Міністерство освіти і науки України

Національний університет водного господарства
та природокористування

Кафедра економічної кібернетики

06-11-51

МЕТОДИЧНІ ВКАЗІВКИ

та варіанти завдань для проведення практичних занять та виконання самостійних завдань із навчальної дисципліни

«Застосування графів в економічних задачах»

для здобувачів вищої освіти другого (магістерського) рівня
усіх освітньо-професійних програм спеціальностей НУВГП
денної та заочної форми навчання

Схвалено
науково-методичною радою НУВГП
Протокол № 5 від 25.09.2019 р.

Рівне – 2019

Методичні вказівки та варіанти завдань для проведення практичних занять та виконання самостійних завдань із навчальної дисципліни «Застосування графів в економічних задачах» для здобувачів вищої освіти другого (магістерського) рівня усіх освітньо-професійних програм спеціальностей НУВГП денної та заочної форми навчання [Електронне видання] / Бабич Т. Ю. – Рівне : НУВГП, 2019. – 40 с.

Укладач: Бабич Т. Ю., к.е.н., старший викладач кафедри економічної кібернетики.

Відповідальний за випуск: Грицюк П. М., д.е.н., професор, завідувач кафедри економічної кібернетики.

Вчений секретар
науково-методичної ради Костюкова Т. А.

ЗМІСТ

ВСТУП

Теорія графів – одна з істотних частин математичного апарату інформатики та кібернетики. У термінах теорії графів можна сформулювати багато задач, пов'язаних з дискретними об'єктами. Такі задачі виникають зокрема в економіці. В даному курсі розглянуто як базові поняття теорії графів, так і основні підходи та алгоритми розв'язування численних задач практичного характеру для економічних та соціальних проблем.

В першій частині навчальної дисципліни здобувачі вищої освіти мають змогу ознайомитися з елементами теорії графів.

Розуміючи під графом певну множину точок, деякі з яких з'єднані лініями, цей незвичний математичний об'єкт можна без перебільшення застосовувати в будь-якій науковій чи практичній галузі. Незвичним в цій "геометрії" є майже все: в ній немає кутів, немає відстаней між точками в звичному розумінні цього слова, рівноправними є будь-які розташування точок на рисунку, не розрізняються лінії (прямі чи криві), що з'єднують точки, специфічними є операції, не схожі на арифметичні, алгебраїчні чи геометричні. Перевагами такого підходу є простота та наочність. Складність полягає у вмінні побачити можливості переформулювання умови задачі мовою графів – постановці так званої графової моделі задачі. Розв'язавши задачу в межах теорії графів, результат необхідно інтерпретувати у вихідних термінах.

Другу частину даного курсу присвячено можливим застосуванням графів в економіці.

Пропоновані завдання розділено на дві частини: першу частину призначено для розбору та виконання студентами під час проведення практичних занять. У більшості випадків наведено необхідні короткі теоретичні відомості, приклади розв'язування аналогічних завдань.

Завдання другої частини студенти мають змогу опанувати самостійно. На самостійне опрацювання винесено деякі задачі оптимізації на графах.

В результаті вивчення дисципліни «Застосування графів в економічних задачах» студенти повинні отримати знання про основні поняття теорії графів, різнопланове застосування графів в різних галузях економіки, застосування графів до мережних систем, постановки основних задач, що зводяться до графів; сформувати вміння і практичні навички з бачення можливості переформулювання умови задачі мовою графів (створення графової моделі задачі), розв'язування задачі в межах теорії графів та інтерпретації результатів у вихідних термінах.

ПРАКТИЧНІ РОБОТИ

Практична робота №1. Операції над графами. Ізоморфність графів

Теоретичні відомості

Граф G(X,W) – пара множин $X,W \subset M_2$. Множина X – це множина вершин, множина W – множина ребер. Якщо $x_i,x_j \in W$, ребро x_i,x_j сполучає вершину x_i з вершиною x_j, інша термінологія – ребро x_i,x_j і вершини x_i та x_j *інцидентні*. Тут $X = x_1,...,x_n$ - деяка скінченна множина (множина вершин), M_2 – множина всіх невпорядкованих пар елементів з X.

$$M_2 = x_i,x_j : x_i \in X, x_j \in X, i \neq j .$$

Граф G(X,W) називається *повним*, якщо $W = M_2$.

Граф G(X,W) називається *порожнім*, якщо $W = \varnothing$.

Вершини x_i та x_j графа G(X,W) *інцидентні*, якщо $x_i,x_j \in W$.

Як правило, граф зображується діаграмою, і саме її часто називають графом. Розрізняють *неорієнтовані* графи, вершини яких з'єднані ненаправленими відрізками — ребрами. Якщо ж пари вершин упорядковані, тобто відмінність між початковою і кінцевою вершинами ребра є істотною, то такі ребра називають *дугами*, а відповідний граф — *орієнтованим*.

Граф G(X, W) є *сумою* (об'єднанням) графів $G(X_1, W_1)$, ... ,$G(X_k, W_k)$, множиною вершин якого є $X = X_1 \cup X_2$, а множиною ребер $W = W_1 \cup W_2$.. Ця сума називається *прямою*, якщо $X_i \cap X_j = \varnothing, i \neq j$.

Перетином і *різницею* графів G_1 (X , W_1) і G_2(X , W_2) з однаковими множинами вершин називаються графи G' $X, W_1 \cap W_2$ і G'' $X, W_1 \backslash W_2$ відповідно; позначаються $G' = G_1 \cap G_2$ і $G'' = G_1 \backslash G_2$.

Доповненням графа G(X,W) називається граф $\overline{G}$ $X, M_2 \backslash W$.

Графи G_1 X_1, W_1 і G_2 X_2, W_2 називаються *ізоморфними,* якщо існує таке взаємно однозначне відображення φ множини вершин X_1 на множину вершин X_2, що ребро (v, w) належить W_1 тоді і тільки тоді, коли ребро (φ(v),φ(w)) належить W_2. Відображення φ називається *ізоморфним відображенням* або *ізоморфізмом* графа G_1 на граф G_2.

Завдання для виконання

1.1. Знайти об'єднання та перетин графів (рис.1.1, а-д):

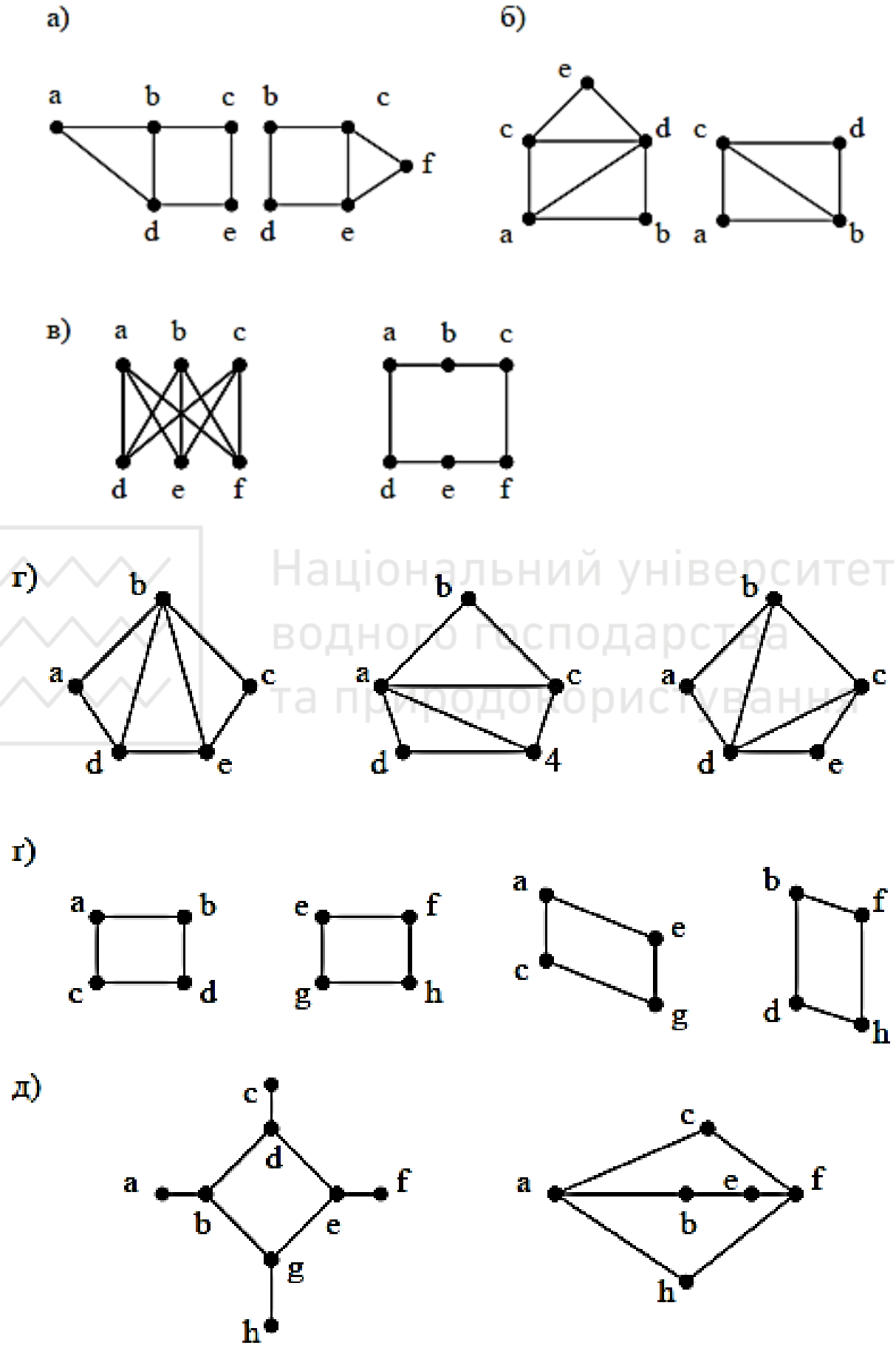


Рис. 1.1.

1.2. Знайти доповнення графів (рис.1.2, а-и):

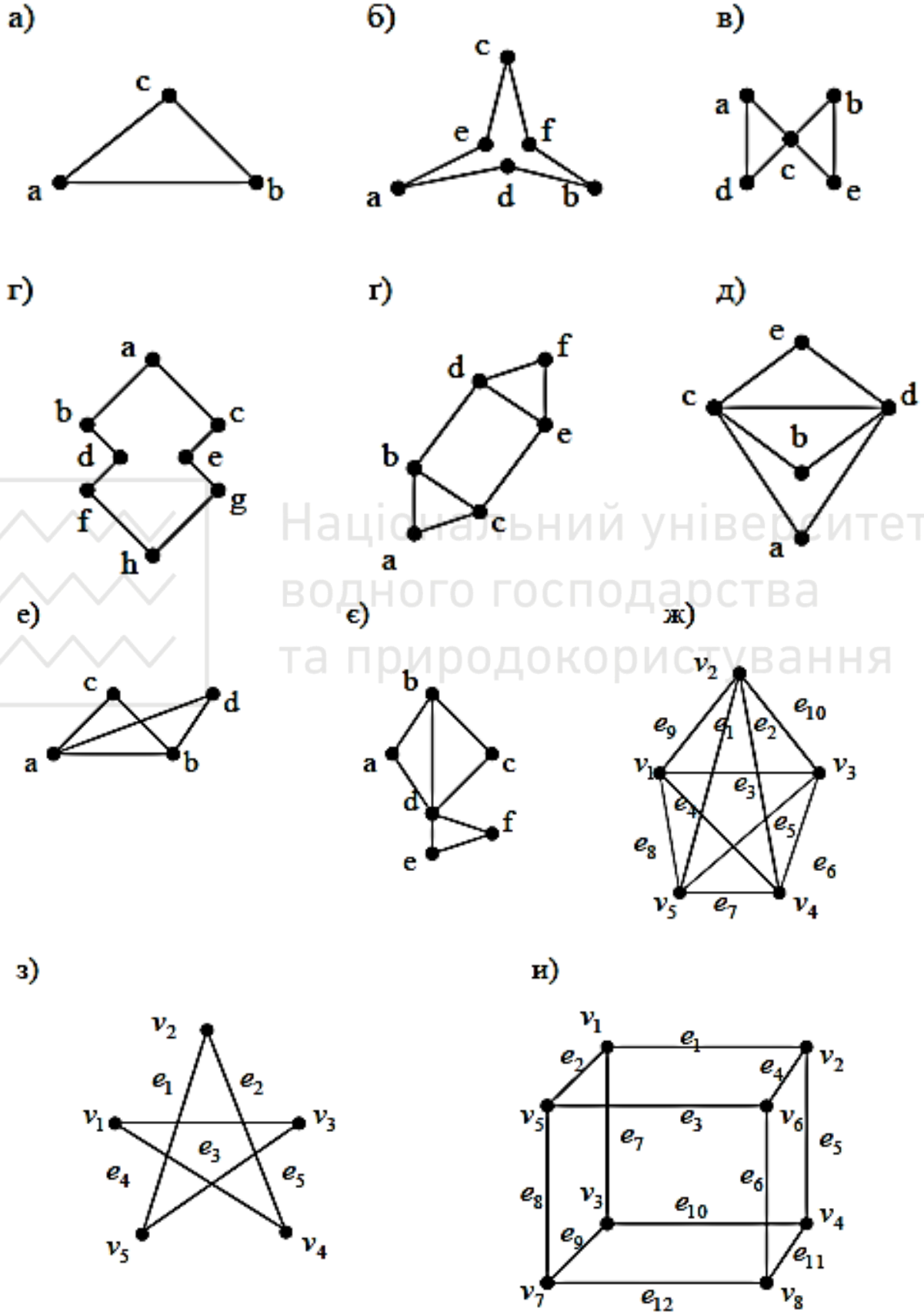


Рис. 1.2.

1.3. Пояснити, чому пари графів, зображені на рис. 1.3, а-е, не є ізоморфними.

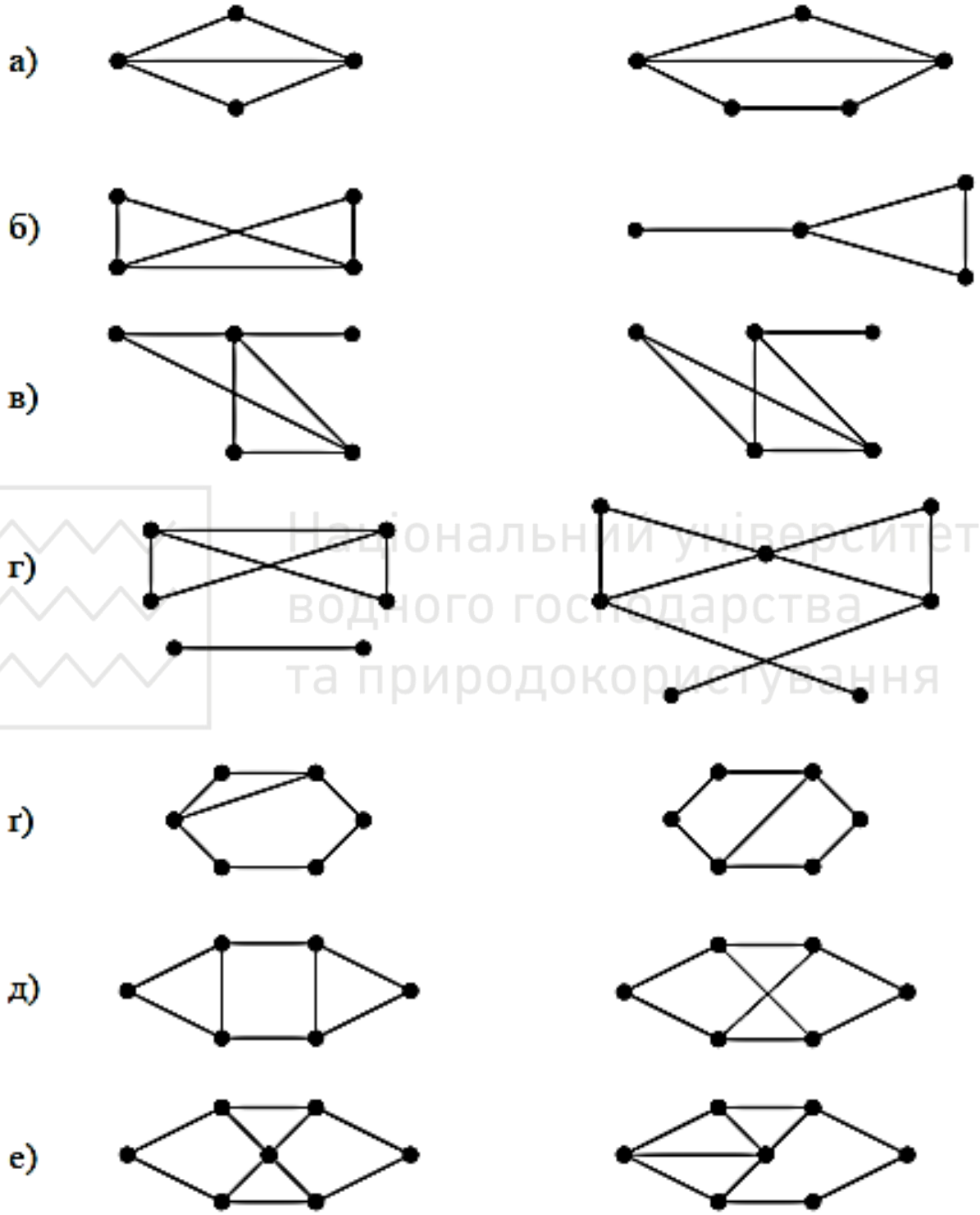


Рис. 1.3.

1.4. Довести, що пари графів, зображені на рис. 1.4, а-в, є ізоморфними.

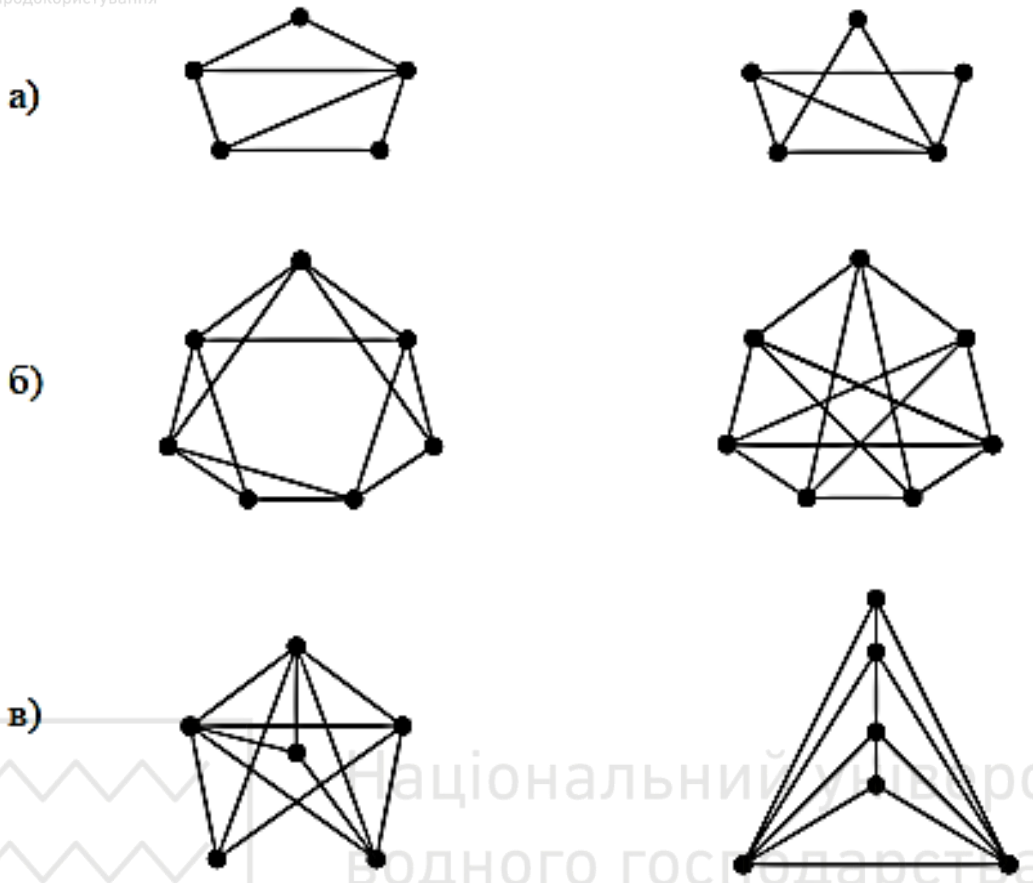


Рис. 1.4.

1.5. Чи є ізоморфними графи (рис.1.5, а-к)? Відповідь обґрунтувати.

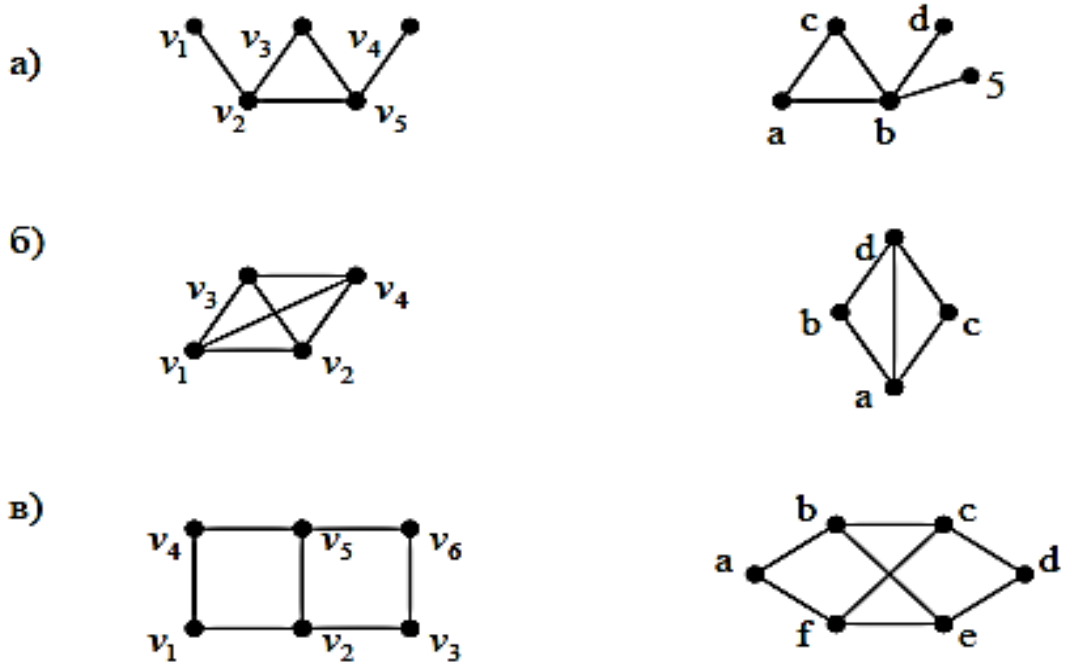
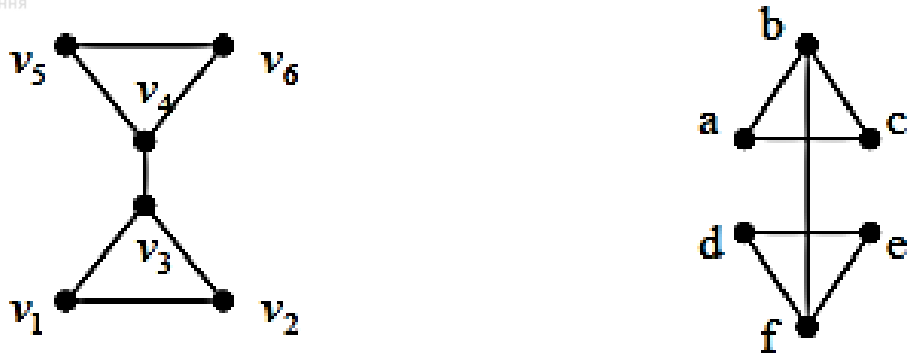
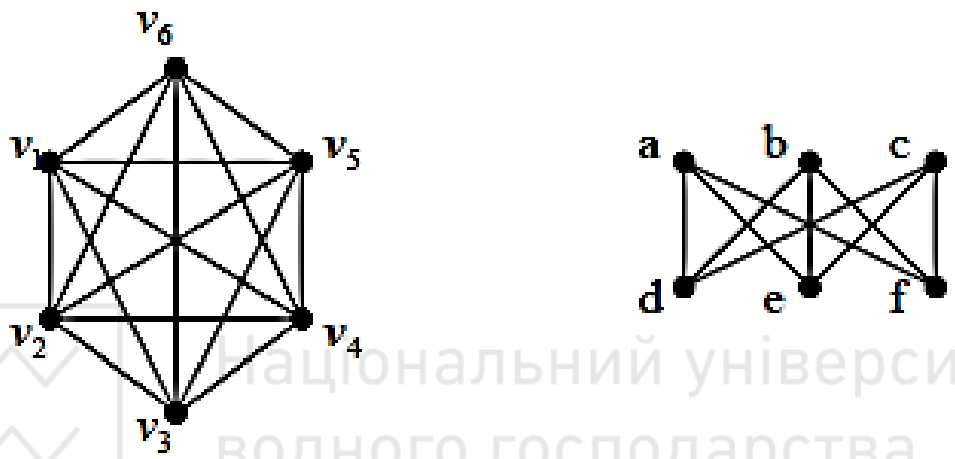


Рис. 1.5 (а-в).

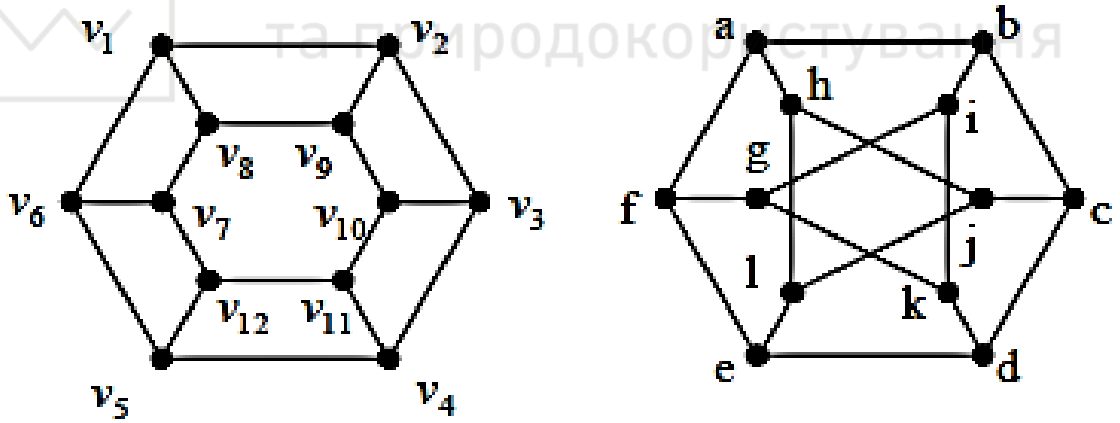
і)



ї)



й)



к)

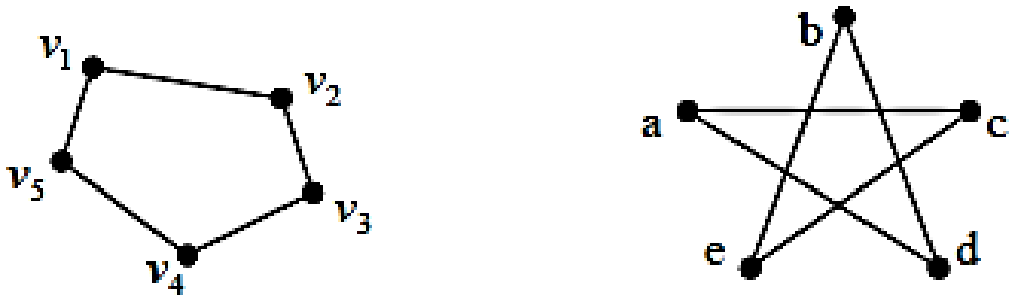


Рис. 1.5 (і-к).

1.6. Перевірити, чи є ізоморфними графи G_1 і G_2, задані своїми матрицями суміжності A_1 і A_2.

а) $A_1 = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \end{pmatrix}$, $A_2 = \begin{pmatrix} 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 \end{pmatrix}$;

б) $A_1 = \begin{pmatrix} 0 & 1 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 \end{pmatrix}$, $A_2 = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 \end{pmatrix}$;

в) $A_1 = \begin{pmatrix} 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 \end{pmatrix}$, $A_2 = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 \end{pmatrix}$;

г) $A_1 = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \end{pmatrix}$, $A_2 = \begin{pmatrix} 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 \end{pmatrix}$.

Практична робота №2. Різні форми подання графів. Локальні ступені вершин графа

Теоретичні відомості

Степенем $d\ x_i$ вершини x_i графа G(X,W) називається число вершин x_j, які інцидентні вершині x_i (число відрізків, що виходять із вершини x_i). Для орієнтованого графа визначаються напівстепені виходу та входу у вершині як кількість дуг, які відповідно виходять та входять у дану вершину.

Граф повністю визначається його суміжностями або інциденціями. Цю інформацію про граф зручно подавати в *матричній* формі. Для певного графа можна скласти кілька матриць, у тому числі матрицю суміжностей, матрицю інциденцій та список ребер. Часто їх використовуються при з'ясуванні окремих властивостей графа.

Відмінність матриці інцидентності орієнтованого графа від неорієнтованого складається у вказівці початку і кінця ребер. Матриця суміжності губить свою симетричність. У списку ребер важливий порядок вказівки вершин, що з'єднують зазначеним ребром (від початку до кінця).

Як відзначалося вище, всі розглянуті способи завдання графів однозначно визначають граф. Виникає питання: чи можливо відновити граф по заданих матрицях інцидентності, суміжності або списку ребер? Очевидна позитивна відповідь.

По матриці інцидентності число ребер і вершин визначається з розмірності матриці: число ребер $|E|$ графа дорівнює числу стовпців m, а число вершин $|V|$ - числу рядків n матриці.

По матриці суміжності число вершин визначається з розмірності матриці. Число ребер графа дорівнює сумі елементів, розташованих на головній діагоналі і у верхньому правому трикутнику. Число ребер орграфа дорівнює сумі всіх елементів матриці суміжності.

Список ребер є скороченим варіантом матриці інцидентності. Кількість ребер очевидна, а кількість вершин дорівнює максимальному номеру всіх перерахованих вершин зі списку.

Знаючи матрицю інцидентності можна записати список ребер, і навпаки.

Побудова матриці інцидентності за списком ребер. Кожен рядок списку ребер відповідає рядку в матриці инцидентності з тим же номером. Для неорієнтованого графа в кожному рядку списку ребер зазначені номери елементів матриці інцидентності, рівні 1 (всі інші елементи – нулі). Для орієнтованого графа першою вказується вершина, що відповідає початку ребра (у матриці інцидентності – елемент -1), а другий – відповідному кінцю ребра (у матриці інцидентності – елемент 1). При збігу елементів у рядку списку ребер, у відповідному рядку матриці інцидентності записується число, відмінне від -1, 0, 1, наприклад, 2. Така ситуація відповідає наявності у графі петель.

Приклади розв'язування задач

Приклад 2.1. Визначити степені вершин графа, що зображено на рис. 2.1.

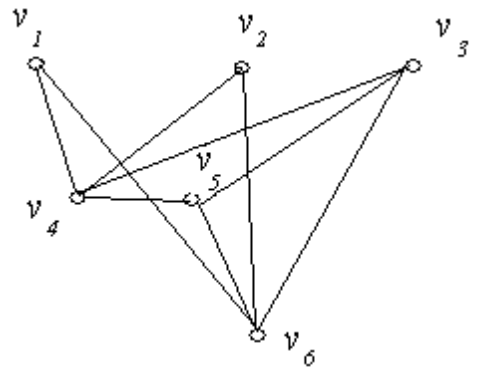


Рис. 2.1.

Розв'язування:

$deg\, v_1 = 2$; $deg\, v_2 = 2$; $deg\, v_3 = 3$; $deg\, v_4 = 4$; $deg\, v_5 = 3$; $deg\, v_6 = 4$.

$\sum_{v \in G} deg\, v = 2+2+3+4+3+4 = 18 = 2 \cdot 9 = 2m$.

У розглянутому графі дев'ять ребер, а вершин непарного степеня дві: v_3; v_5.

Для орієнтованого графа визначаються дві степені вершин: $deg\, v'$ - кількість ребер, що виходять із вершини v і $deg\, v''$ - кількість ребер, що входять у вершину v. Петля дає внесок по одиниці в обидві степені.

В орграфі суми степенів всіх вершин $deg\, v'$ і $deg\, v''$ рівні між собою і дорівнюють кількості ребер m цього графа: $\sum_{v \in G} deg\, v' = \sum_{v \in G} deg\, v'' = m$.

Приклад 2.2. Визначити степені вершин орграфа, зображеного на рис. 2.2.

Розв'язування:

$deg\, a' = 1$, $deg\, b' = 2$; $deg\, c' = 3$; $deg\, d' = 2$; $deg\, e' = 2$;

$deg\, a'' = 2$, $deg\, b'' = 1$; $deg\, c'' = 2$; $deg\, d'' = 2$; $deg\, e'' = 3$;

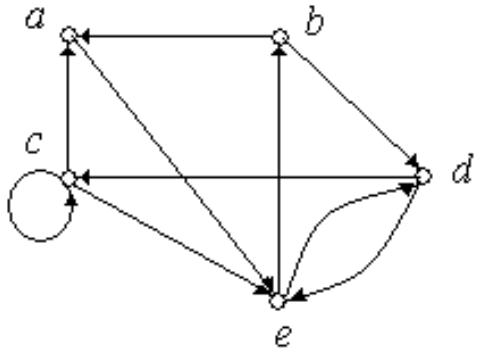


Рис. 2.2.

$\sum_{v \in G} deg\, v' = 1+2+3+2+2 = 10 = \sum_{v \in G} deg\, v'' == 2+1+2+2+3 = 10 = m$.

Приклад 2.3. Задати матрицями інцидентності і суміжності, а також списком ребер, неорієнтований граф, зображений на рис. 2.3.

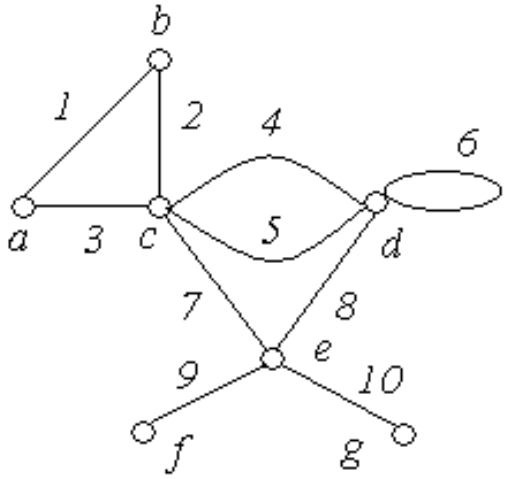


Рис. 2.3.

Розв'язування:

Матриця інцидентності

	1	2	3	4	5	6	7	8	9	10
a	1	0	1	0	0	0	0	0	0	0
b	1	1	0	0	0	0	0	0	0	0
c	0	1	1	1	1	0	1	0	0	0
d	0	0	0	1	1	2	0	1	0	0
e	0	0	0	0	0	0	1	1	1	1
f	0	0	0	0	0	0	0	0	1	0
g	0	0	0	0	0	0	0	0	0	1

Матриця суміжності

	a	*b*	*c*	*d*	*e*	*f*	*g*
a	0	1	1	0	0	0	0
b	1	0	1	0	0	0	0
c	1	1	0	2	1	0	0
d	0	0	2	1	1	0	0
e	0	0	1	1	0	1	1
f	0	0	0	0	1	0	0
g	0	0	0	0	1	0	0

Тут $\alpha = 2$.

Список ребер

Ребро		1	2	3	4	5	6	7	8	9	10
Вершини	Початок	*a*	*b*	*a*	*c*	*c*	*d*	*c*	*d*	*e*	*e*
	Кінець	*b*	*c*	*c*	*d*	*d*	*d*	*e*	*e*	*f*	*g*

Як бачимо, у кожному стовпці матриці інцидентності є тільки два елементи, відмінних від нуля (або один, якщо ребро – петля).

Матриця суміжності симетрична щодо головної діагоналі.

Список ребер є найбільш компактним способом завдання графів.

Кожний із наданих способів однозначно описує граф, що зображено на рис. 2.3.

Приклад 2.4. Задати матрицями інцидентності, суміжності, списком ребер орієнтований граф, що зображено на рис.2.4.

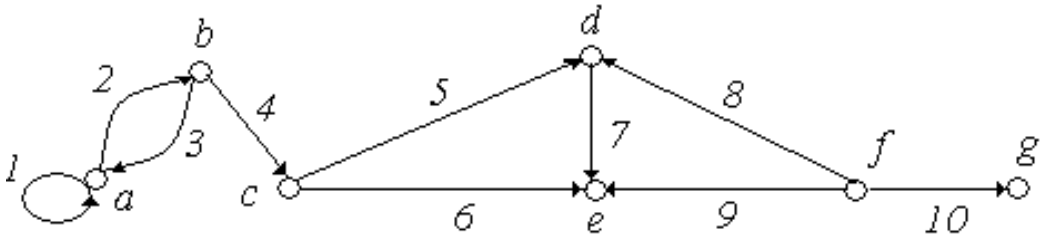


Рис. 2.4.

Розв'язування:

Матриця інцидентності

	1	2	3	4	5	6	7	8	9	10
a	2	−1	1	0	0	0	0	0	0	0
b	0	1	−1	−1	0	0	0	0	0	0
c	0	0	0	1	−1	−1	0	0	0	0
d	0	0	0	0	1	0	−1	1	0	0
e	0	0	0	0	0	1	1	0	1	0
f	0	0	0	0	0	0	0	−1	−1	−1
g	0	0	0	0	0	0	0	0	0	1

Матриця суміжності

	a	*b*	*c*	*d*	*e*	*f*	*g*
a	1	1	0	0	0	0	0
b	1	0	1	0	0	0	0
c	0	0	0	1	1	0	0
d	0	0	0	0	1	0	0
e	0	0	0	0	0	0	0
f	0	0	0	1	1	0	1
g	0	0	0	0	0	0	0

Список ребер

Ребро		1	2	3	4	5	6	7	8	9	10
Вершини	початок	*a*	*a*	*b*	*b*	*c*	*c*	*d*	*f*	*f*	*f*
	кінець	*a*	*b*	*a*	*c*	*d*	*e*	*e*	*d*	*e*	*g*

Приклад 2.5. Побудувати матрицю інцидентності неорієнтованого графа за списком ребер:

Ребро		1	2	3	4	5	6	7	8
Вершини	початок	*a*	*d*	*e*	*c*	*a*	*d*	*b*	*f*
	кінець	*d*	*f*	*f*	*e*	*b*	*c*	*f*	*f*

Розв'язування:

Матриця інцидентності, відповідно до списку ребер, має вигляд:

	1	2	3	4	5	6	7	8
a	1	0	0	0	1	0	0	0
b	0	0	0	0	1	0	1	0
c	0	0	0	1	0	1	0	0
d	1	1	0	0	0	1	0	0
e	0	0	1	1	0	0	0	0
f	0	1	1	0	0	0	1	2

Приклад 2.6. Записати список ребер відповідно до матриці інцидентності орієнтованого графа:

	1	2	3	4	5	6	7	8
a	1	0	0	0	0	0	0	-1
b	-1	-1	0	0	0	0	0	0
c	0	1	2	-1	0	0	0	0
d	0	0	0	1	2	-1	0	0
e	0	0	0	0	0	1	-1	0
f	0	0	0	0	0	0	1	1

Розв'язування:

Список ребер, записаний відповідно до матриці інцидентності орієнтованого графа має вигляд:

Ребро		1	2	3	4	5	6	7	8
Вершини	початок	*b*	*b*	*c*	*C*	*d*	*d*	*e*	*a*
	кінець	*a*	*c*	*c*	*D*	*d*	*e*	*f*	*f*

Завдання для виконання

2.1. Нехай задано граф G(X, W) :

а) $X = \{1,2,3,4\}$, $W = \{(1,3),(2,3),(3,4),(4,1),(4,2)\}$;

б) $X = \{a,b,c,d,e\}$, $W = \{(a,d),(b,c),(b,e),(c,e),(d,b),(d,e),(e,a)$;

в) $X = \{1,2,3\}$, $W = \{(1,2),(1,3),(2,3)\}$;

г) $X = \{A, B, C, D\}$, $W = \{(A, B), (A, D), (B, C), (B, D), (C, A), (C, D)\}$.

Побудувати діаграму, матриці суміжності та інцидентності для кожного із заданих графів.

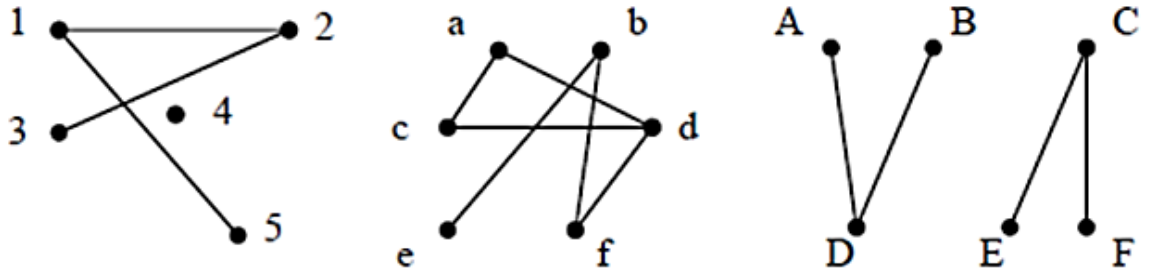
2.2. Нехай $X = \{a, b, c, d, e\}$. Граф G(X, W) задано за допомогою матриці суміжності A. Визначити множину ребер W графа G.

а) $$A = \begin{vmatrix} 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \end{vmatrix}$$

б) $$A = \begin{vmatrix} 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 \end{vmatrix}$$

в) $A = \begin{vmatrix} 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \end{vmatrix}$ г) $A = \begin{vmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{vmatrix}$

2.3. Граф G задано його діаграмою (рис. 2.5). Визначити множину вершин X і множину ребер матриці суміжності та інцидентності графа G.



а) б) в)

Рис. 2.5.

2.4. Нарисувати діаграму повного графа з n вершинами K_n для:

а) $n = 2$; б) $n = 3$; в) $n = 4$; г) $n = 5$.

2.5. Чому дорівнює степінь кожної вершини у повному графі з n вершинами?

2.6. Скільки ребер містить повний граф з n вершинами?

2.7. Довести, що доповненням графа $\overline{G}$ є граф G.

2.8. Чому дорівнює степінь вершини x у графі $\overline{G}$, якщо в графі G з n вершинами:

а) $d(x) = 1$; б) $d(x) = n - 1$; в) $d(x) = 0$; г) $d(x) = k$?

2.9. Чому дорівнює кількість ребер у графі $\overline{G}$, якщо граф G має n вершин і k ребер?

2.10. Довести, що для довільного графа G об'єднання $G \cup \overline{G}$ є повним графом.

2.11. Нехай графи $G_1(X, W_1)$ і $G_2(X, W_2)$ задано за допомогою матриць суміжності A_1 і A_2 відповідно ($|X| = n$). Визначити матрицю суміжності A для графа:

а) $G1 \cup G2$; б) $G1 \cap G2$; в) $G_1 \setminus G_2$; г) $\overline{G_1}$

2.12. Нехай задано матрицю суміжності A деякого графа G. Як за допомогою матриці A визначити:

а) кількість вершин графа G;

б) кількість ребер графа G;

в) степінь $d(x)$ певної вершини x графа G;

г) чи є граф G повним графом;

ґ) матрицю інцидентності графа?

Практична робота №3. Обходи графів. Ейлерові та гамільтонові графи

Теоретичні відомості

Для визначення центра і радіуса графа необхідно побудувати для нього *матрицю відстаней* A, кожен елемент якої a_{ij} описує відстань між вершинами i і j графа G, тобто $a_{ij} = d(v_i, v_j)$. Очевидно, що матриця відстаней A симетрична щодо головної діагоналі (елементи якої дорівнюють нулю, тому що $d(v_i, v_i) = 0$).

Центром графа називається вершина, від якої максимальна з відстаней до інших вершин була б мінімальною.

Периферійною точкою графа називається вершина, від якої максимальна з відстаней до інших вершин була б максимальна.

Максимальна відстань від центра графа G до його вершин називається *радіусом графа* $r(G)$.

Зв'язний граф називається *гамільтоновим графом*, якщо існує замкнений ланцюг, який проходить через кожну вершину графа рівно один раз. Такий ланцюг називатимемо *гамільтоновим ланцюгом*, або *гамільтоновим циклом.*

Теорема. Якщо граф G має ребро розрізу, то він не може мати цикл Гамільтона. Якщо компоненти графа, отримані шляхом видалення ребра розрізу, мають цикл Гамільтона, то граф G має шлях Гамільтона.

Теорема (О. Оре, 1960) Якщо $G(X,W)$ - зв'язний граф з n вершинами $(n \geq 3)$ і для кожної пари несуміжних вершин u і v, сума степенів вершин задовольняє умові $\deg u + \deg v \geq n$, тоді граф $G(X,W)$ гамільтонів.

Теорема (Г. Дірак, 1952 р.) Якщо $G(X,W)$ - зв'язний граф з n вершинами $(n \geq 3)$ і для кожної пари несуміжних вершин u і v, виконується нерівність $\deg v \geq \frac{n}{2}$, то граф $G(X,W)$ гамільтонів.

Приклад 3.1. Визначити центр, периферійні вершини, радіус і діаметр графа G, зображеного на рис. 3.1.

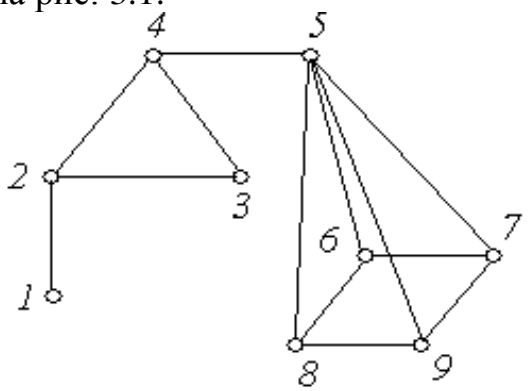


Рис. 3.1.

Розв'язування:

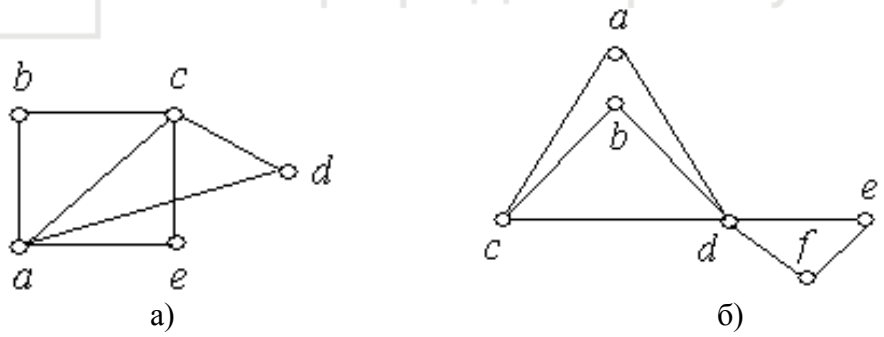
Матриця відстаней графа G має вигляд.

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
1	0	1	1	2	3	4	4	4	4
2	1	0	1	1	2	3	3	3	3
3	1	1	0	1	2	3	3	3	3
4	2	1	1	0	1	2	2	2	2
5	3	2	2	1	0	1	1	1	1
6	4	3	3	2	1	0	1	1	2
7	4	3	3	2	1	1	0	1	1
8	4	3	3	2	1	1	1	0	1
9	4	3	3	2	1	2	1	1	0

Знайдемо максимальну відстань від кожної з вершин графа $l\ v_i$ як $\max\limits_{1\le j\le 9} a_{ij}$: $l\ 1 = 4; l\ 2 = 3; l\ 3 = 3; l\ 4 = 2; l\ 5 = 3; l\ 6 = 4; l\ 7 = 4; l\ 8 = 4;$ $l\ 9 = 4$.

Отже, згідно з визначеннями, що подано вище, центром графа є вершина 4; периферійні вершини – 1, 6, 7, 8, 9. Радіус графа $r\ G = 2$, а діаметр графа $D\ G = 4$.

Приклад 3.2. Чи мають графи, зображені на рис. 3.2 (а, б), ейлеров цикл?



а) б)

Рис. 3.2.

Розв'язування:

Щоб відповістити на поставлене запитання, порахуємо степені вершин графа:

а) $deg\ a = 4; deg\ b = 2; deg\ c = 4; deg\ d = 2; deg\ e = 2$.

Степені всіх вершин графа, зображеного на рис. 3.2 а), парні, отже, граф має ейлеров цикл;

б) $deg\ a = 2; deg\ b = 2; deg\ c = 3; deg\ d = 5; deg\ e = 2; deg\ f = 2$.

Степені вершин c і d графа, що зображено на рис. 3.2 б), непарні, отже, граф не має ейлерового циклу.

Приклад 3.3. Знайдіть цикл Гамільтона, якщо він існує, для графа G, зображеного на рис. 3.3,а.

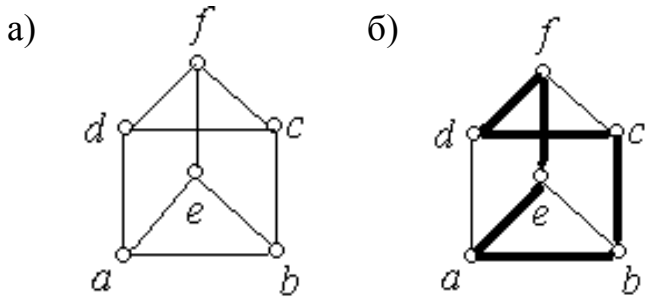


Рис. 3.3.

Розв'язування:

Граф G - зв'язний, кількість вершин графа – $n = 6$. Степінь кожної з вершин дорівнює 3, тобто кожна з несуміжних вершин графа задовольняє умові $\deg v \geq \frac{n}{2}$. Отже, даний граф є гамільтонів, тобто існує цикл Гамільтона. Знайти його можемо тільки методом перебирання. Результати прямого перебирання – цикл $baefdcb$ (рис. 3.3,б).

Завдання для виконання

3.1. Визначити центр, периферійні точки, радіус, діаметр графа G, зображеного на рис. 3.4 (а,б):

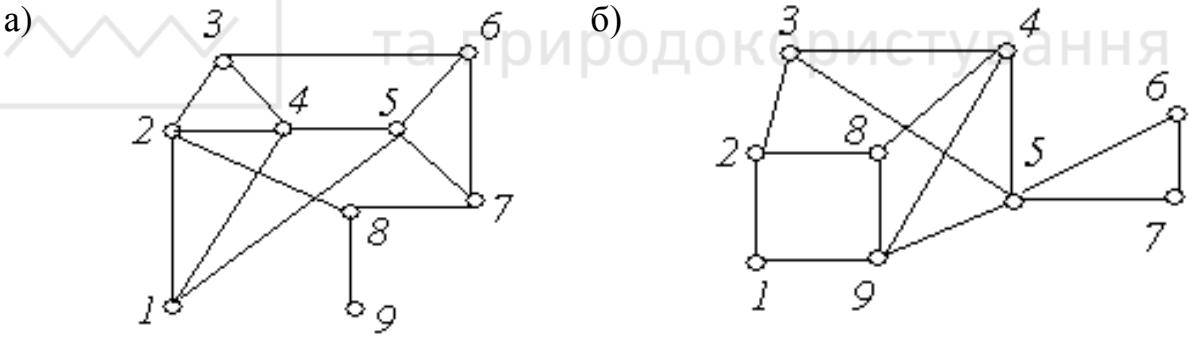
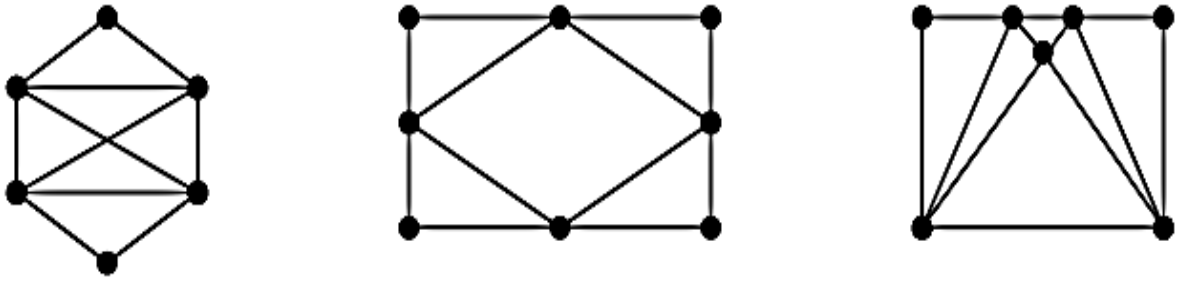


Рис. 3.4.

3.2. Переконатись у тому, що зображені графи ейлерові.



3.3. Серед графів, що наведені на рис. 3.5 (а-г), знайти ті, що мають ейлеров цикл. Обґрунтувати відповідь.

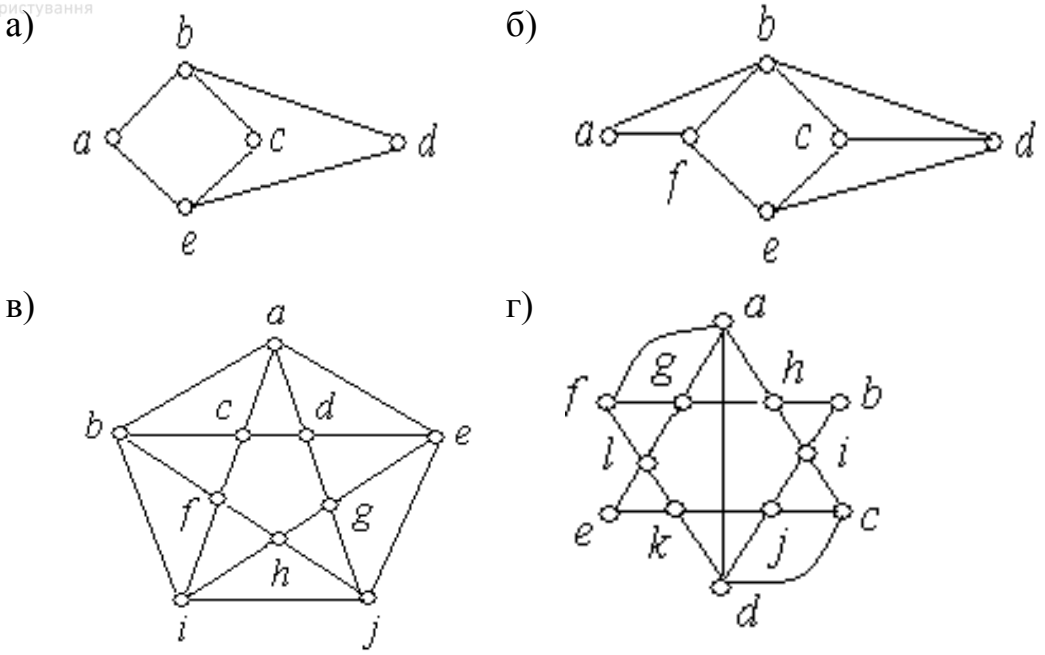


Рис. 3.5.

3.4. На рис. 3.6 (а,б) зображено схеми музеїв, у яких вершинами є зали музеїв, а ребрами – переходи між ними. Визначити, з якого залу потрібно розпочати екскурсію і в якому завершити для того, щоб провести відвідувачів по всіх залах, пройшовши по кожному з переходів один раз. Знайти один з таких маршрутів.

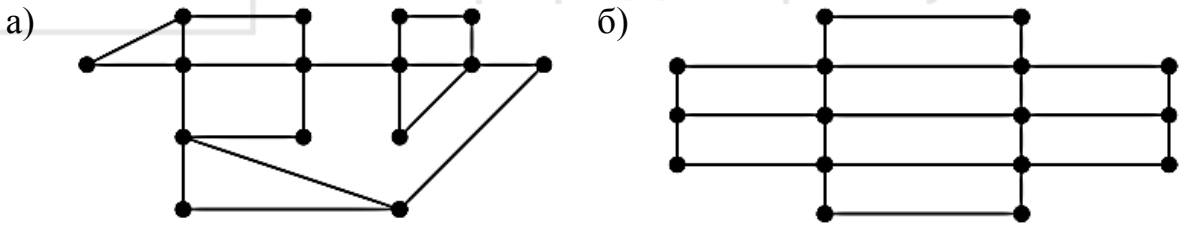


Рис. 3.6.

3.5. На рис. 3.7 зображено лабіринт. Знайти у ньому такий маршрут, щоб, розпочавши шлях з якоїсь кімнати, пройти у ньому по одному разу через усі двері й повернутись у початкову точку.

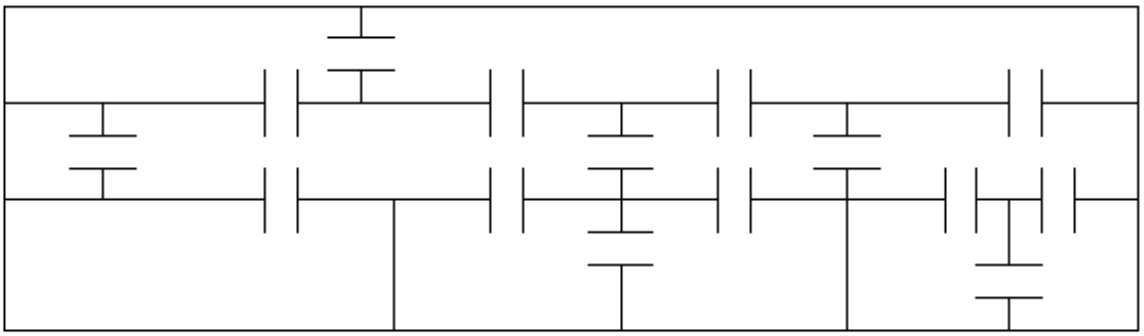


Рис. 3.7.

3.6. Знайти гамільтонові цикли в графах, що зображено на рис. 3.8.

а) б)

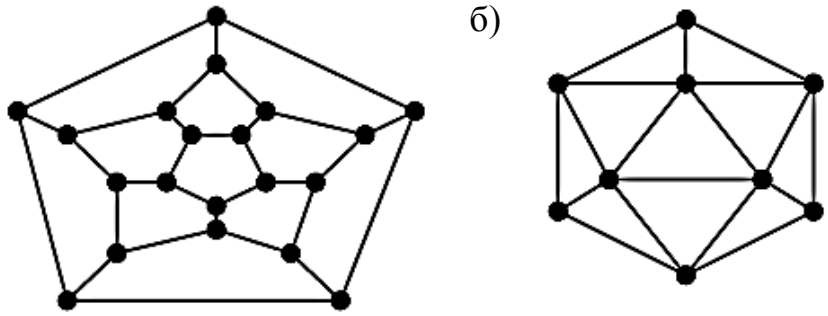


Рис. 3.8.

3.7. Знайдіть цикл Гамільтона, якщо він існує, для кожного з графів наведених на рис. 3.9 (а-ж).

а) б)

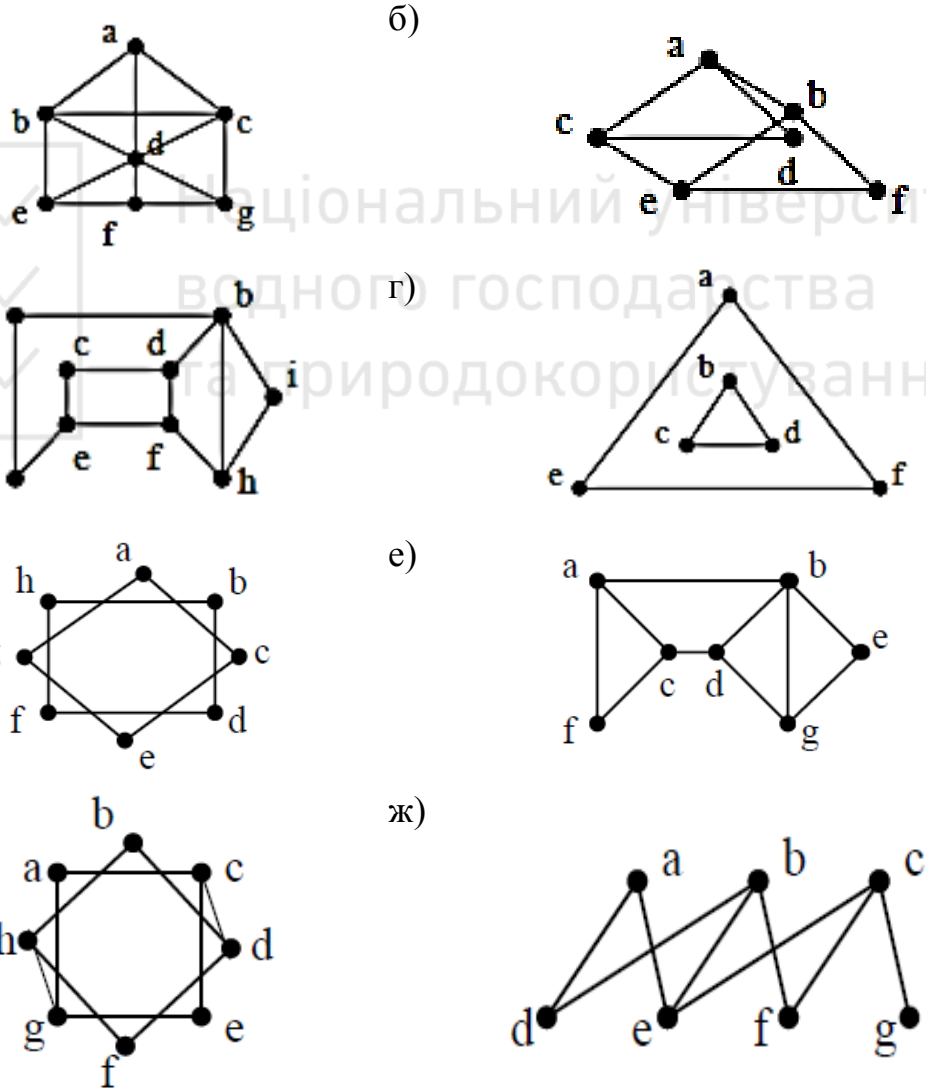


Рис. 3.9.

3.8. Знайти гамільтонів шлях (якщо він існує) в кожному з графів (рис.3.10 а-г):

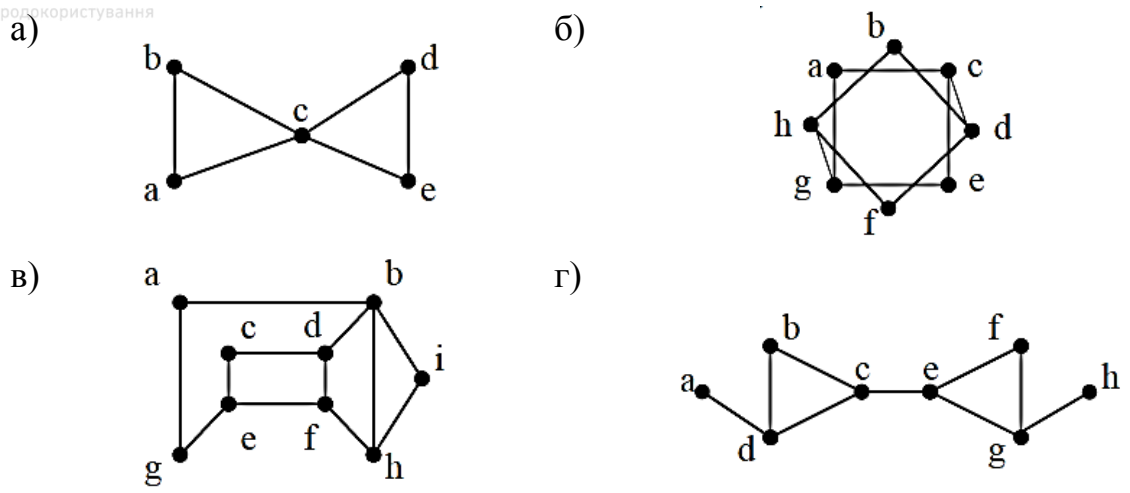
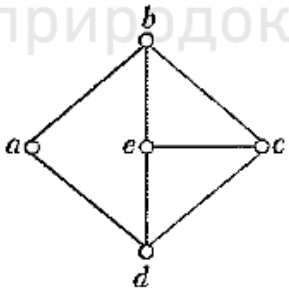


Рис. 3.10.

Практична робота №4. Алгоритми розфарбування графа

Процес розфарбовування графа (вершин або ребер) можна здійснити на основі алгоритму бектрекінгу (або пошуку з поверненням). Проілюструємо алгоритм на прикладі.

Приклад 4.1. Знайти гамільтонів цикл у заданому графі.



Розв’язування:

Починаємо з довільної вершини. Будуємо шлях без повторення вершин, доки це можливо. Якщо вдалося пройти всі вершини, то перевіряємо, чи існує ребро, що з'єднує останню й початкову вершини цього шляху. Якщо описаний процес у певний момент неможливо продовжити, то повертаємося на одну вершину назад і намагаємося продовжити побудову шляху (без повторення вершин) іншим способом.

Пошук усіх гамільтонових циклів у заданому графі проілюстровано за допомогою дерева, зображеного на рис. 4.1. Роботу алгоритму почато з одноелементної послідовності, бо в циклі вибір першої вершини неістотний. Розв'язки задачі обведено, і до них у прямокутних рамках приписано відповідні гамільтонові цикли.

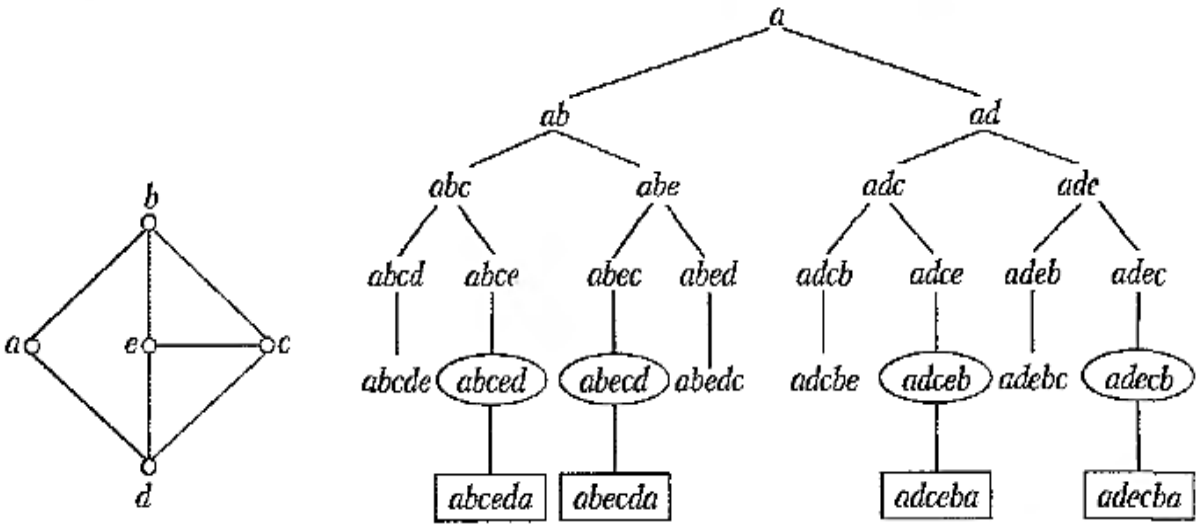
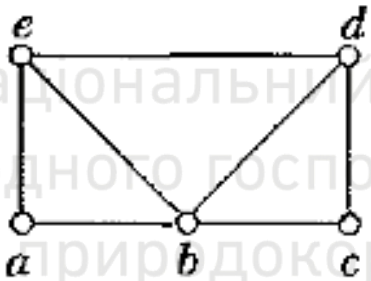


Рис. 4.1.

Приклад 4.2. Розфарбувати граф в 3 кольори.



Розв'язування.

Спочатку розфарбуємо вершину *a* в колір 1, а потім — вершину *b* в той самий колір, якщо *b* не суміжна з вершиною *a*, а ні, то розфарбуємо вершину *b* в колір 2. Перейдемо до третьої вершини *c*. Використаємо для вершини *c* колір 1, якщо це можливо, а ні, то колір 2, якщо це можливо. Тільки якщо жоден із кольорів 1 і 2 не можна використовувати, розфарбуємо вершину *c* в колір 3. Продовжимо цей процес, доки це можливо, використовуючи один з *n* кольорів для кожної повної вершини, причому завжди братимемо перший можливий колір зі списку кольорів. Досягнувши вершини, яку не можна розфарбувати в жоден з *n* кольорів, повертаємося до останньої розфарбованої вершини, відміняємо її колір і присвоюємо наступний можливий колір зі списку. Якщо й це неможливо, то ще раз повертаємося до попередньої вершини, відміняємо її колір і намагаємося присвоїти новий колір, наступний можливий зі списку. Цей процес продовжуємо аналогічно.

На рис. 4.2 зображено граф і процес присвоювання трьох кольорів вершинам із використанням алгоритму бектрекінг.

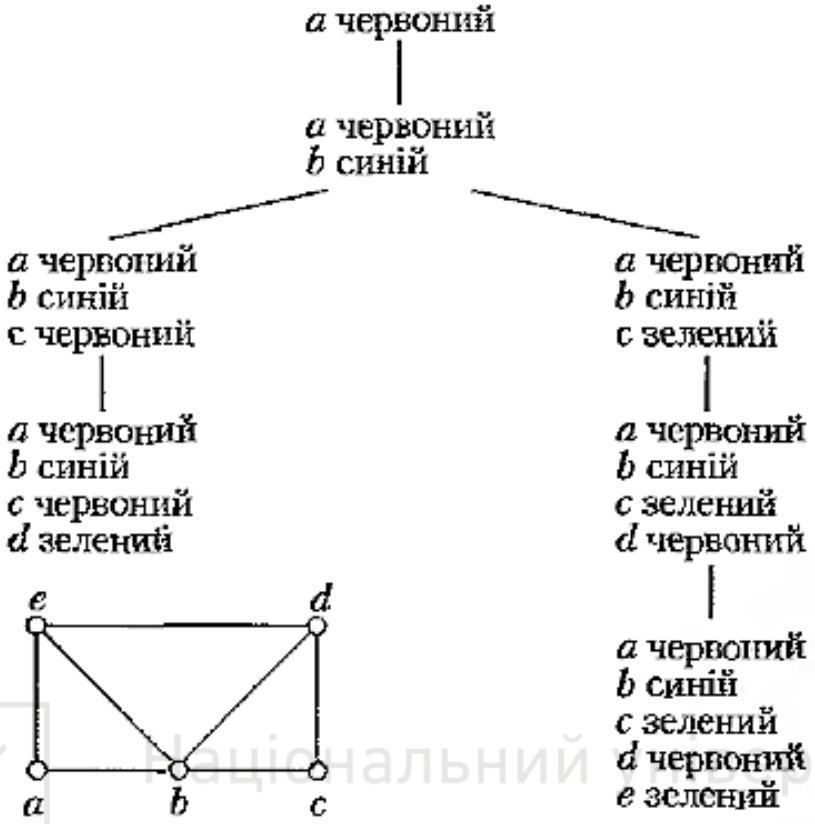


Рис. 4.2.

Завдання для виконання

4.1. Знайти цикл Гамільтона, якщо він існує, для кожного з графів, що наведено у завданні 3.7 за допомогою бектрекінгу.

4.2. Розфарбувати вершини кожного з графів, що наведено у завданні 3.7, в 3 кольори за допомогою бектрекінгу.

Практична робота №5. Побудова покриваючих дерев

Теоретичні відомості

Мережею називається граф, кожному ребру або дузі якого поставлено у відповідність деяке число, що називається *вагою* ребра або дуги і відбиває певні властивості цього ребра чи дуги (наприклад, відстань, швидкість, ціну, пропускну здатність, дохід, прибуток).

Вагою дерева називається сума ваг усіх його ребер. Дерево називається мінімальним (максимальним), якщо воно має мінімальну (максимальну) з можливих ваг.

Покривним деревом називається будь-яке дерево, що містить усі вершини графа.

Орієнтоване дерево – дерево, у якому жодні дві дуги не заходять у ту саму вершину.

Коренем дерева називається єдина його вершина, в яку не заходить жодна дуга. Поняття кореня для неорієнтованих дерев немає. Коренем для неорієнтованих дерев може слугувати будь-яка вершина.

Орієнтований ліс – це ліс, що складається з орієнтованих дерев.

Покривним орієнтованим деревом називається орієнтоване дерево, що є одночасно і покривним деревом.

Покривним орієнтованим лісом називається орієнтований ліс, що містить усі вершини графа.

Завдання для виконання

5.1. Багато країн Європейського Союзу мають спільні кордони. Побудувати граф, вершини якого відповідають різним країнам ЄС. Дві вершини слід з'єднати дугою, якщо відповідні країни мають спільний кордон. Дати відповідь на запитання:

- чи є отриманий граф зв'язним?
- скільки дерев містить граф?

5.2. Чи є в ЄС країна, виключення якої зі співтовариства розірвало б усі наземні комунікації між країнами, що залишилися?

5.3. Чи є граф, що зображено на рис.5.1, мережею? Якщо ні, то чим його потрібно і можна доповнити, щоб вийшла мережа?

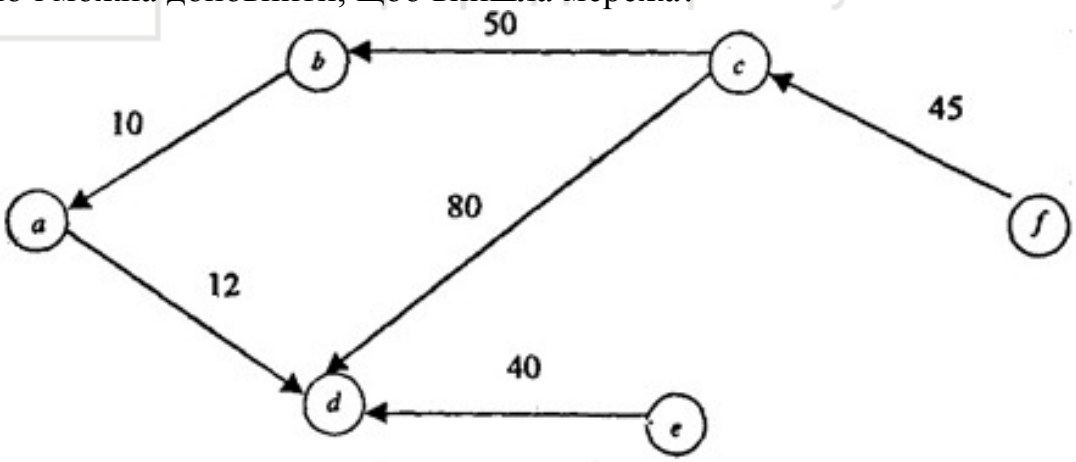


Рис. 5.1.

5.4. Побудувати для графа, що зображено на рис.5.1, покривне дерево. Скільки таких дерев можна побудувати?

5.5. Для графів, що зображено на рис.5.2 а-г, побудувати максимальне та мінімальне покриваючі дерева

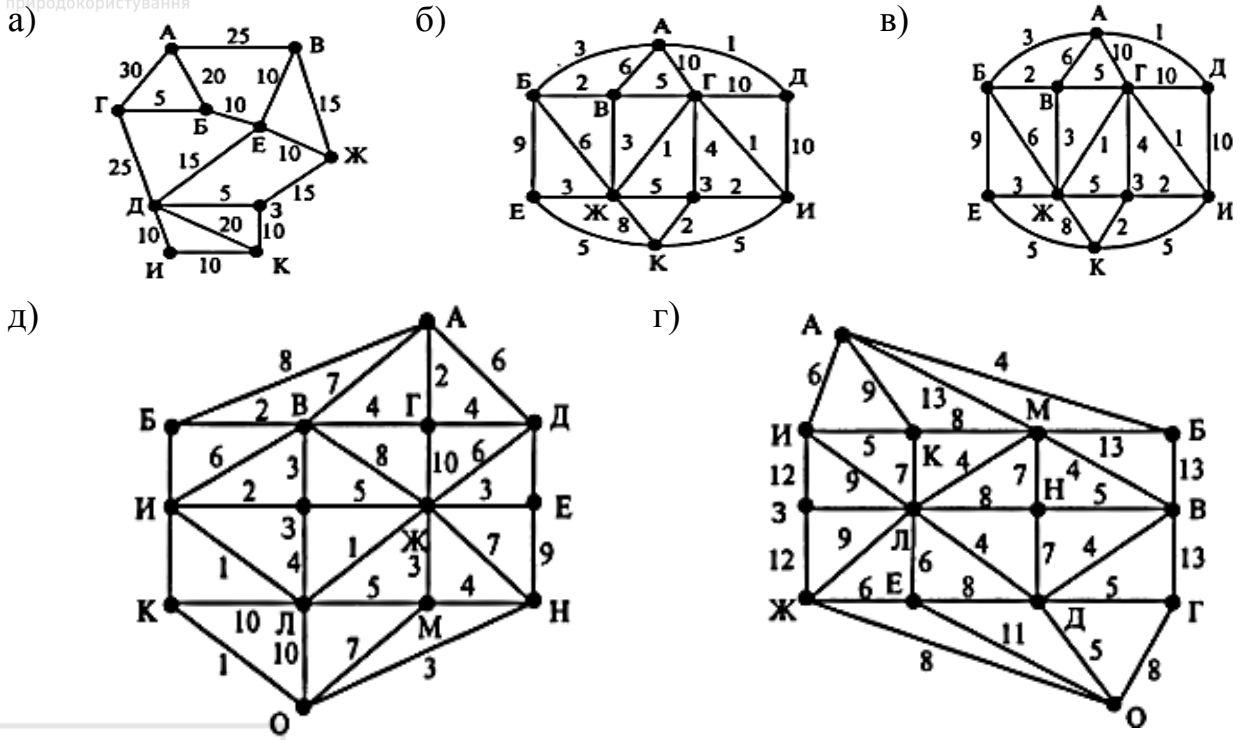


Рис. 5.2.

Практична робота №6. Алгоритми побудови покривного дерева для графа

Теоретичні відомості

Розглянемо найпростіші алгоритми побудови покривних дерев.

Алгоритм побудови покривного дерева для графа. Суть алгоритму полягає у покроковому перегляді в довільному порядку всіх ребер графа. Розглянемо його реалізацію як процес фарбування ребер. Ребро включається в дерево в тому випадку, якщо воно розфарбовується. На кожному кроці перевіряється наявність циклів і не допускається їх присутність у дереві, яке будується. На цій підставі приймається рішення, буде чи не буде ребро включене в дерево.

Зв’язний компонент (або сукупність) ребер при реалізації алгоритму називається *букетом*.

Так зване тіло алгоритму складається з таких кроків. Перед початком роботи алгоритму забираємо всі петлі. Вони в побудові дерева брати участі не будуть.

Крок 1. Вибрати будь-яке ребро, зафарбувати його і почати побудову дерева.

Крок 2. Вибрати будь-яке незафарбоване ребро. Якщо такого ребра немає, закінчити процедуру – вихідний граф не містить покривного дерева. Якщо є, можливі чотири випадки:

а) обидві вершини обраного ребра належать тому самому букету. У цьому випадку ребро не зафарбовується (тому що ми вийшли на цикл). Повернутися до початку кроку 2;

б) одна з вершин ребра належить деякому букету, а інша не належить жодному з букетів. Ребро зафарбувати і включити в букет, якому належить кінцева вершина;

в) жодна з вершин не належить жодному з наявних букетів. Сформувати новий букет і зафарбувати ребро;

г) кінцеві вершини обраного ребра належать різним букетам. Ребро зафарбувати, а обидва букети злити в один.

Крок 3. Якщо усі вершини ввійшли в один букет, закінчити процедуру. Якщо ні, повернутися до кроку 2.

Робота алгоритму закінчується, коли число зафарбованих ребер стає рівним числу, на одиницю меншому від числа вершин, або, що те саме, коли всі вершини зібрані в один букет.

Якщо ж граф не містить покривного дерева, тобто є незв'язним, алгоритм припиняє роботу після зафарбування всіх ребер графа. У цьому випадку робота алгоритму закінчується побудовою покривного лісу.

Алгоритм побудови мінімального (максимального) покривного дерева для мережі. У комплексі оптимізаційних задач частіше розглядаються не графи, а мережі, тобто ті ж графи, кожному ребру чи дузі яких поставлена у відповідність вага.

Алгоритм побудови мінімального (максимального) покривного дерева дуже простий. Він являє собою алгоритм побудови звичайного покривного дерева, у якому ребра передивляються в порядку спадання (зростання) їхніх ваг. Якщо кілька ребер мають однакові ваги, вони упорядковуються довільно.

Зазначимо, що як тільки здійснено формалізацію задачі у вигляді мережі і сформульовано вимоги для її оптимізації, зміст вихідної задачі, по суті, не має значення для самого процесу розв'язання. Зміст вихідної задачі знову буде потрібен для інтерпретації вже отриманого зв'язку.

Мінімальне (максимальне) дерево можна будувати і для тих мереж, у яких певні зв'язки (ребра) вважаються заздалегідь включеними в покривне дерево незалежно від їхніх ваг і побудова починається з мінімального (максимального) з ребер, що залишились.

Завдання для виконання

6.1. Аналіз мережі комунікації. Є мережа доріг, що зв'язує кожен населений пункт із деяким іншим (рис.6.1). Потрібно визначити, чи можна, користуючись цими дорогами, проїхати з кожного населеного пункту в будь-який інший.

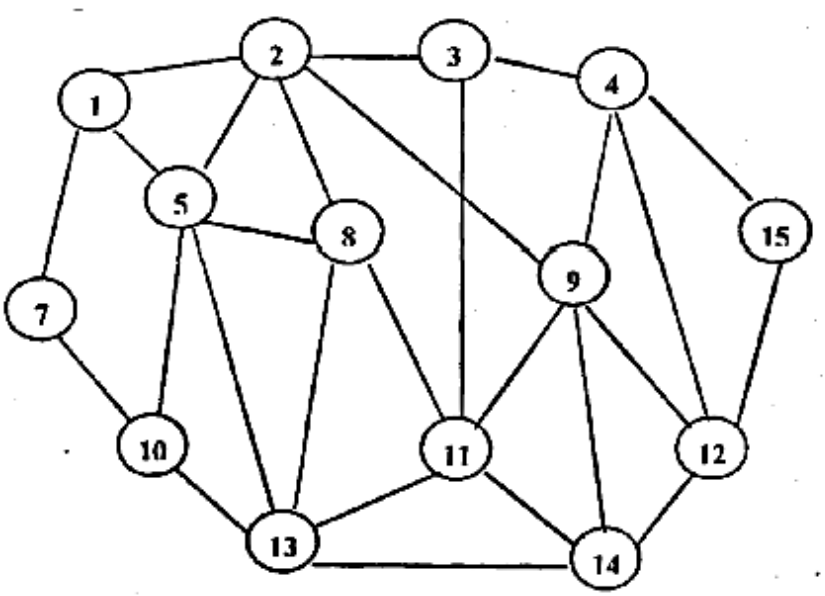


Рис. 6.1.

6.2. Проект реконструкції (будівництва) мережі комунікацій. Для мережі, зображеної на рис.6.2, побудувати мінімальне і максимальне покривні дерева.

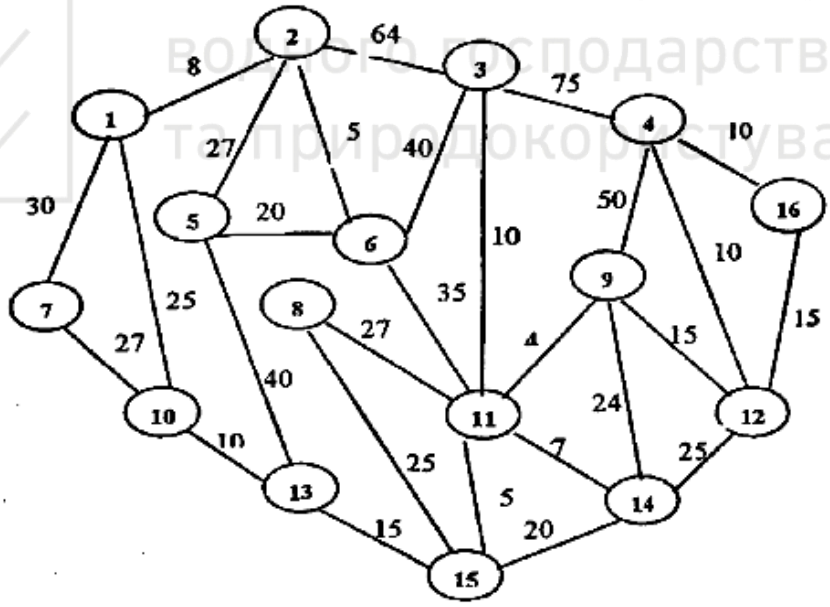


Рис. 6.2.

6.3. Зафіксувати які-небудь зв’язки (рис.6.2) і побудувати мінімальне і максимальне дерева з фіксованими зв’язками. Порівняти результати розв’язання цієї і попередньої задач.

6.4. На рис. 6.3 фіксовані зв’язки зображено жирними лініями. Побудувати для цієї мережі мінімальне і максимальне покривні дерева.

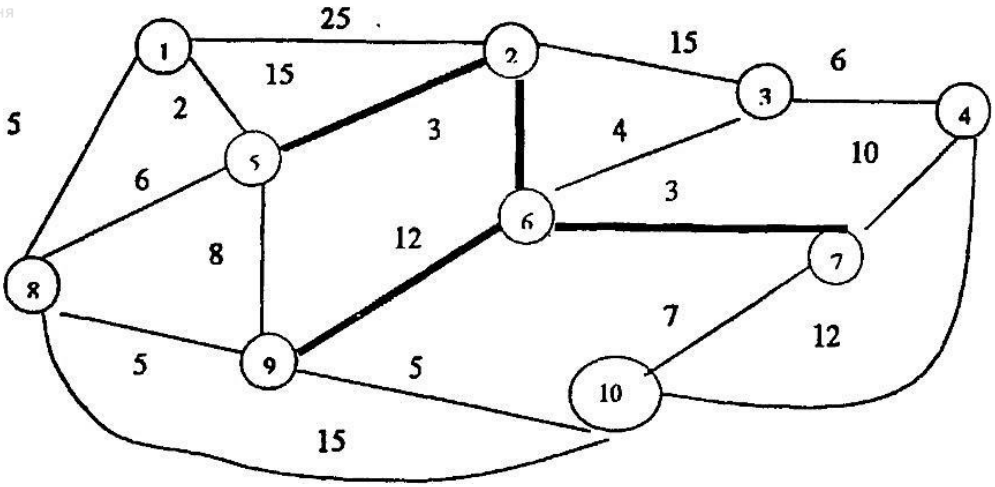


Рис. 6.3.

Практична робота № 7. Розв'язання задачі про лідерство

Є колектив людей, спрямований на виконання певного завдання. Потрібно на основі системи переваг членів колективу вибрати серед них лідера.

Теоретичні відомості

Для розв'язування такої задачі слід побудувати мережу з вершинами, що відповідають кожному члену колективу. Дугами відобразити бажання кожного члена колективу бачити своїм лідером того чи іншого члена. Як ваги дуг узяти кількісну оцінку цього бажання (бальну чи яку-небудь іншу). Отже, розв'язування задачі зведеться до побудови максимального орієнтованого лісу для побудованої мережі. Корені виявлених дерев відповідатимуть ймовірним лідерам.

Для розв'язання задачі використовуємо алгоритм побудови максимального покривного лісу. З цією метою побудуємо мережу. Кожному члену колективу поставимо у відповідність вершину мережі. Зв'язки між вершинами відображатимемо дугами, що відбивають авторитетність кожної людини в очах інших людей чи вплив одних людей на інших відповідно до їхніх оцінок власних уподобань (або наприклад, оцінку якої-небудь доцільності для кожного члена колективу, визначену навіть не самими членами, а кимось ззовні). Як ваги дуг слід взяти ці оцінки.

При цьому будемо виходити з таких базових положень:

- кінцева мета або задача відомі всім членам колективу;
- усі члени колективу зацікавлені в найкращому вирішенні поставленого завдання чи досягненні поставленої мети;
- у складі колективу є особистості, здатні привести його до мети;
- усі члени колективу можуть сформувати свої переваги.

Приклад 7.1. Розглянемо задачу вибору лідерів у колективі з 20 осіб при організації на його основі працездатного підрозділу відповідно до особистого бачення членами власних лідерів. Метою було оголошене створення структури управління колективом для виконання великого проекту. Таким чином, задача зводиться до вибору керівника всього колективу і створення кількох мобільних груп зі своїми внутрішніми керівниками для розв'язання проміжних задач.

Для зручності членам колективу присвоїмо номери. Для спрощення міркувань (і тільки для цього), дозволимо кожній людині віддати свою перевагу тільки двом особам. У принципі, можна запропонувати і себе. Оцінювання в даному випадку пропонується за десятибальною системою, 10 – найвища оцінка. Якщо, скажімо, член колективу під номером 17 згоден бачити лідерами осіб під номерами 5 і 12 і ступінь своєї переваги визначає оцінками 10 і 8 відповідно, то це відображатиметься конструкцією на рис. 7.1.

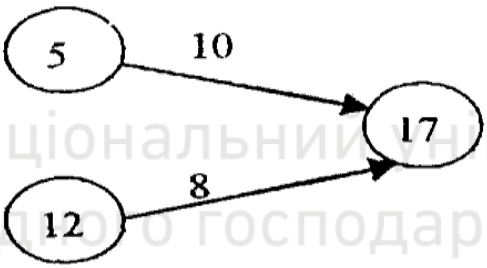


Рис. 7.1. Відображення на мережі переваг

Мережу, отриману в результаті цієї роботи, зображено на рис. 7.2. Поруч з кожною дугою мережі поставлено її вагу. Оскільки рисунок малий і ідентифікація ваг і дуг може бути складною, наводиться таблиця переваг (ваг дуг) кожного члена колективу (табл.7.1).

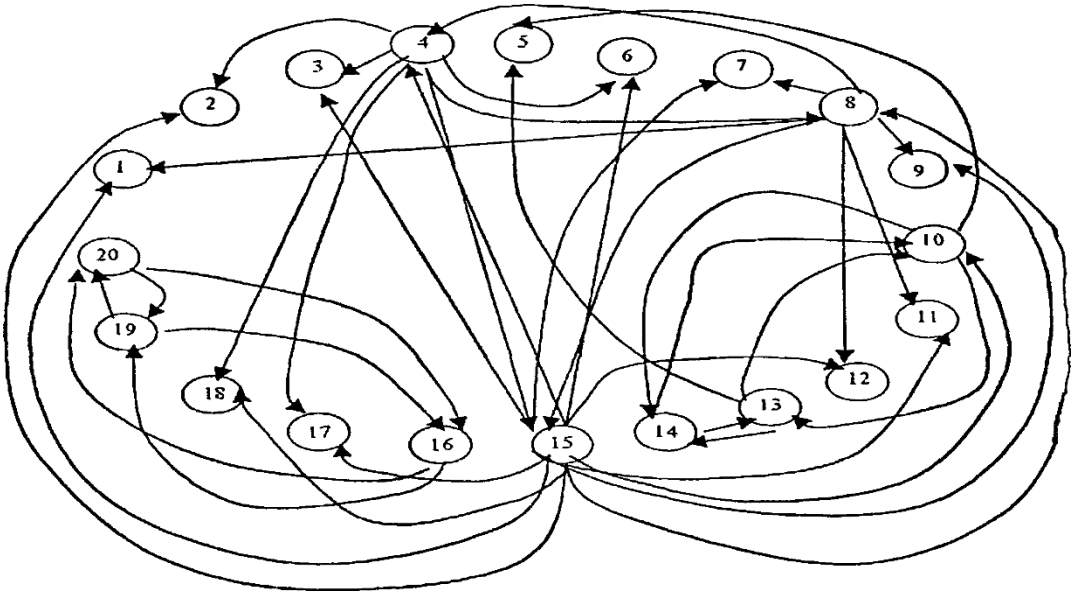


Рис. 7.2. Мережа, що відображає колектив людей з оцінками уподобань

Таблиця 7.1 Таблиця переваг

Дуга	Вага	Дуга	Вага	Дуга	Вага	Дуга	Вага	Дуга	Вага
(15,1)	10	(10,5)	10	(15,9)	10	(10,13)	10	(15,17)	10
(8,1)	8	(13,5)	9	(8,9)	8	(14,13)	9	(4,17)	5
(15,2)	10	(15,6)	10	(14,10)	8	(10,14)	10	(15,18)	10
(4,2)	7	(4,6)	5	(13,10)	7	(15,14)	8	(15,18)	10
(15,3)	10	(15,7)	10	(15,11)	10	(8,15)	10	(16,19)	10
(4,3)	8	(8,7)	9	(8,11)	9	(4,15)	9	(20,19)	6
(15,4)	10	(15,8)	10	(8,12)	10	(20,16)	10	(16,20)	10
(8,4)	6	(4,8)	7	(15,12)	9	(19,16)	5	(19,20)	7

У таблиці 7.1 в кожного члена колективу є дві оцінки переваг: найвища – першого рівня і наступна за значущістю – другого рівня.

При застосуванні алгоритму побудови максимального покривного лісу до мережі на рис. 7.2 одержимо, що вихідна мережа розпадається на три дерева. Кожне дерево відбиває найвищі переваги людей і відповідає наміченим групам, а корені дерев – претендентам на лідерство. На рис. 7.3 ці групи представлено у зручному графічному вигляді з коренями дерев, що відповідають лідерам, угорі.

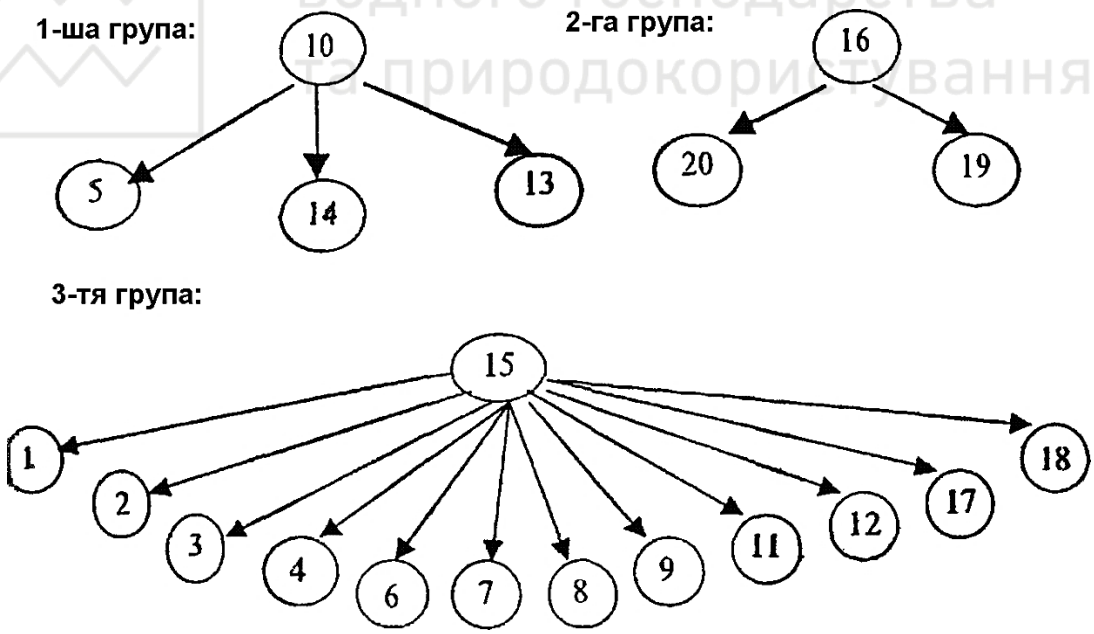


Рис. 7.3. Проміжний етап розв’язання задачі

1-шу і 2-гу групи можна вважати сформованими. Їх члени одержали саме тих лідерів, яких хотіли, знаючи, яке завдання їм доведеться вирішувати. Однак 3-тя група, з огляду на розмір колективу – 20 осіб, досить велика. Відповідно до наперед поставленої мети її слід було б розбити на кілька груп. Пропонований алгоритм дає змогу це зробити.

З цією метою звернемося до системи переваг другого плану. У вихідній мережі на рис. 7.4 заберемо сформовані 1-шу і 2-гу групи, виявленого лідера № 15 і всі зв'язки з ним. Разом з ним підуть всі переваги першого рівня.

До нової мережі знову застосуємо алгоритм Едмондса. Після такої процедури виявляються лідери другого плану. У підсумку одержимо дерево, зображене на рис. 5, у якому присутні тільки переваги другого рівня. На ньому бачимо дві групи з лідерами № 4 і 8, але лідер групи № 8 підпорядковується разом зі своєю групою лідеру № 4.

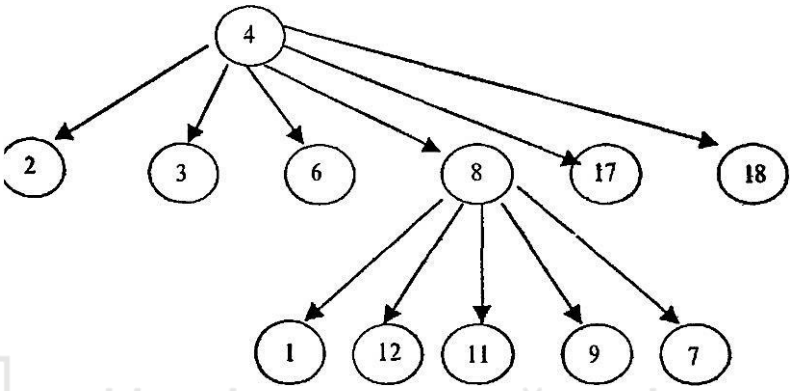


Рис. 7.4. Трансформована 3-тя група

Отже, колектив людей на рис. 7.4 можна організувати відповідно до їхніх особистих бажань у дві групи (рис. 7.5).

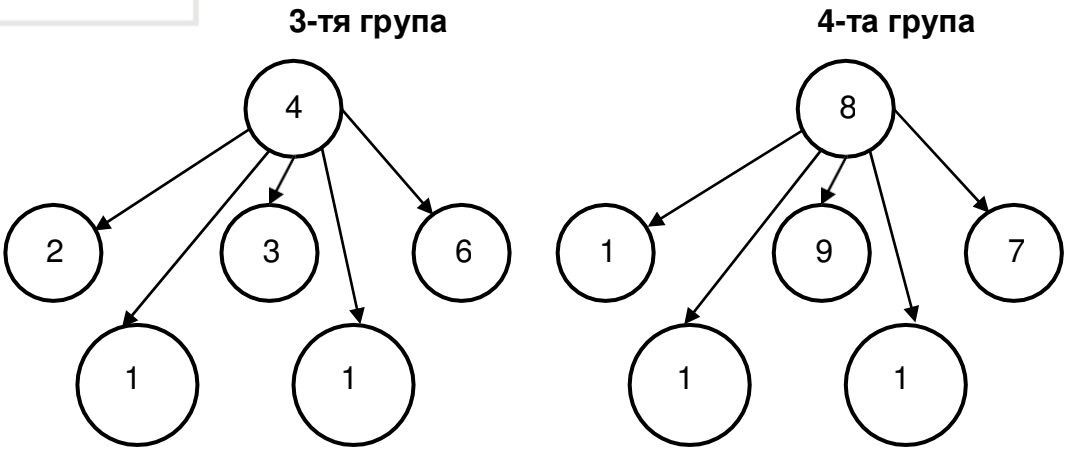


Рис. 7.5. Дві групи, сформовані на основі 3-ї групи

Підіб'ємо підсумок. Для вихідної мережі, зображеної на рис. 3, яка відбиває систему уподобань кожного члена колективу, на основі запропонованого алгоритму сформовано чотири групи. Співпідпорядкованість складових його груп має виглядати так, як показано на рис. 7.3 і 7.5.

Дві групи з лідерами № 4 і 8 мають працювати під керівництвом безумовного лідера № 15. Вони його самі обрали. Дві інші групи з лідерами № 10 і 16 бажають працювати незалежно. В остаточному підсумку

виконавець має вирішити підпорядковувати їх № 15 чи створити окремі підрозділи.

Оскільки в поставленій задачі потрібно вибрати лідера всього підрозділу і лідерів груп, що входять у цей підрозділ, виконавець, швидше за все, оголосить лідером усього підрозділу № 15 і підпорядкує йому групи з лідерами № 10 і 16. Тут вона буде діяти за принципом “вибору більшості голосів”.

Кінцеву підпорядкованість груп показано на рис. 7.6 пунктирною лінією. Групи не вийшли однаковими за кількісним складом, але основного принципу – бажання працювати під керівництвом конкретної, заявленої кожним членом групи особи – дотримано. Зрештою, групам, різним за кількісним складом, можна доручати завдання відповідно до їх можливостей.

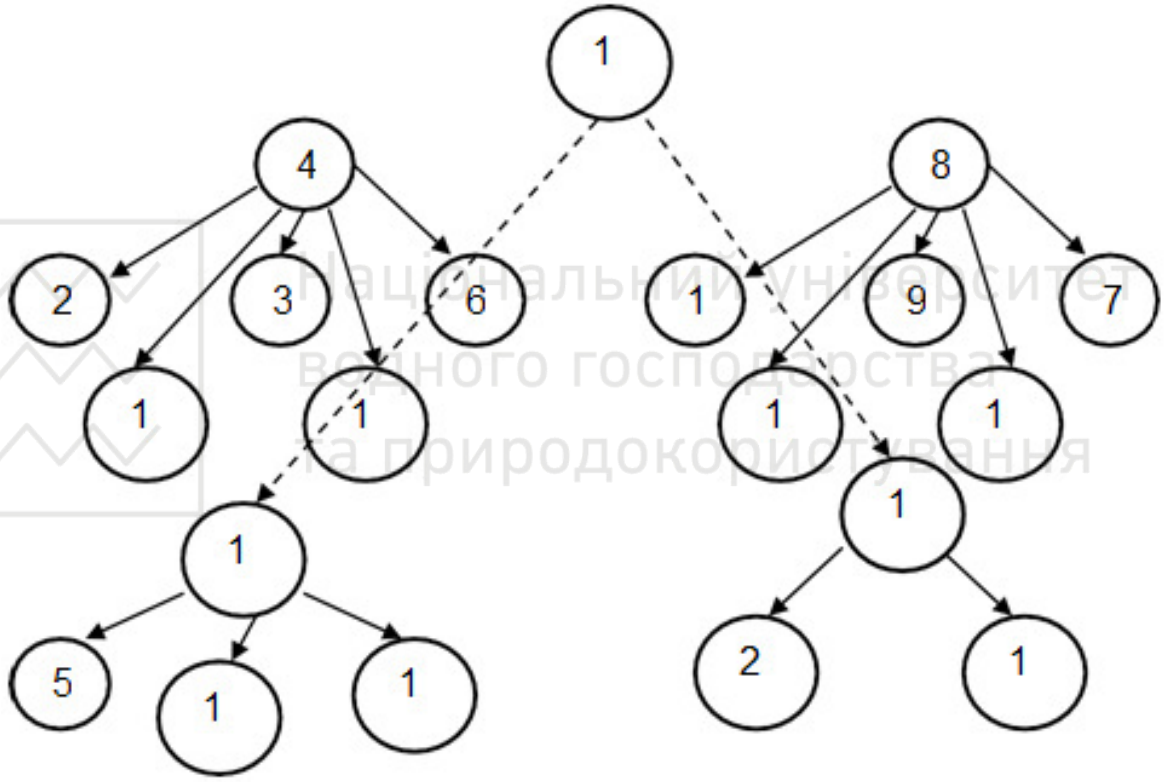


Рис. 7.6. Результат розв’язання задачі про лідерів

Завдання для виконання

7.1. Для мережі, що зображено на рис. 7.7, побудувати максимальний орієнтований ліс.

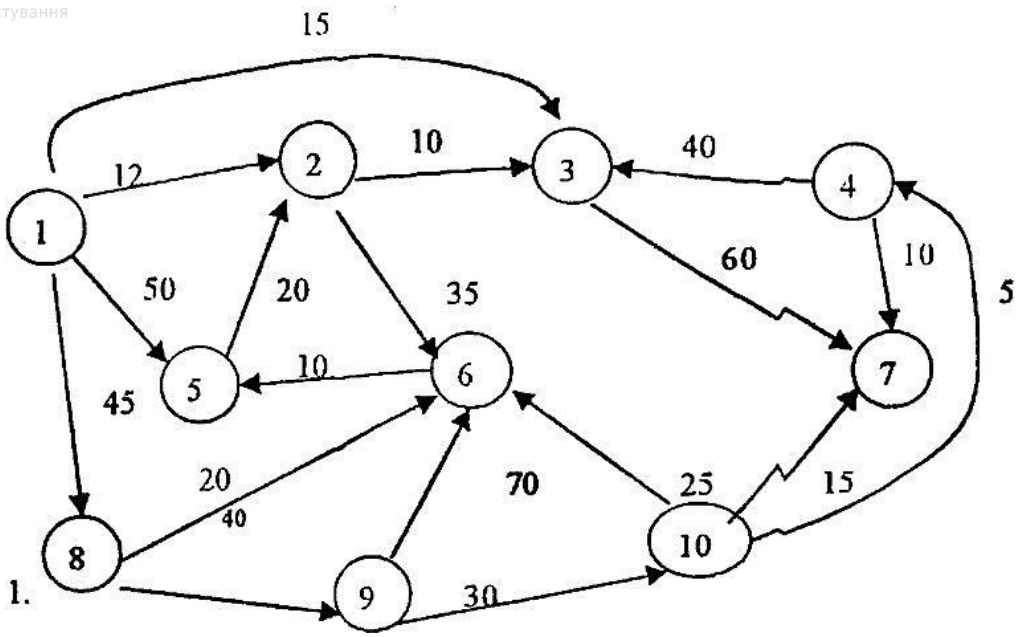


Рис. 7.7. Мережа для задачі 1

7.2. Які умови має задовольняти мережа, щоб максимальний і мінімальний покривний ліс для неї збігалися?

7.3. На мережі, що зображено на рис.7.8, зафіксовано ряд зв'язків. Побудувати для цієї мережі максимальний і мінімальний покривний ліс. Порівняти результати.

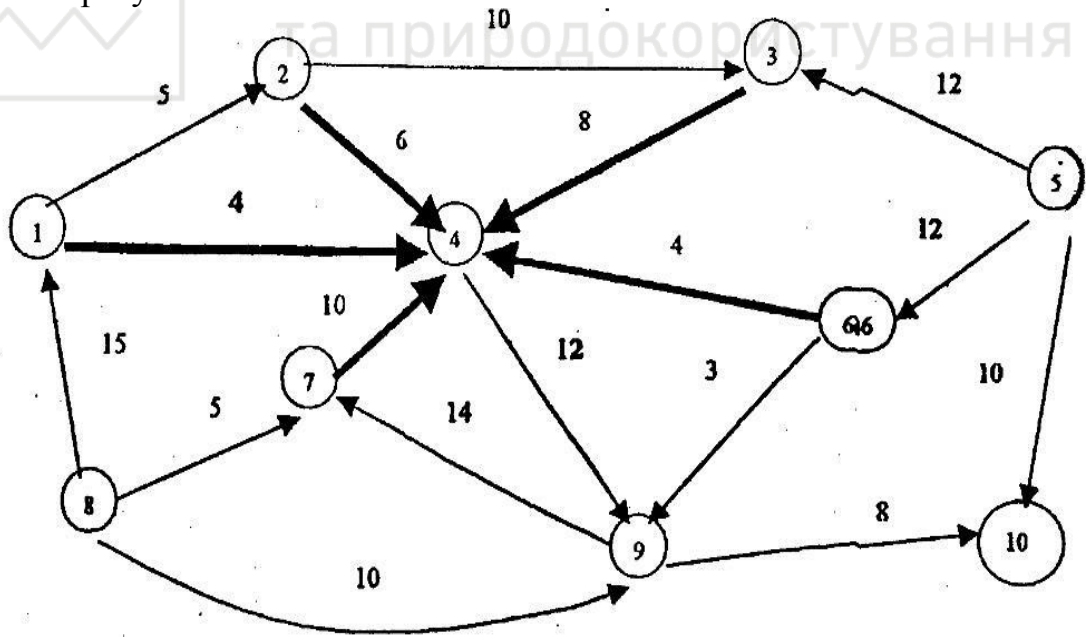


Рис. 7.8. Мережа для задачі 7.3

7.4. На мережі, що зображено на рис. 7.8, фіксовані вершини затушовані. Застосувати до цієї мережі алгоритм Едмондса і знайти для неї максимальний орієнтований ліс.

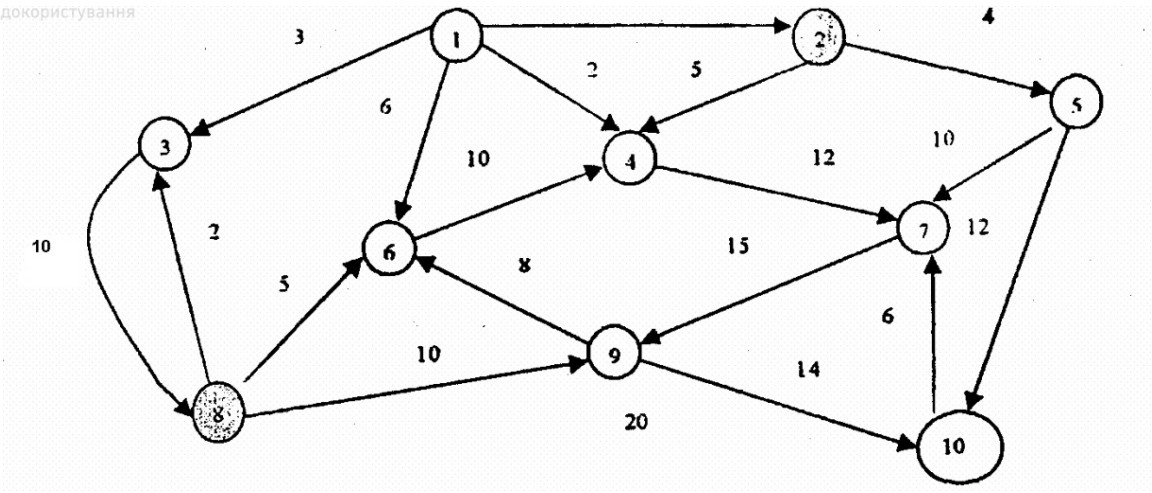


Рис. 7.8. Мережа для задачі 7.4

САМОСТІЙНА РОБОТА

Оптимізація на мережах

Задача С.1. Задача Пріма – Краскала

Приклад С.1. Задано дорожню мережу, що складається з дев'яти населених пунктів, з'єднаних польовими дорогами так, як це показано на рис. С.1.

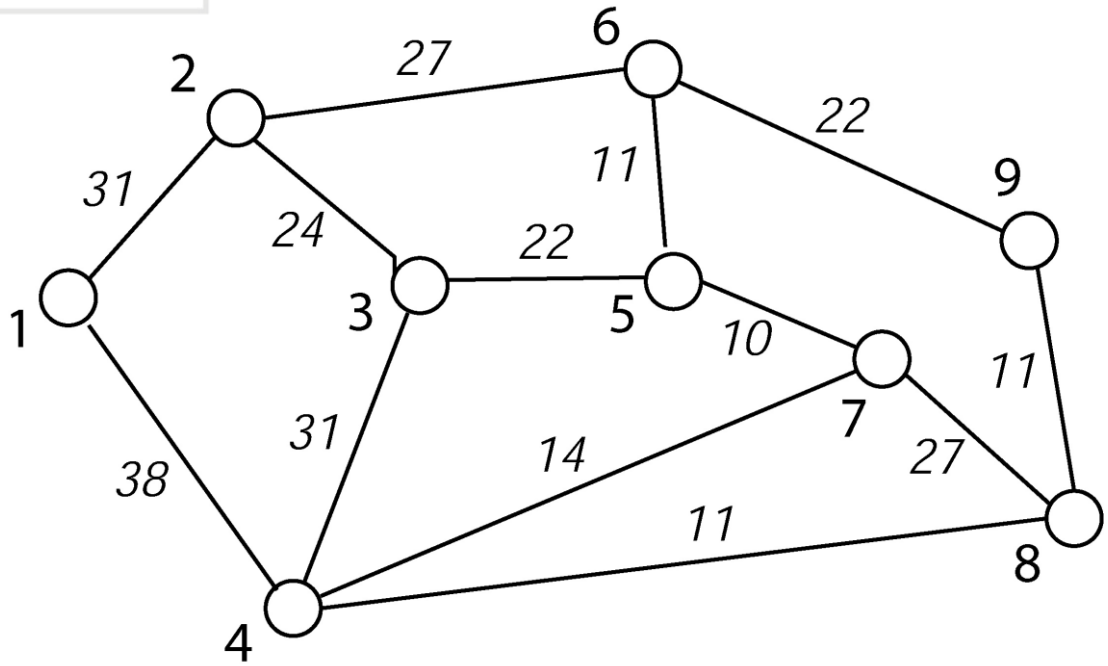


Рис. С.1. Дорожня мережа

Матриця відстаней*, що відповідає рис. С.1, матиме вигляд:

	1	2	3	4	5	6	7	8	9
1	*0*	*31*	*M*	*38*	*M*	*M*	*M*	*M*	*M*
2	*31*	*0*	*24*	*M*	*M*	*27*	*M*	*M*	*M*
3	*M*	*24*	*0*	*31*	*22*	*M*	*M*	*M*	*M*
4	*38*	*M*	*31*	*0*	*M*	*M*	*14*	*11*	*M*
5	*M*	*M*	*22*	*M*	*0*	*11*	*10*	*M*	*M*
6	*M*	*27*	*M*	*M*	*11*	*0*	*M*	*M*	*22*
7	*M*	*M*	*M*	*14*	*10*	*M*	*0*	*27*	*M*
8	*M*	*M*	*M*	*11*	*M*	*M*	*27*	*0*	*11*
9	*M*	*M*	*M*	*M*	*M*	*22*	*M*	*11*	*0*

** Всі інші відстані, що не вказані в таблиці, дорівнюють M ("машинна нескінченність").*

На основі існуючої мережі необхідно запроектувати найдешевшу мережу доріг з твердим покриттям. Вартість будівництва 1 км дороги однакова для всіх сполучень. Для розв'язування задачі використати алгоритм Пріма-Краскала. Зобразити початкову мережу та запроектовану найдешевшу мережу. Вказати загальну довжину найдешевшої мережі.

Порядок виконання роботи:

Побудова найдешевшої дорожної мережі. Застосовуємо алгоритм Пріма – Краскала (жадібний алгоритм).

1) Вибираємо ще не розглянуте ребро мінімальної довжини. У нашому випадку це ребро (5,7). Помічаємо розглянуте ребро хрестиком.
2) Приєднуємо розглянуте ребро (5,7) до раніше вибраних ребер при умові, що при цьому не утвориться цикл. Сполучаємо на схемі відповідні вершини. Якщо не всі ребра переглянуті, повертаємось до пункту 1.

В результаті перегляду всіх ребер із застосуванням алгоритму Пріма – Краскала отримано мінімальне покриваюче дерево для розглядуваного графа, або, іншими словами, найдешевшу дорожню мережу для даної системи населених пунктів. Мережа має вигляд, представлений на рис. С.2.

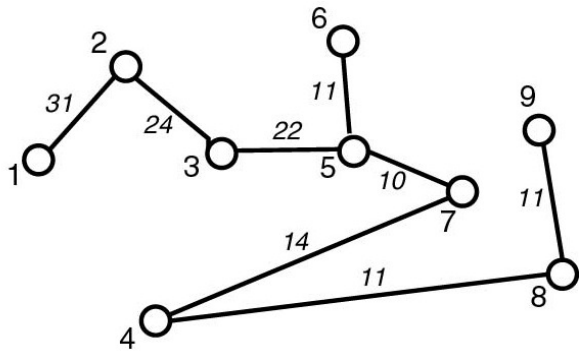


Рис. С.2. Схема найкоротшої дорожньої мережі для задачі Пріма –Краскала

Висновок. Побудовано найкоротшу дорожню мережу загальною довжиною 134 км.

Задача С.2. Задача про оптимальне розташування школи

При розв'язуванні задачі про оптимальне розташування школи необхідно виконати два таких кроки:

1. Побудувати матрицю найкоротших відстаней між пунктами мережі, використовуючи алгоритм Флойда, або ж метод перебору варіантів.
2. Використовуючи дані про кількість учнів у населених пунктах, наведені в таблиці С.3, знайти оптимальне місце для розташування школи.

Теоретичні відомості

В основу алгоритму розв'язування задачі покладене таке міркування: сума відстаней, пройдених всіма учнями, по дорозі до школи повинна бути мінімальною.

Це міркування називають *принципом мінімальної суми*, а саму задачу – *мінісумною*. Доведено теорему, згідно з якою оптимальним місцем розташування школи може бути лише населений пункт. Це значно спрощує розв'язування задачі і, по суті, зводить її до задачі про побудову матриці найкоротших відстаней між вершинами графа.

Для побудови матриці найкоротших відстаней використовують алгоритм Флойда. Основна частина цього алгоритму на мові Pascal має вигляд:

```
for k:=1 to N do
    for i:=1 to N do
        for j:=1 to N do
            If d[i,k] + d[k,j] < d[i,j]   then  d[i,j] := d[i,k] + d[k,j]
```

Тут N – кількість вершин мережі, d[i,j] – відстань від i-ої до j-ої його вершини.

У випадку невеликої кількості вершин для побудови матриці найкоротших відстаней можна використати *метод перебору варіантів*. Розглянемо суть цього методу на прикладі мережі, що зображено на рис.С.2. Метод перебору варіантів проілюструємо на прикладі відшукання найкоротших відстаней від вершини 1 до інших вершин мережі.

Найкоротші відстані до вершин 2 і 4 дорівнюють довжинам відповідних ребер, тобто 31 і 38.

Для знаходження найкоротшої відстані до вершини 3 розглянемо два варіанти: маршрут 1 – 2 – 3 довжиною 55 і маршрут 1 – 4 – 3 довжиною 69. Вибираємо коротший, тобто 1 - 2 – 3 (довжина 55).

Для знаходження найкоротшої відстані до вершини 5 розглянемо три варіанти: маршрут 1 – 2 – 3 – 5 довжиною 77; маршрут 1 – 2 – 6 – 5 довжиною 69 і маршрут 1 – 4 – 7 – 5 довжиною 62. Вибираємо коротший,

тобто 1 - 4 – 7 – 5 (довжина 62). Маршрут 1 – 4 – 3 – 5 розглядати немає змісту, оскільки вже було показано, що маршрут 1 – 2 – 3 є оптимальнішим від маршруту 1 – 4 – 3.

Найкоротший маршрут до вершини 6: 1 – 2 – 6 (довжина 58) і це очевидно.

Найкоротший маршрут до вершини 7: 1 – 4 – 7 (довжина 52) і це також очевидно.

Найкоротший маршрут до вершини 8: 1 – 4 – 8 (довжина 49).

Для знаходження найкоротшої відстані до вершини 9 розглянемо два варіанти: маршрут 1 – 2 – 6 – 9 довжиною 80 і маршрут 1 – 4 – 8 – 9 довжиною 60. Вибираємо коротший, тобто 1 – 4 – 8 – 9 (довжина 60).

Аналогічним чином аналізуємо всі інші вершини. В результаті отримуємо матрицю найкоротших відстаней у вигляді:

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	***к-сть учнів***
1	*0*	*31*	*55*	*38*	*62*	*58*	*52*	*49*	*60*	**50**
2	*31*	*0*	*24*	*55*	*38*	*27*	*48*	*60*	*49*	**20**
3	*55*	*24*	*0*	*31*	*22*	*33*	*32*	*42*	*53*	**10**
4	*38*	*55*	*31*	*0*	*24*	*35*	*14*	*11*	*22*	**40**
5	*62*	*38*	*22*	*24*	*0*	*11*	*10*	*35*	*33*	**80**
6	*58*	*27*	*33*	*35*	*11*	*0*	*21*	*33*	*22*	**30**
7	*52*	*48*	*32*	*14*	*10*	*21*	*0*	*25*	*36*	**60**
8	*49*	*60*	*42*	*11*	*35*	*33*	*25*	*0*	*11*	**90**
9	*60*	*49*	*53*	*22*	*33*	*22*	*36*	*11*	*0*	**70**
	21120	**19550**	**16630**	**9650**	**11430**	**11820**	**10640**	**10570**	**11840**	

Приєднаємо справа до таблиці ще один стовпець, де розмістимо дані про кількість учнів у населених пунктах (табл. С.3). Приєднаємо знизу до таблиці ще один рядок, де розмістимо суму добутків кількості учнів у населених пунктах на найкоротші відстані до цих населених пунктів. Це число характеризуватиме суму відстаней, що пройдено всіма учнями, у випадку розташування школи в даному населеному пункті. Аналіз останнього рядка показує, що найвигідніше розмістити школу у четвертому населеному пункті, оскільки сума у цьому випадку буде найменшою.

Висновок. Найвигідніше розмістити школу в четвертому населеному пункті. Сума відстаней, пройдених всіма учнями при цьому буде мінімальною і становитиме 9650 км.

Задача С.3. Задача про оптимальне розташування пожежної частини

Оптимальний варіант розташування пожежної частини в одному з населених пунктів можна знайти з використанням побудованої попередньо матриці найкоротших відстаней.

Теоретичні відомості

В основу розв'язування задачі покладене наступне міркування: *відстань від пожежної частини до найбільш віддаленого пункту мережі повинна бути мінімальною*. Це міркування називають *принципом мінімаксу*, а саму задачу – мінімаксною.

Аналогічно розв'язуються задачі про розміщення станції швидкої допомоги, служби охорони, тобто тих служб, час виклику яких повинен бути мінімальним. На відміну від задачі про розміщення школи, задача про розміщення пожежної частини допускає два варіанти розв'язування:

- розміщення пожежної частини у населеному пункті;
- розміщення пожежної частини поза населеним пунктом.

Тому необхідно проаналізувати обидва варіанти, а потім вибрати кращий з них. У даному курсі розглядається лише аналіз більш простого першого варіанту.

<u>Задача про розміщення пожежної частини у населеному пункті</u>

Ця задача зводиться до задачі про побудову матриці найкоротших відстаней між вершинами графа. Розглянемо дорожню мережу (рис. С.1). Матриця найкоротших відстаней має вигляд:

	1	2	3	4	5	6	7	8	9	*max*
1	*0*	*31*	*55*	*38*	*62*	*58*	*52*	*49*	*60*	*62*
2	*31*	*0*	*24*	*55*	*38*	*27*	*48*	*60*	*49*	*60*
3	*55*	*24*	*0*	*31*	*22*	*33*	*32*	*42*	*53*	*55*
4	*38*	*55*	*31*	*0*	*24*	*35*	*14*	*11*	*22*	*55*
5	*62*	*38*	*22*	*24*	*0*	*11*	*10*	*35*	*33*	*62*
6	*58*	*27*	*33*	*35*	*11*	*0*	*21*	*33*	*22*	*58*
7	*52*	*48*	*32*	*14*	*10*	*21*	*0*	*25*	*36*	<u>52</u>
8	*49*	*60*	*42*	*11*	*35*	*33*	*25*	*0*	*11*	*60*
9	*60*	*49*	*53*	*22*	*33*	*22*	*36*	*11*	*0*	*60*

Приєднаємо до неї справа ще один стовпець і випишемо в ньому максимальні елементи кожного рядка. Вони відповідатимуть відстані до найбільш віддаленого пункту мережі (розглядається найгірший варіант) у випадку розташування пожежної частини у населеному пункті, номер якого відповідає номеру рядка матриці. Вибираємо мінімальний елемент серед всіх чисел добавленого стовпця. У нашому випадку це число 52 (пункт 7). Це і є розв'язок задачі.

Висновок. Якщо розміщувати пожежну частину у населеному пункті, то найбільш вигідно це зробити у населеному пункті *7*. При цьому відстань до найбільш віддаленого пункту становитиме 52 км.

Завдання для виконання

Використовуючи графи до завдання 5.5, визначити:

С.1. Оптимальне місце для побудови школи. Кількість учнів, які проживають в населених пунктах, наведено в табл. С.1.

С.2. Оптимальне місце для побудови пожежної частини в населеному пункті.

Таблиця С.1. Кількість учнів у населених пунктах

Граф	**Назви населених пунктів**									
	А	**Б**	**В**	**Г**	**Д**	**Е**	**Ж**	**З**	**И**	**К**
а	***40***	***50***	***70***	***45***	***30***	***25***	***20***	***65***	***55***	***35***
б	***50***	***70***	***45***	***30***	***25***	***20***	***55***	***27***	***35***	***40***
в	***70***	***45***	***30***	***25***	***20***	***55***	***35***	***39***	***40***	***50***

Граф	**Назви населених пунктів**													
	А	**Б**	**В**	**Г**	**Д**	**Е**	**Ж**	**З**	**И**	**К**	***Л***	***М***	***Н***	***О***
г	***45***	***30***	***25***	***65***	***20***	***55***	***24***	***35***	***58***	***40***	***48***	***50***	***70***	***34***
д	***30***	***25***	***20***	***38***	***55***	***35***	***32***	***40***	***26***	***50***	***64***	***70***	***45***	***75***

Рекомендована література

1. Базилевич Л. Дискретна математика у прикладах і задачах : підручник. Львів : Видавець І. Е.Чижиков, 2013. 487 с.
2. Грицюк П. М., Бабич Т. Ю. Геоінформаційні системи і технології : навчальний посібник. Рівне : НУВГП, 2014. 239 с.
3. Ільків В. С., Каленюк П. І., Когут І. В., Пукач П. Я., Сохан П. Л., Нитребич З. М., Столярчук Р. Р., Ярка У. Б. Основи дискретної математики. Математична логіка. Теорія графів. Частина 2. Львів : Видавництво Львівської політехніки, 2011. 184 с.
4. Карнаух Т. О., Ставровський А. Б. Теорія графів у задачах : навчальний посібник. Київ : ВПЦ "Київський університет", 2004. 76 с.
5. Кононюк А. Е. Дискретно-непрерывная математика. (Графы. К.7 Ч.2). В 15-и кн. Кн. 7. Київ : Освіта України, 2015. 512 с.
6. Кристофидес Н. Теория графов. Алгоритмический подход. Москва : Мир, 1978. 432 с.
7. Ніколаєва К. В., Койбічук В. В. Дискретний аналіз. Графи та їх застосування в економіці : навчально-методичний посібник. Суми : УАБС НБУ, 2007. 84 с.
8. Нікольський Ю. В., Пасічник В. В., Щербина Ю. М. Дискретна математика : підручник. Львів : «Магнолія 2006», 2007. 608 с.