Міністерство освіти і науки України

Львівський національний університет імені Івана Франка

Факультет прикладної математики та інформатики

Кафедра програмування

Звіт

до лабораторної роботи №7

**“Алгоритм Прима”**

Підготував:

студент групи ПМІ-31

 Процьків Назарій

Львів 2023

***Завдання***

Для зваженого зв'язного неорієнтованого графа G, використовуючи алгоритм Прима, з довільно заданої вершини а побудувати мінімальне кісткове дерево. 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.

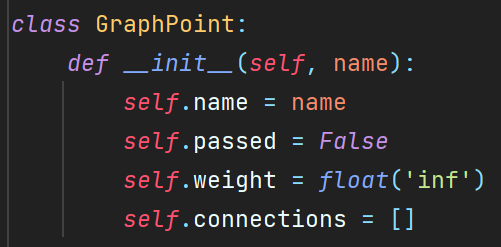
Зваженим називають простий граф, кожному ребру e якого приписано дійсне число w(e). Це число називають вагою ребра e.

***Хід роботи***

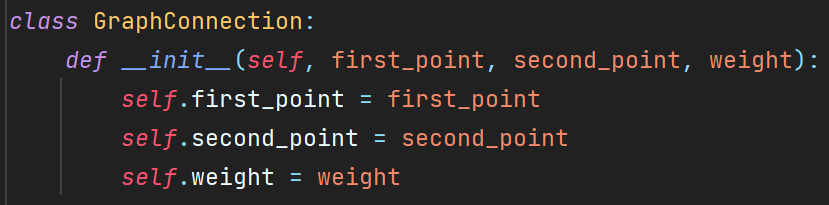
Виконав цю лабораторну мовою програмування Python.

Реалізація:

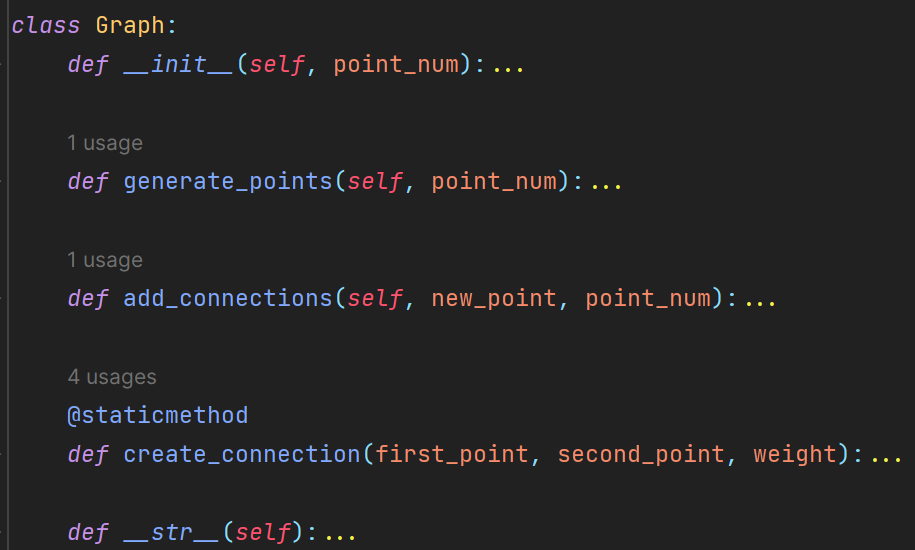
Створив клас GraphPoint:



Створив клас GraphConnection:

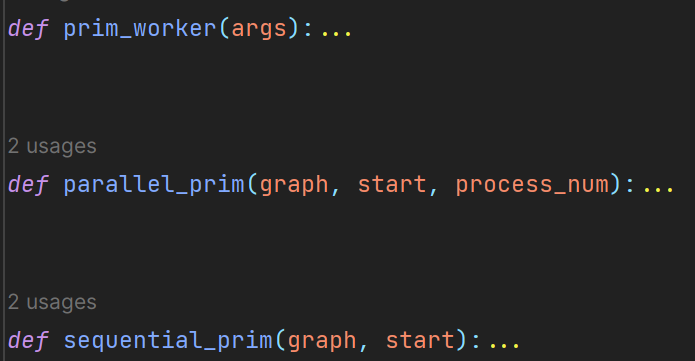


Створив клас Graph:



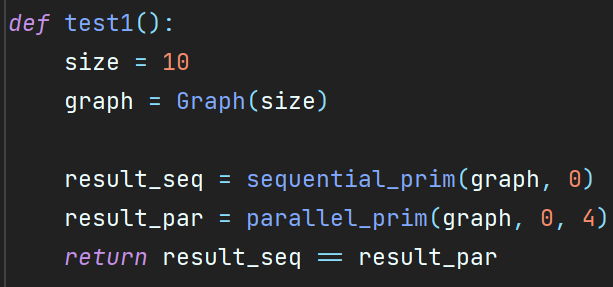
Він генерує рандомний граф, з допомогою методу add\_connections.

Написав три функції:



1. Допоміжна функція для паралельного алгоритму Прима.
2. Функція паралельного алгоритму Прима.
3. Функція послідовного алгоритму Прима.

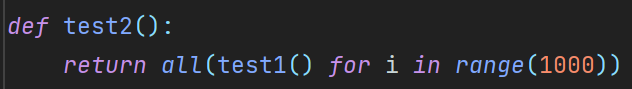
Створив перший тест test1() щоб перевірити роботу алгоритму на малих графах:



Тут створив граф на 10 вершин і порівняв чи результати роботи послідовного та паралельного алгоритмів збігаються.

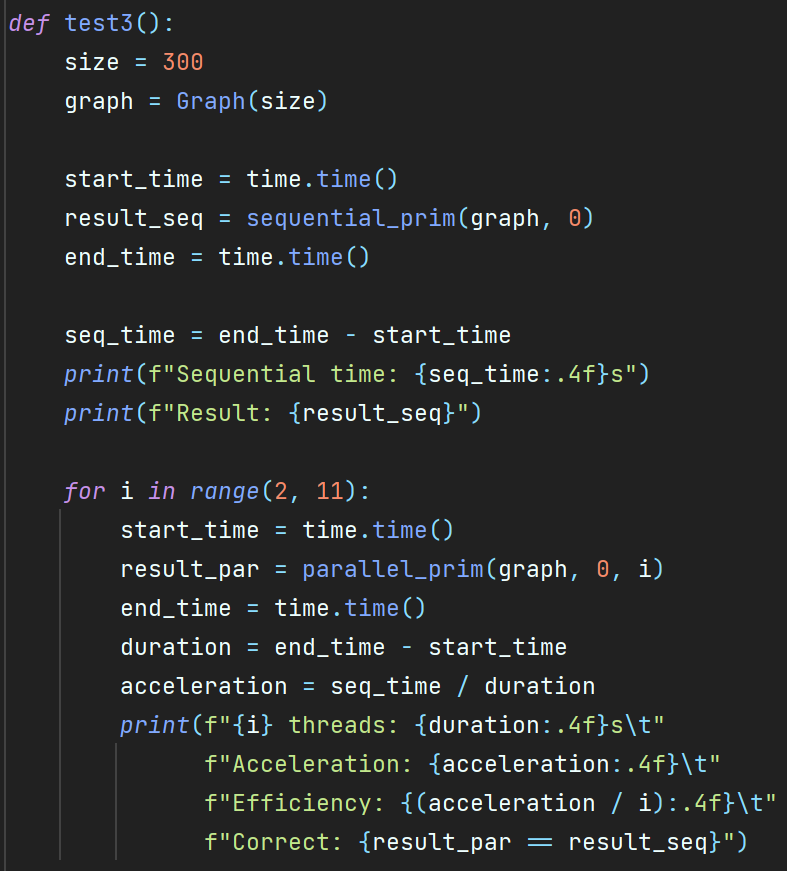
Результат: True

Створив другий тест test2() щоб прогнати test1() багато разів і подивитись чи точно все працює правильно:

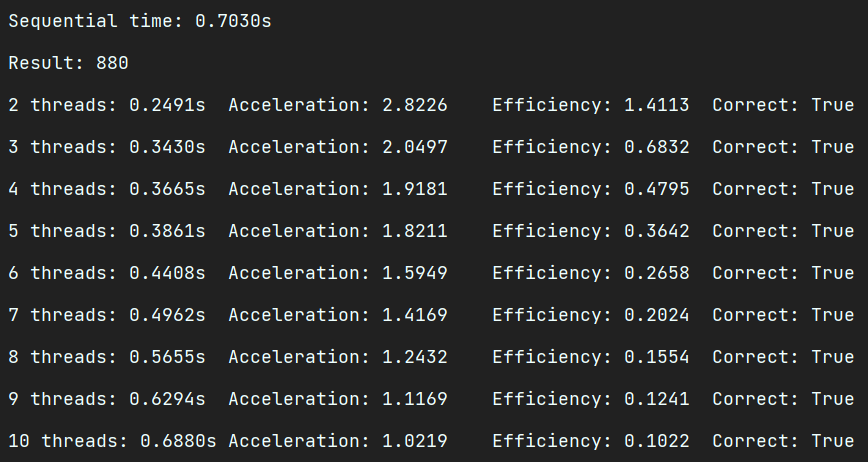


Результат: True

Створив третій тест test3() щоб перевірити правильність роботи алгоритму на великих графах та визначити прискорення та ефективність:



Результат:



В цьому тесті створив граф на 300 вершин. 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. Для цієї лабораторної роботи вирішив для кожного потоку також вивести чи збігається його результат з результатом виконання послідовного алгоритму.

***Висновок***: під час виконання лабораторної роботи №7 написав програму для побудови мінімального кістякового дерева у зваженому неорієнтованому графі, використовуючи алгоритм Прима (послідовний та паралельний), обчислив прискорення та ефективність для різної кількості потоків та навчився аналізувати ці дані.