МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

КРЕМЕНЧУЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

ІМЕНІ МИХАЙЛА ОСТРОГРАДСЬКОГО

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ЕЛЕКТРИЧНОЇ ІНЖЕНЕРІЇ

ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра комп’ютерної інженерії та електроніки

ЗВІТ З ПРАКТИЧНИХ РОБІТ

з навчальної дисципліни

«Алгоритми та методи обчислень»

Тема «Жадібні Алгоритми»

Студентка гр. КН-23-1 ПІБ Варич А.І

Викладач к. т. н., доц. В. М. Сидоренко

Кременчук 2024

**ЗМІСТ**

[1 Завдання 3](#_Toc167124125)

[2 Контрольні запитання 6](#_Toc167124126)

# Завдання

Варіант 6

**Постановка задачі:** 1. Розв’язати задачу комівояжера для графа, заданого варіантом, використовуючи код, наведений вище. 2. Візуалізувати граф. 3. Обґрунтувати асимптотику для обох алгоритмів, неведену в табл. 1.4. Заданий зважений граф: [(1,2,2), (1,3,1), (1,4,4), (1,5,3), (2,3,2), (2,4,3), (2,5,2), (3,4,2), (3,5,2), (4,5,1)]

1. Зважений граф:

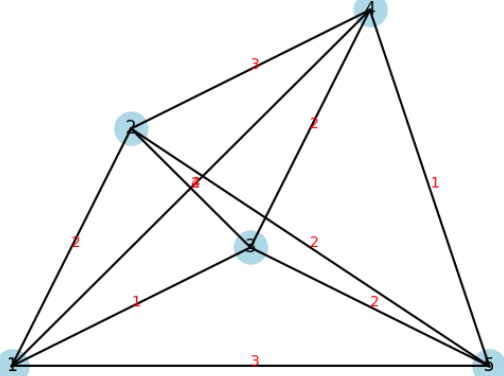


рис. 1 - Зважений граф

1. Цей граф можна представити у вигляді матриці ваг:

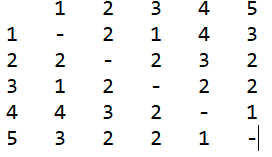


рис. 2 - Матриця ваг

1. Метод грубої сили:

Перестановки для вершин (2, 3, 4, 5) — всього 4! = 24 перестановки.

Обчислюю довжину кожного маршруту (наприклад, 1-2-3-4-5-1).

Знаходжу маршрут з мінімальною довжиною.

1. Алгоритм найближчого сусіда:

Починаю з вершини 1.

Найближча вершина до 1 — це 3 (вага 1).

З 3 найближча вершина — 2 (вага 2).

З 2 найближча вершина — 5 (вага 2).

З 5 найближча вершина — 4 (вага 1).

Повертаюся до 1 (вага 4).

Маршрут: 1 → 3 → 2 → 5 → 4 → 1.

1. Отже, візуалізація графа:

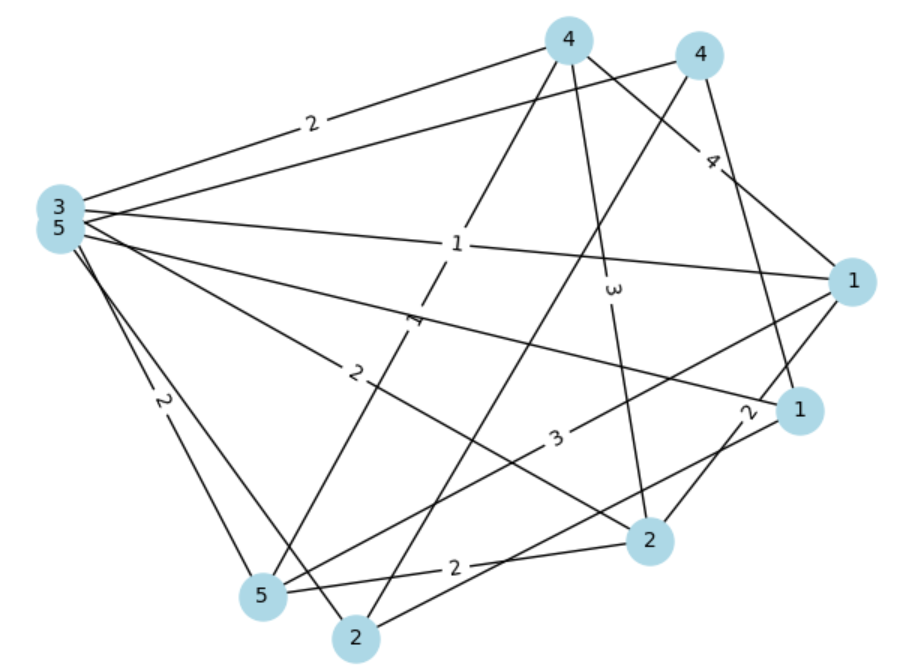


рис. 3 - Візуалізація графа

1. Обґрунтування асимптотичної складності:

Метод грубої сили (О(𝑛!))

Кількість можливих маршрутів для n вершин становить (n-1)!, оскільки необхідно переглянути всі перестановки вершин.

Перебір всіх перестановок має експоненційну складність О(𝑛!), що робить цей метод неефективним для великих n.

Алгоритм найближчого сусіда (О(𝑛^2⋅log𝑛))

На кожному кроці потрібно знайти найближчу невідвідану вершину. Це можна зробити за О(n) для кожної вершини.

Так як у нас n вершин, повна складність пошуку буде О(𝑛^2).

Додатково, для сортування вершин за відстанню використовується логарифмічна складність О(𝑛log𝑛).

Загальна асимптотична складність — О(𝑛^2⋅log𝑛).

Отже, алгоритм грубої сили є значно менш ефективним порівняно з алгоритмом найближчого сусіда, особливо для великих графів.

# Контрольні запитання

1. Що таке жадібний алгоритм?

Жадібний алгоритм - це алгоритм для оптимізації, який обирає найкращий варіант на кожному кроці в надії, що в цілому це приведе до оптимального рішення.

1. Які головні принципи роботи жадібних алгоритмів?

На кожному кроці алгоритм обирає найбільш оптимальний варіант.

Він не повертається назад, щоб переглянути або змінити своє рішення після прийняття кожного кроку.

Жадібні алгоритми не завжди гарантують найкращий результат, але часто працюють ефективно.

1. Яка головна відмінність між жадібними алгоритмами та динамічним програмуванням?

В основі жадібних алгоритмів лежить принцип локальної оптимізації, тоді як динамічне програмування використовує принцип глобальної оптимізації.

1. Наведіть приклади задач, які можна розв’язати за допомогою жадібних алгоритмів.

Задача охолодження квартири (вибір оптимальних пристроїв для кожної кімнати).

Задача організації рюкзака (вибір предметів для рюкзака з максимальною вартістю, яка вміщається).

Задача обрізання кропиви (вибір оптимального моменту та місця для обрізки кропиви).

1. Які можуть бути обмеження у використанні жадібних алгоритмів для розв'язання екстремальних задач?

Жадібні алгоритми не завжди гарантують оптимальне рішення.

Можуть бути ситуації, коли локальна оптимізація не призводить до глобальної оптимізації.

1. Чому жадібні алгоритми часто використовуються для наближеного розв’язання екстремальних задач?

Вони прості в реалізації та мають низьку обчислювальну складність.

Жадібні алгоритми можуть дати швидкий результат, який, хоч і не є оптимальним, може бути достатнім для багатьох задач.