

МЕТОДИЧНІ РЕКОМЕНДАЦІЇ:

**до виконання індивідуального завдання
«Оптимізація маршрутизації у комп'ютерній
мережі» з дисципліни «Моделювання систем»**

Запоріжжя 2024



УДК 004.7(072)
М54

Рекомендовано Науково-методичною радою
ТОВ «ТЕХНІЧНИЙ УНІВЕРСИТЕТ
«МЕТІНВЕСТ ПОЛІТЕХНІКА»
(протокол № 1 від 27.09.2024 р.)

Укладач

Кайдан Н.В., канд. фіз-мат. наук, доцент.

- М54 Методичні рекомендації до виконання індивідуального завдання «Оптимізація маршрутизації у комп'ютерній мережі» з дисципліни «Моделювання систем» (для студентів комп'ютерних спеціальностей усіх форм навчання першого (бакалаврського) рівня вищої освіти) / уклад. Н.В. Кайдан. Запоріжжя : ТОВ «ТЕХНІЧНИЙ УНІВЕРСИТЕТ «МЕТІНВЕСТ ПОЛІТЕХНІКА». 2024. 28 с.

Методичні вказівки містять відомості щодо формування кейсу задач та критеріїв оцінювання індивідуального завдання, приклади їх розв'язання з наведенням основних формул, етапів розв'язання та геометричної ілюстрації в обсязі, необхідному для виконання індивідуального завдання. Матеріал навчального посібника має на меті підвищити якість виконання роботи, виробити навички розв'язання задач прикладного спрямування, в тому числі з використанням системи комп'ютерної математики Maple. Рекомендовано для студентів комп'ютерних спеціальностей усіх форм навчання першого (бакалаврського) рівня вищої освіти.

УДК 004.7(072)

© ТОВ «ТЕХНІЧНИЙ УНІВЕРСИТЕТ МЕТІНВЕСТ ПОЛІТЕХНІКА», 2024



ЗМІСТ

ВСТУП	4
1 ІНДИВІДУАЛЬНЕ ЗАВДАННЯ	5
1.1 Відомості щодо формування кейсу задач та критеріїв оцінювання індивідуального завдання	5
1.2 Кейс задач індивідуального завдання «Оптимізація маршрутизації у комп'ютерній мережі»	6
Завдання 1. Оптимізація мережевої інфраструктури.....	6
Завдання 2. Оптимізація мережевого підключення.....	8
Завдання 3. Оптимізація маршрутизації дорожньої інфраструктури	11
Завдання 4. Планування передачі даних у мережі.....	14
2 РОЗВ'ЯЗАННЯ ТИПОВОГО ВАРІАНТУ ІНДИВІДУАЛЬНОГО ЗАВДАННЯ 1	17
2.1. Приклад розв'язання завдання 1.	17
Завдання 1.	17
Приклад розв'язання завдання 1 з використанням можливостей Maple.	18
2.2. Приклад розв'язання завдання 2.	19
Завдання 2.	19
2.3. Приклад розв'язання завдання 3.	20
Завдання 3.	20
Приклад розв'язання завдання 2-3 з використанням можливостей Maple.	21
2.4. Приклад розв'язання завдання 4.	23
Завдання 4.	23
Приклад розв'язання завдання 4 з використанням можливостей Maple.	25
Перелік використаних джерел	26



ВСТУП

Освітній компонент «Моделювання систем» є ключовим курсом циклу математичної та природничо-наукової підготовки, спрямованим на розвиток у студентів комп'ютерних спеціальностей навичок застосування теоретичних знань і методів для розв'язання математично формалізованих задач, аналізу та моделювання пристроїв, процесів і явищ, а також пошуку оптимальних рішень. Цей курс формує базу для вирішення складних професійних задач та практичних проблем у майбутній діяльності. Його особливістю є прикладна орієнтація математичної підготовки із залученням комп'ютерних технологій та спеціалізованих математичних програмних пакетів, що забезпечує глибоке розуміння та критичне осмислення теорій, методів і концепцій, необхідних у професійній сфері. У процесі вивчення курсу студенти набувають вмінь використовувати сучасні підходи до створення математичних моделей, прогнозування поведінки систем, проведення чисельних експериментів та оптимізації параметрів системи.

У межах вивчення дисципліни «Моделювання систем» студентам пропонується виконати індивідуальне завдання на тему «Оптимізація маршрутизації в комп'ютерній мережі». Ці методичні рекомендації надають інформацію щодо складання кейсу задач, критеріїв оцінювання виконаного завдання, а також містять приклади їх розв'язання. У посібнику представлені основні формули, етапи вирішення задач, а також геометрична ілюстрація результатів, що забезпечує необхідний обсяг знань для успішного виконання індивідуальних завдань. Особливу увагу приділено використанню системи комп'ютерної математики Maple для моделювання та розв'язання задач.



1 ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

1.1 Відомості щодо формування кейсу задач та критеріїв оцінювання індивідуального завдання

Під час виконання індивідуального завдання кожному студенту надається кейс задач, який є однаковим за обсягом і змістом відповідно до варіанту, що визначається його порядковим номером у списку академічної групи та затверджується викладачем у межах відповідного освітнього компонента та змістового модуля. Студентам рекомендується використовувати систему комп'ютерної математики Maple як для спрощення розрахунків чи побудови графіків, так і для розробки модулів автоматизованого розрахунку формалізованих задач прикладного характеру.

Індивідуальне завдання виконується студентом самостійно у зручний час, дотримуючись встановлених термінів подачі, зазначених у розділі «Розподіл балів за контрольними точками та графік їх виконання». Готове розв'язання завантажується у відповідний розділ платформи Moodle у форматі файлу .docx, .pdf, .jpg, .png або .txt. У разі наявності розробленого розрахункового модуля в системі комп'ютерної математики Maple додатково завантажується файл у форматі .mw.

Максимальна кількість балів за кожне завдання визначається залежно від обґрунтованості ходу розв'язання, рівня формалізації задачі, правильності отриманого результату та його аналізу, а також враховує необхідність геометричної інтерпретації чи використання можливостей системи комп'ютерної математики Maple.

Використання штучного інтелекту (ШІ) дозволяється, оскільки якість отриманих відповідей значною мірою залежить від ретельно сформульованого запиту та уточнюючих питань. Проте, якщо відповідь, отримана за допомогою ШІ, містить значні помилки, є неповною, не відповідає встановленим вимогам до оформлення, термінології або іншим



критеріям завдання, оцінка за виконання буде знижена.

Індивідуальне завдання перевіряється протягом тижня після закінчення терміну подачі. У разі виявлення помилок або неповного виконання завдання студент має право на доопрацювання, яке дозволяється завершити до передостаннього тижня навчального семестру.

1.2 Кейс задач індивідуального завдання «Оптимізація маршрутизації у комп'ютерній мережі»

Завдання 1. Оптимізація мережевої інфраструктури.

Компанія займається розробкою нової комп'ютерної мережі, що включає кілька серверів і маршрутизаторів. Мережа має бути надійною і складатися з вузлів (серверів) та зв'язків (кабелів), які їх з'єднують. Вам задано граф $G = (V, E)$, де:

V — множина вершин, що представляє сервери (з номерами від 1 до n),

E — множина ребер, що моделює фізичні з'єднання між серверами.

Ваше завдання:

1. Побудувати діаграму графа G , що моделює фізичну топологію мережі.
2. Побудувати:
 - Матрицю суміжності, яка показує, які сервери з'єднані між собою.
 - Список інцидентності для кожної вершини, що демонструє, які кабелі підключені до кожного сервера.
3. Обчислити:
 - Степені всіх вершин графа G (кількість підключень кожного сервера).

- Кількість компонент зв'язності графа G (окремих підмереж, якщо такі є).


4. Зробити висновок про те, чи всі сервери підключені до єдиної мережі. Якщо ні, запропонувати додаткові з'єднання, щоб забезпечити повну зв'язність.

Застосування в реальності:

Ця задача моделює завдання з планування фізичної комп'ютерної мережі, оптимізації її структури, а також оцінки надійності мережі на основі її компонент зв'язності.

Варіанти:

1. $V = \{1,2,3,4,5,6,7,8\}$, $E = \{(1,7), (3,5), (4,2), (4,6), (6,7), (7,8), (8,2)\}$;
2. $V = \{1,2,3,4,5,6,7,8\}$, $E = \{(1,5), (2,5), (3,1), (4,7), (5,3), (7,8), (8,4)\}$;
3. $V = \{1,2,3,4,5,6,7,8\}$, $E = \{(1,3), (2,4), (3,5), (4,6), (6,8), (7,1), (8,2)\}$;
4. $V = \{1,2,3,4,5,6,7,8\}$, $E = \{(1,4), (2,4), (3,7), (4,3), (4,8), (5,6), (8,1)\}$;
5. $V = \{1,2,3,4,5,6,7,8,9\}$, $E = \{(1,6), (1,9), (2,5), (2,7), (3,4), (6,7), (7,9), (8,5)\}$;
6. $V = \{1,2,3,4,5,6,7,8,9\}$, $E = \{(1,2), (1,8), (2,3), (4,6), (4,9), (6,8), (7,9), (8,9)\}$;
7. $V = \{1,2,3,4,5,6,7,8,9\}$, $E = \{(1,5), (3,8), (4,5), (5,3), (6,7), (6,8), (7,8), (8,4)\}$;
8. $V = \{1,2,3,4,5,6,7,8,9\}$, $E = \{(1,7), (2,8), (3,5), (3,7), (4,2), (5,7), (6,9), (8,4)\}$.
9. $V = \{1,2,3,4,5,6,7,8\}$, $E = \{(1,7), (3,5), (4,2), (4,6), (6,7), (7,8), (8,2)\}$;
10. $V = \{1,2,3,4,5,6,7,8\}$, $E = \{(1,5), (2,5), (3,1), (4,7), (5,3), (7,8), (8,4)\}$;
11. $V = \{1,2,3,4,5,6,7,8\}$, $E = \{(1,3), (2,4), (3,5), (4,6), (6,8), (7,1), (8,2)\}$;
12. $V = \{1,2,3,4,5,6,7,8\}$, $E = \{(1,4), (2,4), (3,7), (4,3), (4,8), (5,6), (8,1)\}$;
13. $V = \{1,2,3,4,5,6,7,8,9\}$, $E = \{(1,6), (1,9), (2,5), (2,7), (3,4), (6,7), (7,9), (8,5)\}$;
14. $V = \{1,2,3,4,5,6,7,8,9\}$, $E = \{(1,2), (1,8), (2,3), (4,6), (4,9), (6,8), (7,9), (8,9)\}$;
15. $V = \{1,2,3,4,5,6,7,8,9\}$, $E = \{(1,5), (3,8), (4,5), (5,3), (6,7), (6,8), (7,8), (8,4)\}$;
16. $V = \{1,2,3,4,5,6,7,8,9\}$, $E = \{(1,7), (2,8), (3,5), (3,7), (4,2), (5,7), (6,9), (8,4)\}$.
17. $V = \{1,2,3,4,5,6,7,8\}$, $E = \{(1,5), (2,5), (3,1), (4,7), (5,3), (7,8), (8,4)\}$;

- 
18. $V = \{1,2,3,4,5,6,7,8\}$, $E = \{(1,4), (2,4), (3,7), (4,3), (4,8), (5,6), (8,1)\}$;
19. $V = \{1,2,3,4,5,6,7,8,9\}$, $E = \{(1,2), (1,8), (2,3), (4,6), (4,9), (6,8), (7,9), (8,9)\}$;
20. $V = \{1,2,3,4,5,6,7,8,9\}$, $E = \{(1,7), (2,8), (3,5), (3,7), (4,2), (5,7), (6,9), (8,4)\}$.

Завдання 2. Оптимізація мережевого підключення

Компанія займається розробкою оптимальної комп'ютерної мережі для підключення 10 серверів. З'єднання між серверами має певну вартість, яка залежить від довжини кабелю, його типу та інших факторів. Інженерам потрібно побудувати мережу таким чином, щоб усі сервери були з'єднані між собою, а загальна вартість прокладки кабелів була мінімальною. Мережу описано у вигляді зваженого графа $G = (V, E)$, де:

$V = \{1,2,3,4,5,6,7,8,9,10\}$, — множина серверів (вершин графа),

E — множина з'єднань (ребер графа), представлена матрицею ваг, де кожен елемент w_{ij} задає вартість з'єднання між сервером i та сервером j . Якщо з'єднання між двома серверами відсутнє, то $w_{ij} = \infty$.

Ваше завдання:

1. Використовуючи алгоритм Краскала, побудувати остове дерево мінімальної вартості для мережі. Це дерево має з'єднати всі сервери з мінімальною сумарною вартістю.
2. Визначити:
 - Які сервери та які з'єднання (кабелі) потрібно використовувати.
 - Загальну вартість побудови такої мережі.
3. Подати результати у вигляді:
 - Переліку обраних з'єднань із зазначенням їх вартості.
 - Візуалізації графа остова мінімальної ваги.

Застосування в реальності:

Ця задача моделює процес оптимізації витрат на прокладання фізичної мережі (кабелів або інших засобів зв'язку) між серверами чи іншими пристроями. Алгоритм Краскала допомагає знайти оптимальне рішення в ситуаціях, коли потрібно з'єднати вузли з мінімальними витратами.

Варіанти:

$$1. E = \begin{pmatrix} - & - & 5 & 6 & - & 7 & 6 & 8 & - & 7 \\ - & - & 3 & - & 2 & 3 & - & 1 & 3 & 6 \\ 5 & 3 & - & 4 & 4 & - & - & 2 & 1 & - \\ 6 & - & 4 & - & 4 & 1 & 2 & - & - & 6 \\ - & 2 & 4 & 4 & - & - & 1 & 3 & 2 & - \\ 7 & 3 & - & 1 & - & - & 2 & - & 5 & 8 \\ 6 & - & - & 2 & 1 & 2 & - & 3 & 3 & - \\ 8 & 1 & 2 & - & 3 & - & 3 & - & - & 5 \\ - & 3 & 1 & - & 2 & 5 & 3 & - & - & 5 \\ 7 & 6 & - & 6 & - & 8 & - & 5 & 5 & - \end{pmatrix}$$

$$2. E = \begin{pmatrix} - & 7 & 6 & - & 6 & - & 5 & - & 7 & 6 \\ 7 & - & 1 & 4 & - & 3 & 2 & 1 & - & - \\ 6 & 1 & - & - & 2 & - & - & 3 & 3 & 2 \\ - & 4 & - & - & 6 & 1 & 4 & - & 4 & 5 \\ 6 & - & 2 & 6 & - & 5 & 1 & - & 2 & - \\ - & 3 & - & 1 & 5 & - & 6 & 1 & - & 5 \\ 5 & 2 & - & 4 & 1 & 6 & - & - & 3 & - \\ - & 1 & 3 & - & - & 1 & - & - & 5 & 3 \\ 7 & - & 3 & 4 & 2 & - & 3 & 5 & - & 3 \\ 6 & - & 2 & 5 & - & 5 & - & 3 & 3 & - \end{pmatrix}$$

$$3. E = \begin{pmatrix} - & - & 8 & - & 5 & 8 & 6 & - & 6 & 4 \\ - & - & 2 & 3 & - & 4 & - & 2 & 1 & 4 \\ 8 & 2 & - & - & 5 & - & 3 & 4 & - & 4 \\ - & 3 & - & - & 3 & 4 & 2 & 5 & 1 & - \\ 5 & - & 5 & 3 & - & 6 & - & 1 & - & 6 \\ 8 & 4 & - & 4 & 6 & - & - & - & 3 & 1 \\ 6 & - & 3 & 2 & - & - & - & 6 & 2 & 2 \\ - & 2 & 4 & 5 & 1 & - & 6 & - & 5 & - \\ 6 & 1 & - & 1 & - & 3 & 2 & 5 & - & - \\ 4 & 4 & 4 & - & 6 & 1 & 2 & - & - & - \end{pmatrix}$$

$$4. E = \begin{pmatrix} - & - & 8 & 6 & 7 & 8 & - & 5 & - & 7 \\ - & - & - & 2 & 3 & - & 3 & 5 & 2 & 2 \\ 8 & - & - & - & 3 & 3 & 3 & 1 & 4 & - \\ 6 & 2 & - & - & - & 4 & 4 & - & 3 & 3 \\ 7 & 3 & 3 & - & - & 3 & - & 1 & 2 & - \\ 8 & - & 3 & 4 & 3 & - & 2 & 3 & - & - \\ - & 3 & 3 & 4 & - & 2 & - & 4 & 1 & 1 \\ 5 & 5 & 1 & - & 1 & 3 & 4 & - & - & - \\ - & 2 & 4 & 3 & 2 & - & 1 & - & - & 1 \\ 7 & 2 & - & 3 & - & - & 1 & - & 1 & - \end{pmatrix}$$

$$5. E = \begin{pmatrix} - & 8 & - & - & 6 & 6 & - & 5 & 7 & 7 \\ 8 & - & 6 & 4 & - & 2 & 2 & - & 4 & - \\ - & 6 & - & 2 & 3 & - & 4 & 4 & - & 5 \\ - & 4 & 2 & - & 1 & - & 1 & 5 & 2 & - \\ 6 & - & 3 & 1 & - & - & 2 & 5 & - & 4 \\ 6 & 2 & - & - & - & - & 2 & 1 & 1 & 3 \\ - & 2 & 4 & 1 & 2 & 2 & - & - & 4 & - \\ 5 & - & 4 & 5 & 5 & 1 & - & - & - & 1 \\ 7 & 4 & - & 2 & - & 1 & 4 & - & - & 4 \\ 7 & - & 5 & - & 4 & - & 3 & 1 & 4 & - \end{pmatrix}$$

$$6. E = \begin{pmatrix} - & 4 & 5 & - & 6 & 8 & 7 & 5 & - & - \\ 4 & - & 1 & 6 & 2 & - & - & - & 4 & 4 \\ 5 & 1 & - & 5 & - & 1 & 2 & 6 & - & 5 \\ - & 6 & 5 & - & - & 6 & 5 & - & 3 & 2 \\ 6 & 2 & - & - & - & 1 & 5 & 3 & 3 & - \\ 8 & - & 1 & 6 & 1 & - & - & 4 & - & 4 \\ 7 & - & 2 & 5 & 5 & - & - & 6 & 2 & - \\ 5 & - & 6 & - & 3 & 4 & 6 & - & 7 & 1 \\ - & 4 & - & 3 & 3 & - & 2 & 7 & - & 3 \\ - & 4 & 5 & 2 & - & 4 & - & 1 & 3 & - \end{pmatrix}$$

$$7. E = \begin{pmatrix} - & - & 5 & 6 & - & 7 & 5 & - & 8 & 7 \\ - & - & - & 2 & 3 & - & 2 & 3 & 2 & 3 \\ 5 & - & - & 3 & 7 & 4 & - & 5 & - & 4 \\ 6 & 2 & 3 & - & - & - & 1 & 3 & 4 & 4 \\ - & 3 & 7 & - & - & 3 & 2 & 5 & 2 & - \\ 7 & - & 4 & - & 3 & - & 2 & 3 & 1 & - \\ 5 & 2 & - & - & 2 & 2 & - & - & 1 & 1 \\ - & 3 & 5 & 1 & 5 & 3 & - & - & - & - \\ 8 & 2 & - & 3 & 2 & 1 & 1 & - & - & 1 \\ 7 & 3 & 4 & 4 & - & - & 1 & - & 1 & - \end{pmatrix}$$

$$8. E = \begin{pmatrix} - & - & 7 & - & 5 & 6 & 7 & 5 & 6 & - \\ - & - & 1 & 4 & 4 & - & 3 & 1 & - & 2 \\ 7 & 1 & - & 2 & - & 3 & 2 & - & 2 & - \\ - & 4 & 2 & - & 3 & 3 & 4 & - & - & 1 \\ 5 & 4 & - & 3 & - & 3 & - & 2 & - & 1 \\ 6 & - & 3 & 3 & 3 & - & - & - & 3 & 4 \\ 7 & 3 & 2 & 4 & - & - & - & 1 & 4 & - \\ 5 & 1 & - & - & 2 & - & 1 & - & 3 & 3 \\ 6 & - & 2 & - & - & 3 & 4 & 3 & - & 5 \\ - & 2 & - & 1 & 1 & 4 & - & 3 & 5 & - \end{pmatrix}$$



Завдання 3. Оптимізація маршрутизації дорожньої інфраструктури

Розроблено проект системи інтелектуального управління дорожньою інфраструктурою для 10 ключових пунктів контролю (камер спостереження, світлофорів або датчиків руху). Між цими пунктами необхідно забезпечити зв'язок, щоб інформація могла надходити в єдиний центр обробки даних. Вартість прокладки кабелів або встановлення бездротового з'єднання залежить від відстані між пунктами та технічних умов.

Інфраструктура описується у вигляді зваженого графа $G = (V, E)$, де: $V = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$, — множина пунктів контролю (вершин графа), E — множина можливих з'єднань між пунктами (ребер графа), заданих матрицею ваг, де елемент w_{ij} описує вартість встановлення з'єднання між пунктами i та j . Якщо з'єднання між двома пунктами неможливе, то $w_{ij} = \infty$.

Ваше завдання:

1. Використовуючи алгоритм Прима, побудувати остове дерево мінімальної вартості для інфраструктури. Це дерево має з'єднати всі пункти контролю з мінімальною сумарною вартістю.
2. Визначити:
 - Які пункти контролю потрібно безпосередньо з'єднати.
 - Загальну вартість створення зв'язку.
3. Подати результати у вигляді:
 - Списку обраних з'єднань із зазначенням їх вартості.
 - Графічної візуалізації остова мінімальної ваги.

Застосування в реальності:

Ця задача моделює оптимізацію витрат на створення системи інтелектуального управління дорожньою інфраструктурою. Вона дозволяє знайти найкраще рішення для з'єднання пунктів контролю з мінімальними витратами, що важливо для муніципалітетів чи компаній, які впроваджують системи «розумного міста».

Варіанти:

$$1. E = \begin{pmatrix} - & - & 5 & 6 & - & 7 & 6 & 8 & - & 7 \\ - & - & 3 & - & 2 & 3 & - & 1 & 3 & 6 \\ 5 & 3 & - & 4 & 4 & - & - & 2 & 1 & - \\ 6 & - & 4 & - & 4 & 1 & 2 & - & - & 6 \\ - & 2 & 4 & 4 & - & - & 1 & 3 & 2 & - \\ 7 & 3 & - & 1 & - & - & 2 & - & 5 & 8 \\ 6 & - & - & 2 & 1 & 2 & - & 3 & 3 & - \\ 8 & 1 & 2 & - & 3 & - & 3 & - & - & 5 \\ - & 3 & 1 & - & 2 & 5 & 3 & - & - & 5 \\ 7 & 6 & - & 6 & - & 8 & - & 5 & 5 & - \end{pmatrix}$$

$$2. E = \begin{pmatrix} - & 7 & 6 & - & 6 & - & 5 & - & 7 & 6 \\ 7 & - & 1 & 4 & - & 3 & 2 & 1 & - & - \\ 6 & 1 & - & - & 2 & - & - & 3 & 3 & 2 \\ - & 4 & - & - & 6 & 1 & 4 & - & 4 & 5 \\ 6 & - & 2 & 6 & - & 5 & 1 & - & 2 & - \\ - & 3 & - & 1 & 5 & - & 6 & 1 & - & 5 \\ 5 & 2 & - & 4 & 1 & 6 & - & - & 3 & - \\ - & 1 & 3 & - & - & 1 & - & - & 5 & 3 \\ 7 & - & 3 & 4 & 2 & - & 3 & 5 & - & 3 \\ 6 & - & 2 & 5 & - & 5 & - & 3 & 3 & - \end{pmatrix}$$

$$3. E = \begin{pmatrix} - & - & 8 & - & 5 & 8 & 6 & - & 6 & 4 \\ - & - & 2 & 3 & - & 4 & - & 2 & 1 & 4 \\ 8 & 2 & - & - & 5 & - & 3 & 4 & - & 4 \\ - & 3 & - & - & 3 & 4 & 2 & 5 & 1 & - \\ 5 & - & 5 & 3 & - & 6 & - & 1 & - & 6 \\ 8 & 4 & - & 4 & 6 & - & - & - & 3 & 1 \\ 6 & - & 3 & 2 & - & - & - & 6 & 2 & 2 \\ - & 2 & 4 & 5 & 1 & - & 6 & - & 5 & - \\ 6 & 1 & - & 1 & - & 3 & 2 & 5 & - & - \\ 4 & 4 & 4 & - & 6 & 1 & 2 & - & - & - \end{pmatrix}$$

$$4. E = \begin{pmatrix} - & - & 8 & 6 & 7 & 8 & - & 5 & - & 7 \\ - & - & - & 2 & 3 & - & 3 & 5 & 2 & 2 \\ 8 & - & - & - & 3 & 3 & 3 & 1 & 4 & - \\ 6 & 2 & - & - & - & 4 & 4 & - & 3 & 3 \\ 7 & 3 & 3 & - & - & 3 & - & 1 & 2 & - \\ 8 & - & 3 & 4 & 3 & - & 2 & 3 & - & - \\ - & 3 & 3 & 4 & - & 2 & - & 4 & 1 & 1 \\ 5 & 5 & 1 & - & 1 & 3 & 4 & - & - & - \\ - & 2 & 4 & 3 & 2 & - & 1 & - & - & 1 \\ 7 & 2 & - & 3 & - & - & 1 & - & 1 & - \end{pmatrix}$$

$$5. E = \begin{pmatrix} - & 8 & - & - & 6 & 6 & - & 5 & 7 & 7 \\ 8 & - & 6 & 4 & - & 2 & 2 & - & 4 & - \\ - & 6 & - & 2 & 3 & - & 4 & 4 & - & 5 \\ - & 4 & 2 & - & 1 & - & 1 & 5 & 2 & - \\ 6 & - & 3 & 1 & - & - & 2 & 5 & - & 4 \\ 6 & 2 & - & - & - & - & 2 & 1 & 1 & 3 \\ - & 2 & 4 & 1 & 2 & 2 & - & - & 4 & - \\ 5 & - & 4 & 5 & 5 & 1 & - & - & - & 1 \\ 7 & 4 & - & 2 & - & 1 & 4 & - & - & 4 \\ 7 & - & 5 & - & 4 & - & 3 & 1 & 4 & - \end{pmatrix}$$

$$6. E = \begin{pmatrix} - & 4 & 5 & - & 6 & 8 & 7 & 5 & - & - \\ 4 & - & 1 & 6 & 2 & - & - & - & 4 & 4 \\ 5 & 1 & - & 5 & - & 1 & 2 & 6 & - & 5 \\ - & 6 & 5 & - & - & 6 & 5 & - & 3 & 2 \\ 6 & 2 & - & - & - & 1 & 5 & 3 & 3 & - \\ 8 & - & 1 & 6 & 1 & - & - & 4 & - & 4 \\ 7 & - & 2 & 5 & 5 & - & - & 6 & 2 & - \\ 5 & - & 6 & - & 3 & 4 & 6 & - & 7 & 1 \\ - & 4 & - & 3 & 3 & - & 2 & 7 & - & 3 \\ - & 4 & 5 & 2 & - & 4 & - & 1 & 3 & - \end{pmatrix}$$

$$7. E = \begin{pmatrix} - & - & 5 & 6 & - & 7 & 5 & - & 8 & 7 \\ - & - & - & 2 & 3 & - & 2 & 3 & 2 & 3 \\ 5 & - & - & 3 & 7 & 4 & - & 5 & - & 4 \\ 6 & 2 & 3 & - & - & - & 1 & 3 & 4 & 4 \\ - & 3 & 7 & - & - & 3 & 2 & 5 & 2 & - \\ 7 & - & 4 & - & 3 & - & 2 & 3 & 1 & - \\ 5 & 2 & - & - & 2 & 2 & - & - & 1 & 1 \\ - & 3 & 5 & 1 & 5 & 3 & - & - & - & - \\ 8 & 2 & - & 3 & 2 & 1 & 1 & - & - & 1 \\ 7 & 3 & 4 & 4 & - & - & 1 & - & 1 & - \end{pmatrix}$$

$$8. E = \begin{pmatrix} - & - & 7 & - & 5 & 6 & 7 & 5 & 6 & - \\ - & - & 1 & 4 & 4 & - & 3 & 1 & - & 2 \\ 7 & 1 & - & 2 & - & 3 & 2 & - & 2 & - \\ - & 4 & 2 & - & 3 & 3 & 4 & - & - & 1 \\ 5 & 4 & - & 3 & - & 3 & - & 2 & - & 1 \\ 6 & - & 3 & 3 & 3 & - & - & - & 3 & 4 \\ 7 & 3 & 2 & 4 & - & - & - & 1 & 4 & - \\ 5 & 1 & - & - & 2 & - & 1 & - & 3 & 3 \\ 6 & - & 2 & - & - & 3 & 4 & 3 & - & 5 \\ - & 2 & - & 1 & 1 & 4 & - & 3 & 5 & - \end{pmatrix}$$

Завдання 4. Планування передачі даних у мережі

Уявіть, що компанія займається проєктуванням мережевої топології для передавання даних між 10 ключовими вузлами (комутаторами, маршрутизаторами чи пристроями зв'язку). Одним із завдань є визначення оптимальних маршрутів передавання даних із головного центру (вузол 1) до всіх інших вузлів.

Мережу описано у вигляді зваженого графа $G = (V, E)$, де:

$V = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$, — множина вузлів,

E — множина можливих з'єднань між вузлами (ребер графа), представлена матрицею ваг. Елемент w_{ij} описує вартість (час затримки, втрати пакета, або іншу метрику) передавання даних між вузлами i та j . Якщо з'єднання між двома вузлами відсутнє, то $w_{ij} = \infty$.

Ваше завдання:

1. Використовуючи алгоритм Форда-Беллмана, знайти найкоротші відстані (мінімальні витрати) передавання даних із вузла 1 до всіх інших вузлів графа G .
2. Побудувати остове дерево найкоротших шляхів, яке з'єднує вузол 1 з усіма іншими вузлами.
3. Визначити:
 - Найкоротші відстані від вузла 1 до кожного іншого вузла.
 - Шляхи передавання даних для кожного вузла.
4. Представити результати у вигляді:
 - Списку найкоротших шляхів і відповідних відстаней.
 - Графічної візуалізації остова найкоротших шляхів.

Застосування в реальності:

Ця задача моделює планування маршрутизації в локальних мережах



(LAN) або великих мережах (WAN). Зокрема:

- Оптимізація потоків даних у телекомунікаційних мережах.
- Побудова маршрутизаторів, які мінімізують затримки або втрати пакета.
- Розробка алгоритмів маршрутизації для протоколів типу OSPF, що використовують графові алгоритми для пошуку найкоротших шляхів.

Варіанти:

$$1. E = \begin{pmatrix} - & - & 8 & 11 & 7 & 8 & - & 9 & - & 11 \\ - & - & - & 2 & 3 & - & 3 & 5 & 2 & -1 \\ 8 & - & - & - & 3 & - & 3 & 1 & 4 & - \\ 11 & 2 & - & - & - & 4 & 4 & - & 3 & 3 \\ 7 & 3 & 3 & - & - & 3 & - & 1 & 2 & - \\ 8 & - & -2 & 4 & 3 & - & 2 & 3 & - & - \\ - & 3 & 3 & 4 & - & 2 & - & 4 & 1 & 1 \\ 9 & 5 & 1 & - & 1 & 3 & 4 & - & - & - \\ - & 2 & 4 & 3 & 2 & - & 1 & - & - & 1 \\ 11 & - & - & 3 & - & - & 1 & - & 1 & - \end{pmatrix}$$

$$2. E = \begin{pmatrix} - & 10 & 10 & - & 6 & 9 & 11 & 5 & - & - \\ 10 & - & 1 & 6 & 2 & - & - & - & 4 & 4 \\ 10 & 1 & - & 5 & - & 1 & 2 & 6 & - & 5 \\ - & 6 & 5 & - & - & 6 & 5 & - & 3 & 2 \\ 6 & 2 & - & - & - & 1 & 5 & -2 & 3 & - \\ 9 & - & 1 & 6 & 1 & - & - & 4 & - & 4 \\ 11 & - & 2 & 5 & 5 & - & - & 6 & - & - \\ 5 & - & 6 & - & - & 4 & 6 & - & 7 & - \\ - & 4 & - & 3 & 3 & - & -1 & 7 & - & 3 \\ - & 4 & 5 & 2 & - & 4 & - & - & 3 & - \end{pmatrix}$$

$$3. E = \begin{pmatrix} - & - & 9 & - & 8 & 9 & 9 & 7 & 8 & - \\ - & - & 1 & 1 & 1 & - & 3 & 1 & - & 2 \\ 9 & 1 & - & 2 & - & 3 & 2 & - & 2 & - \\ - & 1 & 2 & - & 3 & 3 & 4 & - & - & 1 \\ 8 & 1 & - & 3 & - & 3 & - & 2 & - & 1 \\ 9 & - & 3 & 3 & 3 & - & - & - & 3 & - \\ 9 & 3 & 2 & 4 & - & - & - & 1 & 1 & - \\ 7 & 1 & - & 2 & - & 1 & - & -1 & 3 & - \\ 8 & - & 2 & - & - & 3 & 1 & - & - & 5 \\ - & 2 & - & 1 & 1 & -2 & - & 3 & 5 & - \end{pmatrix}$$

$$4. E = \begin{pmatrix} - & 8 & - & - & 9 & 9 & - & 6 & 9 & 7 \\ 8 & - & 6 & 4 & - & - & 2 & - & 4 & - \\ - & 6 & - & 2 & 3 & - & 4 & 4 & - & 5 \\ - & 4 & 2 & - & 1 & - & 1 & 5 & 2 & - \\ 9 & - & 3 & 1 & - & - & 2 & 5 & - & 4 \\ 9 & -1 & - & - & - & - & 2 & 1 & 1 & -1 \\ - & 2 & 4 & 1 & 2 & 2 & - & - & 1 & - \\ 6 & - & 4 & 5 & 5 & 1 & - & - & - & 1 \\ 9 & 4 & - & 2 & - & 1 & 1 & - & - & 4 \\ 7 & - & 5 & - & 4 & - & - & 1 & 4 & - \end{pmatrix}$$

$$5. E = \begin{pmatrix} - & 12 & 11 & - & 10 & - & 8 & - & 7 & 11 \\ 12 & - & 1 & 4 & - & 3 & 2 & 1 & - & - \\ 11 & 1 & - & - & -2 & - & - & 3 & 3 & 2 \\ - & 4 & - & - & 6 & 1 & 4 & - & 7 & 5 \\ 10 & - & - & 6 & - & 5 & 1 & - & 2 & - \\ - & 3 & - & 1 & 5 & - & 6 & 1 & - & 5 \\ 8 & 2 & - & 4 & 1 & 6 & - & - & 3 & - \\ - & 1 & 3 & - & - & 1 & - & - & 5 & -2 \\ 7 & - & 3 & 7 & 2 & - & 3 & 5 & - & 3 \\ 11 & - & 2 & 5 & - & 5 & - & - & 3 & - \end{pmatrix}$$

$$6. E = \begin{pmatrix} - & - & 11 & 6 & - & 9 & 11 & 9 & - & 13 \\ - & - & 3 & - & 2 & 3 & - & 1 & 3 & 6 \\ 11 & 3 & - & 4 & 4 & - & - & 2 & 1 & - \\ 6 & - & 4 & - & 4 & 1 & 2 & - & - & 8 \\ - & 2 & 4 & 4 & - & - & 1 & - & 2 & - \\ 9 & 3 & - & 1 & - & - & - & - & 5 & 8 \\ 11 & - & - & 2 & 1 & -2 & - & 3 & 3 & - \\ 9 & 1 & 2 & - & -1 & - & 3 & - & - & 5 \\ - & 3 & 1 & - & 2 & 5 & 3 & - & - & 5 \\ 13 & 6 & - & 8 & - & 8 & - & 5 & 5 & - \end{pmatrix}$$

$$7. E = \begin{pmatrix} - & - & 8 & - & 11 & 8 & 6 & - & 10 & 9 \\ - & - & 2 & 3 & - & 4 & - & 2 & 1 & 4 \\ 8 & 2 & - & - & 5 & - & 3 & 4 & - & -1 \\ - & 3 & - & - & 3 & 4 & 2 & 5 & 1 & - \\ 11 & - & 5 & 3 & - & 6 & - & 1 & - & 6 \\ 8 & 4 & - & 4 & 6 & - & - & - & 1 & - \\ 6 & - & 3 & 2 & - & - & - & 6 & 2 & 2 \\ - & 2 & 4 & 5 & 1 & - & 6 & - & 5 & - \\ 10 & 1 & - & 1 & - & -1 & 2 & 5 & - & - \\ 9 & 4 & - & - & 6 & 1 & 2 & - & - & - \end{pmatrix}$$

$$8. E = \begin{pmatrix} - & - & 13 & 10 & - & 7 & 10 & - & 10 & 11 \\ - & - & - & 2 & 3 & - & 2 & 3 & 2 & -1 \\ 13 & - & - & 3 & 7 & 4 & - & 5 & - & 4 \\ 10 & 2 & 3 & - & - & - & - & 1 & 3 & 4 \\ - & 3 & 7 & - & - & 3 & 2 & 5 & - & - \\ 7 & - & 4 & - & 3 & - & 2 & 3 & 1 & - \\ 10 & 2 & - & - & 2 & 2 & - & - & 1 & 1 \\ - & 3 & 5 & 1 & 5 & 3 & - & - & - & - \\ 10 & 2 & - & 3 & -2 & 1 & 1 & - & - & 1 \\ 11 & - & 4 & 4 & - & - & 1 & - & 1 & - \end{pmatrix}$$

2 РОЗВ'ЯЗАННЯ ТИПОВОГО ВАРІАНТУ ІНДИВІДУАЛЬНОГО ЗАВДАННЯ 1

2.1. Приклад розв'язання завдання 1.

Завдання 1.

Побудувати діаграму, матрицю суміжності та список неорієнтованого графа $G = (V, E)$, у якого $V = \{1, 2, 3, 4, 5, 6\}$ та $E = \{(1, 3), (1, 5), (2, 4), (3, 5)\}$. Знайти степені всіх вершин графа G та кількість його компонент зв'язності.

Розв'язання:

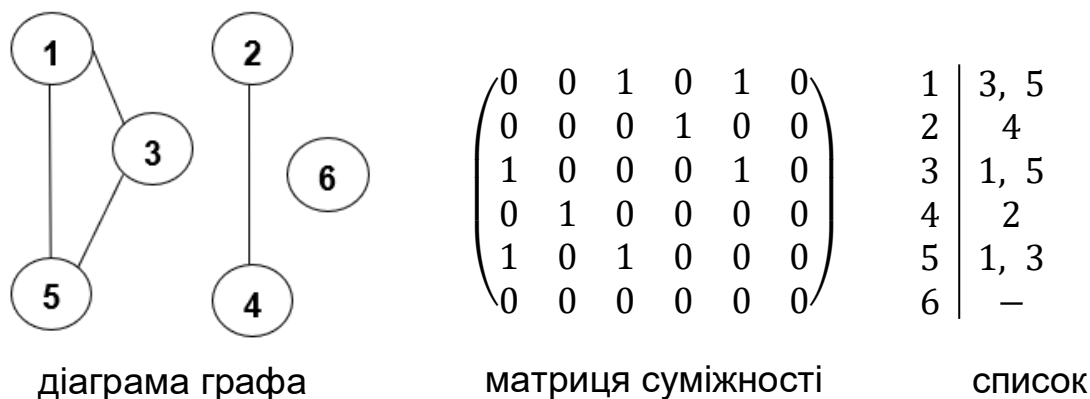


Рис. 2.1. Способи задання графа

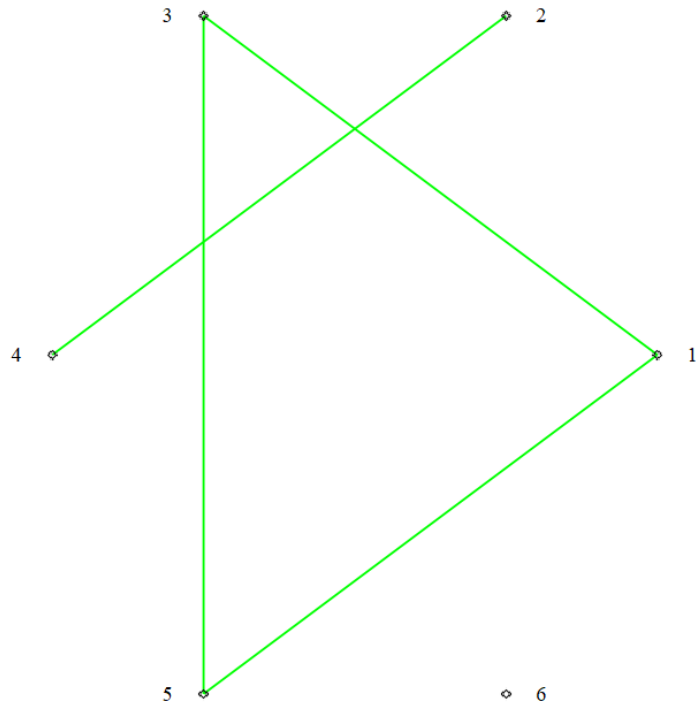
Степенем вершини v називається кількість ребер, інцидентних цій вершині (позначається $d(v)$). Вершина степеня 0 називається ізольованою, а вершина степеня 1 – висячою. В заданому графі $d(1) = 2$; $d(2) = 1$; $d(3) = 2$; $d(4) = 1$; $d(5) = 2$; $d(6) = 0$.

Компонентою зв'язності графа називається максимальний зв'язний підграф у графі. Так заданий граф має три компоненти зв'язності: $\{\{1, 3, 5\}, \{2, 4\}, \{6\}\}$, тобто $K = 3$.



Приклад розв'язання завдання 1 з використанням можливостей Maple.

```
restart :  
with(networks) :  
  
G1 := graph({1,2,3,4,5,6}, {{1,3}, {1,5}, {2,4}, {3,5}}) :  
draw(G1) :
```



```
# Означення множин вершин і ребер  
V := {1,2,3,4,5,6} :  
E := {{1,3}, {1,5}, {2,4}, {3,5}} :  
  
# Побудова матриці суміжності  
AdjMatrix := Matrix(6,6, (i,j) → if {i,j} in E then 1 else 0 fi) :  
AdjMatrix;
```

$$\begin{bmatrix} 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Рис. 2.2. Розв'язання завдання 1 з використанням можливостей СКМ Maple

2.2. Приклад розв'язання завдання 2.

Завдання 2.

Зважений граф $G = (V, E)$, у якого $V = \{1,2,3,4,5,6,7\}$ задано матрицею ваг:

$$E = \begin{pmatrix} - & 7 & 3 & 6 & 7 & 8 & 4 \\ 7 & - & 7 & 1 & 9 & 1 & 6 \\ 3 & 7 & - & 7 & 6 & 7 & 5 \\ 6 & 1 & 7 & - & 8 & 2 & 6 \\ 7 & 9 & 6 & 8 & - & 9 & 2 \\ 8 & 1 & 7 & 2 & 9 & - & 5 \\ 4 & 6 & 5 & 6 & 2 & 5 & - \end{pmatrix}$$

За допомогою алгоритму Краскала знайти для цього графа остів мінімальної ваги.

Розв'язання:

Алгоритм Краскала: Побудова остова мінімальної ваги.

Спочатку множина обраних ребер G' є порожньою.

Крок 1. Сортуємо ребра графа за зростанням ваги.

Крок 2. Поки це можливо, виконуємо таку операцію: додаємо до множини G' ребро найменшої ваги, що не утворює цикл з ребрами, які обрали раніше.

Якщо крок 2 виконати неможливо, то алгоритм закінчує свою роботу.

Хід виконання алгоритму Краскала відобразимо у таблиці 2.1.

Таблиця 2.1 - Ілюстрація виконання алгоритму Краскала для графа G .

Обране ребро	Компоненти	Вага обраного
(2,4)	{2,4}	1
(2,6)	{2,4,6}	2
(5,7)	{2,4,6}, {5,7}	4
(1,3)	{2,4,6}, {5,7}, {1,3}	7
(1,7)	{2,4,6}, {1,3,5,7}	11
(6,7)	{1,2,3,4,5,6,7}	16



На рисунку 2.3 побудуємо діаграму знайденого остову:

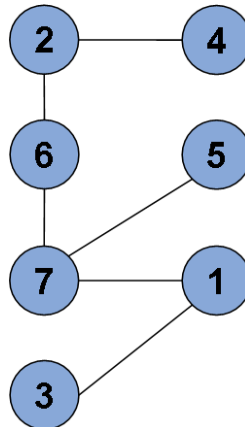


Рис. 2.3. Остів найменшої ваги для графа G .

2.3. Приклад розв'язання завдання 3.

Завдання 3.

Зважений граф $G = (V, E)$, у якого $V = \{1, 2, 3, 4, 5, 6, 7\}$ задано матрицею ваг:

$$E = \begin{pmatrix} - & 7 & 3 & 6 & 7 & 8 & 4 \\ 7 & - & 7 & 1 & 9 & 1 & 6 \\ 3 & 7 & - & 7 & 6 & 7 & 5 \\ 6 & 1 & 7 & - & 8 & 2 & 6 \\ 7 & 9 & 6 & 8 & - & 9 & 2 \\ 8 & 1 & 7 & 2 & 9 & - & 5 \\ 4 & 6 & 5 & 6 & 2 & 5 & - \end{pmatrix}$$

За допомогою алгоритму Прима знайти для цього графа остів мінімальної ваги.

Розв'язання:

Алгоритм Прима: Побудова остова мінімальної ваги.

Спочатку множина обраних ребер G' та множина обраних вершин V' є порожніми.

Крок 1. Сортуємо ребра графа за зростанням ваги.

Крок 2. Обираємо ребро (v', v'') найменшої ваги. Вершини v' та v'' додаємо до множини обраних вершин V' .

Крок 3. Поки це можливо, виконуємо такі операції: додаємо до



множини G' ребро найменшої ваги вигляду (v, w) , де $v \in V'$ та $w \notin V'$ і додаємо вершину w до множини обраних вершин V' .

Якщо крок 3 виконати неможливо, то алгоритм закінчує свою роботу.

Хід виконання алгоритму Прима зручно зображати у вигляді таблиці, що є аналогічною таблиці для алгоритму Краскала.

Таблиця 2.2 - Ілюстрація виконання алгоритму Прима для графа G

Обране ребро	Компонента	Вага обраного
(2,4)	{2,4}	1
(2,6)	{2,4,6}	2
(6,7)	{2,4,6,7}	7
(7,5)	{2,4,5,6,7}	9
(7,1)	{1,2,4,5,6,7}	13
(1,3)	{1,2,3,4,5,6,7}	16

Остів, що знайдено за алгоритмом Прима збігається з остовом, що знайдено за алгоритмом Краскала (рис. 2.3).

Приклад розв'язання завдання 2-3 з використанням можливостей Maple.

#Перезапускає середовище Maple, видаляючи всі попередні змінні, функції та налаштування. Це забезпечує чистий початок.
restart;

with(GraphTheory) :

Визначення матриці ваг

```
W := Matrix([
  [0, 7, 3, 6, 7, 8, 4],
  [7, 0, 1, 9, 7, 9, 6],
  [3, 1, 0, 7, 7, 9, 5],
  [6, 9, 7, 0, 9, 8, 9],
  [7, 7, 7, 9, 0, 9, 2],
  [8, 9, 9, 8, 9, 0, 5],
  [4, 6, 5, 9, 2, 5, 0]
]);
```

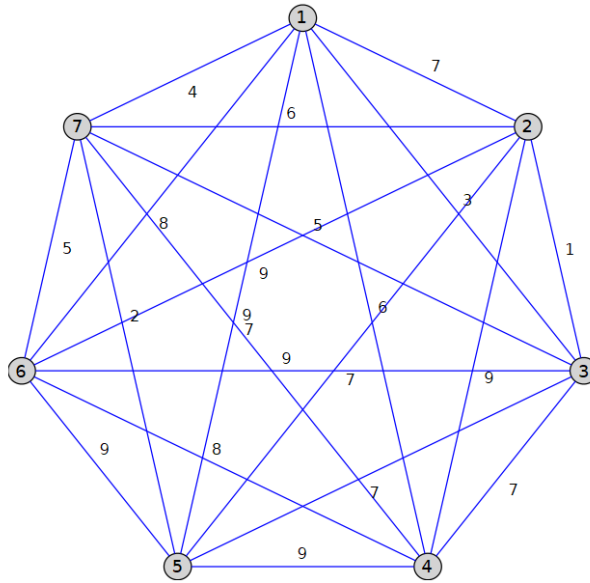
Створення графа з матриці ваг

```
G := Graph(W) :
```

Візуалізація початкового графа

```
DrawGraph(G);
```

Рис. 2.4. Розв'язання завдання 2-3 з використанням можливостей СКМ Maple, лист 1



```

#Перезапускає середовище Maple, видаляючи всі попередні змінні, функції та налаштування.
restart;
#Завантажує пакет networks, який дозволяє працювати з графами.
with(networks):
#Завантажує пакет LinearAlgebra.
with(LinearAlgebra):
#Створює новий об'єкт графа G.
new(G):
#Додає вершини в граф G та створює послідовність чисел від 1 до 7, які стають вершинами графа.
addvertex(seq(i, i=1..7), G):
#Визначає порядок розташування вершин для візуалізації графа. Вершини згруповані у певному порядку: спочатку 4, потім 5 і 3, далі 7, і так далі.
r := [4], [5, 3], [7], [6, 2], [1]:
#Додає ребра до графа G.
addedge([ [1, 2], [1, 3], [1, 4], [1, 5], [1, 6], [1, 7], [2, 1], [2, 3], [2, 4], [2, 5], [2, 6], [2, 7], [3, 1], [3, 2], [3, 4], [3, 5], [3, 6], [3, 7], [4, 1], [4, 2], [4, 3], [4, 4], [4, 5], [4, 6], [4, 7],
[5, 1], [5, 2], [5, 3], [5, 4], [5, 5], [5, 6], [5, 7], [6, 1], [6, 2], [6, 3], [6, 4], [6, 5], [6, 6], [6, 7], [7, 1], [7, 2], [7, 3], [7, 4], [7, 5], [7, 6], [7, 7] ], weights = [7, 3, 6, 7, 8, 4, 7, 7, 1, 9, 1, 6, 3, 7, 7, 6,
7, 5, 6, 1, 7, 8, 2, 6, 7, 9, 6, 8, 9, 2, 8, 1, 7, 2, 9, 5, 4, 6, 5, 6, 2, 5], G):
#Візуалізує граф G, використовуючи порядок вершин r.
draw(Linear(r), G);
#Знаходить мінімальний остов графа G.
T := sprantree(G):
#Отримує список ребер мінімального остова T.
Ed := edges(T):
#Обчислює сумарну вагу мінімального остова T.
MinW = add(eweight(Ed[i], T), i=1..7):
#Малює мінімальний остов T, використовуючи той самий порядок вершин r.
draw(Linear(r), T);

```

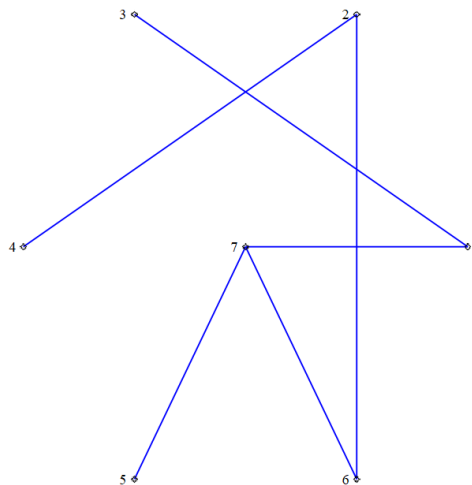


Рис. 2.4. Розв'язання завдання 2-3 з використанням можливостей СКМ Maple, лист 2

2.4. Приклад розв'язання завдання 4.

Завдання 4.

Зважений граф $G = (V, E)$, у якого $V = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$ задано матрицею ваг:

$$E = \begin{pmatrix} - & 11 & 6 & - & 9 & 8 & - & 10 & - \\ 11 & - & - & 3 & -2 & 3 & - & 2 & 1 \\ 6 & - & - & - & 2 & 1 & 4 & - & 2 \\ - & 3 & - & - & - & 5 & - & 1 & 2 \\ 9 & - & 2 & - & - & 1 & 2 & - & 3 \\ 8 & 3 & 1 & 5 & 1 & - & 3 & 4 & - \\ - & - & 4 & - & 2 & 3 & - & - & 1 \\ 10 & 2 & - & 1 & - & 4 & -2 & - & 5 \\ - & 1 & 2 & 2 & 3 & - & 1 & 5 & - \end{pmatrix}$$

За допомогою алгоритму Форда–Беллмана знайти для цього графа найкоротші відстані та шляхи від вершини 1 до усіх інших вершин.

Розв'язання:

В теорії графів орієнтовний цикл, сума ваг ребер якого є від'ємним числом, дістав назву *від'ємного контуру*. Доведено, що у зваженому графі існують найкоротші відстані та шляхи тоді і лише тоді, коли граф є зв'язним та не містить жодного від'ємного контуру.

Для цієї задачі найбільш відомим алгоритмом розв'язку є алгоритм Форда–Беллмана. Цей алгоритм є дуже схожим на алгоритм Дейкстри та відрізняється від нього лише двома моментами:

а) оскільки в графі можуть бути ребра від'ємної ваги, то треба перераховувати відстані до усіх вершин, в тому числі до вершин, які було помічено раніше. До таких вершин також може бути зменшення відстані.

б) якщо відбулося зменшення відстані до вершини, що була поміченою раніше, то треба перерахувати довжину усіх шляхів, що ведуть з цієї вершини. Для цього знімають помітку з цієї вершини.

Запишемо кроки алгоритму Форда–Беллмана.

Алгоритм Форда–Беллмана: Відшукання найкоротших відстаней та шляхів від однієї вершини графа до усіх інших.



Спочатку всі вершини є непоміченими, відстань до вершини 1 дорівнює 0, до всіх інших вершин відстані дорівнюють $+\infty$, а шляхи є порожніми.

Крок 1. Знаходимо непомічену вершину v з найменшою відстанню та помічаємо цю вершину.

Крок 2. Перераховуємо відстані до усіх інших вершин v' за формулою $d(v') = \min\{d(v'); d(v) + \text{вага}(v, v')\}$. Якщо при перерахуванні відстань до вершини v' змінилась, то також змінюється шлях до вершини v' : $p(v') = p(v) + "v"$ (під знаком + розуміємо дописування до шляху $p(v)$ вершини v'). Якщо змінюється відстань до раніше поміченої вершини, то помітка цієї вершини знімається. Після перерахування повертаємося до кроку 1.

Алгоритм закінчує свою роботу у випадку, коли всі вершини є поміченими, та крок 2 не приводить до жодної зміни.

Хід виконання алгоритму Форда–Беллмана зручно ілюструвати у вигляді таблиці. Ця таблиця повністю аналогічна таблиці, що ілюструє виконання алгоритму Дейкстри. Проілюструємо хід виконання алгоритму Форда–Беллмана для нашого завдання.

Таблиця 2.3 - Ілюстрація виконання алгоритму Форда–Беллмана для графа G

2	3	4	5	6	7	8	9
11 10 9	6	12 10	9 8 7	8 7	10 9 8	10	8
1,2 1,3,6,2 1,3,9,2	1,3	1,3,6,4 1,3,9,4	1,5 1,3,5 1,3,9,2,5	1,6 1,3,6	1,3,7 1,3,9,7 1,8,7	1,8	1,3,9

При виконанні алгоритму Форда–Беллмана поточними були вершини у такому порядку: 3, 6, 5, 9, 2, 5, 7, 4, 8, 7. Найкоротші відстані до кожної вершини та шляхи, що їм відповідають, розміщені у відповідному



стовпці таблиці останніми. На рисунку 3.16 зобразимо ці найкоротші шляхи у вигляді дерева.

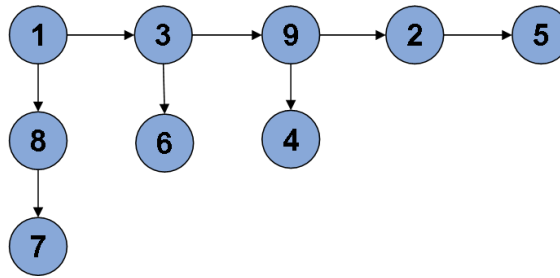


Рис. 2.5. Найкоротші шляхи (від вершини 1) для графа G

Приклад розв'язання завдання 4 з використанням можливостей Maple.

```

restart;
with(GraphTheory) :

# Визначення матриці ваг графа
W := Matrix([
  [infinity, 11, 6, infinity, 9, 8, infinity, 10, infinity],
  [11, infinity, infinity, 3, -2, 3, infinity, 2, 1],
  [6, infinity, infinity, infinity, 2, 1, 4, infinity, 2],
  [infinity, 3, infinity, infinity, infinity, 5, infinity, 1, 2],
  [9, infinity, 2, infinity, infinity, 1, 2, infinity, 3],
  [8, 3, 1, 5, 1, infinity, 3, 4, infinity],
  [infinity, infinity, 4, infinity, 2, 3, infinity, infinity, 1],
  [10, 2, infinity, 1, infinity, 4, -2, infinity, 5],
  [infinity, 1, 2, 2, 3, infinity, 1, 5, infinity],
]);

# Створення графа
G := Graph(W) :

# Визначення початкової вершини
start := 1;

# Виконання алгоритму Форда-Белмана
distances := BellmanFordAlgorithm(G, start);

# Обчислення найкоротших шляхів до всіх вершин
paths := [seq(ShortestPath(G, start, v), v in Vertices(G))];

# Виведення результатів
print("Найкоротші шляхи від вершини", start, ":", paths);
print("Відстані до кожної вершини:", distances);

start := 1
distances := [[[1], 0], [[1, 3, 9, 2], 9], [[1, 3], 6], [[1, 3, 9, 4], 10], [[1, 3, 9, 2, 5], 7], [[1, 3, 6], 7], [[1, 8, 7], 8], [[1, 8], 10], [[1, 3, 9], 8]]
paths := [[1], [1, 3, 9, 2], [1, 3], [1, 3, 9, 4], [1, 3, 9, 2, 5], [1, 3, 6], [1, 8, 7], [1, 8], [1, 3, 9]]
"Найкоротші шляхи від вершини", 1, ":", [[1], [1, 3, 9, 2], [1, 3], [1, 3, 9, 4], [1, 3, 9, 2, 5], [1, 3, 6], [1, 8, 7], [1, 8], [1, 3, 9]]
"Відстані до кожної вершини:", [[[1], 0], [[1, 3, 9, 2], 9], [[1, 3], 6], [[1, 3, 9, 4], 10], [[1, 3, 9, 2, 5], 7], [[1, 3, 6], 7], [[1, 8, 7], 8], [[1, 8], 10], [[1, 3, 9], 8]] (1)

```

Рис. 2.5. Розв'язання завдання 4 з використанням можливостей СКМ Maple

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. McHale A. Foundational Mathematics. LibreTexts. 2021. URL: <https://read.kortext.com/library/books/996663>.
2. Kaidan N. The use of MAPLE CMS in the course of studying the discipline «higher and discrete mathematics» by students of economic specialties. *Міжнародна науково-технічна конференція «MININGMETALTECH 2023 – Гірничо-металургійний комплекс: інтеграція бізнесу, технологій та освіти» 29–30 листопада 2023 р., 2023, С. 243-245.*
3. Loucks D. P. Public Systems Modeling. Springer Nature, 2022. 3098 p. URL: <https://read.kortext.com/library/books/1971910>.
4. Maple : веб-сайт. URL: <https://www.maplesoft.com/products/Maple/> (дата звернення: 15.09.24).
5. Monagan M. B., Geddes K. O., Heal K. M., Labahn G., Vorkoetter S. M., McCarron J., DeMarco P. Maple Advanced Programming Guide Maplesoft (15 version), a division of Waterloo Maple Inc. 2009. 452 p.
6. Strang G., Herman E. Calculus. OpenStax. Volume 2. 2016. 829 p. URL: <https://read.kortext.com/library/books/213032>.
7. Вислоух С. П., Волошко О. В., Тимчик Г. С., Філіппова М. В. Комп'ютерне моделювання процесів та систем. Чисельні методи : підручник. Київ : КПІ ім. Ігоря Сікорського, Вид-во «Політехніка», 2021. 228 с.
8. Дубовой В. М., Никитенко О. Д., Юхимчук М. С., Галушак А. В. Моделювання об'єктів і систем : лабораторний практикум. Вінниця : ВНТУ, 2021. 156 с.
9. Дубовой В. М. Задачі з аналізу і моделювання систем : збірник задач для студентів спеціальності 151 – «Автоматизація та комп'ютеро-інтегровані технології». Вінниця : ВНТУ, 2021. 76 с.



10. Задачин В. М., Конюшенко І. Г. Лабораторний практикум з навчальної дисципліни «Моделювання систем» : навч.-практ. посіб. Харків : Вид. ХНЕУ, 2019. 212 с.

11. Кайдан Н. В. Використання систем комп'ютерної математики Maple при вивченні дисципліни «Моделювання і прогнозування в економіці і менеджменті». *Scientific and pedagogical internship «The latest trends in physical and mathematical education in higher education institutions»: Internship proceedings*, (April 3 – May 14, 2023. Riga, the Republic of Latvia) Riga, Latvia: «Baltija Publishing», 2023. P. 27-30

12. Кайдан В. П., Кайдан Н. В., Колесников С. О. Інтерактивні симуляції як елемент комп'ютерного моделювання при викладанні природничо-математичних дисциплін. *Педагогічна Академія: наукові запуски*. 2024. № 11. DOI: <https://doi.org/10.5281/zenodo.14177740>.

13. Ніколюк П. К. Моделювання систем : навчальний посібник для здобувачів вищої освіти спеціальності 122 Комп'ютерні науки. Вінниця : ДонНУ, 2023. 228 с. URI: <https://r2.donnu.edu.ua/handle/123456789/3259>.

14. Павленко П. М. Основи математичного моделювання систем і процесів : навч. посіб. Київ : Книжкове вид-во НАУ, 2013. 201 с.

15. Томашевський В. М., Жданова О. Г., Жолдакова О. О. Вирішення практичних завдань методами комп'ютерного моделювання : навч. посіб. Київ : Корнійчук, 2019. 214 с.



Навчально-методичне видання

Кайдан Наталія Володимирівна

МЕТОДИЧНІ РЕКОМЕНДАЦІЇ

до виконання індивідуального завдання «Оптимізація маршрутизації у комп'ютерній мережі» з дисципліни «Моделювання систем»

Самостійне електронне мережеве видання

Публікується в авторській редакції