МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ

ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ імені ІВАНА ФРАНКА

Паралельні та розподілені обчислення

ЛАБОРАТОРНА РОБОТА №1

«Додавання та віднімання матриць»

Виконала:

студентка групи ПМі-31

Дудчак Валентина Юріївна

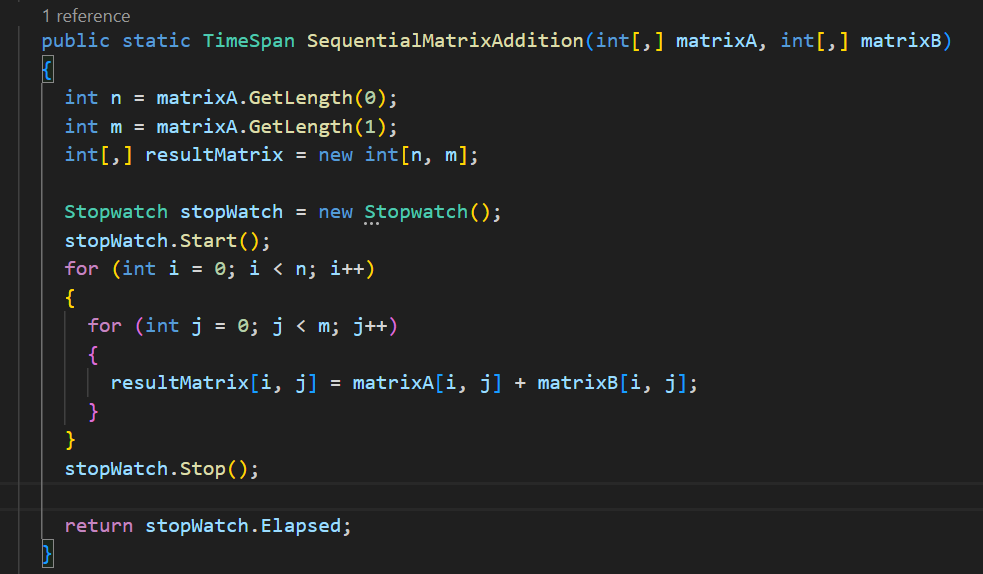
Львів 2024

**Тема**: Розпаралелення додавання/віднімання матриць

**Мета**: Написати програми обчислення суми/різниці двох матриць (послідовний та паралельний алгоритми). Порахувати час роботи кожної з програм, обчислити прискорення та ефективність роботи паралельного алгоритму.

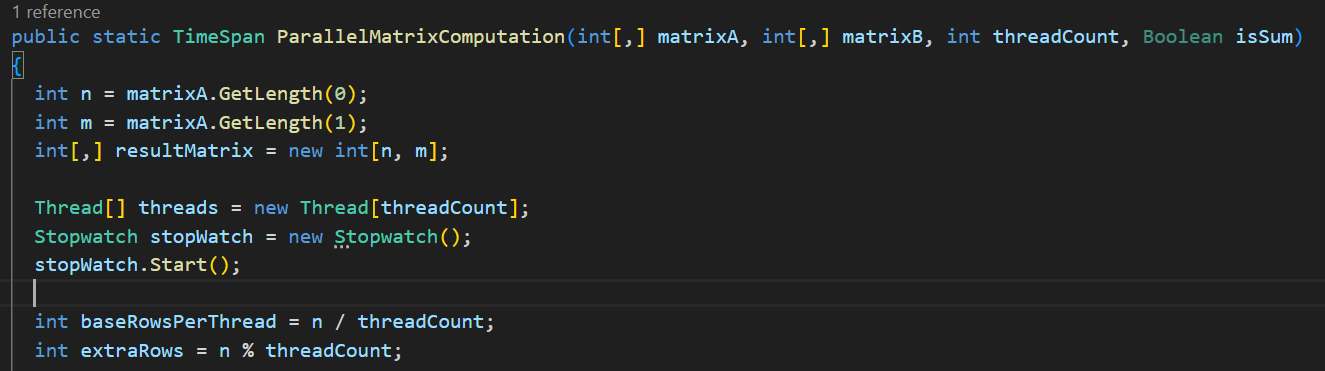
**Послідовний алгоритм:**

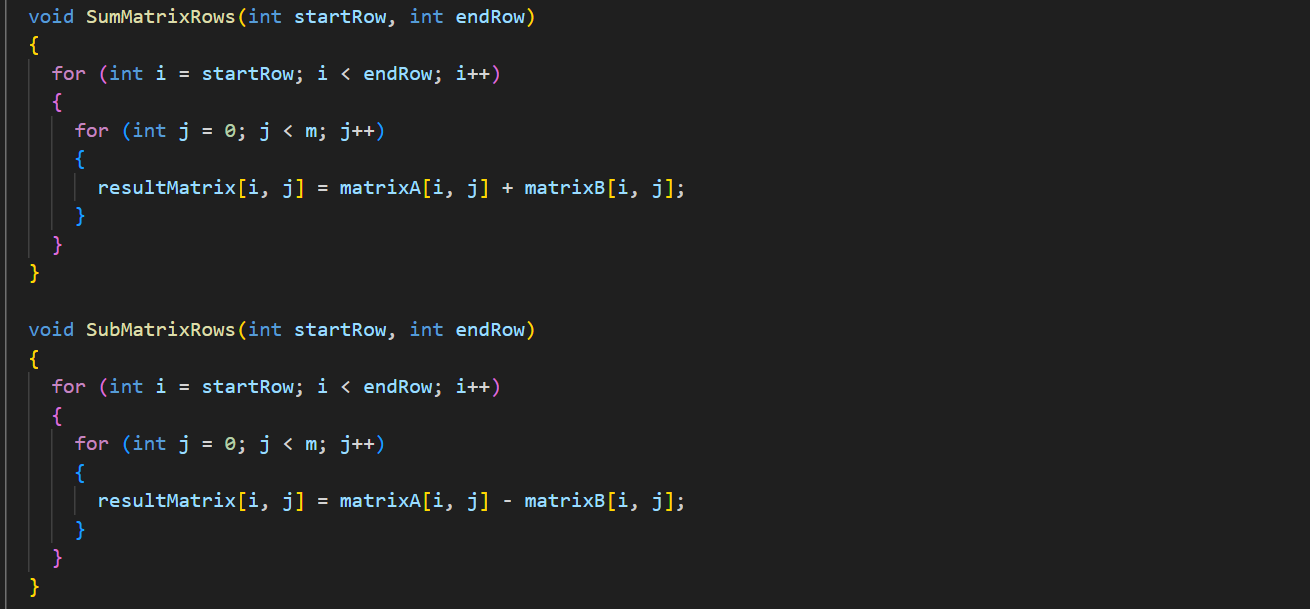
Послідовний алгоритм додавання/віднімання матриць реалізовано у методах SequentialMatrixAddition та SequentialMatrixSubstraction. У них використані вкладені цикли для обрахунків матриць та об’єкт класу Stopwatch для засікання часу (у подальшому потрібно для аналізу результатів):

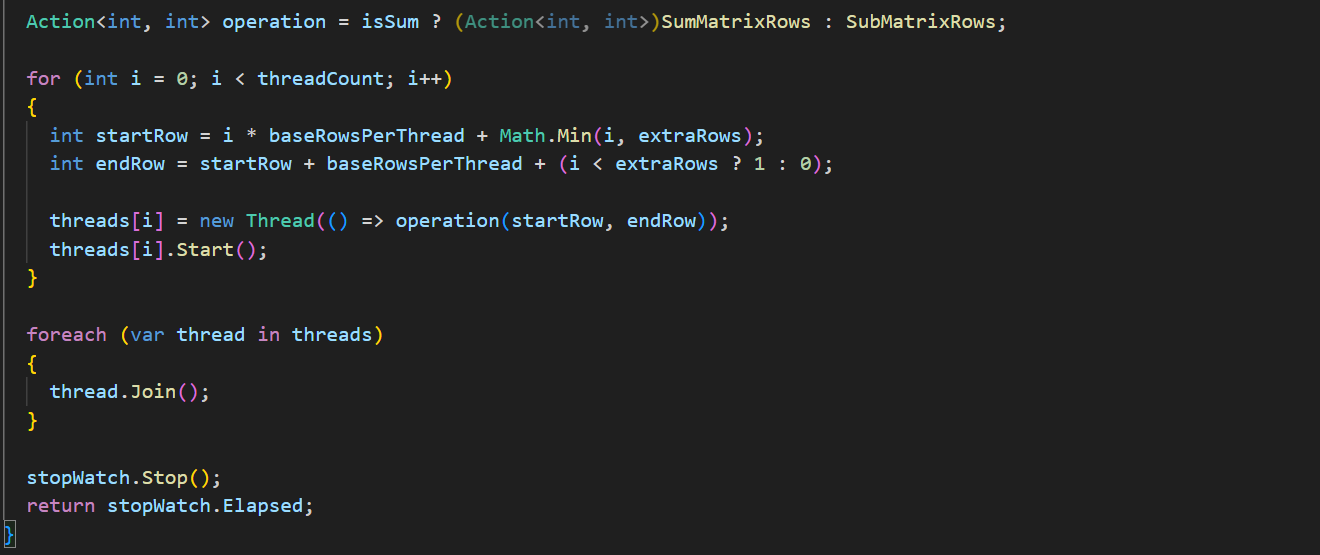


**Паралельний алгоритм:**

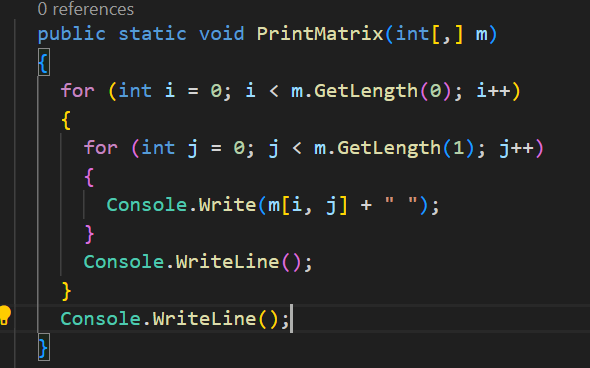
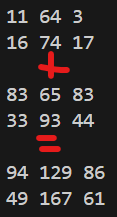
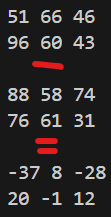
Паралельний алгоритм додавання/віднімання матриць реалізовано у методі ParallelMatrixComputation. Використано об’єкт Thread для розпаралелення процесу обчислення на рядки та функції SumMatrixRows/SubMatrixRows, у яких обчислюються лише потрібні рядки матриці. Кожен потік обчислює цілу частку від ділення кількості рядків на кількість потоків + перші потоки обчислюють ще по одному рядку з залишкових для кращої рівномірності.





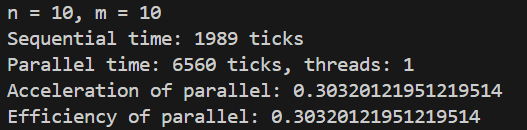


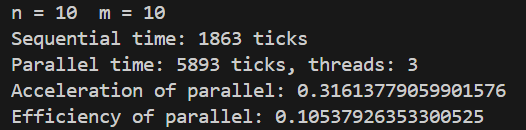
Крім цих двох методів, реалізована також функція **PrintMatrix**, за допомогої якої я перевірила, чи функції додавання/віднімання обчислюються правильно:

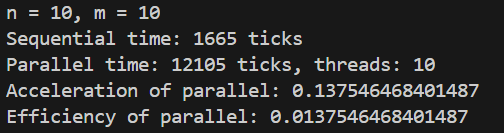
  

**Аналіз результатів:**

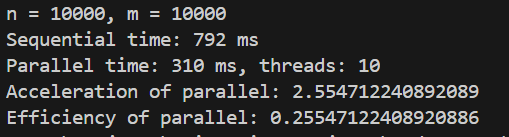
На малих розмірах матриці розпаралелення для додавання матриць неефективне, і стає все більш неефективним зі зростанням кількості потоків:

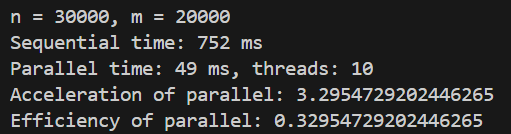




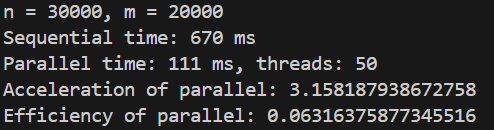


На більших розмірностях розпаралелення дає значно кращий результат:

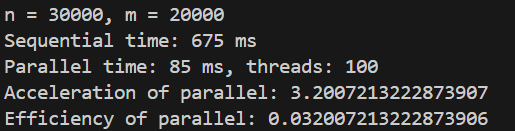


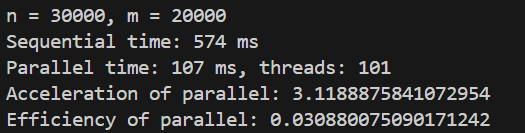


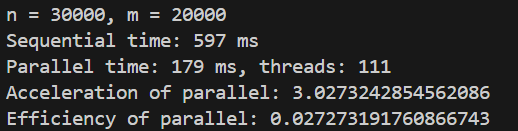
При ~10 потоках для ~20000 рядків ефективність є найбільшою, проте з більшим зростанням кількості потоків погіршується:



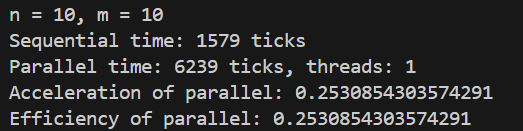
Додатково я порівняла те, як на ефективність впливає кратність розмірності матриці до кількості потоків. Бачимо, що не кратні значення погіршують ефективність паралельного алгоритму, хоча додалася невелика кількість потоків:

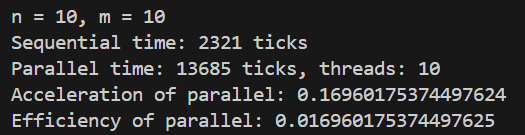


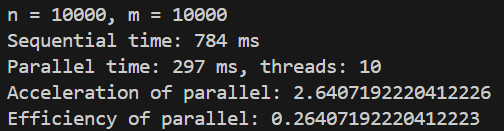


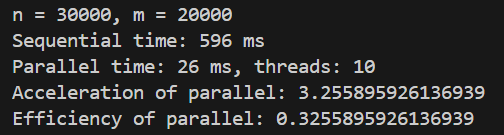


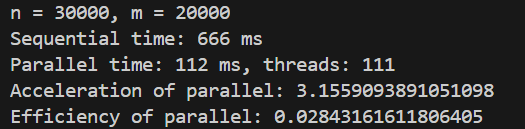
Для алгоритму віднімання матриць результати такі ж самі:











**Висновок:**

Виконуючи лабораторну роботу, я навчилася розпаралелювати алгоритми за допомогою Thread та взнала, у яких випадках це ефективно робити.