Міністерство освіти і науки України

Львівський національний університет імені Івана Франка

Факультет прикладної математики та інформатики

Кафедра програмування

Звіт

до лабораторної роботи №6

**“Алгоритм Дейкстри”**

Підготував:

студент групи ПМІ-31

 Процьків Назарій

Львів 2023

***Завдання***

Для зваженого графа G(V,F), де V={a0, a1,…an} – множина вершин (n –велике число), а F множина ребер між вершинами, використовуючи алгоритм Дейкстри, знайти найкоротший шлях між заданою вершиною а та усіма іншими. Для різної розмірності графів та довільного вузла а 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 Дейкстри – знаходження найкоротшого шляху від заданої вершини графа до всіх інших вершин цього графа.

Швидкодія - O(E \* logV). Об’єм пам’яті - O(V).

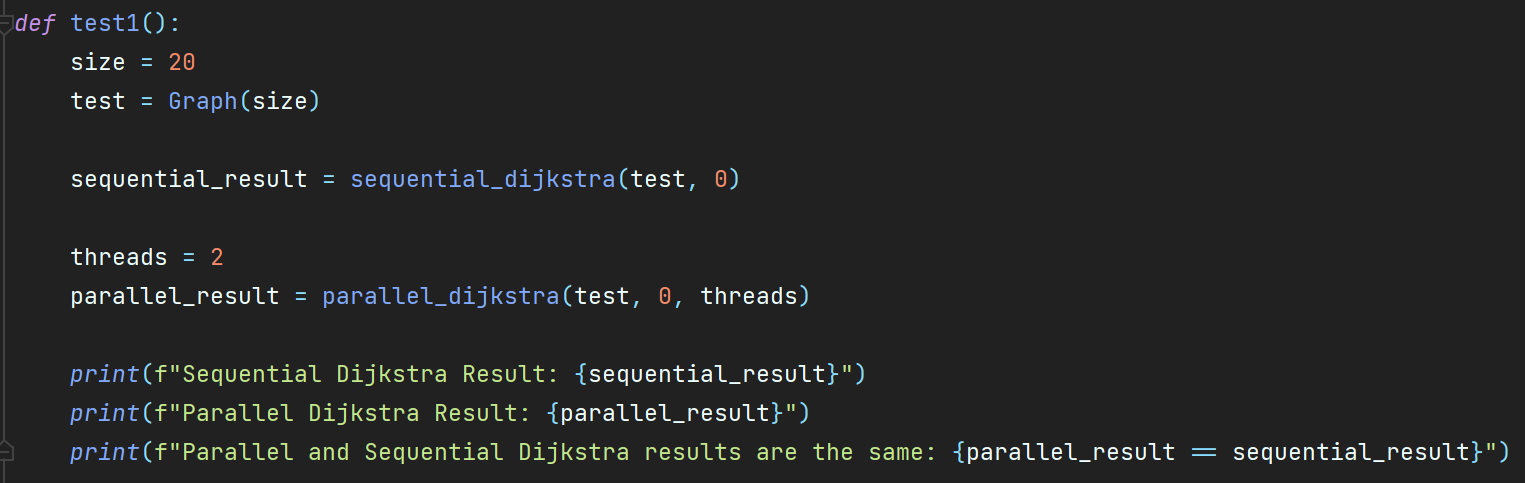
***Хід роботи***

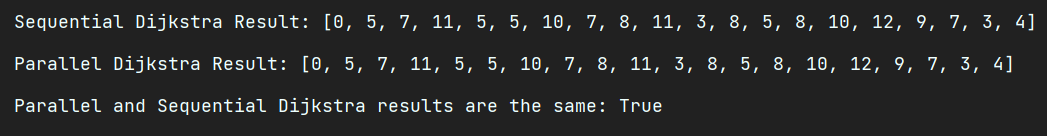
Виконав цю лабораторну мовами програмування C# і Python.

На код на C# можна навіть не дивитись, бо там все добре працює, а от розпаралелити Дейкстри на Python це для мене було справжнім випробуванням. Що я тільки не пробував робити. В мене вийшло зробити цю лабораторну з 4 (четвертого) разу. Це довга історія, в одному звіті не вийде розказати…

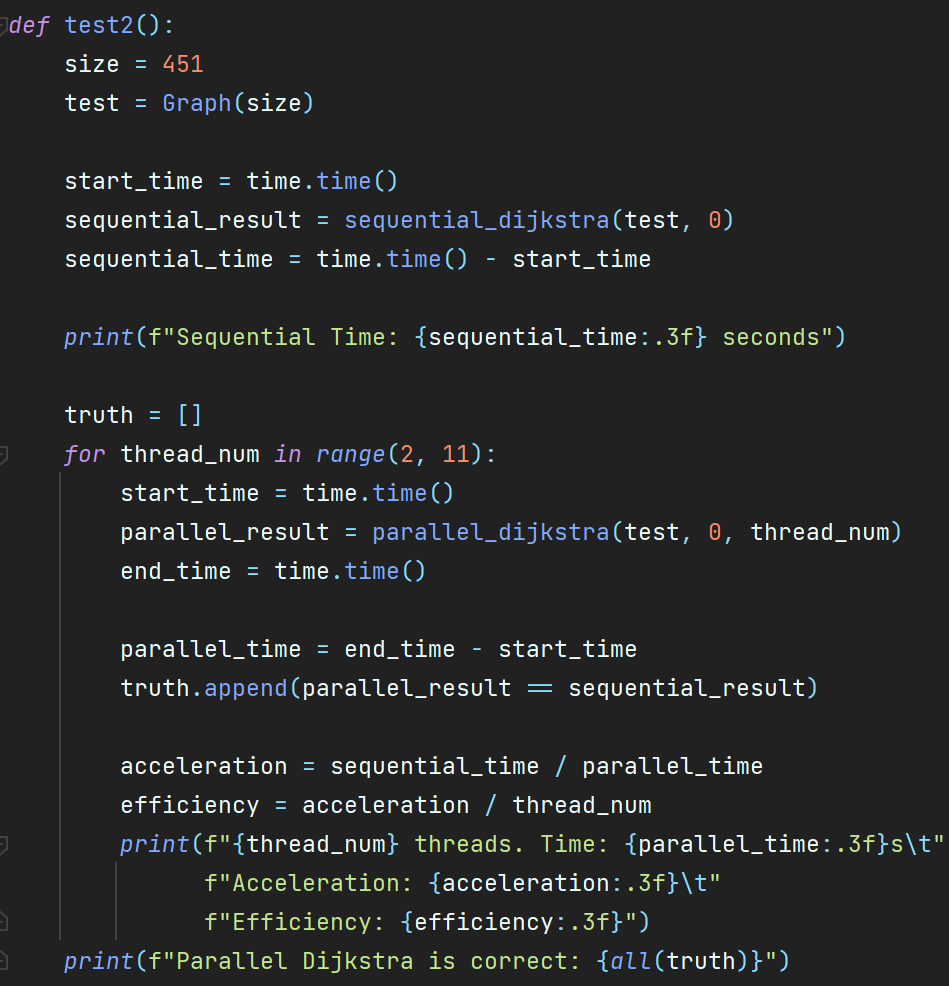
Перейдемо до правильної реалізації Дейкстри на Python.

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



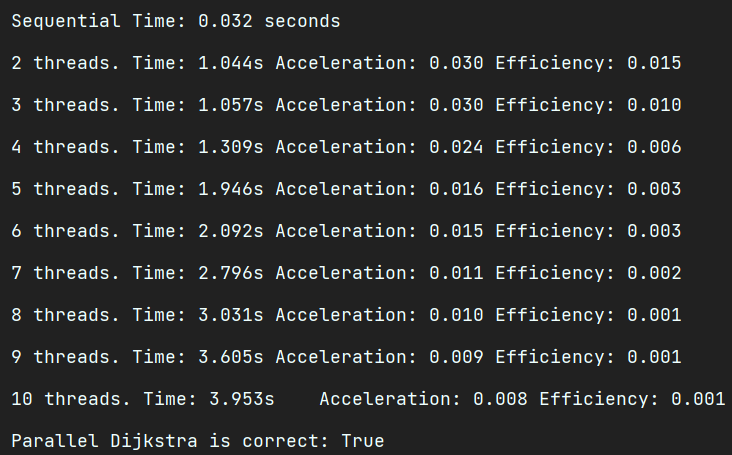
Цей тест проведений навіть без заміряння часу, просто для визначення того чи результати послідовного і паралельного алгоритмів збігаються. Це було зробити найважче, але, як можна бачити на скріншоті нижче, все збіглось:

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



В цьому тесті створив граф на 451 вершину. Виміряв час виконання послідовного алгоритму, вивів його на екран, далі в циклі для кількості потоків від 2 до 10 виміряв час виконання паралельного алгоритму, вивів прискорення та ефективність на екран. І для цієї лабораторної роботи вирішив ще для кожного потоку вивести чи збігається його результат з результатом виконання послідовного алгоритму (останній рядок коду).

Результат:



На жаль, для алгоритму Дейкстри на пайтоні мені не вдалось отримати адекватні значення прискорення та ефективності, зате завжди правильно виконується. Ще я писав інші варіанти виконання цієї лабораторної і там значення прискорення та ефективності виходили набагато кращими, але правильність роботи паралельного алгоритму не вражала.

***Висновок***: під час виконання лабораторної роботи №6 написав програму для знаходження найкоротшого шляху між заданою вершиною та всіма іншими вершинами у зваженому орієнтованому графі, використовуючи алгоритм Дейкстри (послідовний та паралельний), обчислив прискорення та ефективність для різної кількості потоків та навчився аналізувати ці дані.