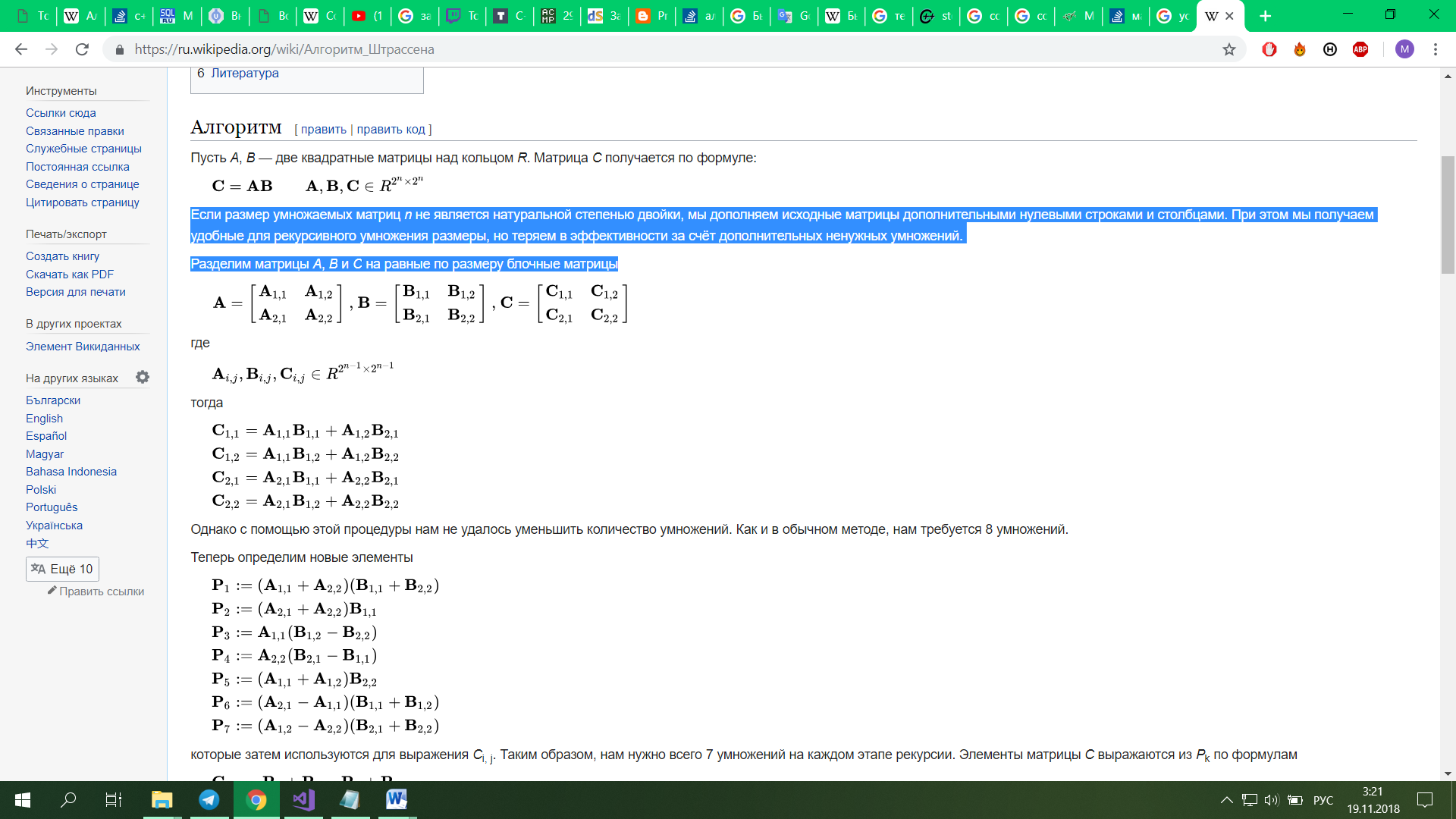
**Ідея алгоритму**

1. Маємо 2 квадратні матриці А та В. Матриця С є результатом перемноження цих матриць.

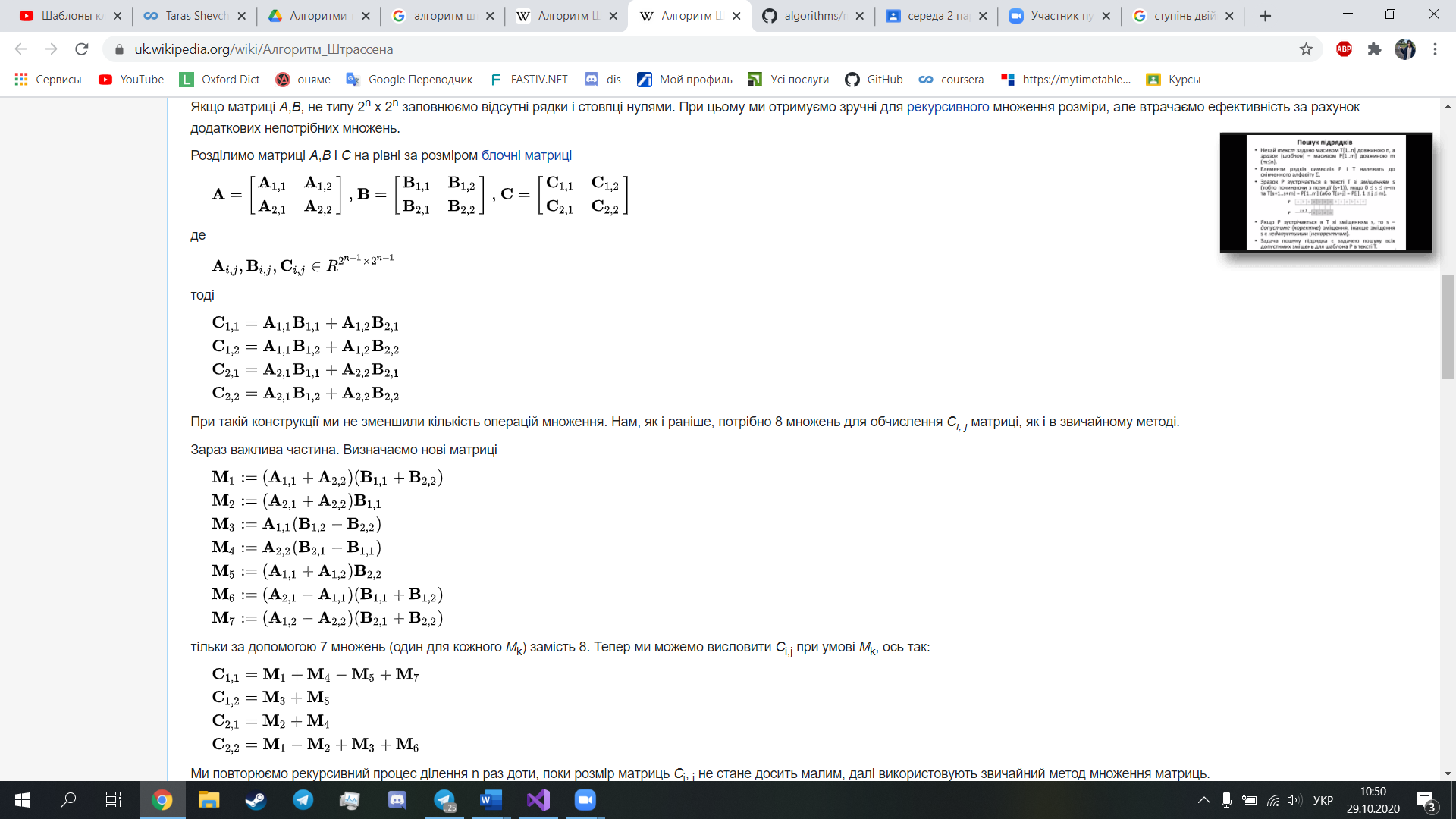
2. Якщо розмір матриць, що перемножаються, не є натуральним степенем двійки, тоді доповнимо А та В нульовими рядками і стовпчиками. При цьому отримаємо зручні для рекурсивного множення розміри, при цьому використавши більше часу за рахунок додаткових множень.

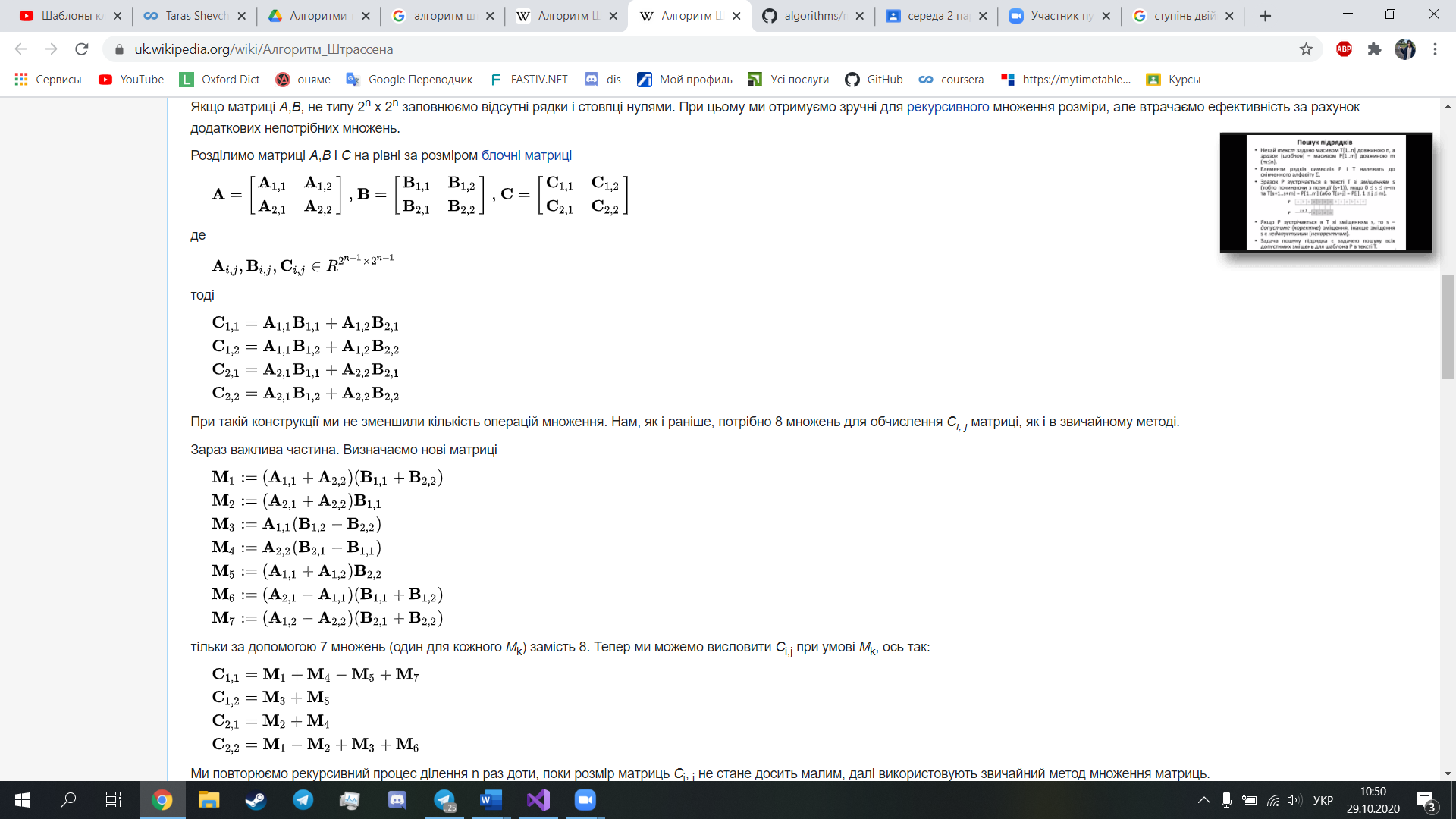
3. Розділимо матриці А,В і С на рівні за розміром матриці.



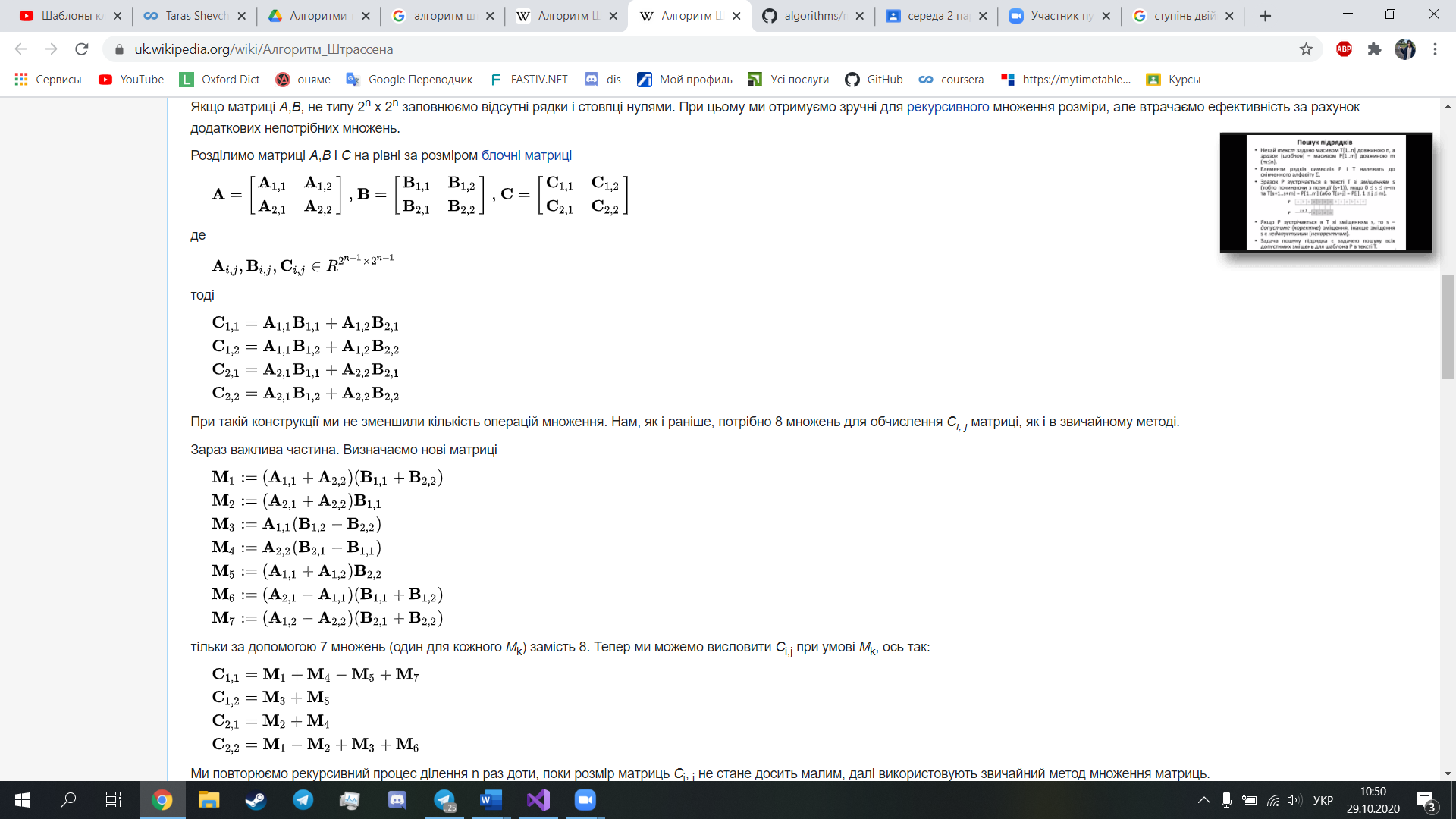
Причому 

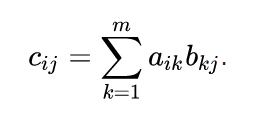
Тоді:

Усе ще маємо 8 операцій множення....

4. Визначимо нові матриці:

Тоді:



Будемо повторювати попередні кроки, починаючи з 1 доти, поки розмір матриці С не стане достатньо малим для використання звичайного методу перемноження матриць за формулою .

Перехід до звичайного методу роблять тому, що алгоритм Штрасена втрачає свою ефективність в порівнянні з ним на малих матрицях через велику кількість додавань. Оптимальний розмір матриць для переходу до звичайного методу лежить в межах від 32 до 128.

**Тестові приклади**

|  |  |  |
| --- | --- | --- |
| A | B | C |
| 1 2 3  4 5 6  7 8 9 | 1 1 1  2 2 2  2 2 2 | 11 11 11  26 26 26  41 41 41 |
| 0 0 0 1 1  1 0 3 0 2  1 1 0 1 2  2 3 0 0 0  1 0 2 3 0 | 0 0 0 0 1  0 0 0 2 2  2 2 0 1 0  1 3 1 0 2  2 0 0 0 0 | 3 3 1 0 2  10 6 0 3 1  5 3 1 2 5  0 0 0 6 8  7 13 3 2 7 |
| 0 0 0 1  4 0 6 2  7 8 0 1  2 3 0 0 | 0 0 0 0  2 2 2 4  2 2 2 3  2 3 4 1 | 2 3 4 1  16 18 20 20  18 19 20 33  6 6 6 12 |

**Складність**

Звичайний алгоритм перемноження матриць виконується за О(n3).

Використовуючи алгоритм Штрассена асимптотична складність для множення матриць розміром N = 2n буде О(), що приблизно дорівнює

О().

**Мова програмування**

С++

**Модулі програми**

Множення матриць класичним способом:

void SimpleMultiply(int A[][N], int B[][N], int C[][N])

Додавання матриць:

void AddMatrix(int n, int X[][N], int Y[][N], int Z[][N])

Віднімання матриць:

void SubMatrix(int n, int X[][N], int Y[][N], int Z[][N])

Алгорим Штрассена:

void Strassen(int n, int A[][N], int B[][N], int C[][N])

**Інтерфейс користувача**

Користувач обирає розмірність матріці, вивід у консоль.

**Висновки**

Потрібний алгоритм був виконаний і «точка перетину» між алгоритмом Штрассена і звичайним була знайдена. Після експериментального визначення «точки перетину» для свого комп’ютера дізналась, що алгоритм Штрассена ефективніше діє для матриць розміром до 64х64. Далі у виграші звичайний алгоритм перемноження.

**Використані джерела**

1. <https://en.wikipedia.org/wiki/Strassen_algorithm>
2. <https://studfile.net/preview/3068589/>