Міністерство освіти і науки України
Львівський національний університет імені Івана Франка
Факультет прикладної математики та інформатики

Звіт

Лабораторна робота №5

Тема: «Алгоритм Флойда»

з дисципліни "Паралельні та розподільні обчислення"

Виконав студент групи ПМі-31
Яцуляк Андрій

Львів 2023 р.

Мета: Написати програми знаходження найкоротшого шляху між всіма парами вершин у зваженому орієнтованому графі, використовуючи алгоритм Флойда (послідовний та паралельний).

Теоретичний матеріал

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, а кожна з пов’язаних пар вершин називається ребром. 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, E) називають сукупність двох множин: скінченої непорожньої множини V **вершин** і скінченої множини E **ребер**, які з'єднують пари вершин. Ребра зображаються 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.

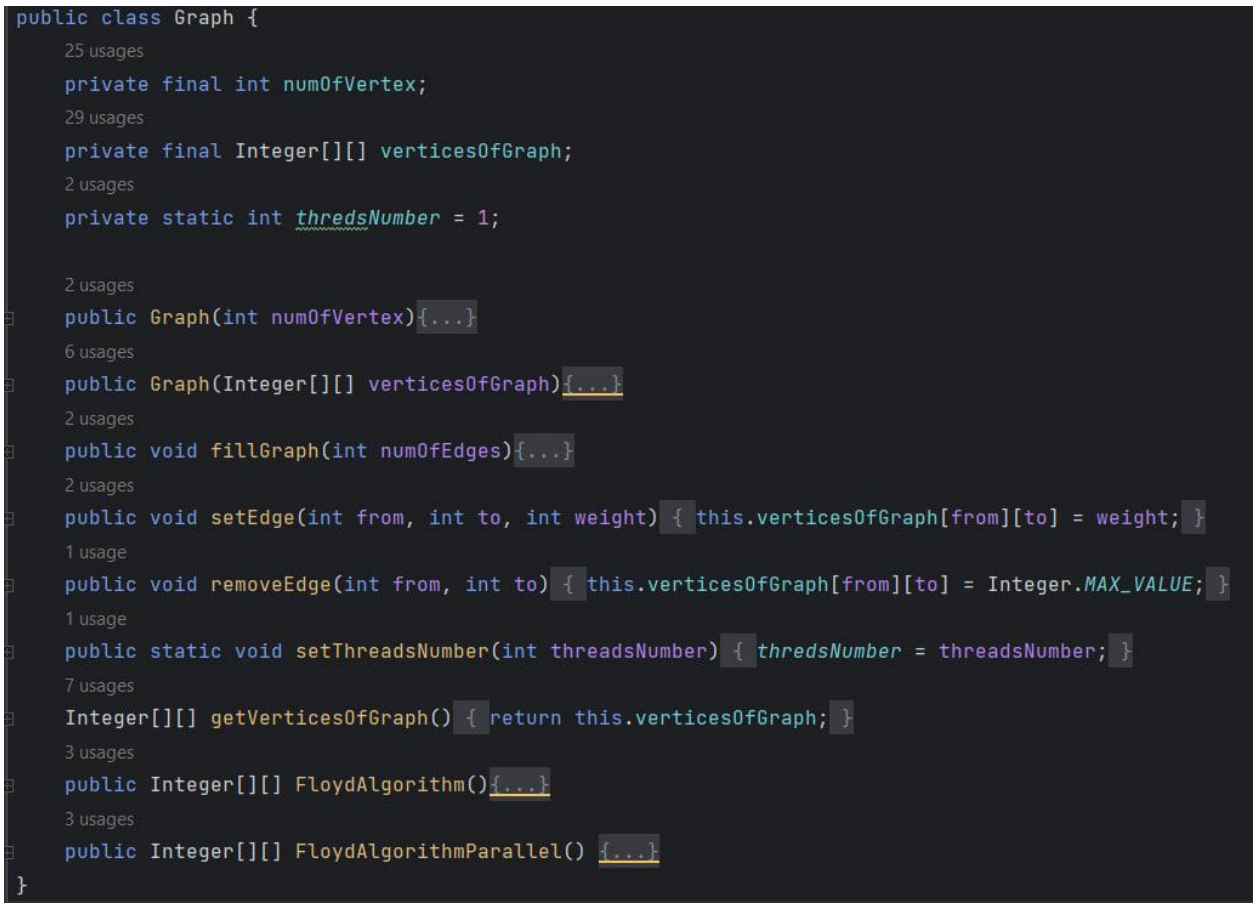
Алгоритм Флойда призначений для знаходження найкоротшого шляху між всім парами вершин у заданому зваженому орієнтованому графі. Цей алгоритм використовує підхід динамічного програмування для пошуку найкоротшого шляху.

Найкраща, найгріша та середня швидкодія $O(|V|^3)$.

Об’єм пам’яті $O(|V|^2)$.

Хід роботи

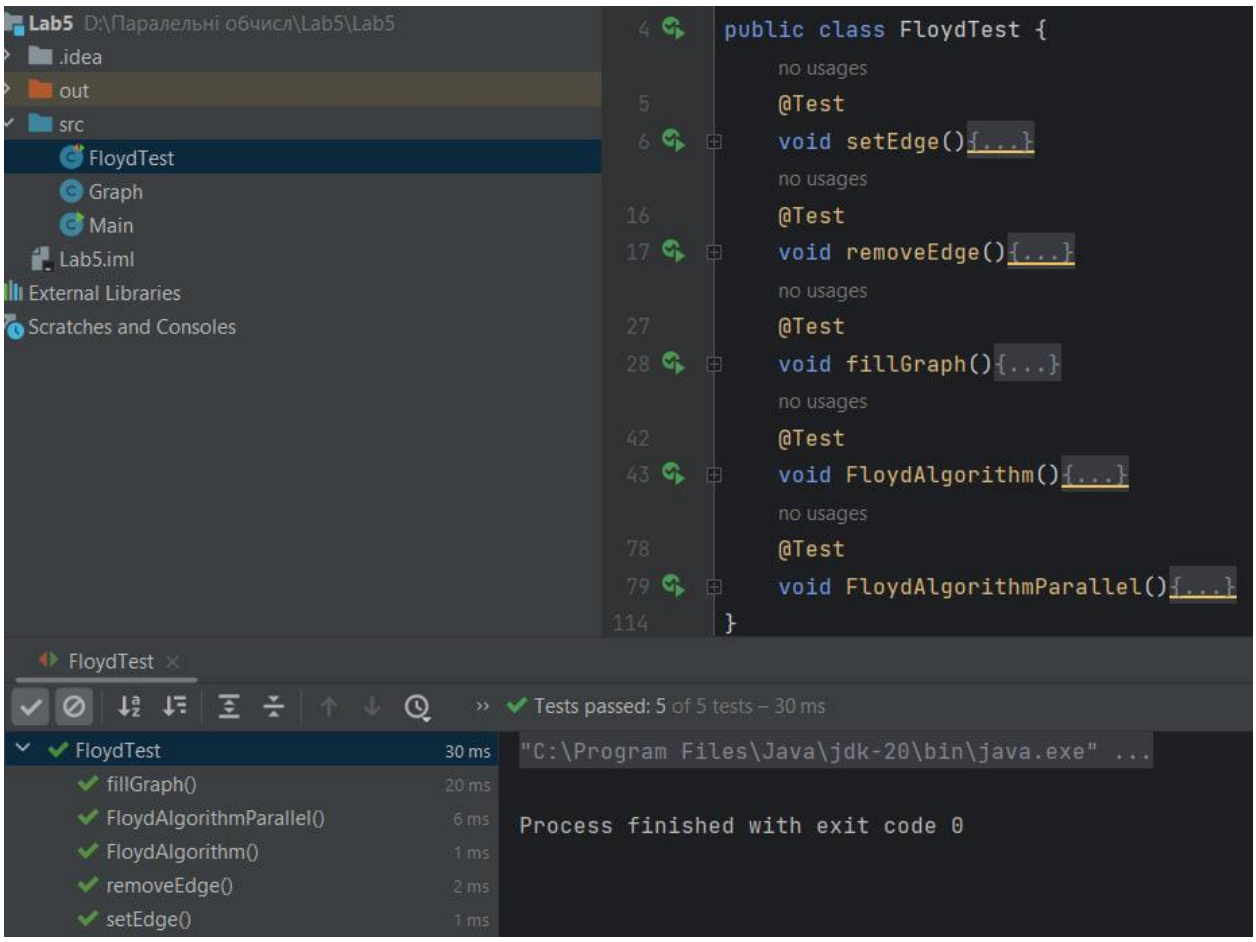
Завдання виконав мовою програмування Java у середовищі IntelliJ IDEA. Написав повноцінну програму для роботи з зваженими орієнтованими графами.



Задається граф матрицею інцидентності, де verticesOfGraph[i][j] – вага орієнтованого ребра від i до j.

Робота з графами

Перед початком основної роботи unit тестів, аби перевірити методи на правильність роботи:



Переконавшись, що все працює правильно, створив граф з 800 вершинами та 15000 орієнтованими ребрами зі значеннями від 1 до 100. Ребра з’єднують випадкові вершини. Циклом для різної кількості ядер обчислив час послідовної та паралельної роботи алгоритму, прискорення, ефективність:

```
Enter the number of vertices: 800
Enter the number of edges: 15000
Sequential time: 1778218100 nanoseconds

Threads number: 2
Parallel time: 483186300 nanoseconds
Speed up: 3.6801914706604886
Efficiency: 1.8400957353302443

Threads number: 3
Parallel time: 266446000 nanoseconds
Speed up: 6.673840477995541
Efficiency: 2.2246134926651804

Threads number: 4
Parallel time: 374110400 nanoseconds
Speed up: 4.7531907693557836
Efficiency: 1.1882976923389459

Threads number: 8
Parallel time: 380321500 nanoseconds
Speed up: 4.6755655412591715
Efficiency: 0.5844456926573964

Threads number: 16
Parallel time: 359670700 nanoseconds
Speed up: 4.944017124553098
Efficiency: 0.30900107028456864
```

Отже, найкращого прискорення з показником 6.67 вдалось досягнути при 3-х потоках, ефективність у 2.22 тут теж є найкращою. При збільшенні кількості потоків ефективність зменшується.

Висновок. Під час виконання лабораторної роботи я написав програму для знаходження найкоротшого шляху між всіма 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.