Львівський національний університет імені Івана Франка

Факультет прикладної математики та інформатики

**Паралельні та розподілені обчислення**

**Лабораторна робота №5**

Виконав:

Студент групи ПМі-33

Тимчишин Ярема

Львів 2023

**Тема:** Алгоритм Флойда.

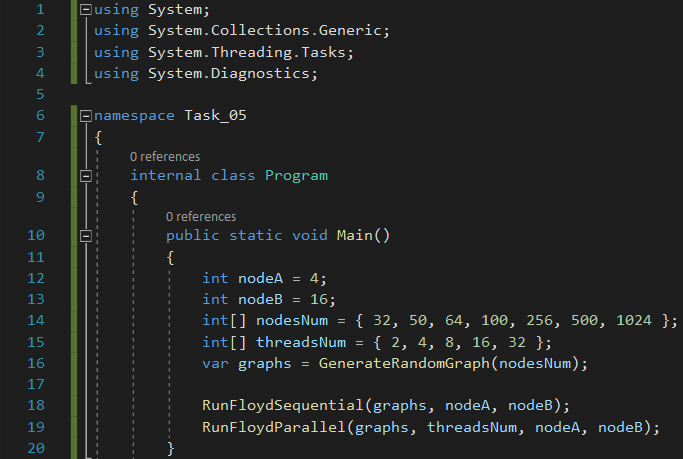
**Завдання:** Для орієнтованого зваженого графа G(V,F), де V={a0, a1,…an} – множина вершин (n –велике число), а F множина орієнтованих ребер (шляхів) між вершинами, використовуючи алгоритм Флойда, знайти найкоротший шлях між заданими вузлами a та b.

Для різної розмірності графів та довільних вузлів a та b порахувати час виконання програми без потоків та при заданих k потоках розпаралелення.

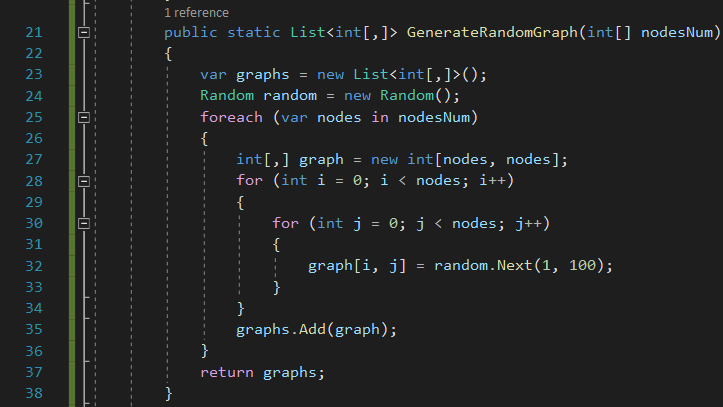
**Хід роботи**

Використовуючи мову програмування C# написав програму для знаходження найкоротшого шляху між заданими вузлами a та b за допомогою розпаралелення методу Флойда.

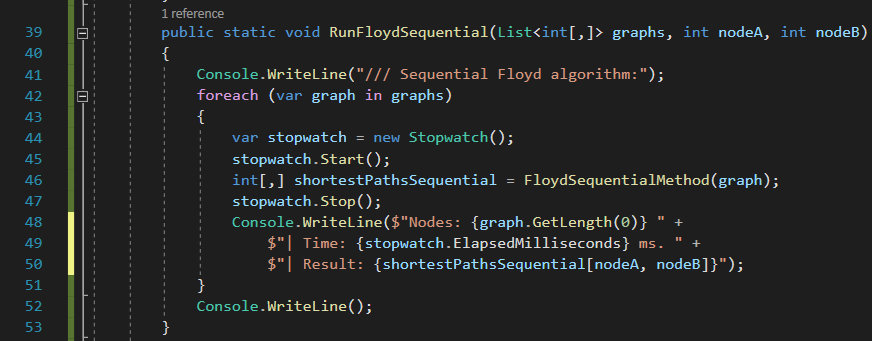
1. Метод **Main()**. Оголошення всіх змінних та виклик двох основних методів RunFloydSequential() та RunFloydParallel().



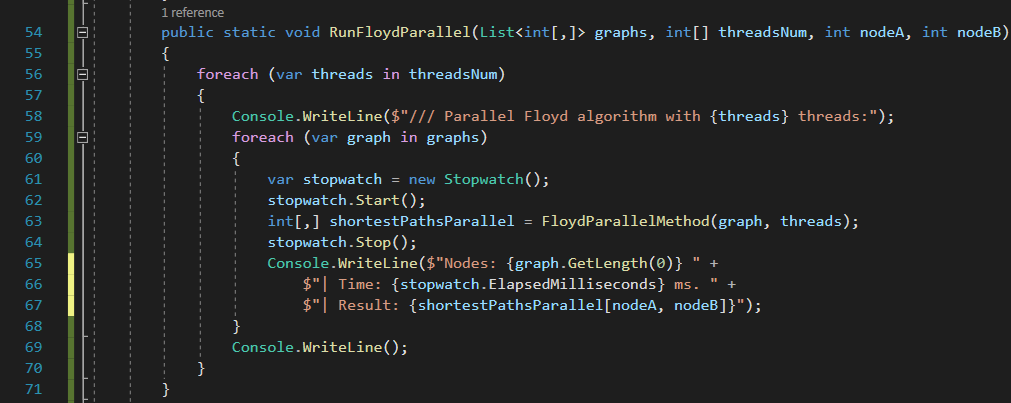
1. Метод **GenerateRandomGraph()**. На основі масиву з кількостями вершин цей метод генерує випадкові графи , заповнюючи матрицю суміжності випадковими вагами ребер, після чого повертає список цих графів.



1. Метод **RunFloydSequential()**. Цей метод викликає послідовний алгоритм Флойда для кожного графа у списку і виводить час виконання та результат - найкоротший шлях між вершинами a та b.



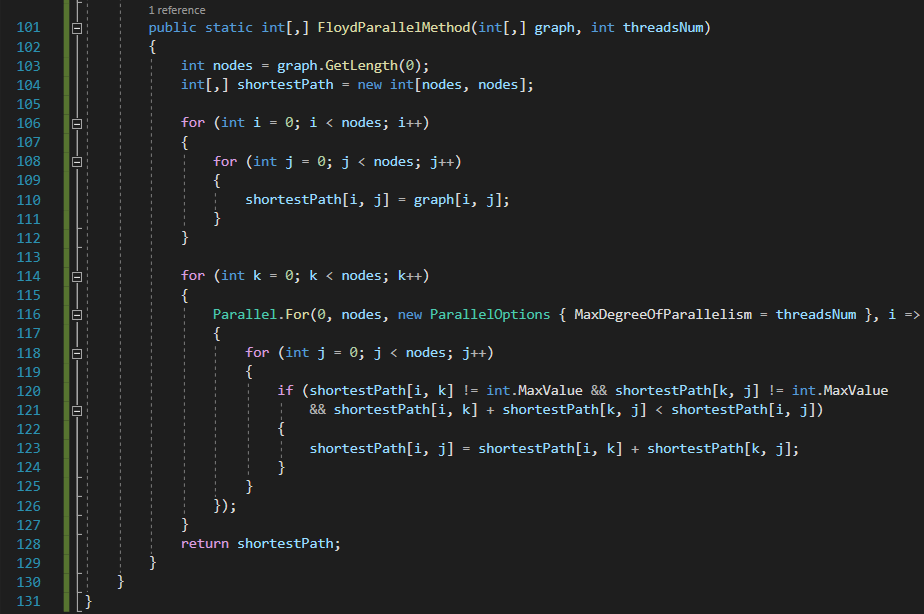
1. Метод **RunFloydParallel()**. Цей метод викликає паралельний алгоритм Флойда з різною кількістю потоків для кожного графа у списку. Він виводить час виконання та результат найкоротшого шляху між вузлами a та b для кожної конфігурації.



1. Метод **FloydSequentialMethod()**. Це реалізація вже самого послідовного алгоритму Флойда. Він обчислює найкоротші шляхи між усіма парами вершин у графі.



1. Метод **FloydParallelMethod()**. Це паралельна реалізація алгоритму Флойда з використанням багатопоточності. Вона використовує цикл *Parallel.For* для розподілу роботи між декількома потоками на основі заданої кількості потоків *threadsNum*. Вона обчислює найкоротші шляхи між усіма парами вершин графа одночасно.



Результат роботи програми

