МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ

ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ імені ІВАНА ФРАНКА

Паралельні та розподілені обчислення

ЛАБОРАТОРНА РОБОТА №5

«Алгоритм Флойда»

Виконала:

студентка групи ПМі-31

Дудчак Валентина Юріївна

Львів 2024

**Тема**: Розпаралелення алгоритму Флойда-Варшалла

**Мета**: Написати програми розв’язування алготму Флойда (послідовний та паралельний алгоритми). Для орієнтованого зваженого графа G(V,F) знайти найкоротший шлях між заданими вузлами *а* та *в*. Порахувати час роботи кожної з програм, обчислити прискорення та ефективність роботи паралельного алгоритму.

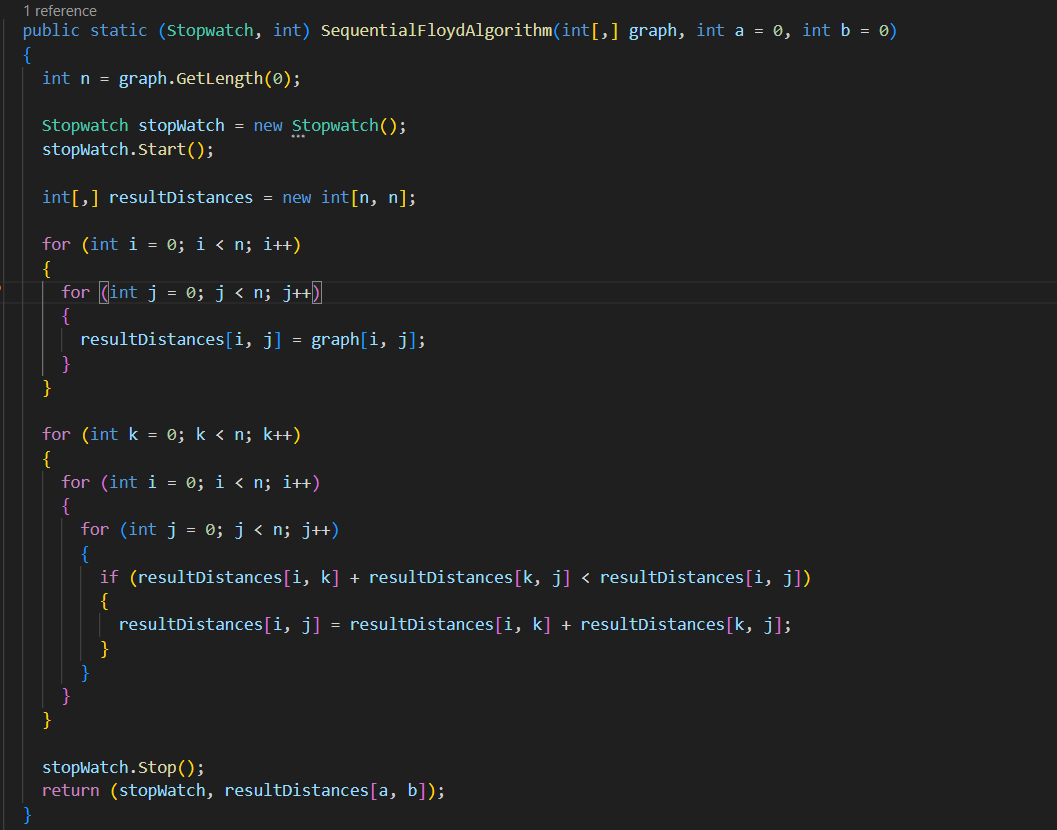
**Хід роботи:**

Для задачі я подаю граф у вигляді двовимірної квадратної матриці n x n, де індекси і та j – вершини, а значення на перетині – вага ребра. Для вершин, між якими немає ребра, присвоюється значення MAX\_VALUE. Для петель – значення 0.

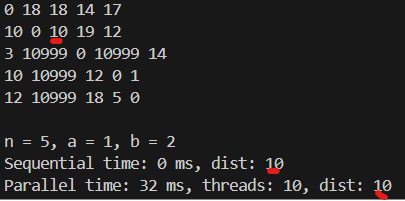
**Послідовний алгоритм**

Послідовний алгоритм Флойда реалізовано у методі SequentialFloydAlgorithm.

У ньому обчислюється найкоротший шлях між усіма парами вершин за допомогою алгоритму Флойда-Варшалла та зберігається у матрицю. Використано також об’єкт класу Stopwatch для засікання часу. Метод повертає об’єкт stopwatch та значення відстані між двома заданими вершинами:



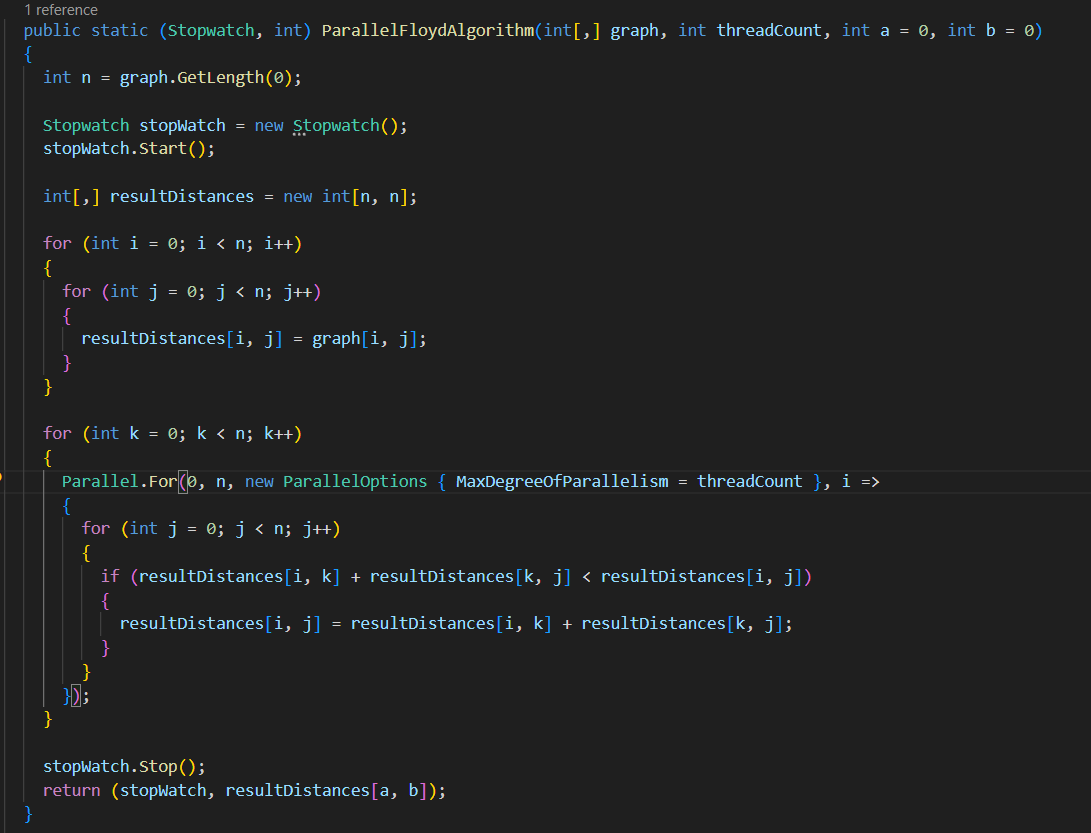
Крім цього, я перевірила, чи результати обчислюються правильно:



**Паралельний алгоритм**

Паралельний алгоритм Флойда реалізовано у методі ParallelFloydAlgorithm.

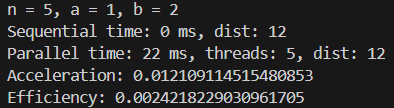
Використано об’єкт класу Tasks, зокрема метод Parallel.For, для розпаралелення обчислення найкоротшого шляху:

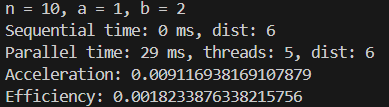


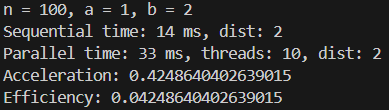
Метод також працює правильно (видно у результатах вище)

**Аналіз результатів**

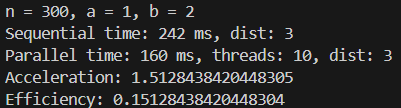
На малих розмірах графа та великій кількості потоків розпаралелення неефективне:

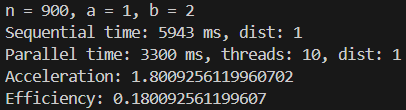


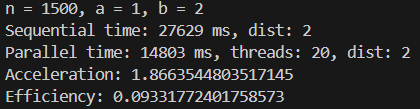




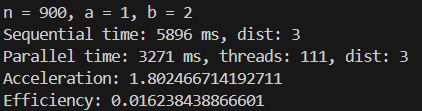
На більших розмірностях розпаралелення дає значно кращий результат:



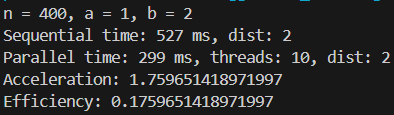


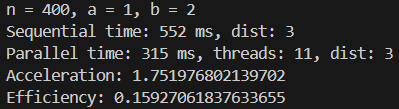


При великій кількості потоків ефективність трохи покращується:



Коли розмірності графа не кратна кількості потоків, ефективність незначно погіршується:





**Висновок:**

Виконуючи лабораторну роботу, я навчилася розпаралелювати алгоритми за допомогою Tasks.Parallel.For і взнала, у яких випадках це ефективно робити.