МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

КРЕМЕНЧУЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ   
ІМЕНІ МИХАЙЛА ОСТРОГРАДСЬКОГО

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ЕЛЕКТРИЧНОЇ ІНЖЕНЕРІЇ   
ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

КАФЕДРА АВТОМАТИЗАЦІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ

НАВЧАЛЬНА ДИСЦИПЛІНА  
Алгоритми та методи обчислень

ЗВІТ

З ПРАКТИЧНИХ РОБІТ

Виконав:

студент групи КН-23-1

Сидоренко А.Ю.

Кременчук 2024

**Практична робота № 6**

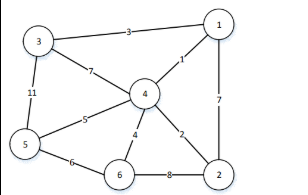
**Тема.** Графи. Найкоротші шляхи

**Мета:** набути практичних навичок розв’язання задач пошуку найкоротших шляхів у графі та оцінювання їх асимптотичної складності.

**Хід роботи**

**Завдання:**

**6. Задача з вар. 2, але за алгоритмом Белмена–Форда**



import networkx as nx //Взято с ЛБ

import matplotlib.pyplot as plt

edges = {(1,2,7),(1,3,3),(1,4,1),(2,4,2),( 2,6,8), (3,4,7), (3,5,11),(4,5,5),(4,6,4),(5,6,6)}

G = nx.Graph()

for edge in edges:

G.add\_edge(edge[0], edge[1], weight=edge[2])

pos = nx.spring\_layout(G)

nx.draw(G, pos, with\_labels=True, node\_size=1500, node\_color="skyblue", font\_size=12, font\_weight="bold", edge\_color="gray", width=2)

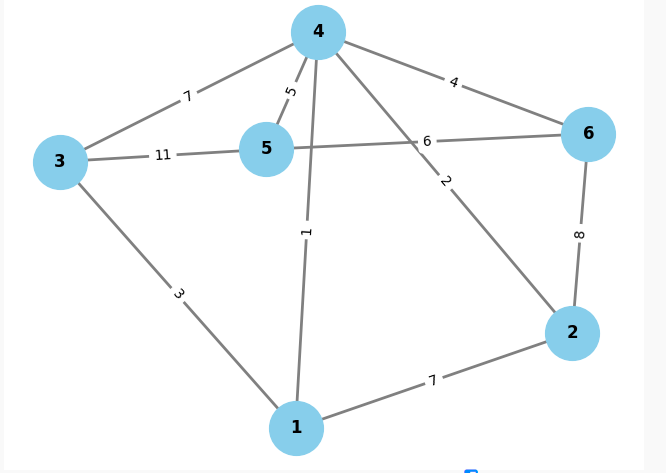
labels = nx.get\_edge\_attributes(G, 'weight')

nx.draw\_networkx\_edge\_labels(G, pos, edge\_labels=labels)

plt.show()

length = nx.single\_source\_bellman\_ford\_path\_length(G, source=1)

print(length)



Найкоротший шлях від т. 1 до всіх інших:

1: 0

2: 3

4: 1

3: 3

6: 5

5: 6

**Контрольні питання**

1. **Що таке граф і які головні складові його структури?**

Граф - це сукупність вершин і ребр. До головних складових графу выдносять вершини, ребра, ваги, напрямок, цикли і орієнтацію.

1. **Які алгоритми використовуються для пошуку найкоротших шляхів у графах?**

Алгоритм Дейкстри, Белмана-Форда, Флойда-Воршала

1. **Як працює алгоритм Дейкстри і які його особливості?**

Алгоритм Дейкстри:

* Обираємо початкову і відстані до всіх інших вершин прописуємо як нескінченність
* Обираємо вершину до якої відстань найменша від початкової
* Оновлюємо відстань обраної вершини і перевіряємо, чи можна скоротити шлях
* Позначаємо вершину як перевірену
* Повторюємо до кінця

Особливості:

Алгоритм працює лише в графах з не від'ємними вагами зв'язків, знаходить найкоротший шлях від початкової вершини до всіх інших вершин.

Якщо в графі є цикли, можливі проблеми в обчисленні коректних відстаней.

1. Що таке алгоритм Белмена–Форда і коли його варто застосовувати?

Алгоритм Беллмана-Форда - це алгоритм пошуку найкоротшого шляху від однієї вершини графа до всіх інших вершин. Цей алгоритм дозволяє працювати з графами, які містять ребра з вагами (як додатні, так і від'ємні), а також знаходить найкоротший шлях, навіть якщо граф містить цикли з від'ємною сумою ваг.

Алгоритм Беллмана-Форда варто застосовувати:

* Якщо граф містить ребра з вагами (додатніми або від'ємними).
* Коли потрібно знайти найкоротший шлях від однієї вершини до всіх інших у графі.
* Якщо потрібно виявити наявність від'ємних циклів у графі.
* Коли граф має невеликий або середній розмір

1. Як працює алгоритм Флойда–Форшала і які його переваги та недоліки?

Алгоритм Флойда–Уоршала - це алгоритм для знаходження найкоротших відстаней між усіма парами вершин у направленому зваженому графі

Переваги алгоритму:

* Здатність знаходити найкоротші шляхи між усіма парами вершин у графі.
* Працює коректно навіть з графами, які містять цикли від'ємної ваги.
* Простота реалізації.

Недоліки алгоритму:

* Часова складність алгоритму - O(V^3), що робить алгоритм менш ефективним для великих графів.
* Потребує квадратичної пам'яті для зберігання матриці відстаней, що може бути проблемою для великих графів.
* Не може коректно працювати з графами, що містять від'ємні цикли, тобто цикли, сума ваг ребер яких менша за нуль. Це може призводити до невірних результатів або зациклення алгоритму.