ПРЕЗЕНТАЦІЯ ДО КУРСОВОЇ РОБОТИ НА ТЕМУ:

ПОРІВНЯЛЬНИЙ АНАЛІЗ АЛГОРИТМІВ ПОШУКУ НАЙКОРОТШОГО ШЛЯХУ В ГРАФАХ З ВАГАМИ

Виконала: Трюхан Тетяна

Науковий керівник: Картавий М. О.

АКТУАЛЬНІСТЬ ТА МЕТА ДОСЛІДЖЕННЯ

Актуальність: існує ряд проблем з реального життя, які можна представити як задачі на графах. Насьогодні недостатньо просто розв’язувати такі задачі, це необхідно робити максимально швидко та з використанням мінімуму ресурсів. Тому немає межі для покращення алгоритмів, завжди можна підійти до проблеми з нового боку. Першим кроком до створення нових чи покращення наявних алгоритмів є поглиблений аналіз вже існуючого.

Метою даної роботи є визначити, як впливають на швидкість прокладання маршрутів обрані алгоритми пошуку найкоротшого шляху в зважених графах та створити демонстраційний застосунок.

ЗАВДАННЯ

- Дослідити графи (їх структуру, різновиди, особливості) та алгоритми пошуку найкоротшого шляху в зважених графах
- Проаналізувати особливості роботи алгоритмів A*, Беллмана-Форда, Флойда-Уоршала
- Реалізувати демонстраційний застосунок
- Порівняти ефективність роботи обраних алгоритмів на різних графах

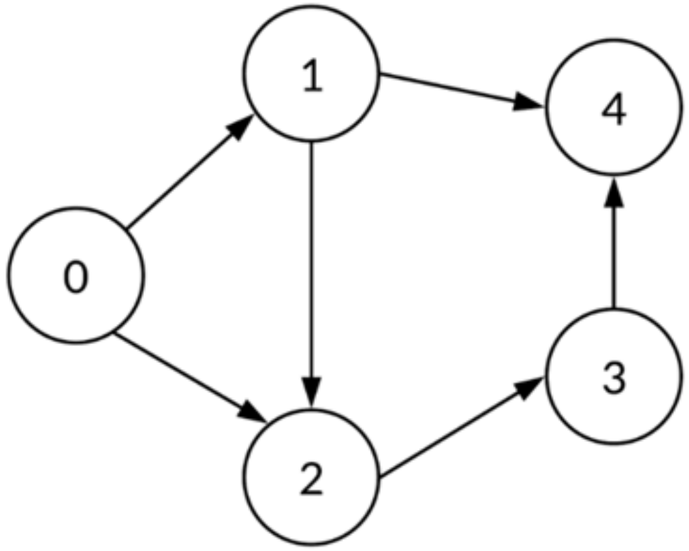
ГРАФИ

Графи - структура даних, яка являє собою скінченну множину вузлів та ребер.

Ключові терміни: вузол, ребро, щільність, маршрут, шлях, цикл

Способи представлення:

- Список суміжності
- Матриця суміжності
- Список ребер
- Матриця інцидентності



Adjacency Matrix

	0	1	2	3	4
0	0	1	1	0	0
1	0	0	1	0	1
2	0	0	0	1	0
3	0	0	0	0	1
4	0	0	0	0	0

Приклад матриці суміжності, джерело: [https://miro.medium.com/v2/resize:fit:1400/format:webp/0*XGraRBR3RaxTaWjQ.png]

АЛГОРИТМИ ПОШУКУ ШЛЯХУ В ЗВАЖЕНИХ ГРАФАХ

Стратегії виконання:

- Жадібні
- Динамічні
- Евристичні

Алгоритм
Дейкстри
Беллмана–Форда

Покращення:

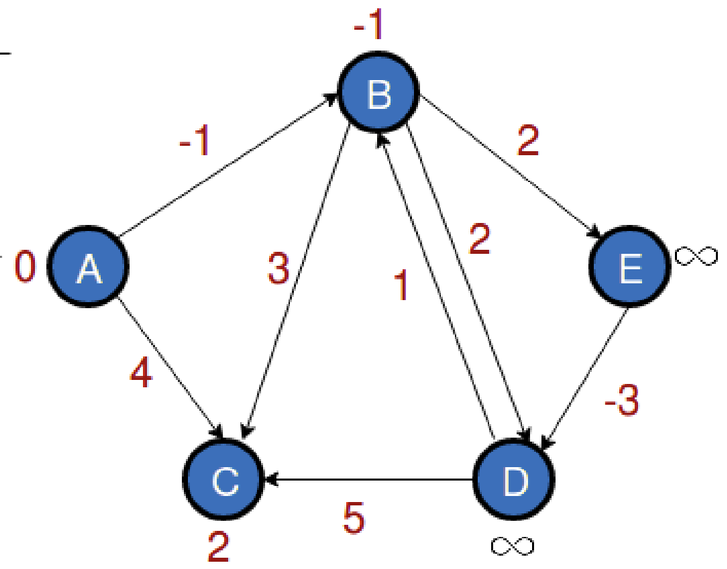
- Дейкстри: A*, двонаправлений
- Беллмана- Форда: SPFA
- A*: ALT, Jump Point Search

АЛГОРИТМ БЕЛЛМАНА-ФОРДА

Стратегія виконання

- Ініціалізація: встановити відстань до всіх вершин як ∞ , стартової - 0
- Релаксація ребер: оновлення оцінки відстані до кінцевого вузла
- Перевірка на наявність циклів з негативною вагою

A	B	C	D	E
0	∞	∞	∞	∞
0	-1	∞	∞	∞
0	-1	4	∞	∞
0	-1	2	∞	∞

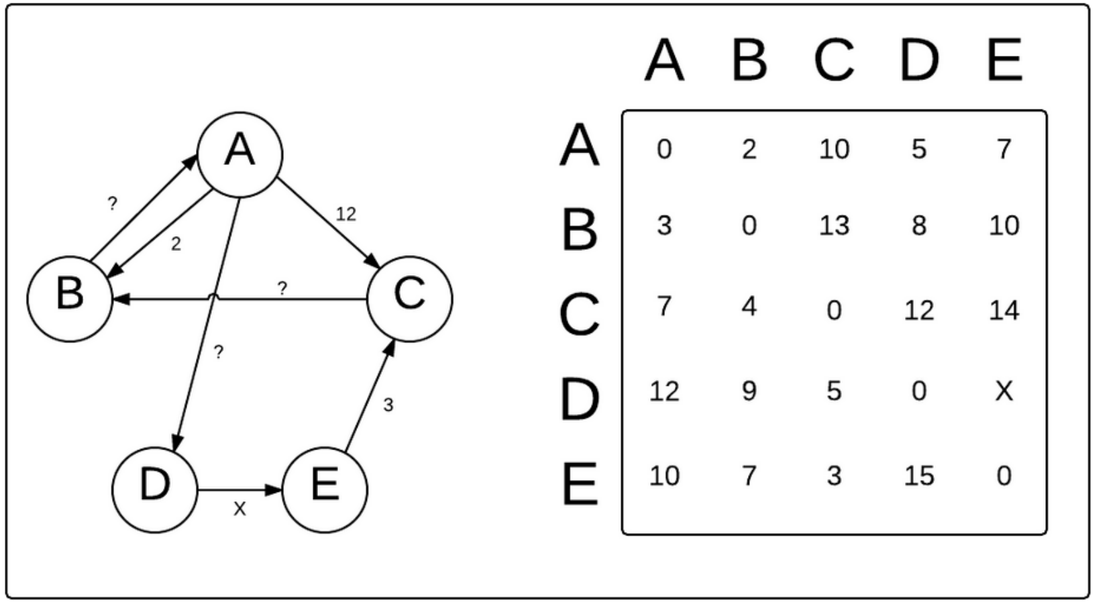


Алгоритм Беллмана- Форда, джерело: [https://media.geeksforgeeks.org/wp-content/uploads/bellmanford2.png]

АЛГОРИТМ ФЛОЙДА-УОРШАЛА

Стратегія виконання

- Ініціалізація: створити матрицю відстаней
- Основний цикл:
  для кожної вершини розглядається можливість, що вона може бути "проміжною ланкою" між іншими двома вершинами
- Перевірка на наявність
  циклів з негативною вагою

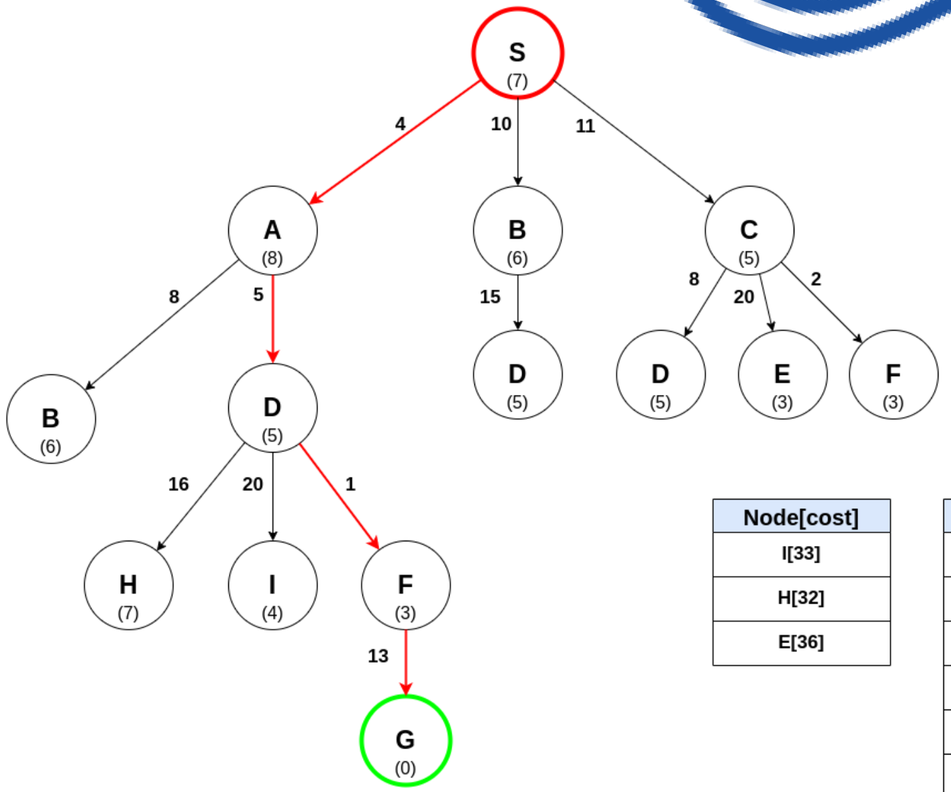


Алгоритм Флойда- Уоршала, джерело: [https://www.google.com/url?sa=i&url=https%3A%2F%2Fbrilliant.org%2Fwiki%2Ffloyd-warshall-algorithm%2F&psig=AOvVaw3LeNllOWr3x55gX-lnGOGj&ust=1746554211837000&source=images&cd=vfe&opi=89978449&ved=0CBQQjRxqFwoTCPC1yNHzjI0DFQAAAAAdAAAAABAE]

АЛГОРИТМ A*

Стратегія виконання

- Ініціалізація: черга з пріоритетом, визначення оцінки відстані для вершин
- Пошук:
  поки черга не пуста, обрати вершину з найменшою відстанню,
  якщо ціль - завершити,
  інакше - перейти до сусідів



Node[cost]
I[33]
H[32]
E[36]

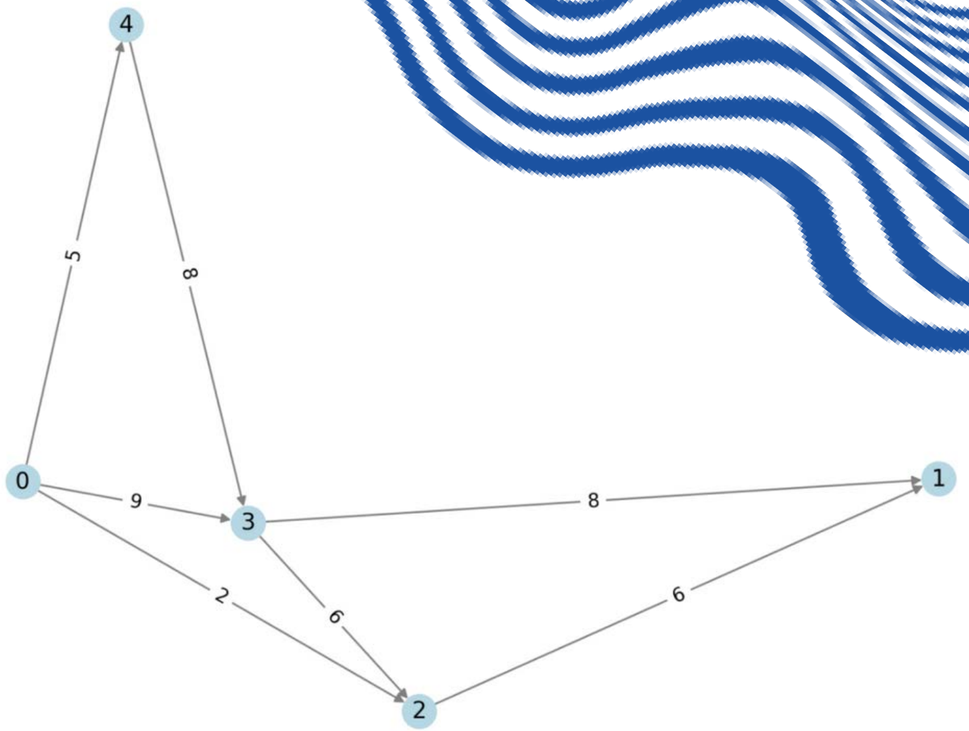
Closed List
S
A
D
F
B
C
G

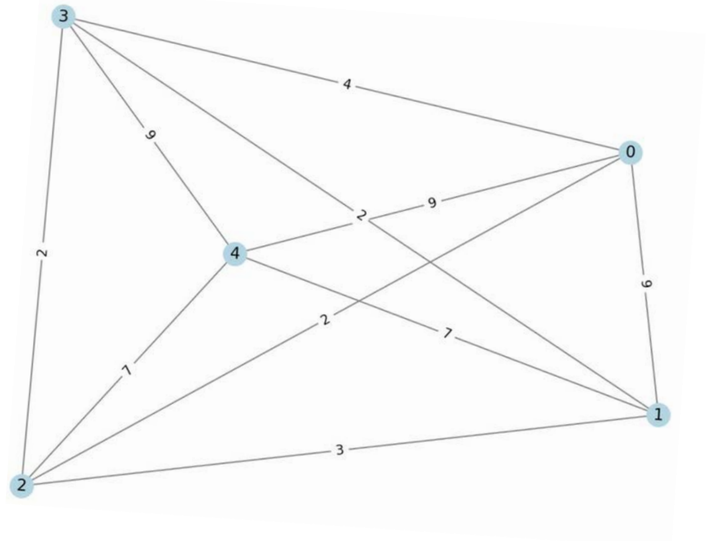
Алгоритм A*, джерело:
[https://www.google.com/url?sa=i&url=https%3A%2F%2Fbrilliant.org%2Fwiki%2Ffloyd-warshall-algorithm%2F&psig=AOvVaw3LeNllOWr3x55gX-lnGOGj&ust=1746554211837000&source=images&cd=vfe&opi=89978449&ved=0CBQQjRxqFwoTCPC1yNHzjI0DFQAAAAAdAAAAABAE]

ЗАСТОСУНОК

Способи представлення графів:

- Список суміжності
- Матриця суміжності





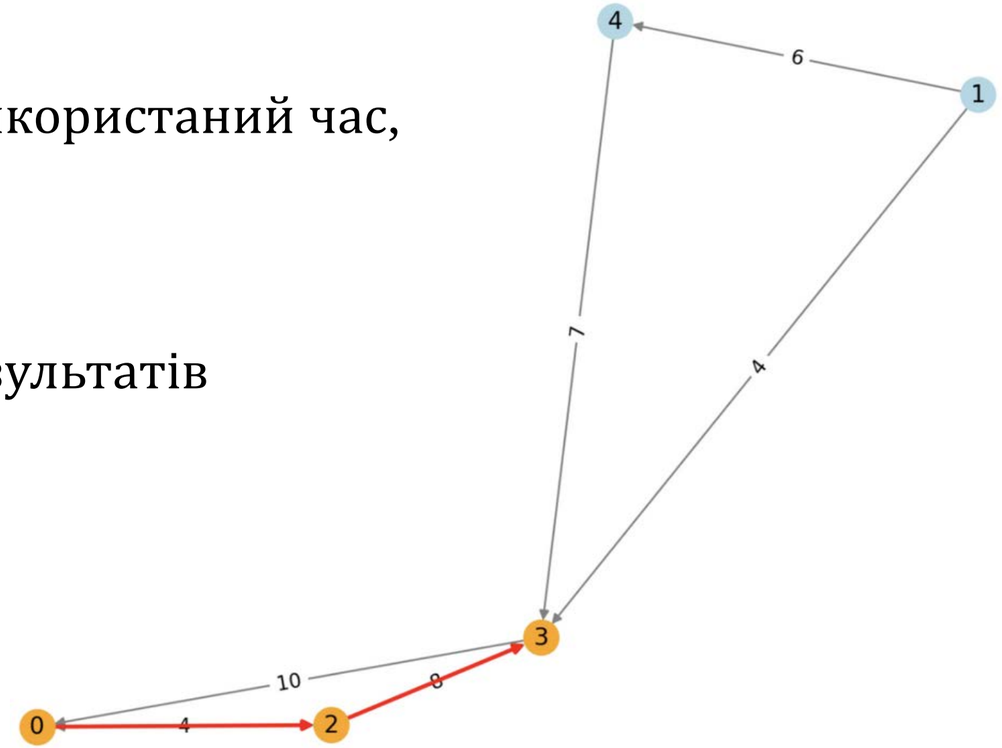
Алгоритми:

- A* з евристикою - Евклідовою відстанню
- Беллмана- Форда на списку суміжності
- Беллмана- Форда на матриці
- Флойда- Уоршала на матриці

ЗАСТОСУНОК

Вимоги до функціоналу:

- Генерація випадкових графів заданого типу, розміру, щільності
- Під час виконання алгоритму має бути заміряно використаний час, пам’ять
- Можливість демонстрації згенерованого графа, результатів
- Збереження графів та результатів у файли



ЗАСТОСУНОК

Приклад:

Параметри графа

Тип графа

Орієнтований

Щільність

high

Кількість вузлів

5

5 30

Згенерувати граф

Початкова вершина

1

Кінцева вершина

4

Запустити Беллмана-Форда

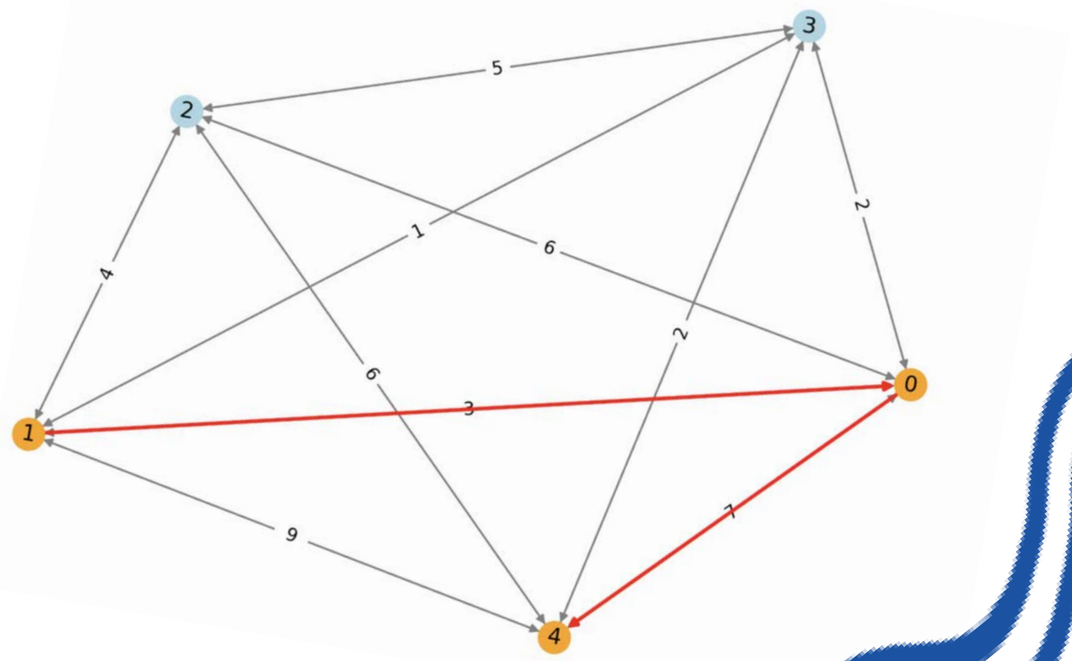
Запустити Флойда-Уоршала

Шлях: [1, 0, 4]

Довжина: 7

Час: 0.0000 сек

Пам'ять: 0.91 KB



ТЕСТУВАННЯ

Критерії оцінки ефективності:

- Застосовність до виду графа
- Час виконання
- Споживання пам’яті
- Кількість перевірених вузлів

Тестові графи:

- розмір: малі, середні, великі
- орієнтованість: так, ні
- щільність: низька, середня, висока
- наявність циклів: так, ні
- графи з координатами

ТЕСТУВАННЯ

- Для кожної тестової множини було створено від тридцяти до п’ятдесяти графів.
- Графи було збережено до файлів.
- На всіх графах з файлу запущено алгоритми.
- Результати збережено в таблицях до файлів.

	A	B	C	D	E	F	G	H
1	graph_id	algorithm	execution_time	memory_kb	restore_time	restore_memory_kb	path_length	path
2	0	bf_l	0,025584681	10,58007813	1,3959E-05	0,0625	4	[0, 35, 41, 99]
3	1	bf_l	0,044970359	10,8125	1,1391E-05	0,0546875	3	[0, 16, 99]
4	2	bf_l	0,018024988	10,125	1,3288E-05	0,0546875	3	[0, 28, 99]
5	3	bf_l	0,016088299	10,125	1,2616E-05	0,046875	2	[0, 99]
6	4	bf_l	0,028173833	10,125	1,3072E-05	0,046875	2	[0, 99]

C
execution_time
0,025584681
0,044970359
0,018024988
0,016088299
0,028173833
0,019839837
0,011464576
0,014627689
0,007726559
0,046735662
0,023868886
0,011564479
0,035514003
0,012426729
0,014310872
0,025486665
0,018592173
0,018144921
0,005235105

ПОРІВНЯННЯ

На основі зібраних з джерел та отриманих шляхом практичних перевірок даних, помітно що:

- На графах з подібною до транспортних систем будовою: евристика забезпечує колосальний відрив у часі на розріджених графах
- A* використовує менше пам’яті і обходить наближену до довжини шляху кількість вершин
- Зі збільшенням щільності A* демонструє менший відрив від інших відносно часу
- Вплив зміни щільності на ефективність Флойда- Уоршала є мінімальним
- На показники ефективності алгоритмів практично не впливає орієнтованість графів та наявність циклів
- На графах розміром від 100 вузлів ефективність алгоритму Флойда- Уоршала різко знижується

ПОРІВНЯННЯ

Робота з ребрами з від'ємними вагами	
Робота з негативними циклами	

ПОРІВНЯННЯ

Тип графа	На
Малий граф, низька та середня щільність	
Середні та великі графи, низька та середня щільність	

- Наведена таблиця вказує найефективніші алгоритми для задач з умовою пошуку шляху між двома вершинами на графах з щільністю, наближеною до транспортних систем. Зважаючи на зменшення відношення часу виконання Беллмана- Форда до A* з наближенням щільності до максимальної щільності, для середніх та великих повних і майже повних графів він буде швидшим за A*.

ПОРІВНЯННЯ

- Для задачі на пошук шляхів з одного вузла в інші стандартним вибором є алгоритм Беллмана- Форда. Однак беручи до уваги на значну різницю часу між A* та Беллмана-Форда, на середніх розріджених графах A*, запущений до кожної з кінцевих вершин, може бути ефективнішим.

- В задачах на пошук шляхів між всіма парами вершин для графів з високою щільністю доцільно використовувати алгоритм Флойда- Уоршала. Але для великих розріджених графів, залежно від кількості вершин, може бути ефективнішим запустити Беллмана-Форда з усіх вузлів, особливо, якщо його оптимізувати.

ВИСНОВКИ

Під час роботи над курсовою було:

- Досліджено основні поняття з теорії графів
- Розглянуто найпоширеніші види графів та способи їх представлення
- Оглянуто алгоритми пошуку найкоротшого шляху в зважених графах
- Досліджено процес виконання алгоритмів A*, Беллмана- Форда та Флойда-Уоршала
- Створено демонстраційний застосунок з реалізацією алгоритмів та створенням випадкових графів для їх тестування
- Зібрано таблиці зі статистикою виконання алгоритмів на графах
- Проаналізовано та порівняно ефективність алгоритмів на різних графах

Дякую за увагу