Міністерство освіти і науки України

Львівський національний університет імені Івана Франка

Факультет прикладної математики та інформатики

Кафедра програмування

Звіт

до лабораторної роботи №5

**“Алгоритм Флойда”**

Підготував:

студент групи ПМІ-31

 Процьків Назарій

Львів 2023

***Завдання***

Для орієнтованого зваженого графа G(V,F), де V={a0, a1,…an} – множина вершин (n –велике число), а F множина орієнтованих ребер (шляхів) між вершинами, використовуючи алгоритм Флойда, знайти найкоротший шлях між заданими вузлами а та в. Для різної розмірності графів та довільних вузлів а та в порахувати час виконання програми без потоків та при заданих k потоках розпаралелення.

***Теоретичні відомості***

Граф — це структура, що складається з набору об’єктів, у якому деякі пари об’єктів у певному сенсі «пов’язані». Об’єкти відповідають математичним абстракціям, які називаються вершинами (також називаються вузлами або точками), а кожна з пов’язаних пар вершин називається ребром (також називається ланкою або лінією). Як правило, граф зображується у вигляді діаграми як набір точок або кіл для вершин, з’єднаних лініями або кривими для ребер. Графи є одним з об’єктів вивчення дискретної математики.

Графом G = (V, Е) називають сукупність двох множин: скінченної непорожньої множини V вершин і скінченої множини Е ребер, які з'єднують пари вершин. Ребра зображуються невпорядкованими парами вершин (u, v).

У графі можуть бути петлі — ребра, що починаються і закінчуються в одній вершині, а також повторювані ребра (кратні, або паралельні). Якщо в графі немає петель і кратних ребер, то такий граф називають простим. Якщо граф містить кратні ребра, то граф називають мультиграфом.

Ребра вважаються неорієнтованими в тому сенсі, що пари (u, v) та (v,u) вважаються одним і тим самим ребром.

Зваженим називають простий граф, кожному ребру e якого приписано дійсне число w(e). Це число називають вагою ребра e.

Алгоритм Флойда призначений для знаходження найкоротшого шляху між усіма парами вершин у заданому зваженому орієнтованому графі. Цей алгоритм використовує підхід динамічного програмування для пошуку найкоротшого шляху.

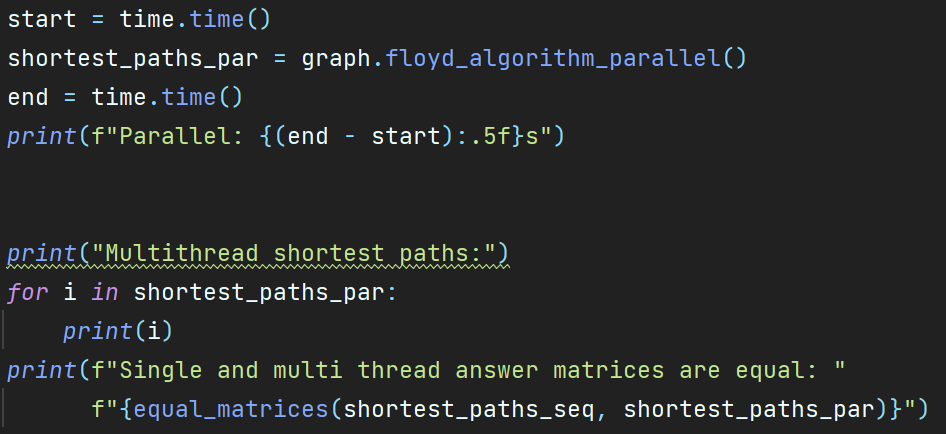
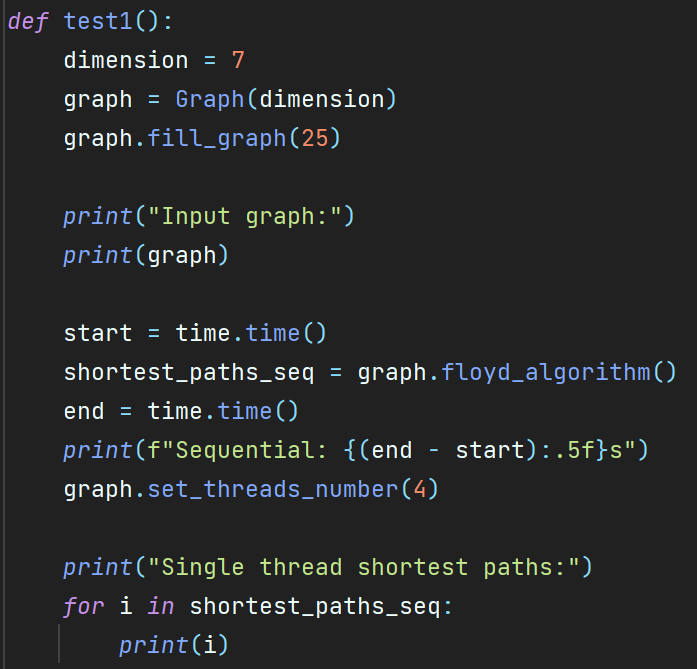
Найкраща, найгірша та середня швидкодія O(V^3).

Об’єм пам’яті O(V^2).

***Хід роботи***

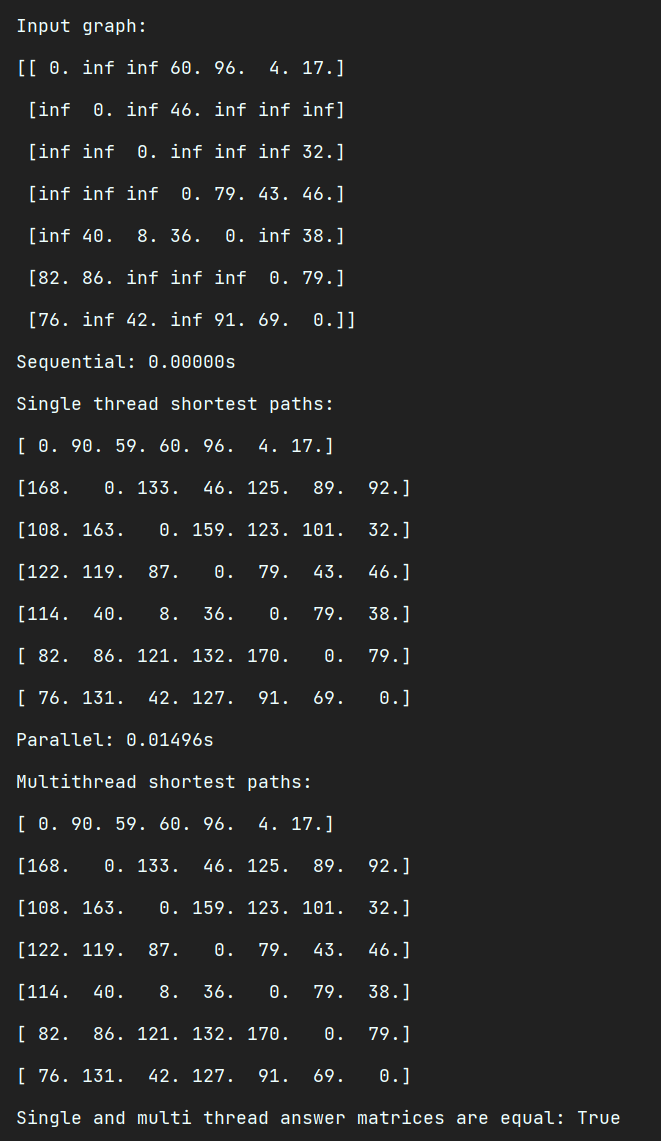
Виконав цю лабораторну мовою програмування Python.

Створив перший тест test1() щоб перевірити роботу алгоритму на малих графах:

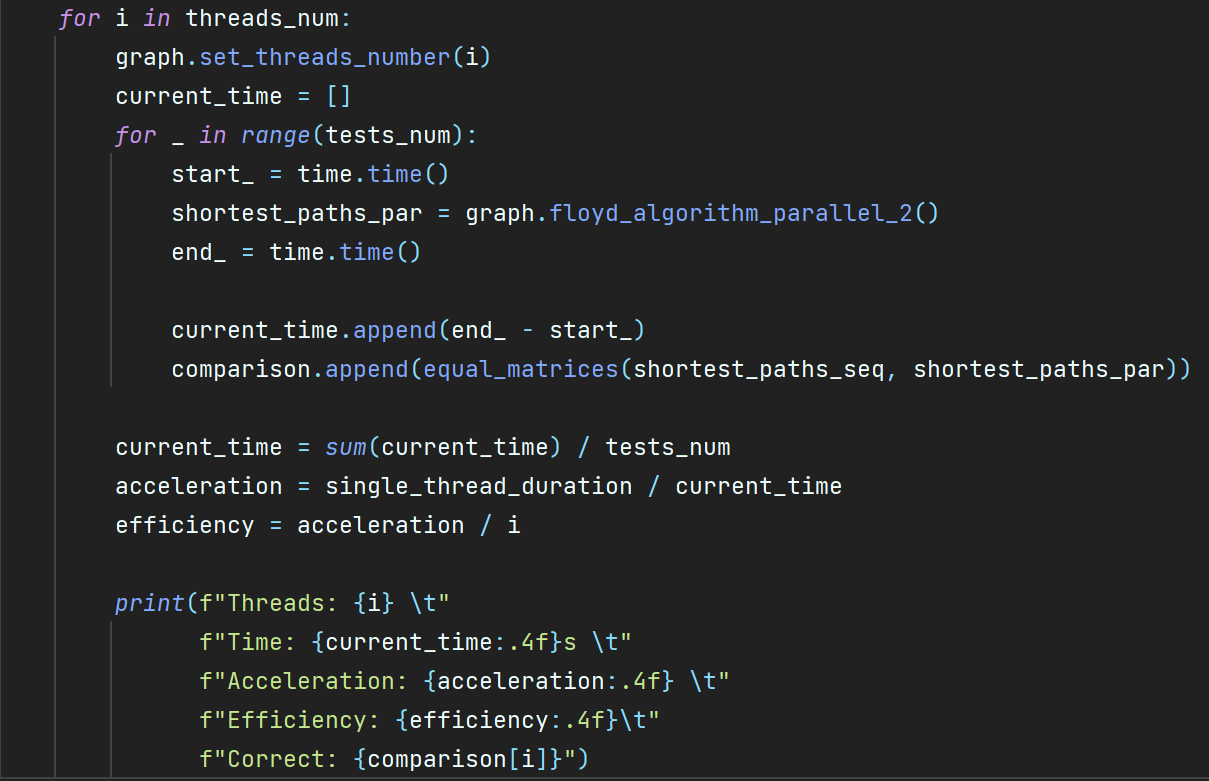
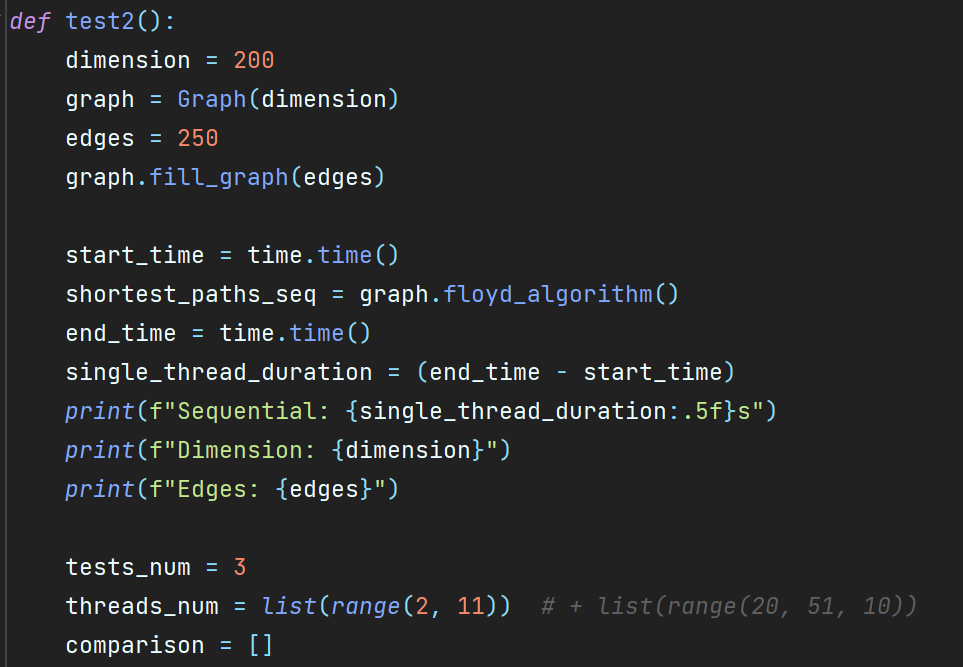


Тут створив граф з 7-ма вершинами і 25-ма ребрами, вивів його на екран, заміряв час виконання послідовного алгоритму, вивів час і граф на екран, побачив, що алгоритм працює правильно і заміряв час виконання паралельного алгоритму, вивів час і граф на екран. Побачив що результуючі графи після і паралельного алгоритмів збігаються.

Результат:

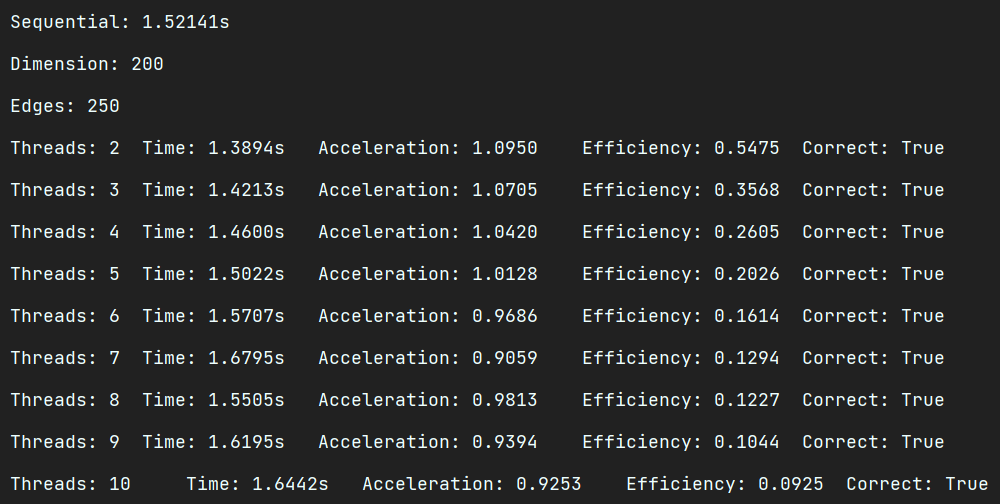


Створив другий тест test2(), щоб заміряти роботу алгоритму на великих графах:



В цьому тесті створив граф на 200 вершин і заповнив його 250 ребрами. Заміряв час виконання послідовного алгоритму, вивів його на екран, далі в циклі для кількості потоків від 2 до 10 заміряв час виконання паралельного алгоритму, для кожної кількості потоків провів по 3 експерименти і взяв з них середнє арифметичне та вивів на екран прискорення та ефективність. І для цієї лабораторної роботи вирішив ще для кожного потоку вивести чи збігається його результат з результатом виконання послідовного алгоритму (Correct).

Результат:



***Висновок***: під час виконання лабораторної роботи №5 написав програму для знаходження найкоротшого шляху між усіма парами вершин у зваженому орієнтованому графі, використовуючи алгоритм Флойда (послідовний та паралельний), обчислив прискорення та ефективність для різної кількості потоків та навчився аналізувати ці дані.