Українська академія банківської справи
Національного банку України
Кафедра вищої математики та інформатики

К.В. Ніколаєва, В.В. Койбічук

ДИСКРЕТНИЙ АНАЛІЗ
Графи та їх застосування в економіці

Навчально-методичний посібник

Для студентів економічних спеціальностей
денної форми навчання

Суми
УАБС НБУ
2007

УДК 510(076.5)
Н63

Рекомендовано до видання методичною радою обліково-фінансового факультету Української академії банківської справи Національного банку України, протокол № 10 від 29.06.2006.

Розглянуто і схвалено на засіданні кафедри вищої математики та інформатики, протокол № 8 від 24.05.2006.

Автори:
кандидат фізико-математичних наук, доцент
К.В. Ніколаєва;

викладач-стажист
В.В. Койбічук

Рецензенти:
кандидат фізико-математичних наук,
доцент кафедри математичного аналізу
Київського національного університету ім. Тараса Шевченка
О.О. Купченко;

кандидат фізико-математичних наук,
доцент кафедри інформатики
Сумського державного університету
С.П. Шаповалов

Відповідальний за випуск
кандидат технічних наук, доцент
В.В. Яценко

Н63 **Ніколаєва К.В., Койбічук В.В. Дискретний аналіз. Графи та їх застосування в економіці:** Навчально-методичний посібник. – Суми: УАБС НБУ, 2007. – 84 с.

Мета посібника – ознайомити з основами теорії графів та їх різноплановими застосуваннями, навчити студентів бачити, ставити та розв’язувати практичні задачі, що зводяться до графів, а також інтерпретувати результати їх розв’язання. Основну увагу приділено застосуванням графів до мережних систем.

Посібник призначений для студентів економічних спеціальностей денної форми навчання.

УДК 510(076.5)

ВСТУП

Посібник поєднує дві науки – математику та економіку, більш конкретно – дискретний аналіз і логістику. Ще конкретніше – теорію графів і теорію мережних систем.

Термін "граф" вперше з'явився в науковій літературі в 1936 році в роботах угорського математика Д. Кьоніга, хоча елементи теорії графів були відомі та широко використовувались ще у XVIII столітті, зокрема в роботах Л. Ейлера.

Незалежно від уподобань, з графами, мабуть, зустрічався кожний. 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.

З елементами теорії графів знайомить І розділ посібника.

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. Переваги – простота та наочність. Складність – вміння побачити можливість переформулювання умови задачі мовою графів – створення так званої графової моделі задачі. Розв'язавши задачу в межах теорії графів, результат необхідно інтерпретувати у вихідних термінах.

Можливим застосуванням графів в економіці присвячений другий розділ посібника. Серед майже неосяжної їх кількості виділені:

- задачі розподілу ресурсів, які виникають при певному наборі операцій (робіт), що необхідно виконати за обмежених ресурсів;
- задачі управління запасами, що полягають у знаходженні оптимальних значень рівня запасів і розміру замовлення для забезпечення безперебійного виробничого процесу;
- задачі мережного планування і управління, в яких розглядаються співвідношення між термінами закінчення великого комплексу операцій і моментами початку всіх операцій комплексу. Потрібно знайти мінімальні тривалості комплексу операцій, оптимальні співвідношення вартості і термінів виконання;
- мережні задачі, що полягають у оптимізації процесу обслуговування на мережах чи самої структури мережі;
- задачі дослідження конфліктних ситуацій, які полягають у виборі оптимальних стратегій поведінки учасників конфлікту;
- задачі масового обслуговування систем з чергами заявок;

- задачі ремонту і заміни устаткування, присвячені зносу і старінню устаткування з необхідністю його заміни з часом;
- задачі складання розкладів полягають у визначенні оптимальної черговості виконання операцій на різних видах устаткування чи при певному способі надання послуг;
- задачі планування і розміщення, пов'язані з визначенням оптимального числа і місця розміщення нових об'єктів з урахуванням їх взаємодії з наявними об'єктами і між собою.

Коло подібних питань об'єднує логістика – наука про планування, організацію, управління, контроль і регулювання руху матеріальних та інформаційних потоків у просторі й часі від їх первинного джерела до кінцевого споживача. До логістики відносять і питання сервісу, прогнозування попиту, розміщення виробничих підприємств і складів, процеси постачання, утилізацію відходів і т.п. Ці складні масштабні проблеми внаслідок великої кількості їх складових і зв'язків між ними, що часто не піддаються опису і навіть простому перелічуванню, вимагають комплексного підходу, і економісти досліджують їх не одне століття. Оскільки в реальному житті такими системами управляють, і часто дуже успішно, це означає, що в принципі задачі оптимального управління для них розв'язувані. Для цього потрібен тільки придатний метод їх формалізації. І такий метод використовувався на чисто практичному рівні давно, задовго до появи наукових теорій і методів дослідження.

З метою вирішення багатьох задач, пов'язаних зі складними виробничими структурами, останні поділяються на сукупність дрібніших, але все-таки досить великих елементів для того, щоб отримана формалізована структура по суті "перестала" б бути великою системою, тобто описувалася б досить обмеженою кількістю параметрів. Такою структурою вже можна оперативно керувати і розв'язувати для неї оптимізаційні задачі, як це й робиться на практиці. Такі розподілені системи називають мережними, а сукупність методів, що застосовують до їх аналізу, – методами (чи алгоритмами) оптимізації на мережах чи графах.

При підборі матеріалу ми обмежились лише найпростішими питаннями теорії графів. Вибираючи їх, ми ставили за мету:

- ознайомити читача з базовими поняттями теорії графів, формуючи запас необхідної для подальшого самостійного вивчення спеціальної математичної, програмістської, економічної літератури;
- розглянути деякі конкретні задачі, які можуть бути розв'язані графовими методами;

- дати деяке уявлення про застосування графів, звертаючи увагу як на побудову математичної графової моделі, так і на характер досліджень, які можуть проводитись за допомогою графів.

Посібник насичений прикладами конкретних алгоритмів від простих до досить складних і специфічних, особливо в другому розділі, і спрямований на самостійну роботу студентів. Тому кожний його параграф завершується великою добіркою завдань для самостійної роботи.

Необхідні для опанування матеріалом як економічні, так і математичні знання, не виходять за межі програми молодших курсів економічних спеціальностей, але передбачають широкий кругозір і розвинуту здатність до творчого і абстрактного мислення.

Рекомендована література допоможе читачеві поглибити та поширити власну поінформованість з питань, що його зацікавили.

В посібнику наведені основні означення та факти, що, на нашу думку, сприяє полегшенню вивчення матеріалу.

Нумерація прикладів, формул, рисунків в кожному параграфі своя. Позначення та скорочення, що використовуються в посібнику, є, в основному, загальновживаними.

РОЗДІЛ І. ТЕОРІЯ ГРАФІВ

§ 1. Основні поняття та факти теорії графів. Операції над графами

Наочно граф можна уявити як геометричну конфігурацію, яка складається з точок (вершин) і ребер (ліній або відрізків, які сполучають деякі точки).

Дамо формальне означення графа.

Нехай $X = \{x_1, \dots, x_n\}$ – деяка скінченна множина (множина вершин), M_2 – множина всіх невпорядкованих пар елементів з X,

$$M_2 = \{(x_i, x_j) : x_i \in X,\ x_j \in X,\ i \neq j\}.$$

О. 1. Граф $G(X, W)$ – пара множин $X, W \subset M_2$. Множина X – це множина вершин, множина W – це множина ребер. Якщо $(x_i, x_j) \in W$, то ми говоримо, що 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*.

О. 2. Граф $\mathrm{G}(X, W)$ називається повним, якщо $W = M_2$.

Якщо множина X містить n вершин, то, очевидно, число ребер повного графа дорівнює C_n^2. Повний граф з n вершинами позначається K_n.

О. 3. Граф $G(X, W)$ називається порожнім, якщо $W = \varnothing$.

Теорема 1. Число усіх різних графів з n вершинами дорівнює

$$2^{C_n^2} = 2^{\frac{n(n-1)}{2}}. \qquad (1)$$

О. 4. Вершини x_i та x_j графа $G(X, W)$ інцидентні, якщо $(x_i, x_j) \in$ $\in W$.

О. 5. Степенем $d(x_i)$ вершини x_i графа $G(X, W)$ називається число вершин x_j, які інцидентні вершині x_i (число відрізків, які виходять з вершини x_i).

О. 6. Якщо $d(x_i) = 1$, то вершина x_i називається *кінцевою* вершиною графа $G(X, W)$. Якщо $d(x_i) = 0$, то вершина x_i називається *ізольованою*.

Граф $G(X, W)$ зручно зображати за допомогою рисунка на площині, який називають *діаграмою* графа G.

Графи можна задавати за допомогою матриць.

Занумеруємо всі вершини графа G натуральними числами від 1 до *n*. *Матрицею суміжності A* графа G називається квадратна $n \times n$-матриця, в якій елемент a_{ij} *i*-го рядка та *j*-го стовпчика дорівнює 1, якщо вершини x_i та x_j з номерами i та j суміжні, і дорівнює 0 – якщо ні.

Занумеруємо всі вершини графа G натуральними числами від 1 до *n* і всі його ребра числами від 1 до *m*. Матрицею інцидентності *B* графа G називається $n \times m$-матриця, в якій елемент b_{ij} *i*-го рядка і *j*-го стовпчика дорівнює 1, якщо вершина x_i з номером *i* інцидентна ребру w_j з номером j, і дорівнює 0 в протилежному випадку. Отже,

$$a_{ij} = \begin{cases} 1, & (x_i, x_j) \in W, \\ 0, & (x_i, x_j) \notin W; \end{cases} \qquad b_{ij} = \begin{cases} 1, & x_i \in w_j, \\ 0, & x_i \notin w_j. \end{cases}$$

Лема про рукостискання. Нехай $G(X, W)$ граф з множиною вершин $X = \{x_1, \dots, x_n\}$. Тоді

$$\sum_{i=1}^{n} d(x_i) = 2N(W). \qquad (2)$$

О. 7. Послідовність ребер $(x_{i_1}, x_{i_2}), (x_{i_2}, x_{i_3}), \dots, (x_{i_{l-2}}, x_{i_{l-1}}), (x_{i_{l-1}}, x_{i_l})$, в якій сусідні ребра інцидентні одній і тій же вершині, називають *ланцюгом*. Ланцюг називається *простим*, якщо всі вершини, належні йому (крім, можливо, першої і останньої), різні; число ребер у цьому випадку називають *довжиною* ланцюга.

Якщо $x_{i_1} = x_{i_l}$, то ланцюг називається *циклом*. Цикл, в якому всі вершини різні, називається *простим*.

О. 8. Граф $G(X', W')$ є *підграфом* графа $G(X, W)$, якщо $X' \subset X$, $W' \subset W$. Якщо $X' \subset X$, то підграф $G(X', W')$ називається *остовним підграфом*.

О. 9. Граф $G(X, W)$ є *сумою* (об'єднанням) графів $G(X_1, W_1), \ldots,$ $G(X_k, W_k)$, якщо $X = \bigcup_{i=1}^{k} X_i$, $W = \bigcup_{i=1}^{k} W_i$. Ця сума називається прямою, якщо $X_i \cap X_j = \varnothing$, $i \neq j$.

О. 10. Граф $G(X, W)$ називається *зв'язним*, якщо будь-які дві вершини x_i та x_j $(x_i \in X, x_j \in X)$ сполучені ланцюгом з початком в x_i і кінцем в x_j. З симетрії випливає, що в цьому випадку і вершина x_j сполучена з вершиною x_i.

О. 11. *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 $G''(X, W_1 \setminus W_2)$ відповідно; позначаються $G' = G_1 \cap G_2$ і $G'' = G_1 \setminus G_2$.

О. 12. *Доповненням* графа $G(X, W)$ називається граф $\overline{G} =$ $= (X, M_2 \setminus W)$; отже, граф $\overline{G}$ має ту саму множину вершин X, що й граф G, а будь-які дві вершини графа $\overline{G}$ суміжні тоді і тільки тоді, коли вони несуміжні в G.

О. 13. Граф $G(X, W)$ називається *двочастковим*, якщо існує таке розбиття множини його вершин X на дві підмножини (частки) X_1 і X_2, що кінці будь-якого ребра графа G належать різним часткам.

Двочастковий граф називається *повним двочастковим* графом, якщо будь-які дві його вершини, що належать різним часткам, є суміжними. Повний двочастковий граф, частки якого X_1 і X_2 складаються з n і m вершин відповідно, позначається $K_{n,m}$.

О. 14. *Операція вилучення вершини* x із графа $G(X, W)$ полягає у вилученні з множини X елемента x, а з множини W – усіх ребер, інцидентних x.

Операція вилучення ребра w з графа $G(X, W)$ полягає у вилученні елемента w з множини W. При цьому всі вершини зберігаються.

Теорема 2. Кожен граф є прямою сумою зв'язних графів.
Ці графи називаються компонентами зв'язності.

Теорема 3. Нехай $G(X,W)$ граф, який складається з n вершин, m ребер і k компонент зв'язності. Тоді виконуються нерівності

$$n-k \le m \le C_{n-k+1}^{2}. \tag{3}$$

Наслідок. Будь-який граф з n вершинами і більше ніж $C_{n-1}^{2} = \frac{(n-1)(n-2)}{2}$ ребрами є зв'язним.

Завдання для самостійної роботи

1. Нехай задано граф $G(X,W)$:
   а) $X=\{1,2,3,4\}, \quad W=\{(1,3),(2,3),(3,4),(4,1),(4,2)\}$;
   б) $X=\{a,b,c,d,e\}, \; W=\{(a,d),(b,c),(b,e),(c,e),(d,b),(d,e),(e,a)\}$;
   в) $X=\{1,2,3\}, \quad W=\{(1,2),(1,3),(2,3)\}$;
   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. Нехай $X=\{a,b,c,d,e\}$. Граф $G(X,W)$ задано за допомогою матриці суміжності A.

а) $A=\begin{pmatrix} 0&1&1&0&1\\ 1&0&0&1&1\\ 1&0&0&0&1\\ 0&1&0&0&0\\ 1&1&1&0&0 \end{pmatrix}$; б) $A=\begin{pmatrix} 0&1&0&0&1\\ 1&0&1&0&0\\ 0&1&0&1&0\\ 0&0&1&0&1\\ 1&0&0&1&0 \end{pmatrix}$;

в) $A=\begin{pmatrix} 0&0&1&0&1\\ 0&0&1&0&0\\ 1&1&0&1&1\\ 0&0&1&0&1\\ 1&0&1&1&0 \end{pmatrix}$; г) $A=\begin{pmatrix} 0&1&1&0&0\\ 1&0&1&0&0\\ 1&1&0&0&0\\ 0&0&0&0&0\\ 0&0&0&0&0 \end{pmatrix}$.

Визначити множину ребер W графа G.

3. Граф G задано його діаграмою (рис. 1).

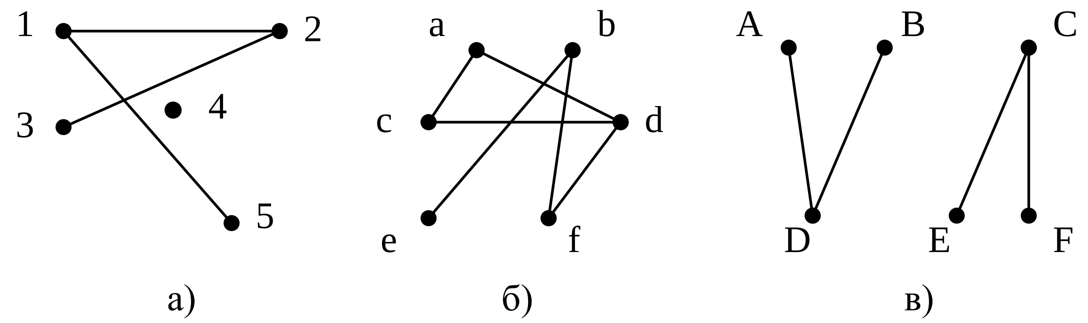


Рис. 1

Визначити множину вершин X і множину ребер W, матриці суміжності та інцидентності графа G.

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

   а) $n=2$; б) $n=3$; в) $n=4$; г) $n=5$.
5. Чому дорівнює степінь кожної вершини у повному графі з n вершинами?
6. Скільки ребер містить повний граф з n вершинами?
7. Чи існує повний граф, у якого кількість ребер дорівнює:

   а) 15; б) 18; в) 199…900…0 г) $8k^2+2k, k \in N$?
8. Довести, що доповненням графа $\overline{G}$ є граф G.
9. 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) $d(x)=k$?
10. Чому дорівнює кількість ребер у графі $\overline{G}$, якщо граф G має n вершин і k ребер?
11. Довести, що для довільного графа G об'єднання $G \cup \overline{G}$ є повним графом.
12. 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) $G_1 \cup G_2$; б) $G_1 \cap G_2$; в) $G_1 \setminus G_2$; г) $\overline{G_1}$.
13. 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?

14. Нехай A – матриця суміжності графа G з n вершинами. Довести, що i-й діагональний елемент матриці A^2 дорівнює $d(x_i)$ i-ї вершини графа $G,\ i = 1, 2, \ldots, n.$
15. Довести, що у будь-якому графі кількість вершин, степінь яких непарний, є парною.
16. Нехай у графі G з n вершинами і m ребрами є p вершин степеня $t,$ а всі інші вершини мають степінь $t+1.$ Довести, що $p = (t+1)n - 2m.$
17. 29 команд беруть участь у футбольному турнірі. Довести, що в будь-який момент знайдеться команда, яка зіграла парну кількість матчів.
18. Чи існує граф з n вершинами, усі вершини якого є кінцевими, якщо:
    а) $n = 10;$ б) $n = 11;$ в) $n = 2k - 1;$ г) $n = 2k$?
19. Скільки вершин може мати граф, усі вершини якого є кінцевими? Скільки ребер у такому графі?
20. Чи можна з повного графа K_{17} вилучити деякі ребра так, щоб степінь кожної з вершин дорівнювала:
    а) 15; б) 2; в) 1?
21. Чи існує кубічний граф із n вершинами, якщо
    а) $n = 100;$ б) $n = 101;$ в) $n = 102;$ г) $n = 2k - 1;$ ґ) $n = 2k$?
22. Скільки вершин повинен мати кубічний граф? Скільки ребер у такому графі?
23. Побудувати кубічний граф із:
    а) чотирма вершинами;
    б) шістьма вершинами;
    в) вісьмома вершинами.
24. Довести, що доповнення жодного кубічного графа не є кубічним графом.
25. У певному товаристві з n осіб кожен є знайомим з k і тільки k іншими особами. Чи можливе таке товариство для:
    а) $n = 5,\ k = 2;$
    б) $n = 5,\ k = 3;$
    в) $n = 2m,\ k = 1;$
    г) $n = 2m,\ k = 3$?
26. Скільки ребер містить повний двочастковий граф $K_{n,\,m}$?
27. Які особливості має матриця суміжності двочасткового графа?
28. Який вигляд має доповнення графа $K_{n,\,m}$?
29. Довести, що в будь-якому графі з n вершинами $(n > 2)$ завжди знайдуться принаймні дві вершини з однаковими степенями.

30. Побудуйте граф із п'ятьма вершинами, в якому тільки дві вершини мають однакові степені.
31. У графі з п'ятьма вершинами тільки дві вершини мають однакові степені. Чи можуть обидві ці вершини мати степінь 0 або степінь 4?
32. Декілька осіб проводять шаховий турнір в одне коло. У деякий момент виявилося, що тільки двоє шахістів зіграли однакову кількість партій. Довести, що тоді є або тільки один учасник, який не зіграв жодної партії, або тільки один, який зіграв усі партії турніру.
33. Скільки вершин із однаковими степенями має граф $\overline{G}$, якщо граф G має тільки 2 вершини з однаковими степенями?
34. Довести, що в довільному графі G із шістьма вершинами завжди знайдуться три вершини, які є або попарно суміжними, або попарно несуміжними. (Це математичне формулювання відомої задачі: довести, що серед будь-яких шести осіб знайдуться три особи, що попарно знайомі між собою, або три особи, попарно незнайомі).
35. Чи існує граф із шістьма вершинами, степені яких дорівнюють:
   а) 2, 3, 3, 4, 4, 4; б) 2, 2, 2, 4, 5, 5?
   Відповідь обґрунтувати.
36. На рис. 2 зображено граф G.
   а) знайти всі ланцюги, що ведуть із вершини v_1 у v_{12};
   б) знайти всі прості ланцюги, що ведуть із вершини v_1 у v_{12};
   в) знайти ланцюг, що веде із вершини v_1 у v_{12} і містить усі вершини графа G;
   г) чи існує в графі G простий ланцюг, що веде з вершини v_1 у v_{12} і містить усі вершини графа G?
   ґ) знайти маршрут у графі G, який веде з v_1 у v_{12} і не є ланцюгом;
   д) знайти маршрут у графі G, який веде з v_1 у v_{12}, містить усі вершини графа G і не є ланцюгом;
   е) знайти який-небудь цикл у графі G;
   є) знайти всі прості цикли графа G.

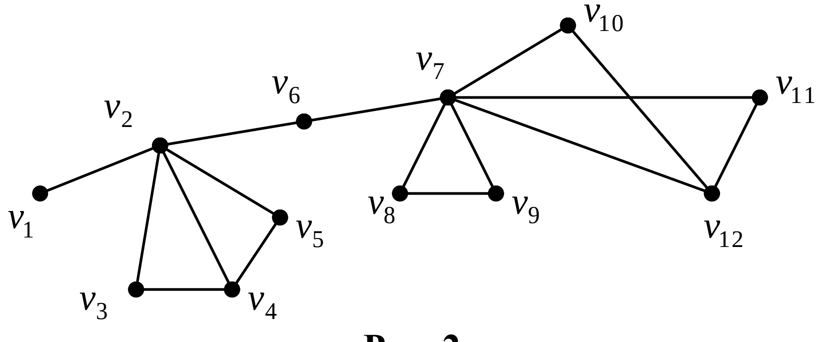


Рис. 2

37. Знайти в графі K_5 цикли довжини:

а) 3; б) 4; в) 5; г) 6; ґ) 10.

Які з цих циклів є простими?

38. Чи існує в графі K_5 цикл довжини 9? Відповідь обґрунтувати.

39. Скільки ребер містить:

а) простий ланцюг із k вершин;

б) простий цикл із k вершин;

в) найкоротший простий цикл?

40. Довести, що будь-який замкнений маршрут непарної довжини містить простий цикл. Чи справедливим є аналогічне твердження для замкнених маршрутів парної довжини?

41. Довести, що в графі G, степені всіх вершин якого не менше від k, існує ланцюг довжини не менше k.

42. Довести, що в графі, степені всіх вершин якого більші 1, є цикл.

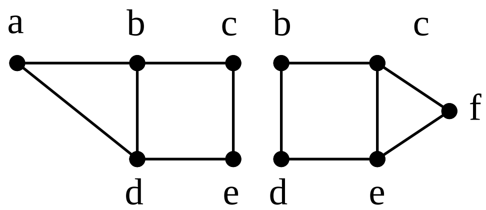
43. Довести, що зв'язний граф є простим циклом тоді і тільки тоді, коли кожна його вершина має степінь 2.

44. Довести, що зв'язний граф із n вершинами містить не менше ніж $n-1$ ребро.

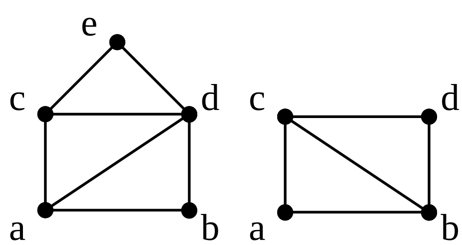
45. Довести, що коли в графі G з n вершинами кількість ребер більша ніж $(n-1)(n-2)/2$, тоді граф G зв'язний.

46. Знайти об'єднання та перетин графів:

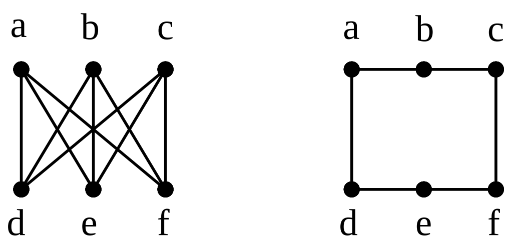
а)



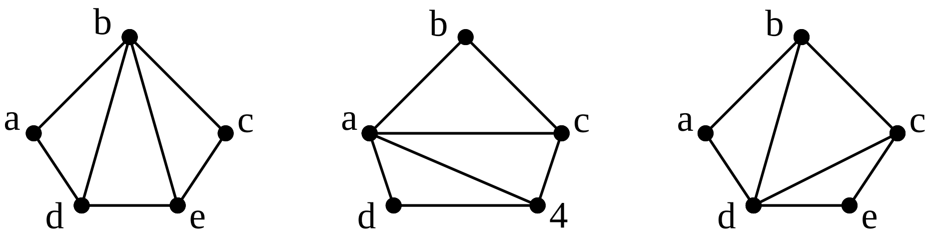
б)



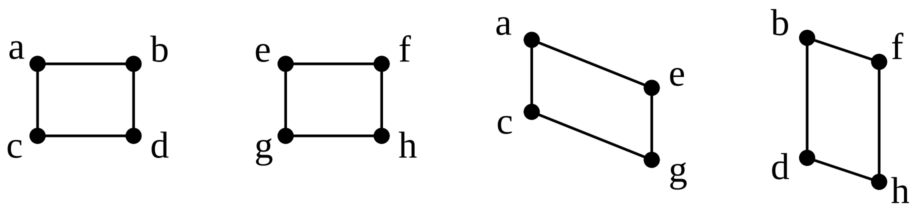
в)



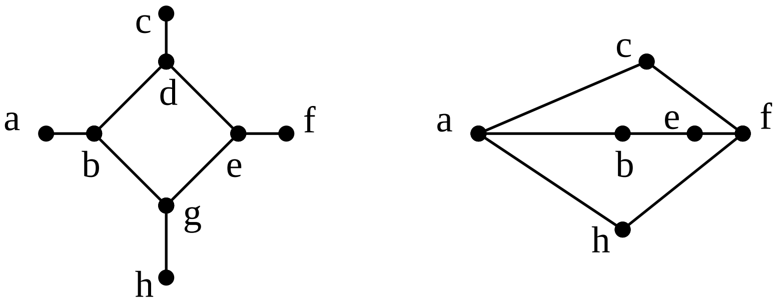
г)



ґ)

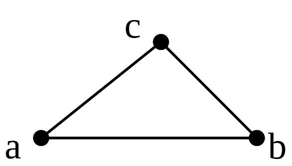


д)

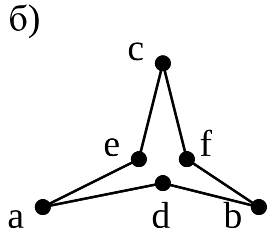


47.Знайти доповнення графів:

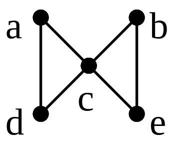
а)



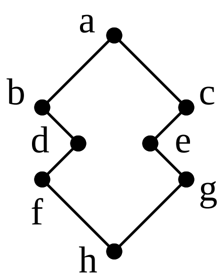
б)



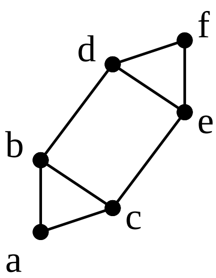
в)



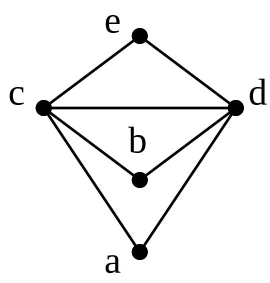
г)



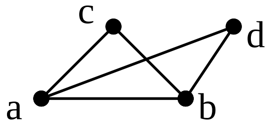
ґ)



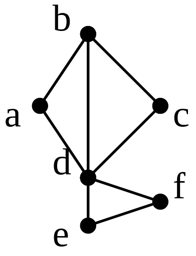
д)



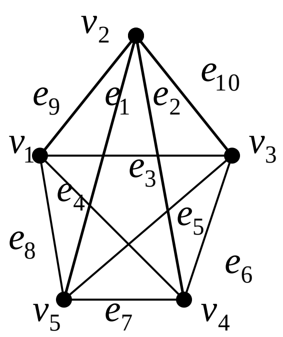
е)



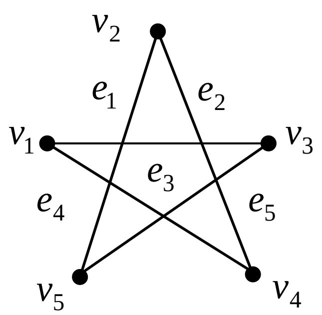
є)



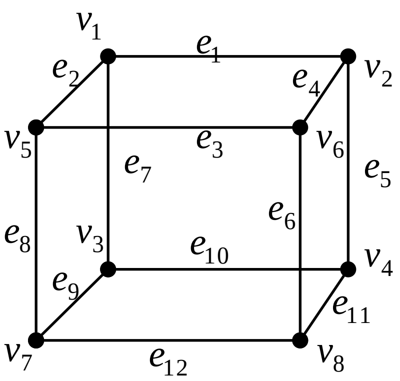
ж)



з)

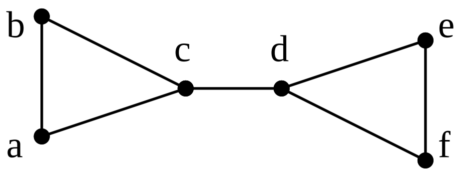


и)

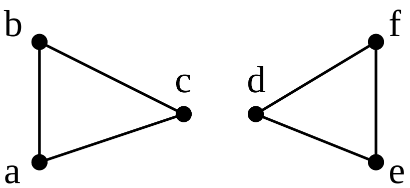


48. Задані графи G_1 і G_2. Які з наведених нижче графів є їх основними графами:

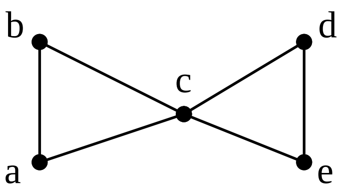
G_1



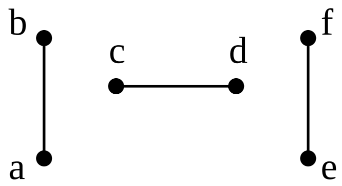
а)



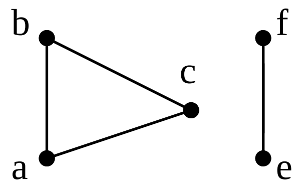
б)



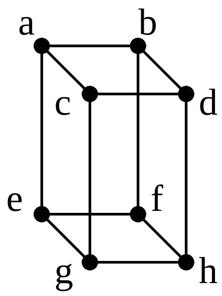
в)



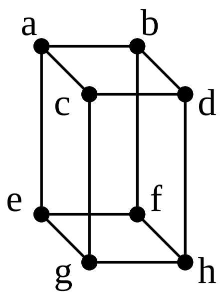
г)



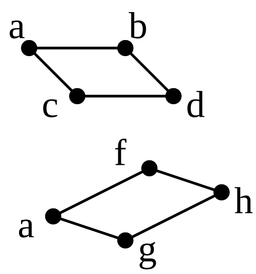
G_2



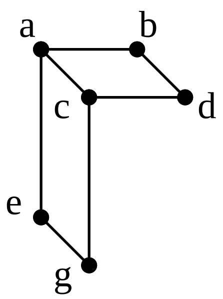
а)



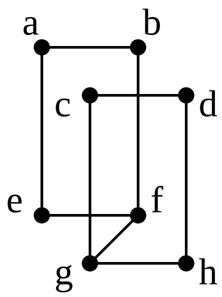
б)



в)



г)



§ 2. Ізоморфізм графів. Орієнтовані графи

О. 15. 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 $(\varphi(v), \varphi(w))$ належить W_2. Відображення φ називається *ізоморфним відображенням* або *ізоморфізмом* графа G_1 на граф G_2.

О. 16. Граф G, ізоморфний своєму доповненню $\overline{G}$, називається *самодоповнювальним*.

О. 17. *Реберним графом* $L(G)$ графа $G(X, W)$ називається граф, множиною вершин якого є множина ребер W графа G, а вершини x_1 і x_2 є суміжними в графі $L(G)$ тоді і тільки тоді, коли ребра w_1 і w_2 є суміжними в G.

О. 18. Нехай $X = \{X_1, \ldots, X_n\}$ – множина вершин, $\widetilde{M}_2$ – множина *впорядкованих* пар елементів з X (будемо називати їх *дугами*). *Орієнтованим графом* $\mathrm{G}(X, W)$ називатимемо пару множин (X, W), де $W \subset \widetilde{M}_2$.

Теорема 4. Число усіх орієнтованих графів з n вершинами дорівнює $2^{n(n-1)}$.

О. 19. Нехай $X = \{X_1, \ldots, X_n\}$ – множина вершин. *Орієнтованим графом з петлями* будемо називати пару множин (X, W), де $W \subset X \times X$.

Теорем 5. Число орієнтованих графів з петлями, які мають n вершин, дорівнює 2^{n^2}.

Теорема 6. Число всіх простих графів з n вершинами і петлями дорівнює $2^n 2^{\frac{n(n-1)}{2}} = 2^{C_{n+1}^2}$.

Завдання для самостійної роботи

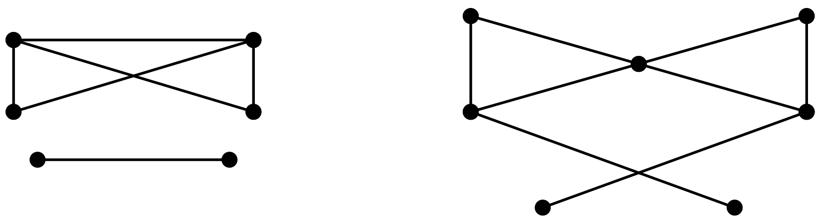
1. Довести, що всі ізоморфні графи мають однакову кількість вершин і однакову кількість ребер.
2. Пояснити, чому пари графів, зображені на рис. 3, не є ізоморфними.

а)

б)

в)

г)



ґ)

д)

е)

є)

Рис. 3

3. Довести, що пари графів, зображені на рис. 4, є ізоморфними.

а)

б)

в)

Рис. 4

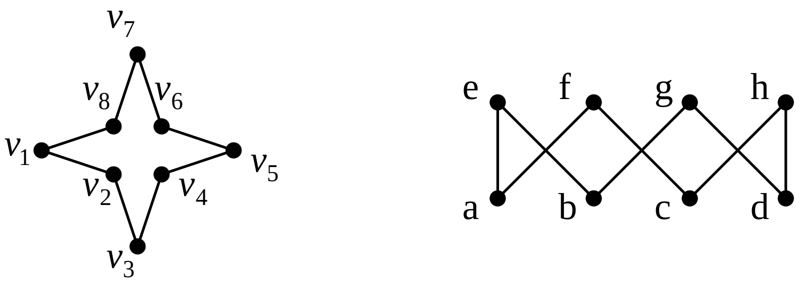
4. Чи є ізоморфними графи? Відповідь обґрунтувати.

а)

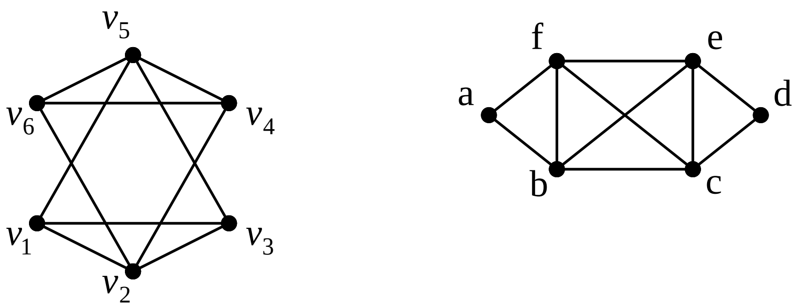
б)

в)

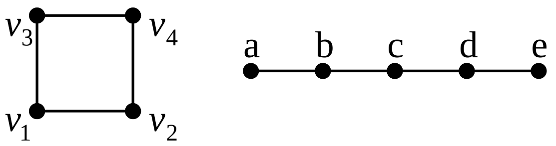
г)


v_7
v_8 v_6
v_1 v_5
v_2 v_4
v_3
e f g h
a b c d

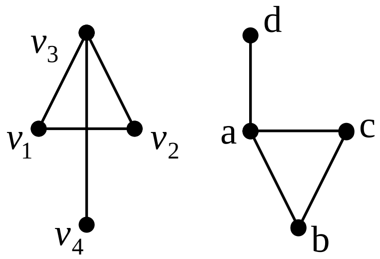
ґ)


v_5
v_6 v_4
v_1 v_3
v_2
f e
a d
b c

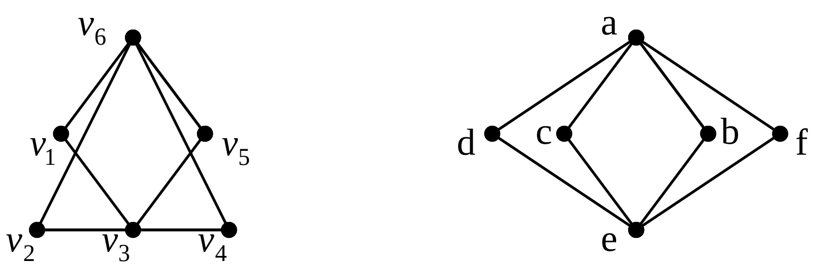
д)


v_3 v_4
v_1 v_2
a b c d e

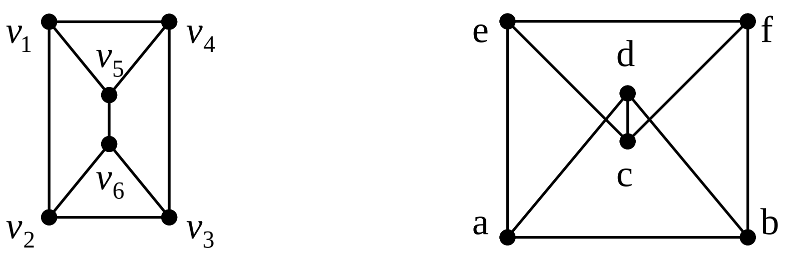
е)


v_3
v_1 v_2
v_4
d
a c
b

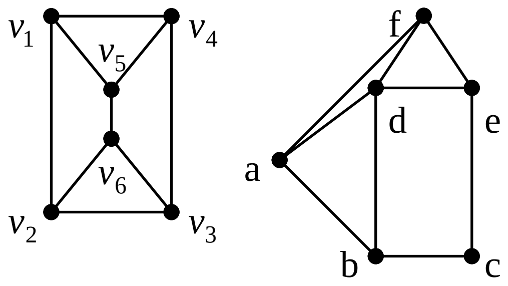
є)


v_6
v_1 v_5
v_2 v_3 v_4
a
d c b f
e

ж)

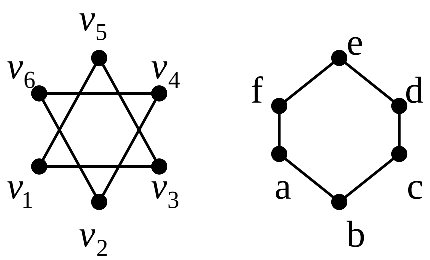

v_1 v_4
v_5
v_6
v_2 v_3
e f
d
c
a b

з)



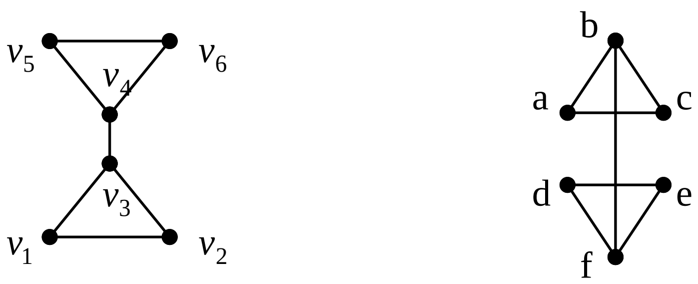
v_1 v_4 v_5 v_6 v_2 v_3
f d e a b c

и)



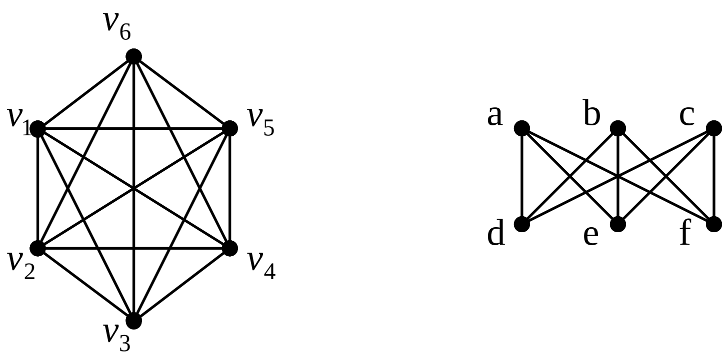
v_5 v_6 v_4 v_1 v_3 v_2
e f d a c b

і)



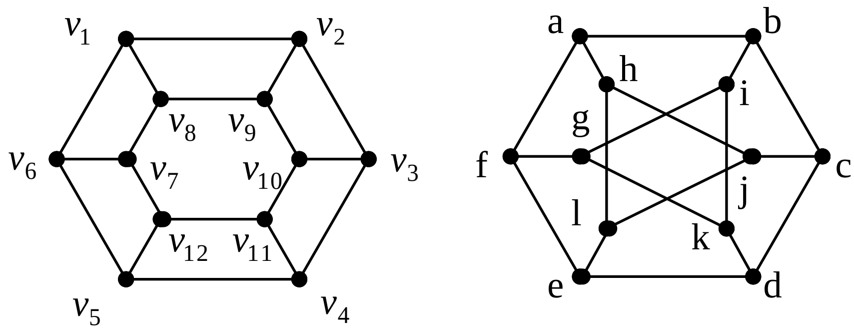
v_5 v_6 v_4 v_3 v_1 v_2
b a c d e f

ї)



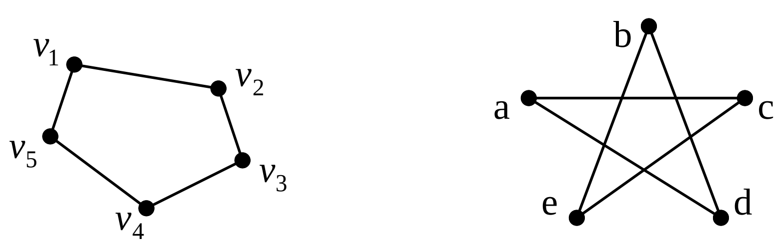
v_6 v_1 v_5 v_2 v_4 v_3
a b c d e f

й)



v_1 v_2 v_8 v_9 v_6 v_7 v_{10} v_3 v_{12} v_{11} v_5 v_4
a b h i g f c j l k e d

к)



v_1 v_2 v_5 v_3 v_4
b a c e d

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

6. Довести, що графи G_1 і G_2 ізоморфні тоді і тільки тоді, коли матрицю суміжності (матрицю інцидентності) одного з цих графів можна одержати з матриці суміжності (матриці інцидентності) іншого за допомогою відповідних перестановок рядків і стовпчиків.

7. Довести, що графи G_1 і G_2 ізоморфні тоді і тільки тоді, коли ізоморфні їхні доповнення $\overline{G_1}$ і $\overline{G_2}$.
8. Знайти нетривіальний самодоповнювальний граф із найменшою кількістю вершин.
9. Довести, що існує тільки один самодоповнювальний граф із чотирма вершинами.
10. Довести, що існує тільки два самодоповнювальні графи з п'ятьма вершинами.
11. Чи існує самодоповнювальний граф, у якого кількість ребер дорівнює:
    а) 5; б) 7; в) $4k^2 - 2,\ k \in N$?
12. Чи існує самодоповнювальний граф, у якого кількість ребер дорівнює:
    а) 6; б) 7; в) $4k - 1,\ k \in N$?
13. Довести, що кількість вершин будь-якого самодоповнювального графа дорівнює або $4k,$ або $4k+1,\ k \in N.$
14. Довести, що довільний самодоповнювальний граф містить або $4k^2 - k,$ або $4k^2 + k$ ребер, $k \in N.$
15. Побудувати чотири попарно неізоморфні самодоповнювальні графи з вісьмома вершинами.
16. Нарисувати всі попарно неізоморфні самодоповнювальні графи з n вершинами для:
    а) $n = 2$; б) $n = 3$; в) $n = 4$; г) $n = 5$.
17. Побудувати всі попарно неізоморфні графи із шістьма вершинами, степені яких дорівнюють $2,\ 2,\ 3,\ 3,\ 3,\ 5.$
18. Скільки існує попарно неізоморфних графів, які мають:
    а) 6 вершин і 11 ребер;
    б) 7 вершин і 18 ребер;
    в) 8 вершин і 24 ребра;
    г) 8 вершин, сума всіх степенів яких не менша від 53;
    ґ) 10 вершин і 43 ребра;
    д) n вершин і $n(n-1)/2 - 2$ ребра?
19. Побудувати реберні графи для графів, зображених на рис. 3.
20. Побудувати реберні графи для графів K_2, K_3, K_4, K_5, $K_{3,3}$ і $K_{1,3}$.
21. Довести, що реберні графи графів K_3 та $K_{1,3}$ ізоморфні.
22. Визначити кількість вершин та степені вершин для реберного графа повного графа K_n.
23. Побудувати три попарно неізоморфні графи, які ізоморфні своїм реберним графам.
24. Довести, що граф G є ізоморфний своєму реберному графу $L(G)$ тоді і тільки тоді, коли степені всіх вершин графа G дорівнюють 2.

§ 3. Обходи графів. Ейлерові та гамільтонові графи

О. 20. Зв'язний граф називається *ейлеровим графом*, якщо існує замкнений ланцюг, який проходить через кожне ребро. Такий ланцюг називатимемо *ейлеровим ланцюгом*, або *ейлеровим циклом*.

О. 21. Граф називається напівейлеровим, якщо існує ланцюг, який проходить через кожне його ребро рівно один раз.

Лема 2. Якщо степінь кожної вершини графа $\mathrm{G}(X,W)$ не менше двох, то граф містить цикл.

Теорема 7. Для зв'язного графа $\mathrm{G}(X,W)$ наступні умови еквівалентні:

1. $\mathrm{G}(X,W)$ – ейлерів граф;
2. кожна вершина $\mathrm{G}(X,W)$ має парний степінь;
3. множину ребер графа $\mathrm{G}(X,W)$ можна розбити на прості цикли.

Теорема 8. Зв'язний граф є ейлеровим тоді і тільки тоді, коли кожна його вершина має парний степінь.

Теорема 9. Зв'язний граф є напівейлеровим тоді і тільки тоді, коли в ньому не більше двох вершин непарного степеня.

О. 22. 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.*

О. 23. Зв'язний граф називається *напівгамільтоновим*, якщо існує ланцюг, який проходить через кожну його вершину рівно один раз.

Теорема 10. (О. Оре, 1960) Якщо для будь-якої пари u і v несуміжних вершин графа $\mathrm{G}(X,W)$ з n вершинами $(n \geq 3)$ виконується нерівність

$$d(u) + d(v) \geq n,$$

то граф $\mathrm{G}(X,W)$ гамільтонів.

Теорема 11. (Г. Дірак, 1952 р.) Якщо для будь-якої пари u і v несуміжних вершин графа $\mathrm{G}(X,W)$ з n вершинами $(n \geq 3)$ виконується нерівність $d(u) \geq \frac{n}{2}$, то граф $\mathrm{G}(X,W)$ гамільтонів.

Завдання для самостійної роботи

1. Побудувати три попарно неізоморфні графи з вісьмома вершинами, які мають ейлерові цикли.
2. Переконатись у тому, що графи, зображені на рис. 5, ейлерові.

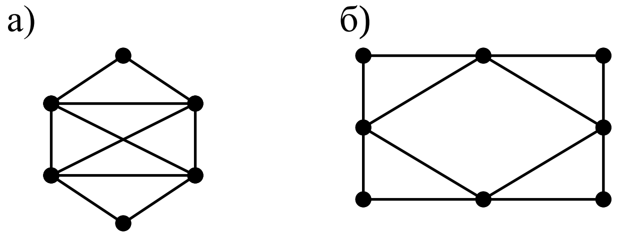


Рис. 5

3. Визначити, які з повних графів K_n є ейлеровими?
4. Побудувати граф із вісьмома вершинами, який не має ейлерового циклу, але має ейлерів ланцюг.
5. На рис. 6 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.

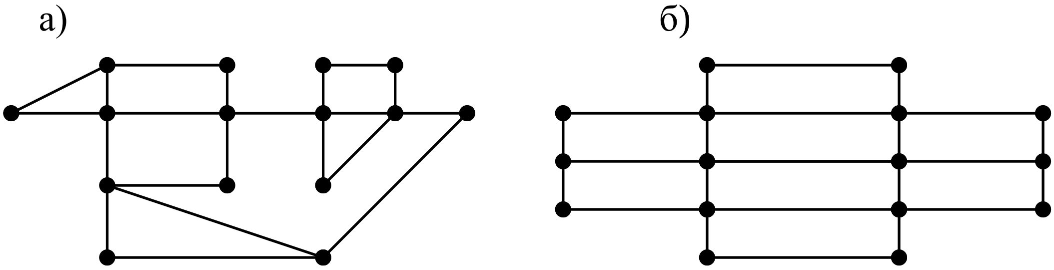


Рис. 6

6. На рис. 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.

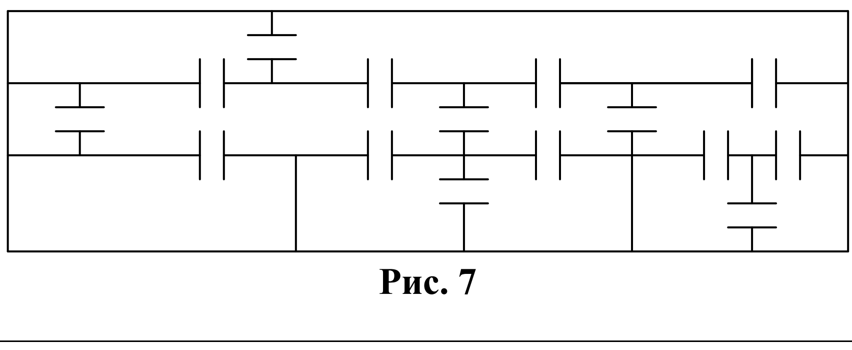


Рис. 7

7. Довести, що для довільного зв'язного графа існує маршрут, який містить усі ребра графа, причому кожне з них не більше двох разів.
8. Знайти гамільтонові цикли в графах, зображених на рис. 8.

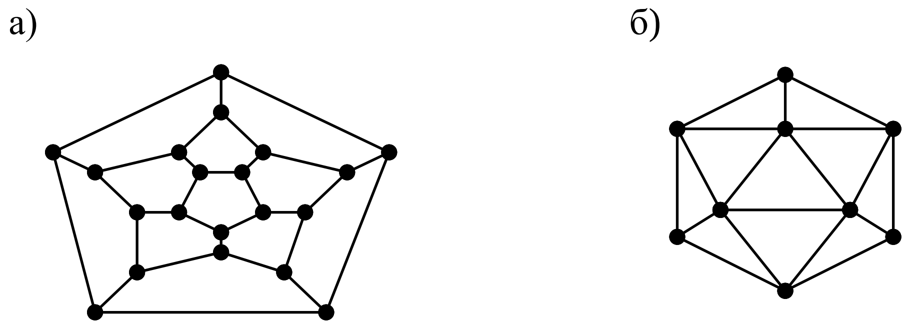
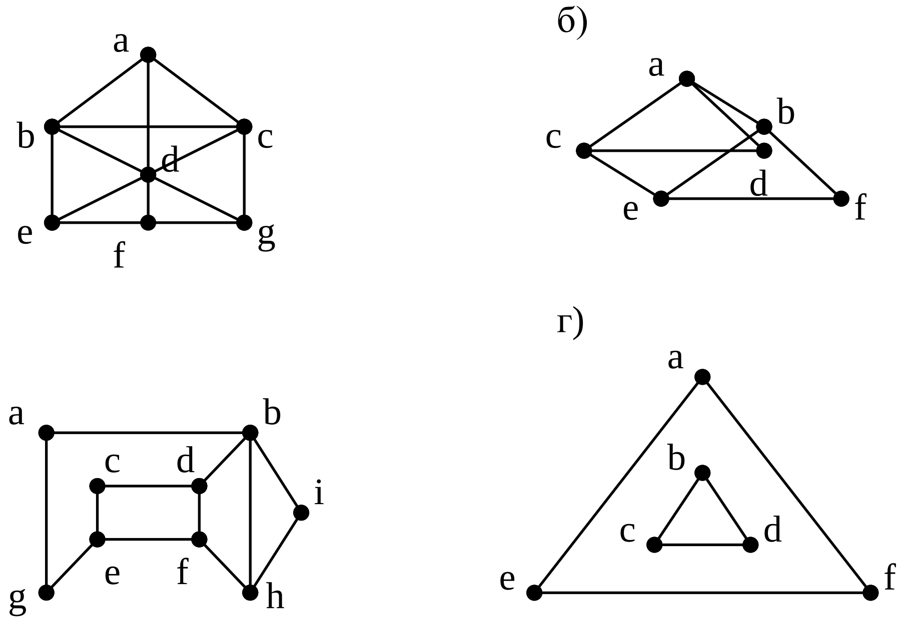
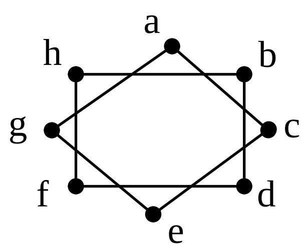


Рис. 8

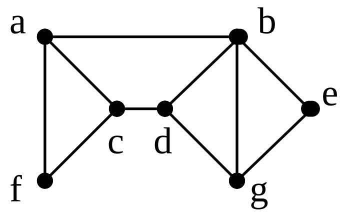
9. Довести, що в повному графі K_n існує гамільтонів цикл для довільного $n > 3$.
10. Занумеруємо вершини графа K_n.
- скільки різних ейлерових циклів має цей граф?
- скільки різних гамільтонових циклів має цей граф?
11. В кожному з наведених графів знайти гамільтонів цикл (якщо він існує).



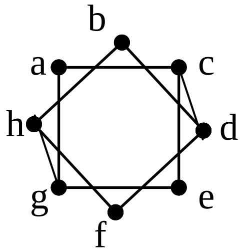
ґ)



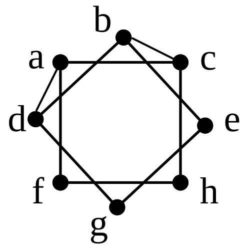
д)



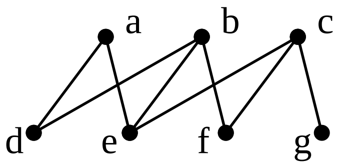
е)



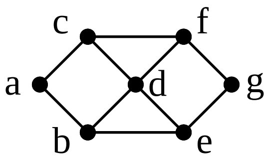
є)



ж)

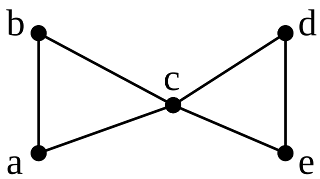


з)

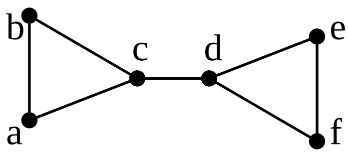


12.Знайти гамільтонів шлях (якщо він існує) в кожному з графів:

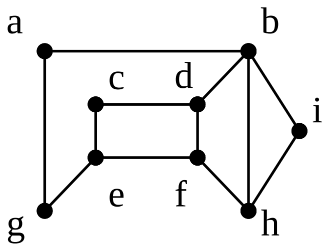
а)



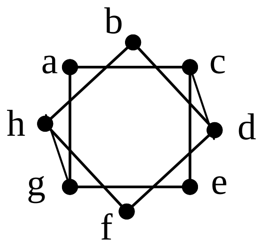
б)



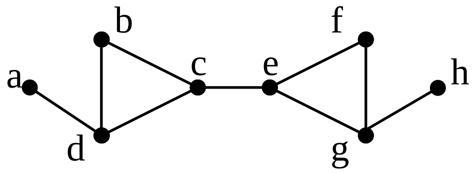
в)



г)



ґ)



13. Навести приклади графа, який є ейлеровим, але не є гамільтоновим, а також гамільтонового графа, який не є ейлеровим.
14. Охарактеризувати графи, які є одночасно ейлеровими і гамільтоновими.
15. Яким повинен бути граф G, щоб його реберний граф $L(G)$ був ейлеровим?
16. Довести, що коли граф G ейлерів, тоді реберний граф $L(G)$ є ейлеровим і гамільтоновим графом.
17. Довести, що коли G – гамільтонів граф, тоді реберний граф $L(G)$ є також гамільтоновим графом.
18. Побудувати граф, який не має ейлерового циклу, але такий, що його реберний граф $L(G)$
а) ейлерів;
б) гамільтонів.
19. Побудувати граф G, який не має гамільтонового циклу, але такий, що його реберний граф $L(G)$ є гамільтоновим.
20. Довести, що реберний граф $L(G)$ є гамільтоновим тоді і тільки тоді, коли граф G має простий цикл, який містить принаймні по одній вершині з кожного ребра графа G.
21. Довести, що для зв'язного графа G реберний граф $L(G)$ є ейлеровим тоді і тільки тоді, коли степені всіх вершин графа мають однакову парність.

§ 4. Дерева. Ліс

О. 24. *Лісом* називається граф, який не містить циклів. Зв'язний ліс називається *деревом.*

Теорема 12. Нехай $\mathrm{G}(X,W)$ граф, що має n вершин та m ребер. Тоді такі умови еквівалентні:

1. $\mathrm{G}(X,W)$ – дерево;
2. $\mathrm{G}(X,W)$ – зв'язний граф і $m=n-1$;
3. $\mathrm{G}(X,W)$ – ациклічний граф і $m=n-1$;
4. будь-які дві вершини графа $\mathrm{G}(X,W)$ сполучаються єдиним простим ланцюгом;
5. $\mathrm{G}(X,W)$ – ациклічний граф, який має властивість: сполучивши ребром будь-яку пару його несуміжних вершин, отриманий граф міститиме рівно один цикл.

О. 25. Вершина $v \in W$ називається кінцевою вершиною, якщо $d(v) = 1$.

Теорема 13. У будь-якому дереві з n вершинами, $(n \geq 2)$ є принаймні дві кінцеві вершини.

Теорема 14. Число різних дерев, які можна побудувати на n даних вершинах, дорівнює $t_n = n^{n-2}$.

Завдання для самостійної роботи

1. Побудувати всі попарно неізоморфні зв'язні графи без циклів із п'ятьма вершинами. Скільки ребер мають ці графи?
2. Побудувати зв'язний граф із сімома вершинами й шістьма ребрами.
3. Чи може зв'язний граф із n вершинами і $n-1$ ребром мати цикл?
4. Побудувати всі попарно неізоморфні дерева, які мають:
   а) 6 ребер та 3 кінцеві вершини;
   б) 6 ребер та 4 кінцеві вершини;
   в) 7 ребер та 3 кінцеві вершини;
   г) 8 ребер та 3 вершини степеня 3.
5. Побудувати три попарно неізоморфні дерева, в яких для будь-якого $k > 0$ кількість вершин степеня k однакова.
6. Довести, що будь-яке дерево з n вершинами $(n > 2)$ має принаймні дві кінцеві вершини.
7. Довести, що дерево має тільки дві кінцеві вершини тоді і тільки тоді, коли воно є простим ланцюгом.
8. Довести, що після вилучення в дереві будь-якої кінцевої вершини отримуємо дерево.
9. Яку найбільшу та яку найменшу кількість кінцевих вершин може мати дерево з n вершинами? Яку структуру мають відповідні дерева?
10. Описати всі дерева, доповнення яких також є деревами.
11. Описати всі дерева, які є самодоповнювальними графами.
12. Довести, що в дереві T кожне ребро є мостом.
13. Довести, що ліс, який має n вершин і складається з k дерев, містить $n-k$ ребер.

§ 5. Планарність графів

О. 26. *Плоским графом* називається граф, у діаграмі якого лінії, що відповідають ребрам, перетинаються лише в точках, які відповідають вершинам графа.

О. 27. *Планарним графом* називається граф G, ізоморфний деякому плоскому графу. Останній називається *плоскою картою* графа G.

О. 28. *Внутрішньою гранню* плоского зв'язного графа називається скінченна область площини, що обмежена замкненим маршрутом графа і не містить усередині ні вершин, ні ребер графа. Частина площини, яка складається з точок, що не належать жодній внутрішній грані, називається *зовнішньою гранню*.

О. 29. Множина всіх граней плоского зв'язного графа позначається P. Замкнений маршрут, що обмежує грань, називається *межею грані*, а довжина цього маршруту – *степенем грані*. Степінь грані $r \in P$ позначається P_r.

О. 30. *Максимальним планарним графом* називається планарний граф, який при додаванні до нього будь-якого ребра перестає бути планарним.

О. 31. Плоский зв'язний граф, кожна грань якого (включаючи й зовнішню) обмежена трикутником, називається *триангуляцією.*

Завдання для самостійної роботи

1. Переконатись у тому, що всі графи, зображені на рис. 3, є планарні. Побудувати для них плоскі карти.
2. Показати, що всі графи, зображені на рис. 9, планарні.

а) б) в)

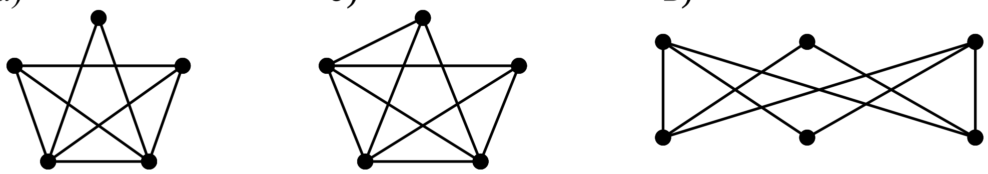
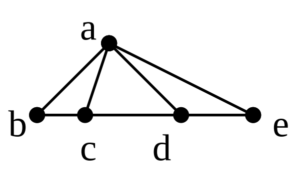


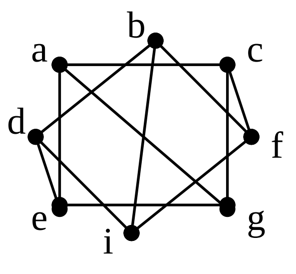
Рис. 9

3. Чи існує непланарний граф із чотирма вершинами?
4. Перевірити графи на планарність:

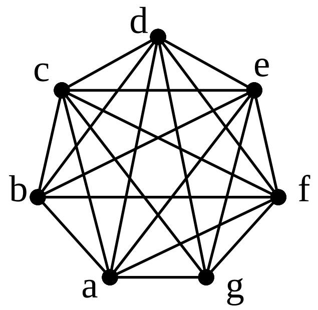
а)



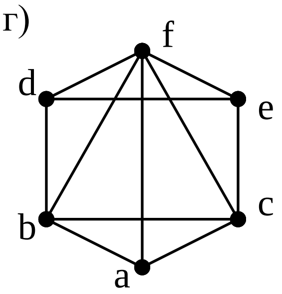
б)



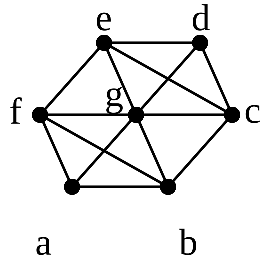
в)



г)

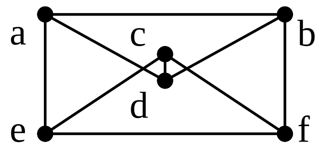


ґ)

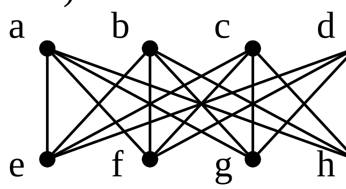


5. Для кожного планарного графа перевірити рівність $|X| - |W| + |P| = 2$.
6. Чи є планарними графи? Відповідь аргументувати.

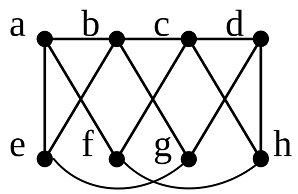
а)



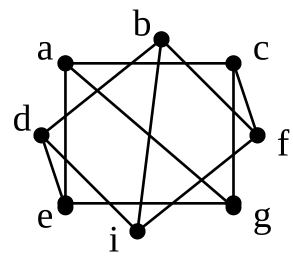
б)



в)



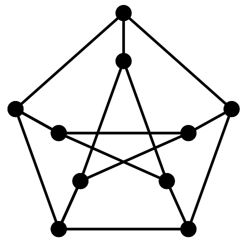
г)



7. Скільком граням може належати вершина степеня k плоского графа?
8. Довести, що будь-яке дерево є планарним графом. Скільки граней має дерево?
9. Довести, що для будь-якого зв'язного планарного графа $G(X, W)$ виконується рівність $|X| - |W| + |P| = 2$ (*теорема Ейлера*).
10. Знайти зв'язний плоский граф із n вершинами і m ребрами, для якого $m > 3n - 6$.
11. Чи існує планарний граф, який має:
    а) 7 вершин і 16 ребер;
    б) 8 вершин і 17 ребер?

12. Яку найбільшу кількість граней може мати плоский граф із п'ятьма вершинами? Побудувати цей граф.
13. Чи існує плоский граф із шістьма вершинами, що має 9 граней?
14. Побудувати всі попарно неізоморфні плоскі графи з шістьма вершинами, що мають 8 граней.
15. Довести, що повний граф із п'ятьма вершинами K_5 не є планарним.
16. Побудувати всі непланарні графи з 6 вершинами та 11 ребрами.
17. Яку найменшу кількість вершин потрібно вилучити з графів, зображених на рис. 10, щоб кожен із них перетворився в планарний граф?

а)



б)

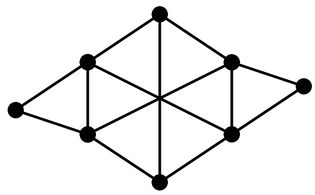


Рис. 10

18. Яку найменшу кількість вершин потрібно вилучити з графів, зображених на рис. 10, щоб вони стали планарними?
19. Довести, що для довільного графа G з n вершинами при $n < 8$ принаймні один із графів – G або $\overline{G}$ є планарним.
20. Довести, що графи G та $\overline{G}$ не можуть бути одночасно планарними, якщо кількість вершин у них не менше 11.
21. Побудувати планарний граф із вісьмома вершинами, доповнення якого є також планарним графом.
22. Довести, що в будь-якому планарному графі є принаймні одна вершина, степінь якої не перевищує 5.
23. Побудувати плоский граф, що має чотири вершини, степені яких не перевищують 5.
24. Довести, що існує тільки одна триангуляція з чотирма вершинами.
25. Довести, що існує тільки одна триангуляція з п'ятьма вершинами.
26. Побудувати дві неізоморфні триангуляції з шістьма вершинами.
27. Чи можна до плоского графа G, зображеного на рис. 11, додати нові ребра так, щоб отриманий граф залишився плоским? Якщо можна, то які ребра і скільки?

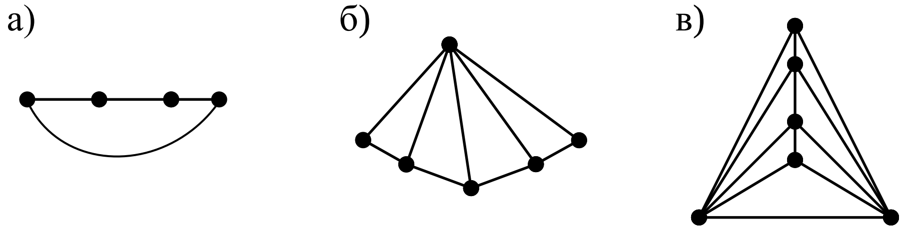


Рис. 11

28. Довести, що для будь-якого планарного графа існує плоска карта, усі ребра якої є відрізками прямих ліній.

ВАРІАНТИ ЗАВДАНЬ ЗАКЛЮЧНОГО СЕМІНАРСЬКОГО ЗАНЯТТЯ

Варіант 1

1. В бюро по туризму складаються маршрути, за якими треба проїхати з пункту S у пункт R і по дорозі подивитися місцеві визначні пам'ятки. Пункти і шосейні дороги представлені схемою (рис. 1). Скласти такий маршрут, щоб туристи в кожний з пунктів попадали тільки один раз. Чи існує такий маршрут? Скільки може бути таких доріг? Виписати послідовність пунктів для кожного маршруту.

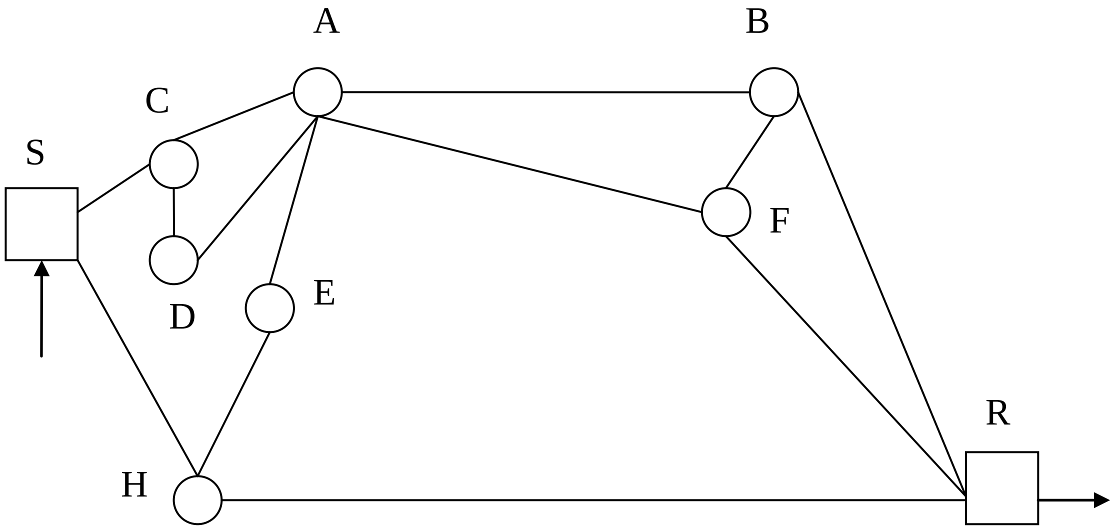


Рис. 1

2. Скільки вийде шматків паперу, якщо спочатку було 4 шматки, деякі розрізали на 4 частини, а всього було розрізано 15 шматків?
3. Намалювати граф із сімома вершинами і шістьма ребрами, який не має жодного циклу.

4. Із графа G (рис. 2) вилучити частину ребер так, щоб новий граф був деревом, яке містить усі вершини графа G.

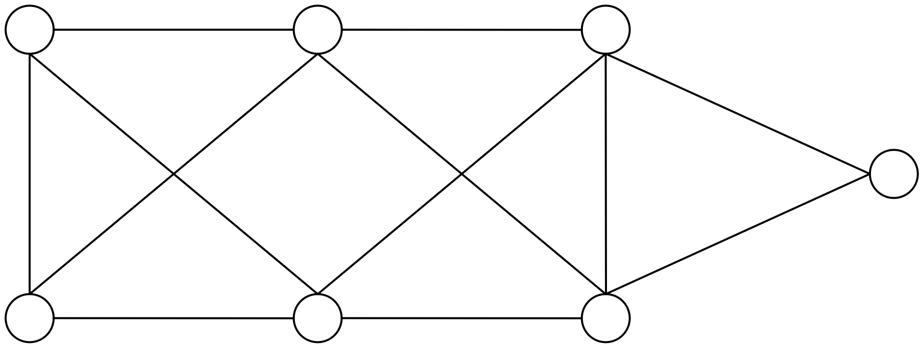
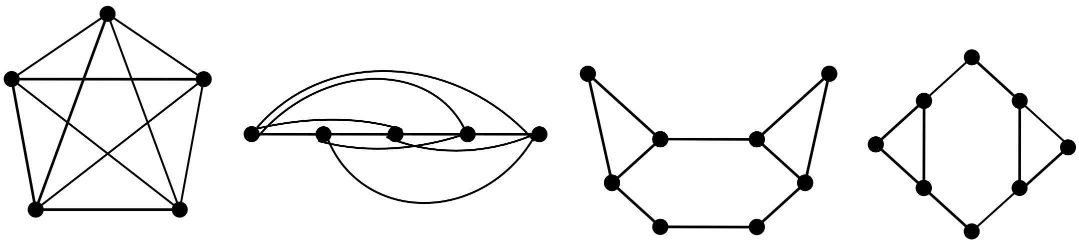


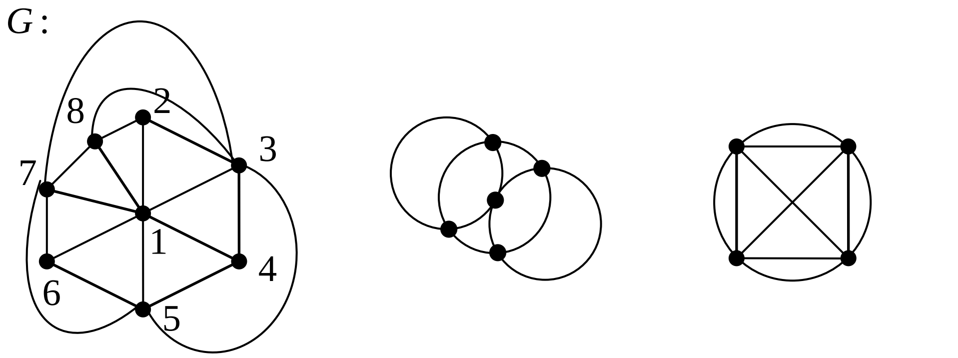
Рис. 2

5. Чи ізоморфні дані пари графів?

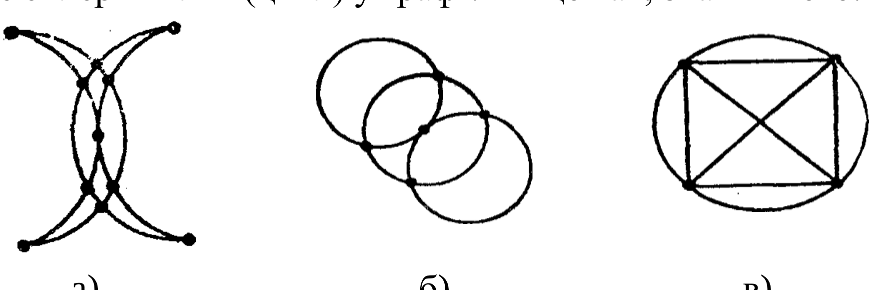


а) б) в) г)

6. Намалювати плоске представлення з прямолінійними ребрами графа G.

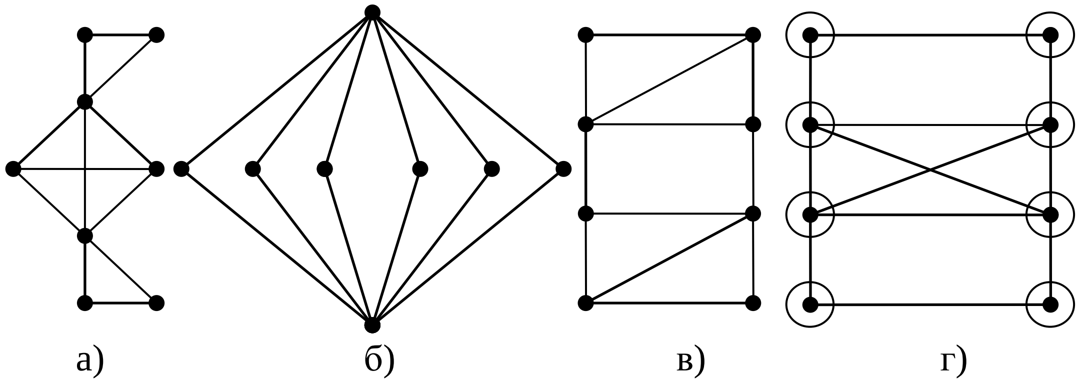


7. Чи існує ейлерів шлях (цикл) у графі? Якщо так, знайти його.



а) б) в)

8. Намалювати граф із вісьмома вершинами, який: а) має ейлерів цикл; б) має ейлерів шлях; в) не має ні ейлерового циклу, ні ейлеревого шляху; г) має простий шлях, який містить усі ребра графа.
9. Навести приклад графа з 6 (7) ребрами, в якому є замкнений маршрут, що містить усі ребра в точності 2 рази і: а) є маршрут, що містить усі ребра в точності 3 рази; б) не має маршруту, який містить усі ребра по 3 рази.
10. Який з графів а)-г) є ейлеровим або гамільтоновим?



Варіант 2

1. В бюро по туризму складаються маршрути, за якими треба проїхати з пункту S у пункт R і по дорозі подивитися місцеві визначні пам'ятки. Пункти і шосейні дороги представлені схемою (рис. 1). Скласти такий маршрут, щоб туристи в кожний з пунктів попадали тільки один раз. Чи існує такий маршрут? Скільки може бути таких доріг? Виписати послідовність пунктів для кожного маршруту.

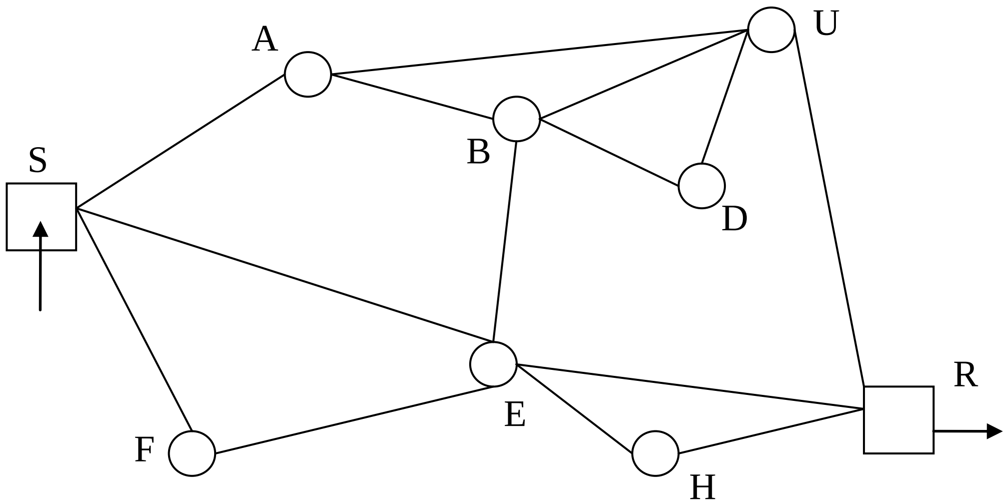
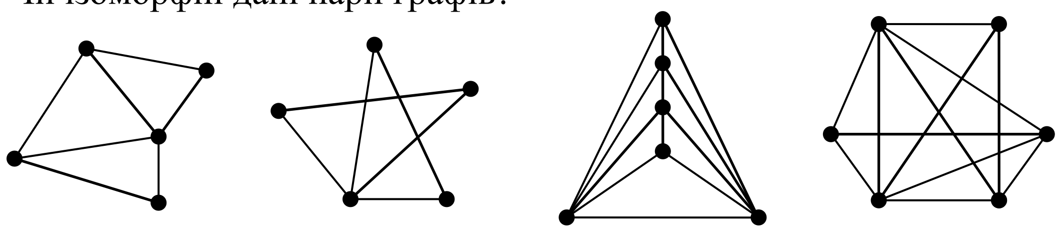


Рис. 1

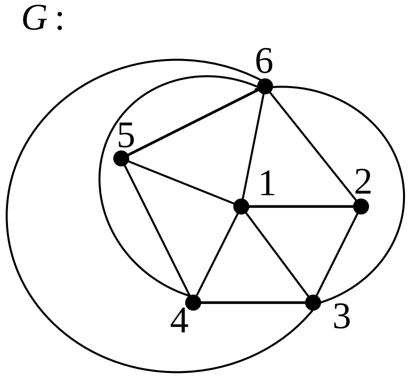
2. Скільки вийде шматків паперу, якщо спочатку було 4 шматки, деякі розрізали на 4 частини, а всього було розрізано 13 шматків?

3. Намалювати граф із сімома вершинами, в якому для будь-яких вершин існує один і тільки один шлях, що їх сполучає.
4. Скільки ребер треба видалити зі зв'язаного графа, який має W ребер і V вершин, щоб одержати дерево, яке містить усі вершини цього графа?
5. Чи ізоморфні дані пари графів?

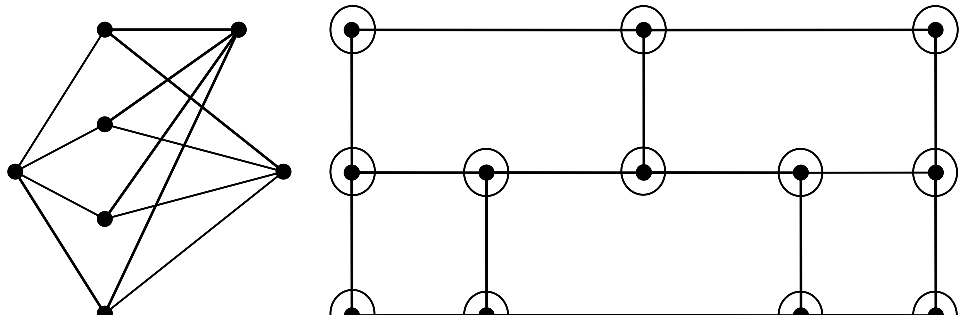


а) б) а) б)

6. Намалювати плоске представлення з прямолінійними ребрами графа G.



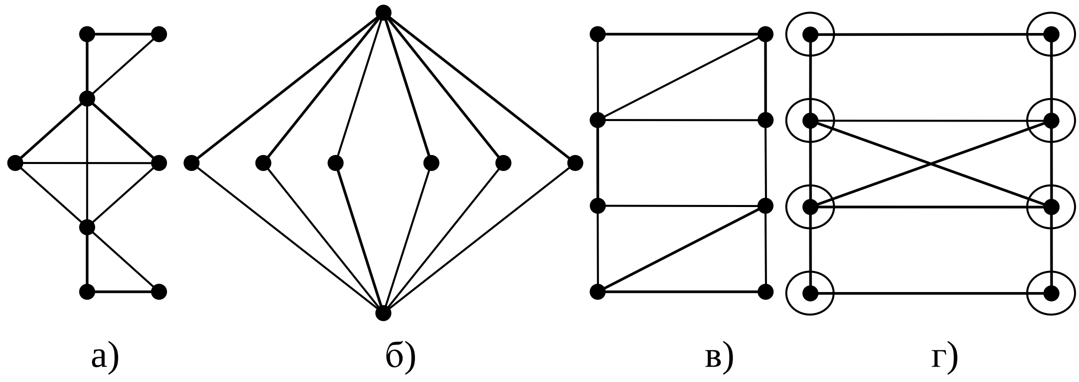
7. Чи існує ейлерів шлях (цикл) у графі? Якщо так, знайти його.



а) б)

8. Намалювати граф із вісьмома вершинами, який: а) має ейлерів цикл; б) має ейлерів шлях; в) не має ні ейлерового циклу, ні ейлерового шляху; г) має простий шлях, який містить усі ребра графа.
9. Навести приклад графа з 6 (7) ребрами, в якому є замкнений маршрут, що містить усі ребра в точності 2 рази і: а) є маршрут, що містить усі ребра в точності 3 рази; б) не має маршруту, який містить усі ребра по 3 рази.

10.Який з графів а)-г) є ейлеровим або гамільтоновим?



Варіант 3

1. В бюро по туризму складаються маршрути, за якими треба проїхати з пункту S у пункт R і по дорозі подивитися місцеві визначні пам'ятки. Пункти і шосейні дороги представлені схемою (рис. 1). Скласти такий маршрут, щоб туристи в кожний з пунктів попадали тільки один раз. Чи існує такий маршрут? Скільки може бути таких доріг? Випивати послідовність пунктів для кожного маршруту.

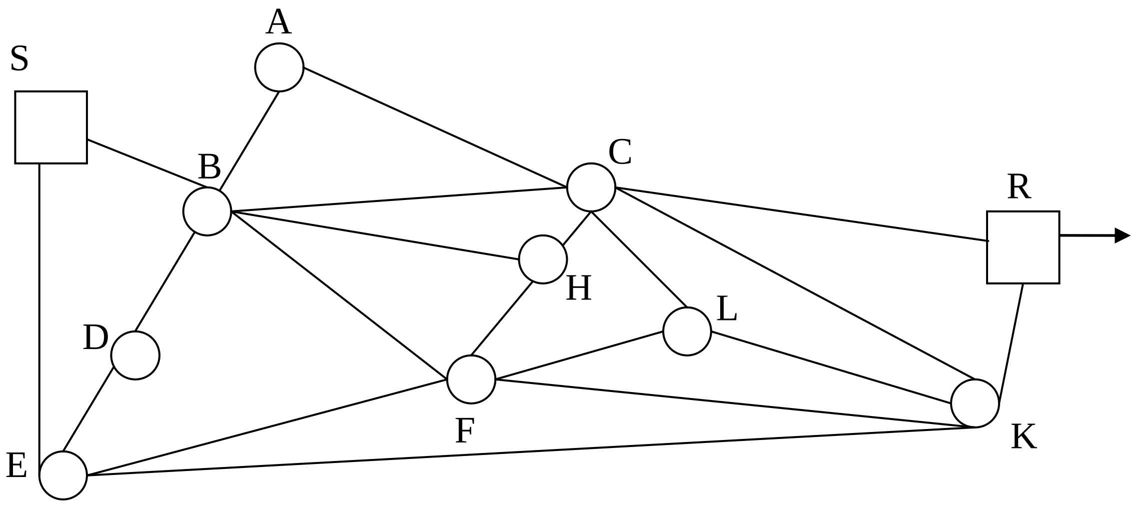
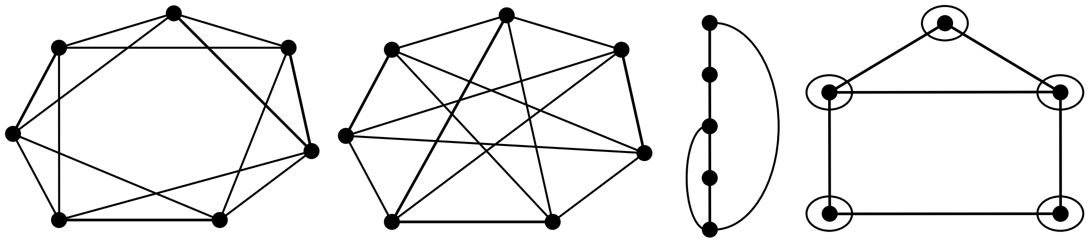


Рис. 1

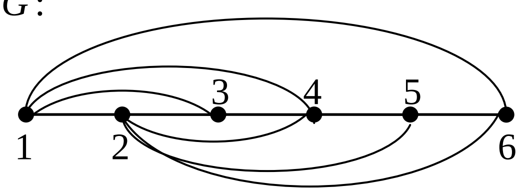
2. Скільки вийде шматків паперу, якщо спочатку було 5 шматків, деякі розрізали на 3 частини, а всього було розрізано 14 шматків?
3. Побудувати зв'язний граф із сімома вершинами, кожне ребро якого – міст.
4. Навести приклад графа, із якого не можна виділити дерево, яке містить усі вершини графа.
5. Чи ізоморфні дані пари графів?



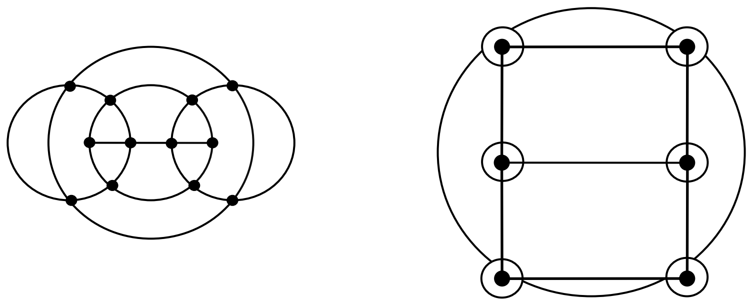
а) б) б) а)

6. Намалювати плоске представлення з прямолінійними ребрами графа G.

G:

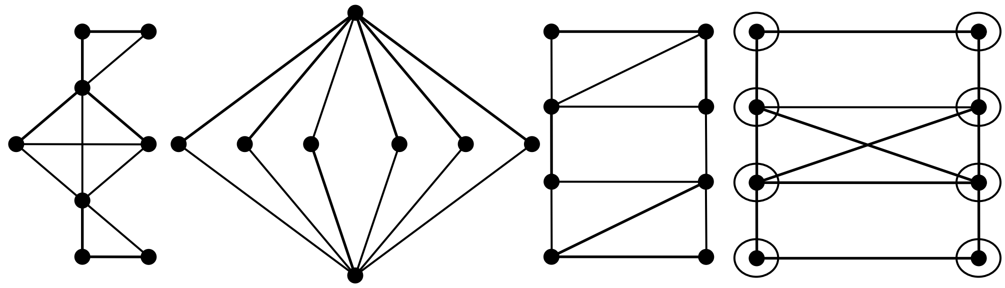


7. Чи існує ейлерів шлях (цикл) у графі? Якщо так, знайти його.



а) б)

8. Намалювати граф із вісьмома вершинами, який: а) має ейлерів цикл; б) має ейлерів шлях; в) не має ні ейлерового циклу, ні ейлерового шляху; г) має простий шлях, який містить усі ребра графа.
9. Навести приклад графа з 6 (7) ребрами, в якому є замкнений маршрут, що містить усі ребра в точності 2 рази і: а) є маршрут, що містить усі ребра в точності 3 рази; б) не має маршруту, який містить усі ребра по 3 рази.
10. Який з графів а)-г) є ейлеровим або гамільтоновим?



а) б) в) г)

РОЗДІЛ II. МЕРЕЖНІ СИСТЕМИ

§ 1. Основні поняття та задачі теорії мережних систем

Узагальненням поняття графа є поняття мережі.

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).

Мережі на практиці, природно, використовуються набагато частіше, ніж графи як такі, тому що ваги ребер чи дуг визначають процес функціонування об'єкта, представленого мережею.

Усі поняття, введені для графів, стосуються і мереж. Мережі, як і графи, можуть бути орієнтованими або ні. Вигляд орієнтованої мережі зображено на рис. 1. На ньому поруч з кожною дугою проставлено її вагу.

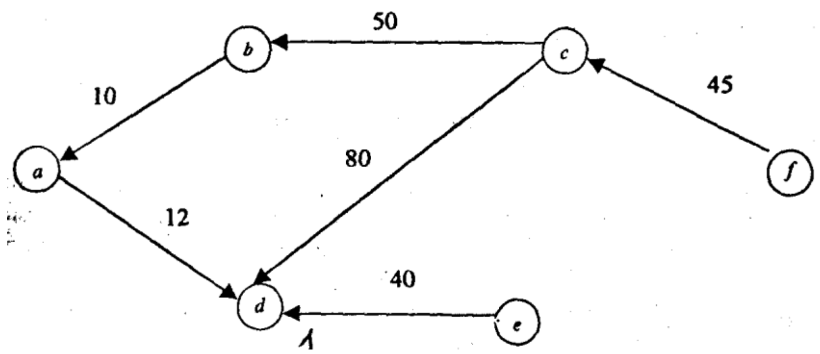


Рис. 1. Зображення мережі

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. Зрозуміло, що для будь-якого зв'язного неорієнтованого графа покривне дерево завжди існує. Для зв'язного орієнтованого – далеко не завжди (рис. 2). Для орієнтованого графа на цьому рисунку не існує покривного дерева, хоча за відсутності стрілок на дугах воно складалося б з ребер (1, 2), (2, 3) і (3,4).

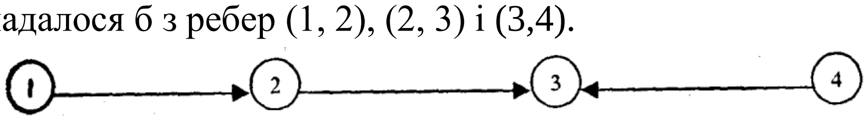


Рис. 2. Приклад графа, для якого не існує покривного дерева

Тому для орієнтованих графів на першому місці є поняття лісу, а не дерева. Для того, щоб чітко визначитися в термінах, введемо їх окремо для орієнтованих графів.

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.

На рис. 2, як вже зазначалося, показаний граф, що не має покривного дерева. Покривний же ліс для нього складається з двох дерев: дерева (1, 2), (2, 3) і дерева (4, 3). Останнє дерево містить тільки одну дугу, але це не заважає йому бути відповідно до вищенаведеного визначення деревом.

Граф, що має покривне дерево, є однозв’язним. Справджується і протилежне: для того, щоб граф був однозв’язним, він має мати покривне дерево.

У загальному випадку граф може мати кілька покривних дерев. Особливо показове це явище для неорієнтованих графів, хоча можливе і для орієнтованих. Для того, щоб це побачити, розглянемо граф на рис. 3.

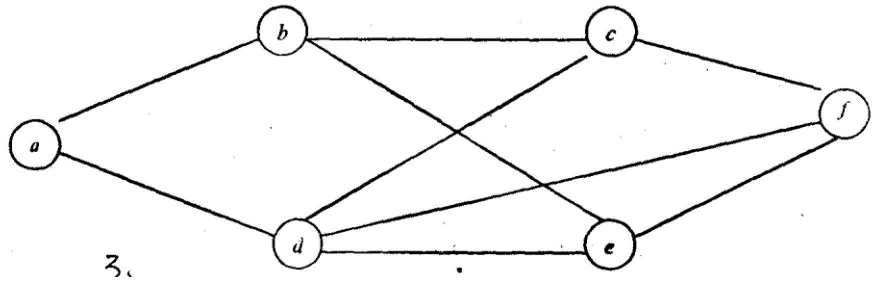


Рис. 3. Приклад графа, для якого існує кілька покривних дерев

Для графа на рис. 3 покривними будуть і дерево (*a, b*), (*b, c*), (*c, f*), (*f, e*), (*e, d*), і дерево (*a, b*), (*b, c*), (*c, d*), (*d, f*), (*f, e*). Можна знайти й інші покривні дерева. Пропонуємо перевірити це.

Кожне дерево можна представити (розгорнути) таким чином, щоб воно мало вигляд відомого в історичному плані “генеалогічного” дерева, за допомогою якого зображають наявність родинних зв’язків. Вигляд генеалогічного дерева зображено на рис. 4.

Так само відбивають різну ієрархію і супідрядність для широкого кола реальних об’єктів. При цьому для неорієнтованого дерева як корінь можна вибрати будь-яку вершину, для орієнтованого – одну, а саме – ту, в котру не заходять дуги.

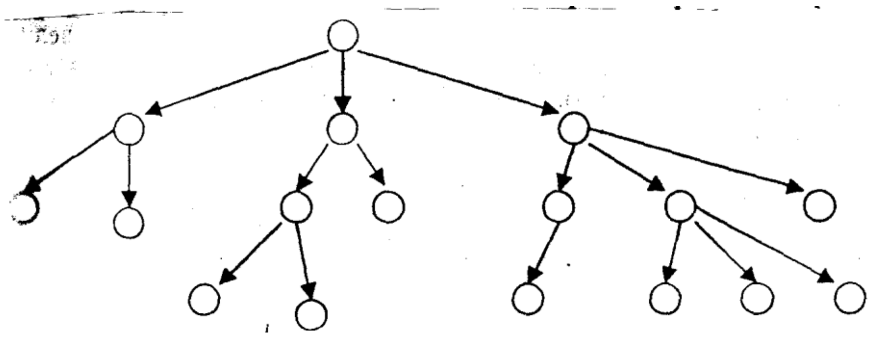


Рис. 4. Вигляд генеалогічного дерева

Множина дуг, виключення яких із графа збільшує число його компонентів, називається розрізом. Розріз, що не містить як власну підмножину жодного іншого розрізу, називається *простим.* Таким чином, простий розріз це той, котрий містить найменшу кількість дуг.

Завдання для самостійної роботи

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?
2. Чи є в ЄС країна, виключення якої зі співтовариства розірвало б усі наземні комунікації між країнами, що залишилися?
3. Чи є граф, отриманий у прикладі 1, мережею? Якщо ні, то чим його потрібно і можна доповнити, щоб вийшла мережа?
4. Побудувати для графа, отриманого в прикладі 1, покривне дерево. Скільки таких дерев можна побудувати?
5. Чи можна для країн ЄС побудувати граф, що відбиває які-небудь інші зв'язки між країнами? Якщо так, то спробуйте його побудувати хоча б на принциповому рівні. Чи має новий граф той самий вигляд, що й попередній? Проілюструвати на отриманому графі всі основні поняття теорії графів.
6. Побудувати граф, множина вершин якого відповідає множині навчальних курсів протягом усіх років навчання у вищому навчальному закладі. При цьому дугу від вершини *x* до вершини *y* включайте в тому випадку, якщо знання курсу *x* необхідні для вивчення курсу *y*. Виконати завдання:
   - інтерпретувати на графі поняття шляху, ланцюга, циклу, контуру;
   - дослідити граф на зв'язність;

- на основі аналізу графа спробувати відповісти на запитання, чи є в інститутському курсі "замкнуті класи" дисциплін, що передбачають знання тільки в межах своїх "класів"?

7. Чи будь-яка частина дерева є деревом?
8. Чи будь-яка частина лісу є лісом?
9. Інтерпретувати для неорієнтованого графа поняття ланцюга і циклу, шляху і контуру. Чи є в цьому випадку ланцюг шляхом, цикл контуром? Якщо так, то чому?
10. Подати в узагальненому вигляді систему міських комунікацій. Для яких комунікацій така мережа зв'язна, а для яких ні? Як можна інтерпретувати розріз у таких мережах? Де місце для простого, тобто найменшого за кількістю ребер розрізу?
11. Зобразити свій родовід у вигляді генеалогічного дерева. Дати відповідь на запитання:
    - чи можуть на цьому графі бути цикли?
    - якщо так, то що це має означати в рамках родинних зв'язків?
    - чи можна дугам цього графа приписати ваги, тобто перетворити його на мережу? Що можуть означати ваги її дуг? Чи існує тільки одне тлумачення цих ваг?
12. Чи може генеалогічне дерево бути багатозв'язним графом? Якщо так, то що це може означати в системі родинних зв'язків?

§ 2. Організація зв'язків у неорієнтованих мережних системах

Такі задачі постають перед користувачем різних мереж, який ставить перед собою питання про досягнення (будь-яким способом) якогось вузла мережі з погляду розв'язуваної ним задачі.

Проблеми вирішуються шляхом побудови для відповідних графів покривних дерев, що відповідають різним вимогам оптимальності.

Розглянемо дві типові задачі, розв'язувані в рамках теорії покривних дерев.

Задача 1 (Поширення чуток). Розглянемо населений пункт, у якому кожний із жителів має зустрічі з деякими іншими жителями. Чи може в цьому населеному пункті поширитися чутка?

Вказівка. Щоб відповісти на запитання задачі, поставимо у відповідність кожному жителю вершину графа. З'єднаємо дві вершини ребром у тому випадку, якщо відповідні жителі спілкуються один з одним. За умови зв'язності отриманого графа на поставлене запитання можна відповісти позитивно. У свою чергу граф буде зв'язним, якщо для нього існує покривне дерево. Отже, щоб розв'язати задачу, потрібно перевірити, чи є в цього графа покривне дерево.

Наступна задача має багато практичних узагальнень. Подібні задачі можна ставити для мереж будь-яких комунікацій (мереж водопроводів, газо- і нафтопроводів, інформаційних мереж, телефонних систем, систем управління і ряду інших).

Задача 2 (Аналіз мережі комунікації). Є мережа доріг, що зв'язує кожен населений пункт із деяким іншим. Потрібно визначити, чи можна, користуючись цими дорогами, проїхати з кожного населеного пункту в будь-який інший.

Вказівка. Для розв'язання задачі поставимо у відповідність кожному населеному пункту і точці перетину наявних доріг вершину графа. Дві вершини з'єднаємо ребром у тому випадку, якщо між пунктами існує дорога. Якщо граф містить покривне дерево, між будь-якими пунктами можна проїхати по існуючій мережі доріг.

Розглянемо найпростіші алгоритми побудови покривних дерев.

1. 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.

2. 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.

Типові задачі на побудову мінімальних (максимальних) покривних дерев

Задача 3 (Проект реконструкції (будівництва) мережі комунікацій мінімальної вартості). Планується реконструкція мережі доріг, що мають зв'язувати задану кількість населених пунктів. Вартість будівництва або реконструкції доріг між будь-якими двома пунктами відома. Необхідно реконструювати (чи побудувати) мережу доріг так, щоб вартість будівництва була мінімальною і по цій мережі можна було проїхати між будь-якими двома пунктами.

Вказівка. Для розв'язання задачі потрібно побудувати мережу, вершини якої відповідають населеним пунктам і точками перетину доріг, а ребра – дорогам. Кожному ребру слід приписати вагу, рівну вартості будівництва (чи реконструкції) цієї ділянки дороги. Проект зводиться до задачі побудови для отриманої мережі покривного дерева мінімальної вартості.

Задача 4 (Проект реконструкції мережі комунікацій мінімальної вартості з фіксованими зв'язками). За умовою попередньої задачі потрібно реконструювати мережу доріг так, щоб деякі ділянки доріг включилися у створювану мережу незалежно від вартості їхньої реконструкції. Наприклад, з причини зручності користування ними.

Вказівка. Проект зводиться до задачі побудови для отриманої мережі покривного дерева мінімальної вартості із заздалегідь зафіксованими зв'язками. Фіксовані зв'язки зафарбовуються безпосередньо перед розв'язанням задачі.

Типові задачі на побудову максимального покривного дерева найчастіше належать до сфери проектування мереж за умови максимізації прибутку від експлуатації мережі.

Приклади побудови покривних дерев

Усі сформульовані вище типові задачі розв'язуються абсолютно однаково після формалізації їх у мережному вигляді, а саме – за допомогою побудови для мережі покривного дерева. Тому ми не будемо формулювати задачі на предметному рівні, а тільки у вигляді задач для конкретної мережі.

Задача 5. Побудувати мінімальне, мінімальне з фіксованими зв'язками і максимальне покривні дерева для мережі, зображеної на рисунку 1.

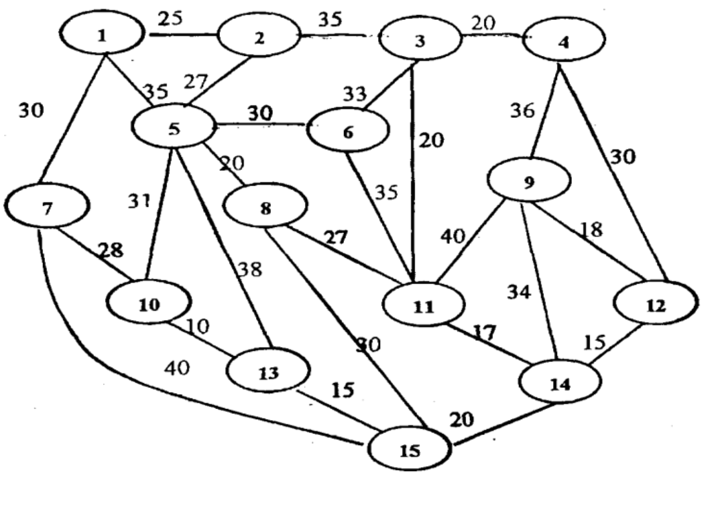


Рис. 1

Розв'язання

Застосовуючи алгоритм побудови мінімального покривного дерева, тобто переглядаючи всі ребра графа в порядку зростання їхніх ваг і не допускаючи циклів, одержуємо розв'язок задачі. Вибір починаємо з ребра (10,13), тому що воно має мінімальну вагу, яка дорівнює 10. Далі вибираємо ребро з вагою 15. Оскільки таких ребер два, першим буде будь-яке. Потім вибираємо, якщо дозволяють умови алгоритму, наступне ребро і так далі.

Внаслідок того, що ребра з однаковими вагами вибираються в довільному порядку, можна отримати кілька варіантів розв'язання. Рішення про те, який з варіантів найбільш прийнятний, має обумовлюватися змістом практичної задачі і прийматися виконавцем. Рекомендуємо одержати кілька рішень і порівняти результати.

У підсумку, зберігаючи конфігурацію вихідної мережі, одержуємо як шуканий розв'язок покривне дерево, зображене на рис. 2.

Зауважимо, що для неорієнтованої мережі отриманий результат можна подати по-різному, оскільки для неї байдуже, яку вершину вибирати як корінь. Це може визначатися тільки особистими уподобаннями і змістом розв'язуваної задачі. На рис. 3 отриманий розв'язок (див. рис. 2) зображено у вигляді дерева, у якого вершина 5 розташована вгорі.

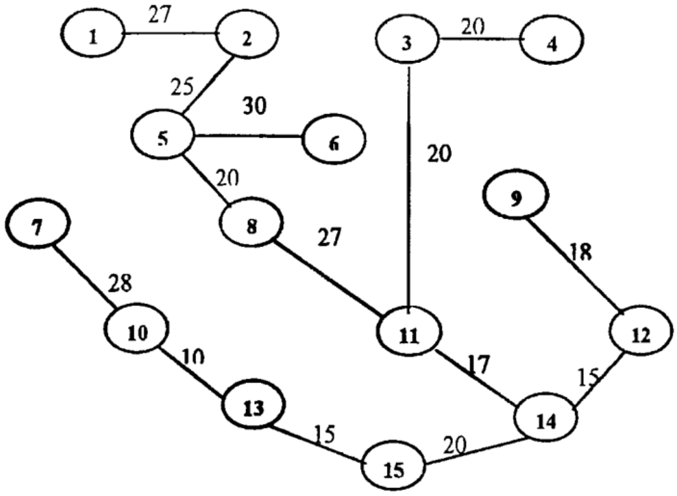


Рис. 2. Мінімальне покривне дерево для мережі на рис. 1

Так само, вибираючи ребра в порядку зменшення їхніх ваг, одержимо для цієї мережі максимальне покривне дерево, зображене на рис. 4. Слід звернути увагу, на те якою різною є їх конфігурація.

Припустимо тепер, що на початковій мережі (рис. 1) ряд зв'язків зафіксований. Така умова може бути включена з різних причин. Нехай, наприклад, попередня задача з цього підрозділу була задачею проекту реконструкції мережі районних доріг, що вимагає мінімальних витрат. Зрозуміло, що отримане рішення буде найкращим не з будь-якого погляду. Так, наприклад, якщо центральна садиба в районі розміщена в пункті 5, то такий проект зовсім не зручний для керівників району.

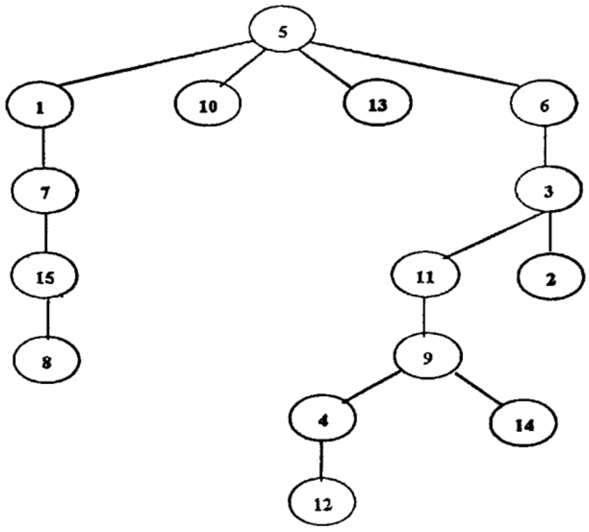


Рис. 3. Інший вигляд мінімального покривного дерева для мережі на рис. 1

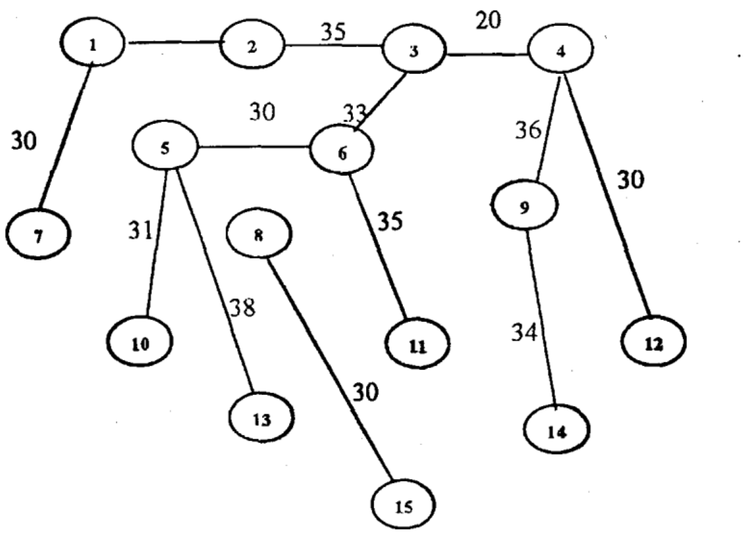


Рис. 4. Максимальне покривне дерево для мережі на рис. 1

Для задоволення цієї вимоги припустимо, що було вирішено ділянки доріг, зв’язані з центральною садибою (вершина 5), включити у майбутню мережу в будь-якому випадку, незалежно від вартості їх реконструкції.

Мережа в цьому випадку набуває вигляду, показаного на рис. 5. Фіксовані зв’язки виділено жирними лініями.

Мінімальне покривне дерево для цієї мережі, побудоване відповідно до розглянутого алгоритму, але з умовою, що фіксовані зв’язки наперед включаються в дерево, подано на рис. 6.

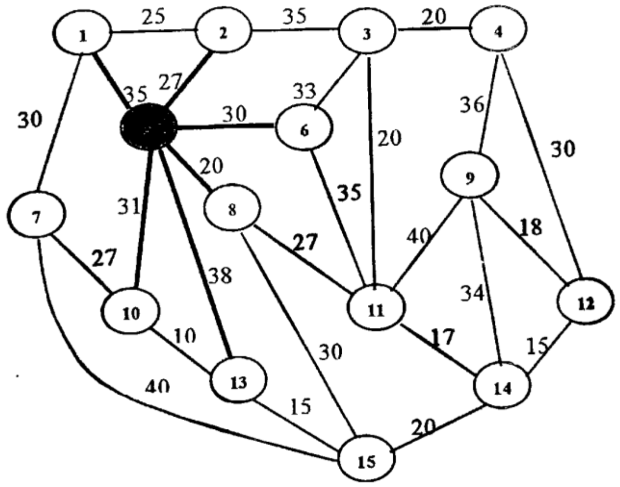


Рис. 5. Мережа з фіксованими зв'язками

Будується воно за тими ж правилами, але фіксовані ребра не зачіпаються, а вважаються відразу ж (до застосування алгоритму) включеними в мінімальне дерево. Виконання алгоритму починається з незафарбованого ребра мінімальної ваги і продовжується доти, доки усі вершини не увійдуть у шукане дерево.

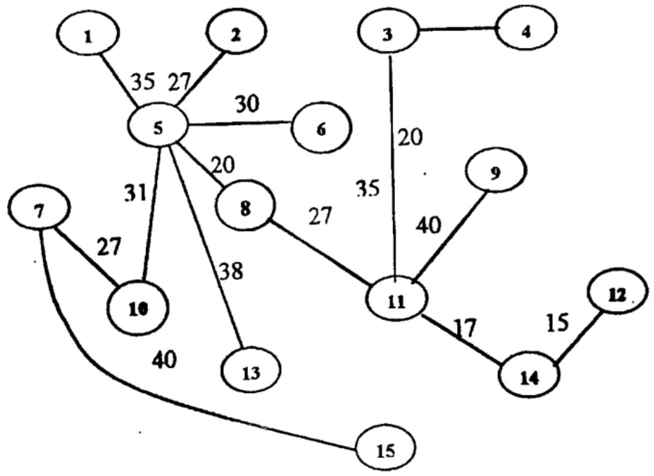


Рис. 6. Мінімальне покривне дерево для мережі на рис. 5 (з фіксованими зв'язками)

Зауважимо, що фіксація зв'язків, звичайно ж, залежить від постановки задачі. Мережа на рис. 6 може бути зручною для керівників району, але видається не досить зручною для жителів при переміщенні між будь-якими двома пунктами. Щоб потрапити, наприклад, з вершини 13 у вершину 15, треба пройти невиправдано довгий шлях.

Але в цій задачі ми ставили умову реалізації проекту мінімальної вартості, а також виконання умови зручності для жителів центральної садиби. Якщо б ми забажали найбільшого комфорту у переміщеннях для всіх жителів району, фіксувати слід було б інші зв'язки.

Завдання для самостійної роботи

1. Для графа, зображеного на рис. 7, побудувати кілька покривних дерев.

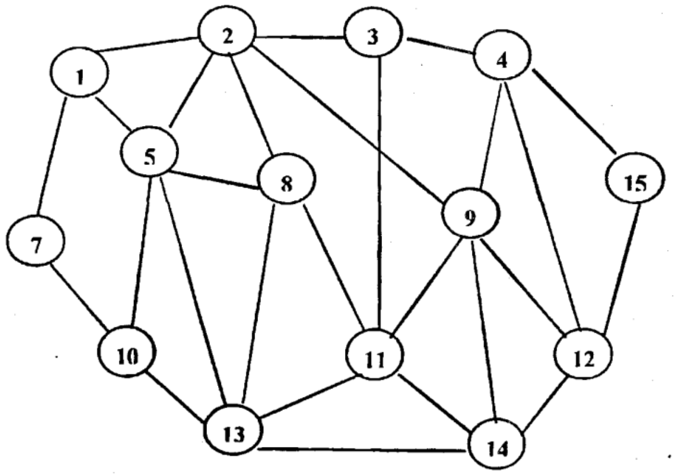


Рис. 7. Граф для задачі 1

2. Намалювати будь-який граф і побудувати для нього покривне дерево відповідно до розглянутого алгоритму.
3. Для мережі, зображеної на рис. 8, побудувати мінімальне і максимальне покривні дерева.

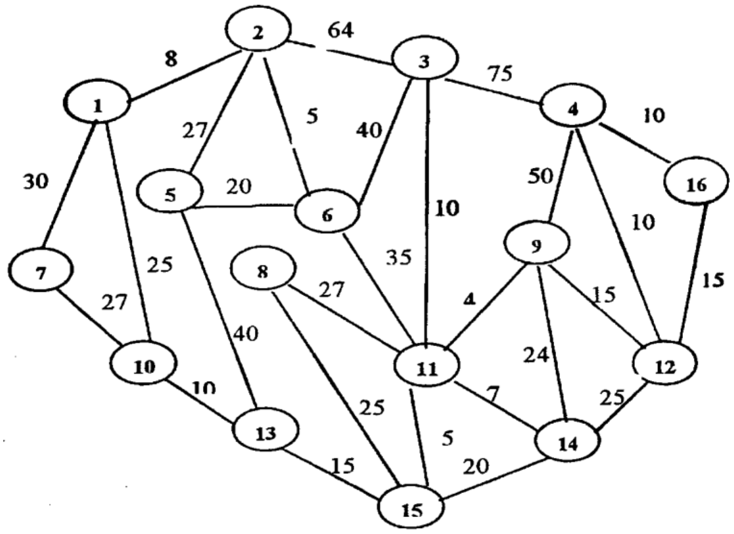


Рис. 8. Мережа для задачі 3

4. Зафіксувати які-небудь зв'язки на рис. 8 і побудувати мінімальне і максимальне дерева з фіксованими зв'язками. Порівняти результати розв'язання цієї і попередньої задач.
5. Намалювати будь-яку мережу і побудувати для неї мінімальне, максимальне і максимальне з фіксованими зв'язками дерева.

6. Постановці яких практичних задач могла б відповідати задача побудови цих дерев? Сформулювати їх. Пояснити, чому на отриманих мережах відповідають вершини і ребра відповідно до Вашої конкретної задачі.
7. Для мережі, зображеної на рис. 9, побудувати максимальне і мінімальне покривні дерева. Ви можете отримати той самий результат. Пояснити, чому. Які умови має задовольняти вихідна мережа, щоб максимальне і мінімальні покривні дерева для неї могли збігатися?
8. На рис. 10 фіксовані зв’язки зображено жирними лініями. Побудувати для цієї мережі мінімальне і максимальне покривні дерева.

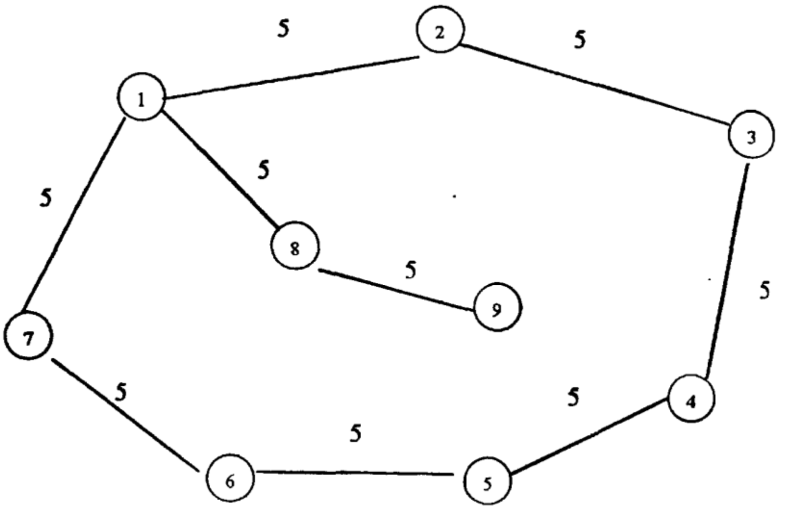


Рис. 9. Мережа для задачі 7

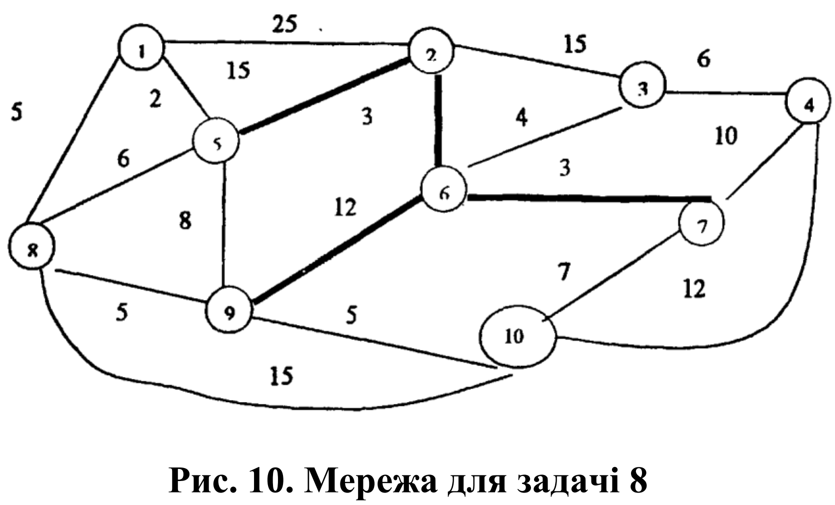


Рис. 10. Мережа для задачі 8

§ 3. Організація зв'язків у орієнтованих мережах

Під час аналізу систем на зв'язність істотною є відмінність орієнтованих графів від неорієнтованих. Для будь-якого зв'язного неорієнтованого графа покривне дерево завжди існує. Для зв'язного орграфа – не завжди.

Тому для орієнтованих графів на першому місці є поняття лісу.

Алгоритми побудови орієнтованого лісу значно складніші аналогічних алгоритмів для неорієнтованих мереж. Існує кілька таких алгоритмів. Найпростішим є алгоритм Едмондса.

Алгоритм побудови максимального орієнтованого лісу

Цей алгоритм розроблений для мереж з позитивними вагами дуг. Однак його можна застосовувати також для побудови мереж, що задовольняють інші умови оптимізації, а саме:

1. *Мінімального орієнтованого лісу.* Знаки ваг кожної дуги при цьому міняються на протилежні.
2. *Максимального покривного дерева з вагами дуг будь-якого знака.* Для цього до ваги кожної дуги потрібно додати деяку позитивну константу, а саме – найбільшу за абсолютною величиною серед усіх абсолютних величин ваг негативних дуг. Це зробить ваги всіх дуг позитивними. По закінченні роботи алгоритму потрібно повернутися до колишніх ваг.
3. *Мінімального покривного дерева.* У цьому випадку спочатку знаки усіх ваг змінюються на протилежні, потім до кожної ваги додається необхідна позитивна константа.
4. *Максимального чи мінімального покривного дерева з коренем у даній вершині.* У цьому випадку, як буде показано нижче, у мережу потрібно включити додаткову фіктивну вершину.

Основну ідею алгоритму – стягування дуг і вершин виявленого контуру у фіктивну вершину – показано на рис. 1.

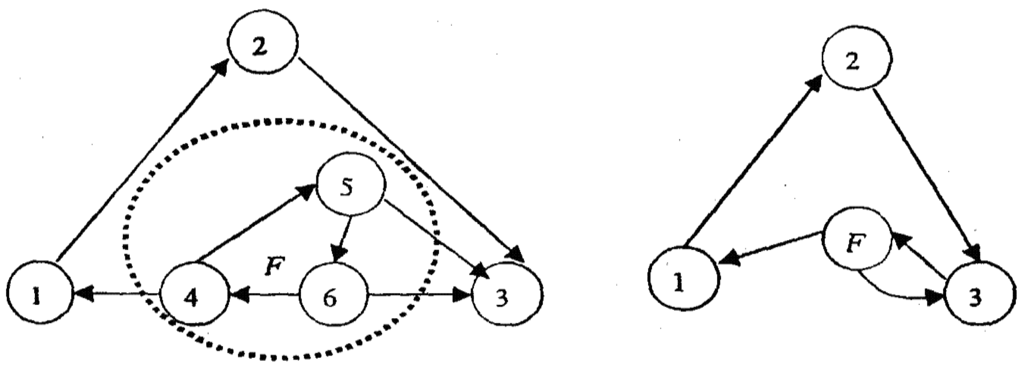


Рис. 1. Стягування контуру у фіктивну вершину F

Контур разом зі своїми дугами і вершинами стягується в одну фіктивну вершину. Всі інші дуги і вершини мережі, що не входять у контур, залишаються такими, як були.

У процесі побудови максимального орієнтованого лісу вихідна мережа G_0 перетвориться в мережі G_1, G_2 і так далі. Сукупності вже обраних вершин і дуг називаються букетами.

Етапи роботи алгоритму відповідають включенню нової фіктивної вершини і регулюються змінним індексом. На початку роботи алгоритму покладається $i = 0$, а букети вершин і дуг вважаються порожніми. Після кожного включення фіктивної вершини, тобто з кожним новим етапом, до індексу i додається 1. Як тільки подальше збільшення i стає неможливим, починається зворотний процес – зменшення i до нуля.

Крок 1. Якщо всі вершини мережі G_i зібрані в букет, перейти до кроку 3. Якщо ні, то вибрати в G_i будь-яку вершину, що не входить в утворений букет, і включити її в букет. Далі серед дуг, що заходять у цю вершину, вибрати дугу з максимальною позитивною вагою.

Якщо такої дуги немає, повернутися до початку кроку. У протилежному випадку включити цю дугу у сформований букет дуг. Якщо після включення дуги не виявлено контуру, повернутися до початку даного кроку. Інакше перейти до кроку 2.

Крок 2. Стягнути всі дуги і вершини контуру в одну вершину, позначити її F_i. Отриману в результаті цього мережу вважати мережею G_{i+1}.

Для всіх мереж G_{i+1}, що заходять у фіктивну вершину, перерахувати ваги. А саме: для кожної дуги (x, y) мережі C_i, що переходить у мережі G_{i+1} у дугу (x, F_i), покласти

$$\alpha(x, F_i) = \alpha(x, y) + \alpha(r, s) - \alpha(t, y), \qquad (1)$$

де (r, s) – дуга, що має в контурі C_i мінімальну вагу;

(t, y) – дуга контуру C_i, що заходить у вершину y.

Інакше формулу (1) можна інтерпретувати так:

Нова вага = стара вага +
+ мінімальна вага дуги в контурі – (2)
– вага дуги контуру, що заходить у дану вершину.

Ваги інших дуг залишити без змін.

Сформувати множину V_{i+1}, включивши в неї всі вершини графа G_{i+1}, що вже містилися у сформованому букеті вершин, і фіктивну вершину.

Збільшити i на одиницю і перейти до кроку 1.

Крок 3. Цей крок виконується тільки тоді, коли всі вершини мережі G_i потрапляють у букет, а дуги утворять у мережі G_i орієнтований ліс.

Якщо $i = 0$, закінчити процедуру, тому що отримано максимальний орієнтований ліс.

Якщо $i \neq 0$, можливі два випадки:

а) фіктивна вершина F_{i-1} на розглянутому етапі є коренем деякого дерева.

Для цього випадку розглянути дуги з букета A_i в сукупності з дугами з контуру C_{i-1} і видалити з контуру C_{i-1} дугу з мінімальною вагою;

б) фіктивна вершина F_{i-1} не є коренем дерева.

Тоді, відповідно до алгоритму, у деяку вершину контуру заходить дві дуги: одна з них належить контуру, а інша – ні. Видалити дугу, що належить контуру. Зменшити i на одиницю і повторити крок 3.

Зауваження:

1. За потреби знайти максимальне (чи мінімальне) орієнтоване дерево з коренем у заданій вершині a, до вихідної мережі додається додаткова вершина a' і дуга (a', a) з довільною дуже великою вагою.

Якщо отримана в такий спосіб мережа містить покривне дерево, то це дерево має мати корінь у вершині a', оскільки в неї не заходять дуги. У вихідній мережі знайденому дереву відповідає дерево з коренем у вершині a.

2. Алгоритм не вказує, у якій послідовності вибирати вершини. З погляду ідеології алгоритму порядок вибору не має значення. Однак практика свідчить, що краще це робити в якому-небудь заздалегідь обговореному порядку. Наприклад, якщо вершини позначено цифрами – у порядку зростання номерів, якщо вершини позначено буквами – за абеткою.

Алгоритм побудови покривного лісу дає змогу розв’язувати найрізноманітніші задачі, які, на перший погляд, його не передбачають.

Розглянемо і розв’яжемо одну з них – задачу вибору лідера в колективі людей, які об’єднуються з метою досягнення конкретної мети. Це можуть бути наукові чи виробничі підрозділи, вибір громадського чи політичного лідера.

Розв’язання задачі про лідерство

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

Вказівка. 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). Розв’язання задачі зводиться 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. Як ваги дуг слід взяти ці оцінки.

Зрештою, немає принципової різниці, який показник буде взято як ваги. Все залежить від ідеології, покладеної в основу формування структури управління. Це може бути, наприклад, оцінка якої-небудь доцільності для кожного члена колективу, визначена навіть не самими членами, а кимось ззовні. Однак у даній роботі ми розглядаємо уподобання самих членів колективу.

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.

Як приклад 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. 2.

Мережу, отриману в результаті цієї роботи, зображено на рис. 3. 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.

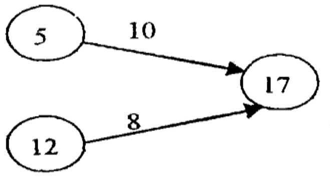


Рис. 2. Відображення на мережі переваг

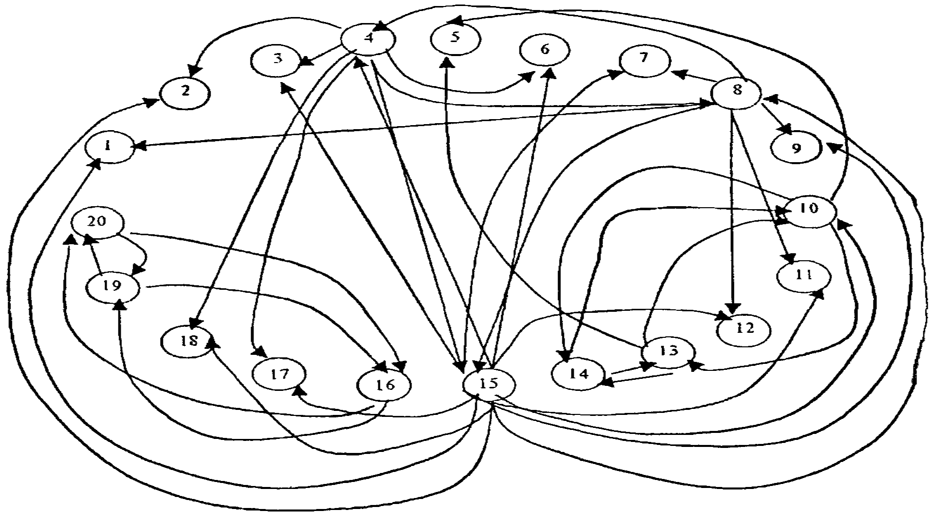


Рис. 3. Мережа, що відображає колектив людей з оцінками уподобань

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

Застосовуючи алгоритм побудови максимального покривного лісу до мережі на рис. 3, 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. 4 ці групи представлено у зручному графічному вигляді з коренями дерев, що відповідають лідерам, угорі.

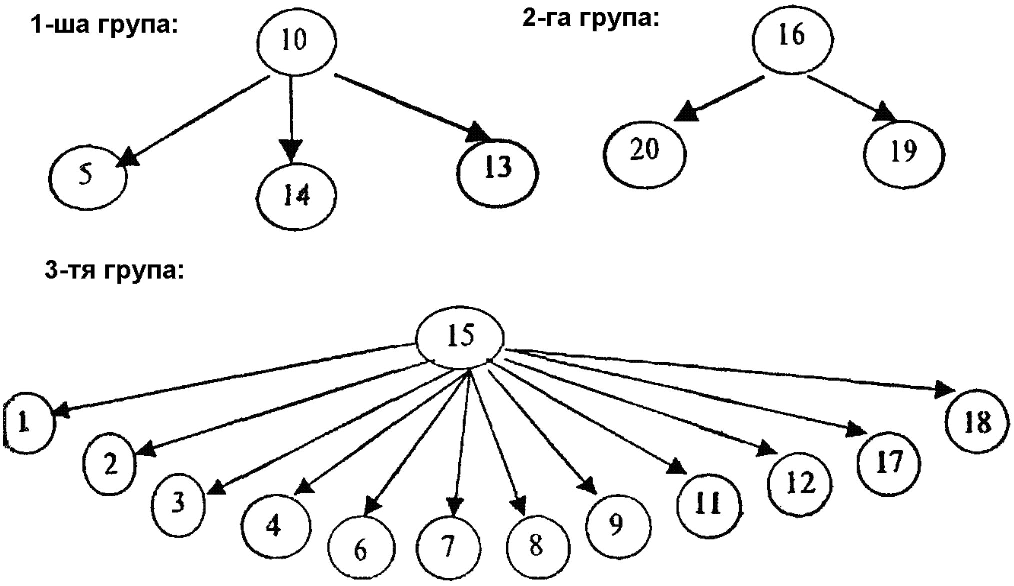


Рис. 4. Проміжний етап розв'язання задачі

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. 3 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.

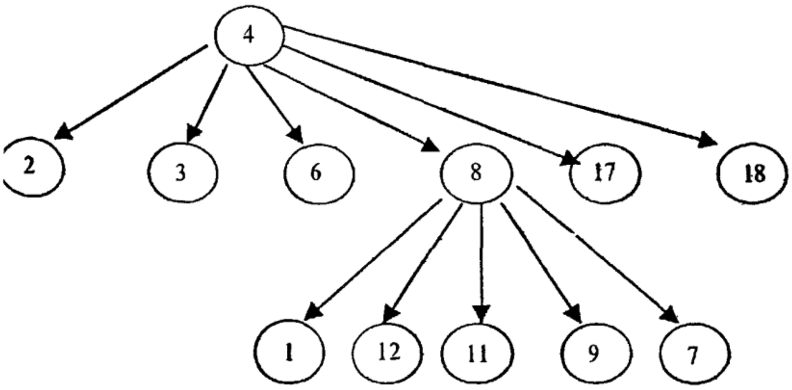


Рис. 5. Трансформована 3-тя група

Отже, колектив людей на рис. 5 можна організувати відповідно до їхніх особистих бажань у дві групи (рис. 6).

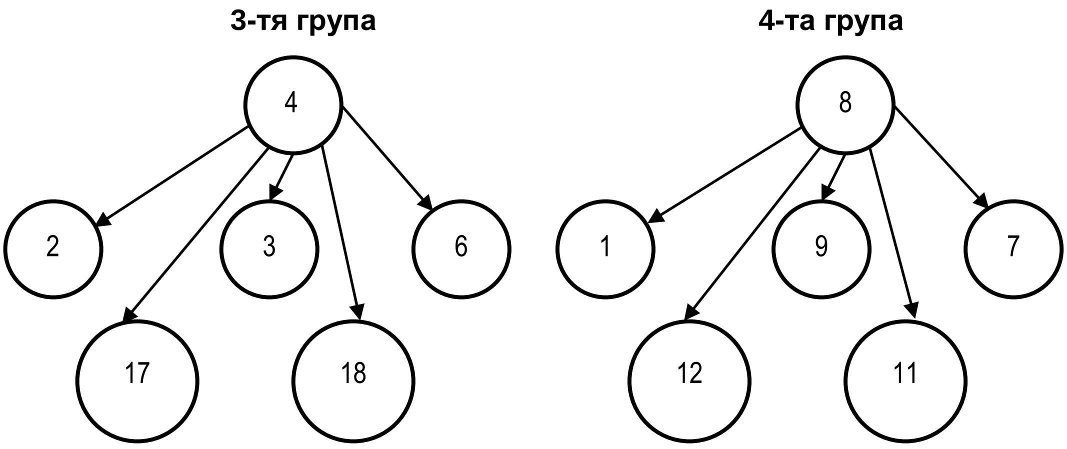


Рис. 6. Дві групи, сформовані на основі 3-ї групи

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. 4 і 6.

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 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.

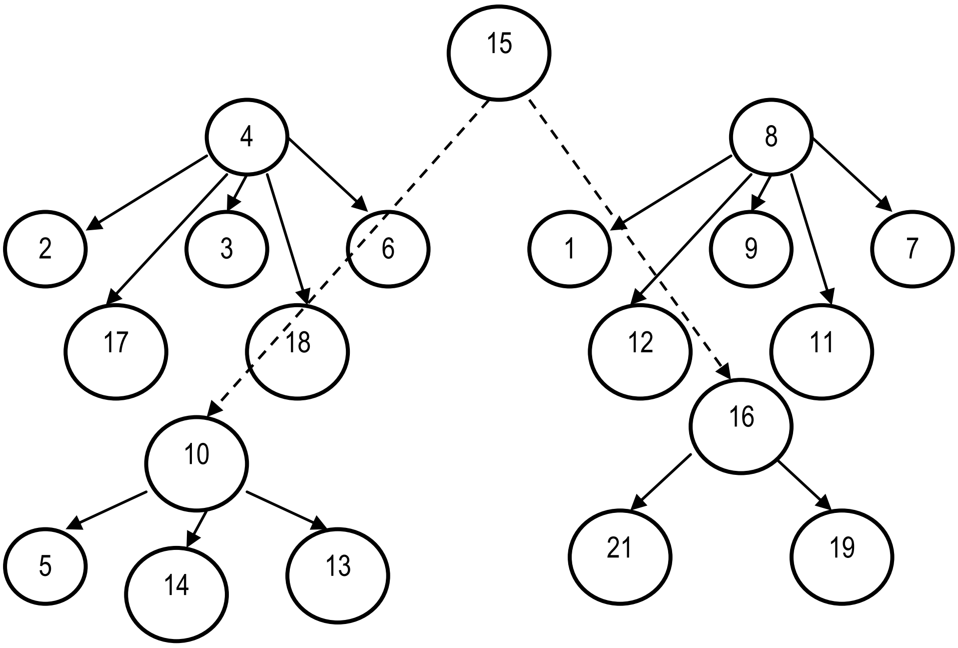


Рис. 7. Результат розв'язання задачі про лідерів

Завдання для самостійної роботи

1. Для мережі, зображеної на рис. 8, побудувати максимальний орієнтований ліс.

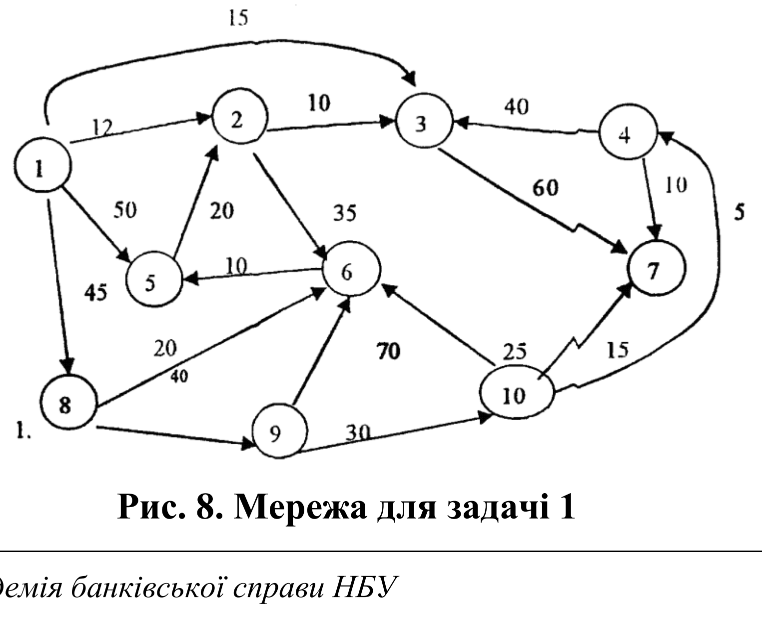


Рис. 8. Мережа для задачі 1

2. Які умови має задовольняти мережа, щоб максимальний і мінімальний покривний ліс для неї збігалися?
3. На мережі, зображеній на рис. 9, зафіксовано ряд зв'язків. Побудувати для цієї мережі максимальний і мінімальний покривний ліс. Порівняти результати.

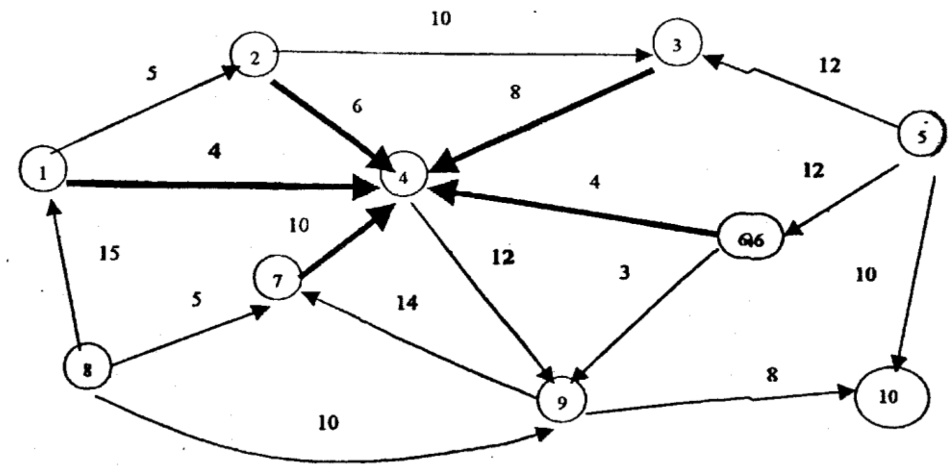


Рис. 9. Мережа для задачі 3

4. На мережі, зображеній на рис. 10, фіксовані вершини затушовані. Застосувати до цієї мережі алгоритм Едмондса і знайти для неї максимальний орієнтований ліс.

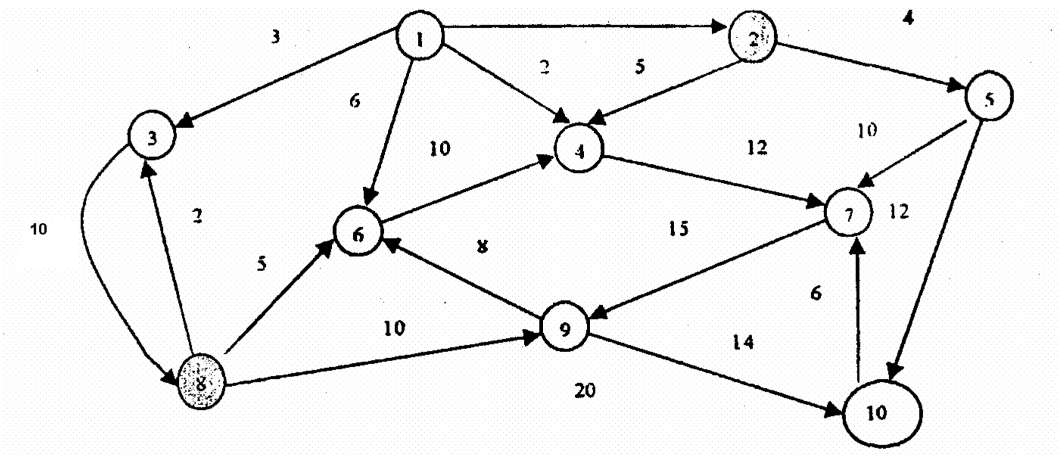


Рис. 10. Мережа для задачі 4

§ 4. Оптимальні шляхи на мережах

Задачі цієї тематики належать до найпопулярніших задач, розв'язуваних у рамках оптимізаційної теорії мереж. Охарактеризуємо лише деякі з них, охопивши по можливості задачі з різних сфер діяльності і різного підходу до оптимізації.

Задача 1 (Вибір маршруту найменшої довжини). Потрібно знайти найкоротший маршрут від Києва до Москви по шосейних дорогах.

Вказівка. Для розв'язання побудувати мережу, вершинами якої будуть усі міста, що зустрічаються на можливому шляху від Києва до Москви, а також перехрестя доріг. Дуги відповідатимуть ділянкам шосе, що з'єднує вершини, а ваги дуг – довжинам ділянок. У цій формалізації задача зводиться до відшукання на мережі найкоротшого шляху між вершинами, що відповідають Києву і Москві.

Задача 2 (Вибір маршруту, що займає найменший час). Пасажир звертається за послугами в авіакомпанію. Він бажає потрапити з Києва в Делі за найкоротший час, однак безпересадочних рейсів не існує. Потрібно визначити маршрут, що має запропонувати йому компанія.

Вказівка. Щоб вирішити це питання, необхідно побудувати мережу з вершинами в аеропортах, через які може проходити політ. Дуги мають відповідати повітряним лініям між аеропортами. Довжиною дуги слід вважати час польоту плюс час очікування в аеропорту. Якби пасажир ставив умову не найменшого часу в дорозі, а найменшого часу, проведеного в повітрі (наприклад, через самопочуття), то за вагу дуги слід було б узяти тільки час самого польоту. У цілому задача зводиться до відшукання найкоротшого шляху між вершинами, що відповідають Києву і Делі. Однак на відміну від попередньої задачі цей шлях вже буде тлумачитиметься не мінімальним за довжиною, а мінімальним за часом.

Задачі пошуку шляхів у мережних системах, оптимальних у тому чи іншому сенсі, різноманітні. Перш ніж їх розглядати, введемо необхідні поняття.

Нагадаємо, що кожній дузі (x, y) мережі ставиться у відповідність вага $\alpha(x, y)$. Якщо яка-небудь дуга відсутня, їй приписується вага, яка дорівнює ∞, тобто покладається $\alpha(x, y) = \infty$.

У даному випадку число $\alpha(x, y)$ буде інтерпретуватиметься як довжина дуги (x, y), хоча в широкому сенсі під довжиною можна розуміти й інші характеристики обслуговування, наприклад час, витрати, прибуток.

Нагадаємо, що шляхом називається зв’язна послідовність дуг (x_1, x_2), (x_2, x_3), $(x_3, x_4), \ldots, (x_{m1}, x_m)$, таких, що кінець попередньої дуги є початком наступної.

Довжиною шляху називається сума довжин дуг – його складових:

$$l(x_1, x_m) = \sum_{i=1}^{m\ 1} \alpha(x_i, x_{i+1}) =$$

$$= \alpha(x_1, x_2) + \alpha(x_2, x_3) + \ldots + \alpha(x_{m\ 1}, x_m). \qquad (1)$$

Надалі дві вершини, між якими знаходиться шлях, позначатимуться літерами s і t від англійських слів: source – джерело і terminal – стік.

Основна задача знаходження шляхів на мережах формулюється так: найти на мережі шлях між двома чи кількома вершинами, оптимальний в обумовленому сенсі.

Як приклад розглянемо схему автомобільних доріг між Москвою і Таллінном, наведену на рис. 1. Просте порівняння маршрутів Москва – Санкт-Петербург – Таллінн і Москва – Вітебськ – Рига – Таллінн показує, що шлях через Санкт-Петербург (1049 км) оптимальніший, якщо задача полягає у знаходженні найкоротшого шляху. Для мереж великих розмінностей цей вибір не такий наочний.

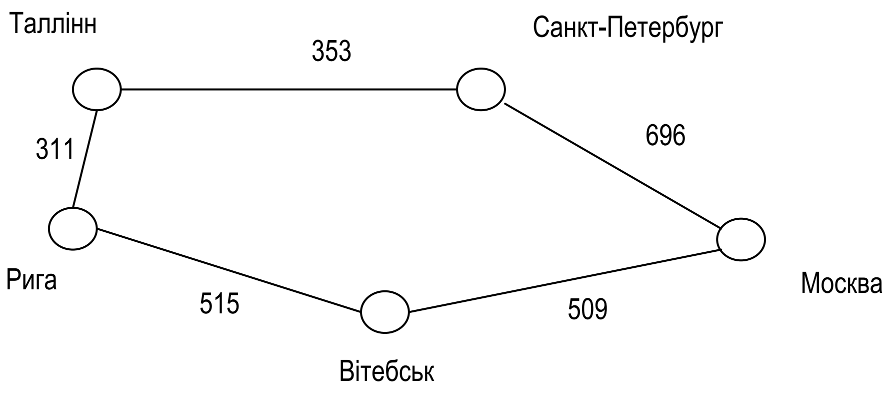


Рис. 1. Схема автомобільних доріг між Москвою і Таллінном

Один з найпростіших алгоритмів знаходження найкоротшого шляху з вершини s у вершину t – алгоритм Дейкстри. Він застосовний тільки для негативних дуг. Як і всі попередні алгоритми, він виконується покроково.

Алгоритм знаходження найкоротших шляхів

Крок 1. Перед початком виконання алгоритму усі вершини і дуги незафарбовані. Кожній вершині x у ході алгоритму присвоюється число $d(x)$, яке дорівнює довжині найкоротшого шляху з s у x, що включає тільки зафарбовані вершини.

Покладається $d(s) = 0$, $d(x) = \infty$ для будь-якого x, $x \neq s$. Зафарбувати вершину s і покласти $y = s$, де y – остання зафарбована вершина.

Крок 2. Для всіх незафарбованих вершин x (реально – тільки для тих, з якими y має зв'язки) слід перерахувати величину $d(x)$ за формулою:

$$d(x) = \min\{d(x), d(y) + a(y, x\}. \qquad (2)$$

Ця формула розшифровується так:

нове число d дорівнює мінімуму з двох величин: старого d *і суми* d *для* y *і ваги дуги з* y *в* x.

Якщо $d(x) = \infty$ для всіх незафарбованих вершин, закінчити алгоритм: у графі немає шляхів з s у незафарбовані вершини. В іншому разі зафарбувати ту з вершин x, для якої величина $d(x)$ є найменшою. Якщо таких вершин кілька, вибираємо будь-яку. Крім того, зафарбовуємо дугу, що веде у вершину x на тому етапі, на якому <u>вперше</u> отримано значення $d(x)$. Покласти $y = x$.

Крок 3. Якщо $y = t$, закінчити процедуру. Найкоротший шлях знайдено. Якщо ні, то перейти до кроку 2.

По закінченні алгоритму отримуємо дерево, що називається орієнтованим деревом найкоротших шляхів. Наголосимо: дерево найкоротших шляхів і мінімальне покривне дерево – не те саме. У цьому легко переконатися, побудувавши для однієї і тієї ж мережі обидва дерева.

Якщо потрібно визначити найкоротші шляхи з s в усі вершини графа, процедуру потрібно продовжувати доти, доки всі вершини графа не зафарбуються. Для цього крок 3 потрібно скоригувати в такий спосіб:

Модифікований крок 3. Якщо всі вершини графа зафарбовані, закінчити процедуру, інакше перейти до кроку 3.

Зауваження 1. Узагальнення алгоритму Дейкстри на випадок негативних ваг належить Фордові. Існують і інші алгоритми для розв'язання подібних задач. Їх опис можна знайти у відповідних джерелах.

Однак і без залучення інших алгоритмів можна запропонувати такий шлях знаходження найкоротших шляхів у випадку негативних ваг: потрібно до усіх дуг додати одне й те саме число, вибране таким чином, щоб після його додавання ваги всіх дуг стали позитивними. Після цього застосовується алгоритм Дейкстри, а по закінченні його роботи повертаються до старих ваг.

Зауваження 2. Алгоритм Дейкстри не застосовний до випадку, коли між двома вершинами є пряма і зворотна дуги, як показано на рис. 2. У принципі, його можна було б модифікувати і для цього випадку, але тут ми не будемо цього робити, щоб не обтяжувати міркувань.

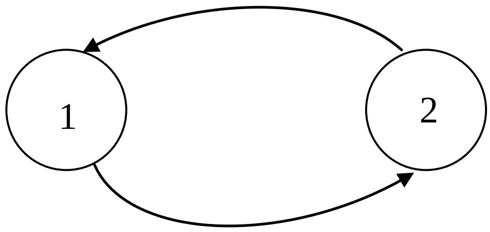


Рис. 2. До питання застосовності алгоритму Дейкстри

Приклад знаходження дерева найкоротших шляхів

Усі сформульовані вище задачі і подібні до них розв'язуються однаковим чином за допомогою алгоритму Дейкстри за умови формалізації їх у вигляді мережі. Тому як ілюстрацію його роботи наведемо метод знаходження найкоротших шляхів для заданої мережі безвідносно до предметної постановки задачі.

Задача 3. Для мережі, зображеної на рис. 3, побудувати дерево найкоротших шляхів для вершини s.

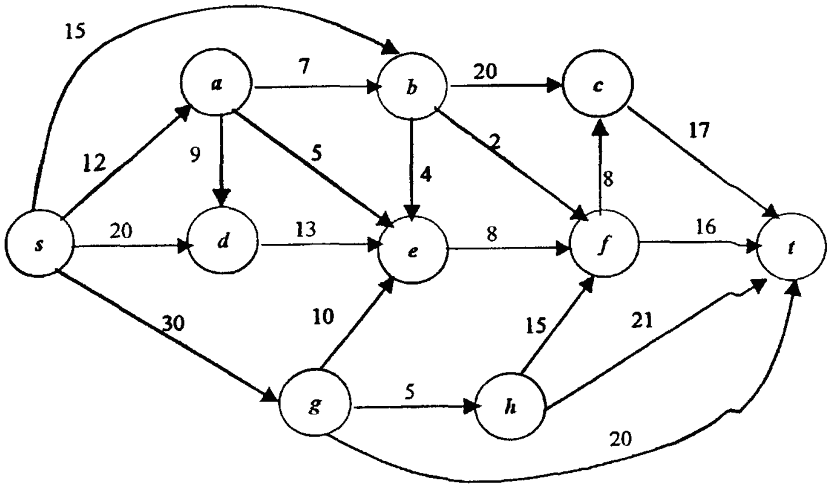


Рис. 3

Користуючись алгоритмом, найкоротші шляхи на рис. 4 виділено жирними лініями. На рис. 5 зображено це ж дерево, але в більш наочному вигляді.

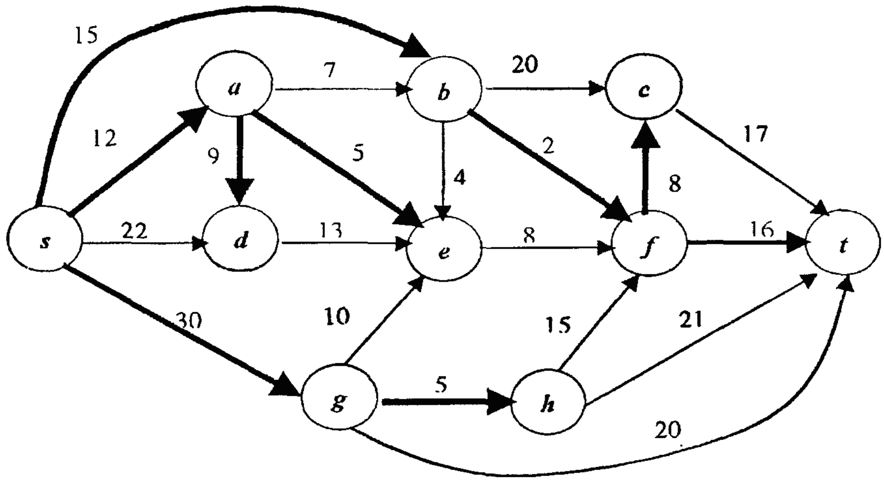


Рис. 4. Мережа з побудованим деревом найкоротших шляхів

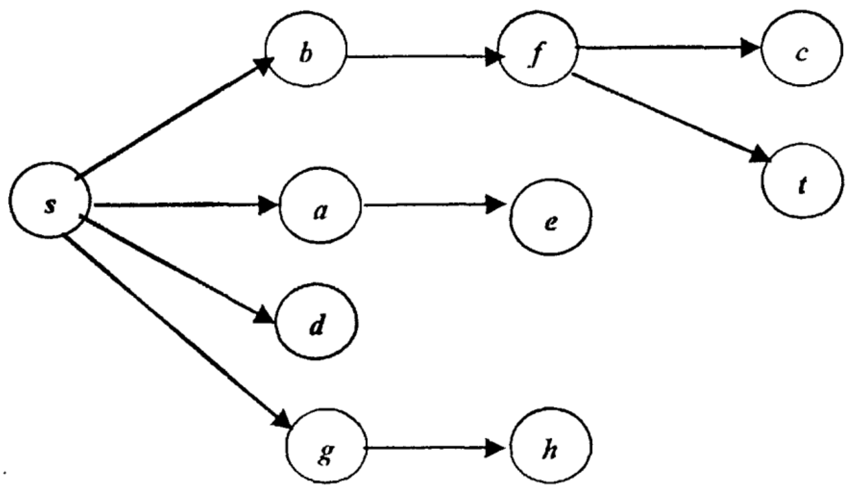


Рис. 5. Дерево найкоротших шляхів для вершини s
мережі на рис. 4

Найкоротші шляхи, що містять задані відрізки

Приклад розв'язання задачі із заданими відрізками шляху наведено на тій самій мережі.

Задача 4. Для мережі, зображеної на рис. 3, побудувати найкоротший шлях з вершини s у вершину t, що включає дугу (e, f).

Така задача може мати численні практичні застосування, в яких реальний шлях шукається, виходячи з якихось додаткових умов. Наприклад, шлях має проходити повз заданий пункт чи, навпаки, в об'їзд основної дороги по вказаному мосту.

Ідея побудови такого шляху проста. Спочатку треба знайти найкоротший шлях P_1 між s і e, далі знайти найкоротший шлях P_2 між

f і t і, нарешті, скласти загальний шлях, який складається зі шляху P_1, дуги (e, f) і шляху P_2.

Шуканий шлях P, отже, дорівнюватиме:

$$P = P_1 + (e, f) + P_2. \qquad (3)$$

Так само можна конструювати шляхи, що містять кілька заздалегідь намічених відрізків.

Ідею цього способу ілюструє рис. 6. На ньому пунктиром показано найкоротші шляхи між початковою і кінцевою вершинами шуканого шляху і вершинами, що відповідають початку і кінцю заданої дуги.

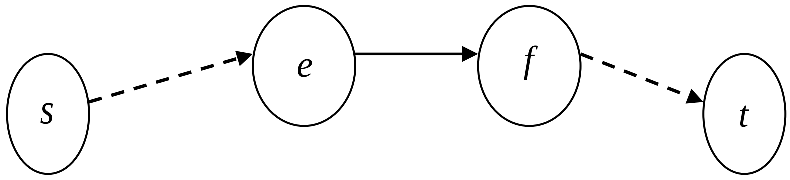


Рис. 6. Ідея методу пошуку найкоротшого шляху, що містить дугу (e, f)

Що стосується мережі на рис. 3, то шуканий шлях для неї, знайдений таким способом, зображено на рис. 7. На ньому пунктиром, відповідно до ідеї, представленої на рис. 6, зображено відрізки найкоротших шляхів між кінцевими вершинами шуканого шляху і кінцевими вершинами заданої дуги.

Природно, що в даному випадку (як і в більшості навчальних задач через їхню малу розмірність), шуканий шлях можна було б знайти без застосування алгоритму. Але нам важливо проілюструвати метод, який для великих мереж далеко не очевидний.

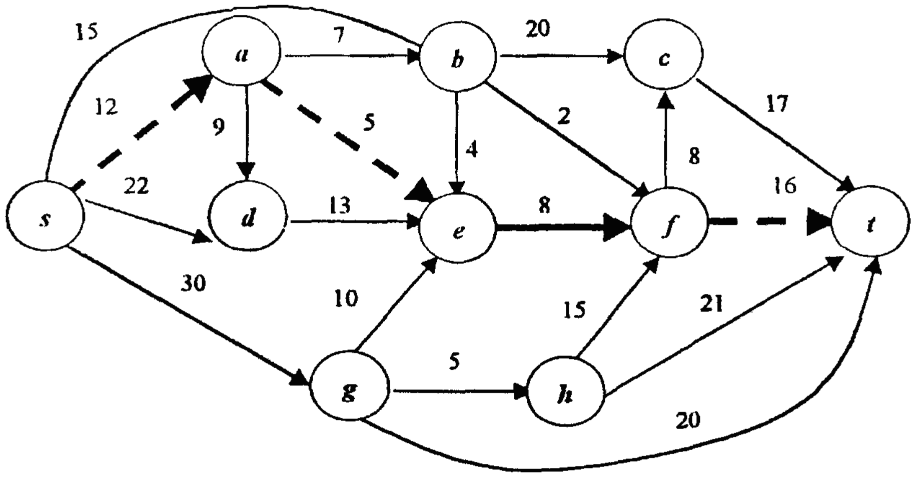


Рис. 7. Найкоротший шлях між вершинами s і t мережі на рис. 3, що містить дугу (e, f)

Порівнюючи рисунки 5 і 7, бачимо відмінності в отриманих найкоротших шляхах між вершинами s i t. У першому випадку це дійсно найкоротший шлях, у другому – найкоротший шлях, що проходить через задану дугу.

На практиці трапляються задачі, у яких потрібно знайти шляхи з іншими вимогами до оптимізації. Алгоритми, призначені для їх розв'язання, через брак місця наводити не будемо. Зазначимо лише, що розглянуті вище алгоритми дають змогу у разі потреби самостійно опанувати їх за літературними джерелами, наведеними наприкінці посібника.

Завдання для самостійної роботи

1. Знайти найкоротші шляхи з вершини s в усі інші вершини (дерево найкоротших шляхів) для мережі, зображеної на рис. 7.
2. Для мережі на рис. 8 зафіксувати один чи кілька зв'язків і знайти найкоротший шлях, що містить фіксовані відрізки. Нехай вона перетвориться в мережу на рис. 9. Порівняти результати задач 1 і 2.

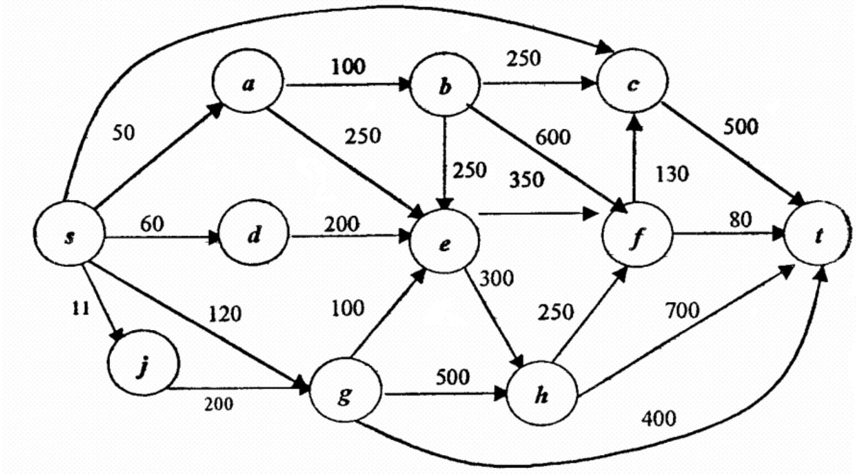


Рис. 8. Мережа для задачі 1

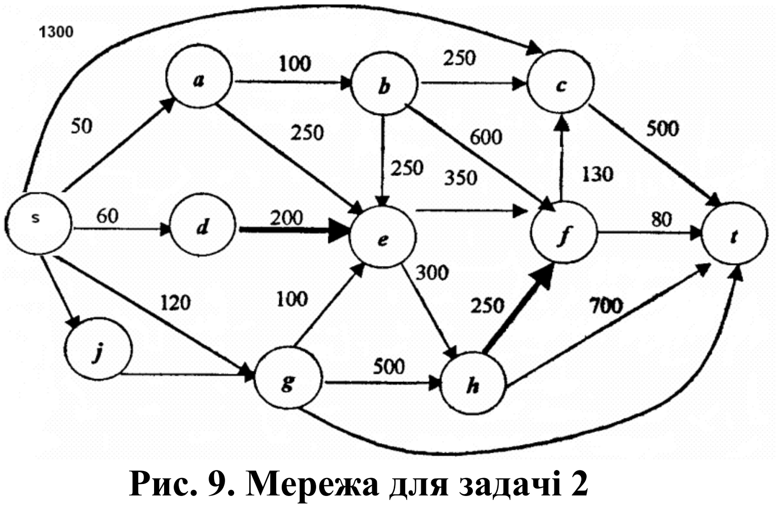


Рис. 9. Мережа для задачі 2

3. Які ще постановки задач на пошук оптимальних шляхів Ви знаєте?
4. Які умови має задовольняти вихідна мережа, щоб у ній існував єдиний найкоротший шлях між двома заданими вершинами?
5. Для розв'язання яких практичних задач потрібно знання наступних за значущістю найкоротших шляхів між вершинами?

§ 5. Загальний огляд алгоритмів оптимізації на мережах і графах

Задачі, розв'язувані в межах оптимізаційної теорії графів і мереж, надзвичайно різноманітні. І вони, звичайно ж, не вичерпуються тими алгоритмами, що розглянуті в попередніх розділах. Теорія графів і мереж, розроблена до нинішнього часу, дає можливість розв'язувати багато інших цікавих і корисних з практичної точки зору задач у принципово інших постановках.

У цьому розділі пропонується короткий огляд задач, які традиційно розв'язуються за допомогою графів і мереж.

Мета цього розділу – допомогти читачеві зорієнтуватися у великому списку можливих задач, розв'язуваних у межах теорії графів і мереж. Головна перевага застосовуваних методів – наочність і орієнтованість на використання комп'ютерів для мереж великих розмірностей. Крім того, багато зі сформульованих нижче задач для мереж великих розмірностей не розв'язуються жодними іншими методами.

Всі алгоритми оптимізаційної теорії графів і мереж формулюються з використанням одних і тих самих принципів і, навчившись застосовувати описані алгоритми, можна без особливих труднощів засвоїти інші.

Розміщення

Задачі розміщення пов'язані з вирішенням проблем оптимального розташування у певних регіонах пунктів обслуговування, таких як: торговельні центри, станції техобслуговування, автозаправні станції, пости пожежної охорони, лікарні, туристичні центри, склади, пошти, готелі і т.п. Математична структура таких задач визначається конфігурацією області припустимих точок розміщення і способом оцінки якості розміщення. Внаслідок цього існує багато різноманітних задач розміщення, що стосуються різних сфер знань і у технічній літературі пропонується чимало методів їх розв'язання.

Для того, щоб формулювати і розв'язувати задачі на розміщення, необхідно ввести нові поняття теорії графів.

Центром графа називається будь-яка його вершина, відстань від якої до найвіддаленішої від неї вершини є мінімальною.

Головним центром графа називається будь-яка його вершина, відстань від якої до найвіддаленішої точки на дугах графа є мінімальною.

Абсолютним центром графа називається будь-яка точка на дузі, відстань від якої до найвіддаленішої вершини графа є мінімальною.

Головним абсолютним центром графа називається будь-яка точка, відстань від якої до найвіддаленішої точки є мінімальною.

Медіаною графа називається точка, в якій сума відстаней до усіх вершин графа є мінімальною.

Аналогічно центрам можна ввести поняття головної, абсолютної і головної абсолютної медіани.

При розв'язанні задач на знаходження розміщення центрів обслуговування слід домовитися про область, у якій їх можна розміщувати. Обмежимося тільки тими задачами, для яких областю припустимих точок розміщення центрів обслуговування є деякий граф. Центри можуть розташовуватися або у вершинах, або на дугах графа і ніде більше. Крім того, ми обмежимося тільки не дуже складними з математичної точки зору задачами.

З подібними задачами ми вже зустрічалися при вивченні алгоритму побудови оптимального орієнтованого лісу. Однак у тому випадку місця вибору розміщень шуканих пунктів були обмежені вершинами графа й орієнтаціями зв'язків між вершинами. У цьому розділі вивчається найбільш загальний випадок.

Розглянемо типові практичні задачі, розв'язувані в межах теорії розміщень.

Задача 1 (Розміщення торгового центру). Адміністрація регіону планує будівництво нового торгового центру, що має обслуговувати кілька районів. Центр вирішено розташувати біля якої-небудь магістралі таким чином, щоб мінімізувати відстань до найбільш віддаленої від нього точки.

Вказівка. Для розв'язання задачі райони слід зобразити вершинами графа, автомагістралі – ребрами. Тоді задача зводиться до відшукання центра отриманого графа.

Задача 2 (Розміщення автозаправної станції). У цьому ж регіоні прийнято рішення організувати автозаправну станцію таким чином, щоб мінімізувати сумарну відстань від неї до всіх районів.

Вказівка. Як і раніше, райони слід зобразити вершинами графа, автомагістралі – ребрами. Задача зводиться до відшукання медіани графа.

Задача 3 (Розміщення станції техобслуговування). У межах тієї ж задачі в регіоні вирішено побудувати станцію техобслуговування для надання допомоги водіям, що потрапили в аварію на якій-небудь автомагістралі регіону. Критерієм якості розміщення станції прийнято мінімум максимальної відстані, яку ремонтники мають здолати до можливого місця аварії.

Вказівка. Як і в попередній задачі, районам слід поставити у відповідність вершини графа, автомагістралям – ребра. Задача зводиться до відшукання для отриманого графа головного центра.

Задача 4 (Розміщення складу). Підприємець, що має мережу магазинів, вирішив побудувати склад для постачання своїх торгових точок, відстань від яких до найбільш віддаленої точки буде мінімальною. Як знайти місце розташування складу?

Вказівка. Для розв'язання задачі потрібно побудувати граф, вершинам якого поставити у відповідність магазини, дугам – відрізки доріг, по яких можна переїжджати між магазинами. Задача зводиться до відшукання для отриманого графа головного абсолютного центра.

Паросполучення

Паросполученням графа називають деяку множину його дуг, таку, що кожна вершина графа інцидентна (зв'язана) не більш ніж одній дузі цієї множини. Зміст цього поняття говорить сам за себе: сполучення парами. Оскільки наш світ переважно двополюсний, інтуїтивно зрозуміло, що теорія паросполучення має численні практичні застосування.

Приклад паросполучення, що складається з дуг $\beta, \ \varepsilon, \ \varphi$, наведено на рис. 1.

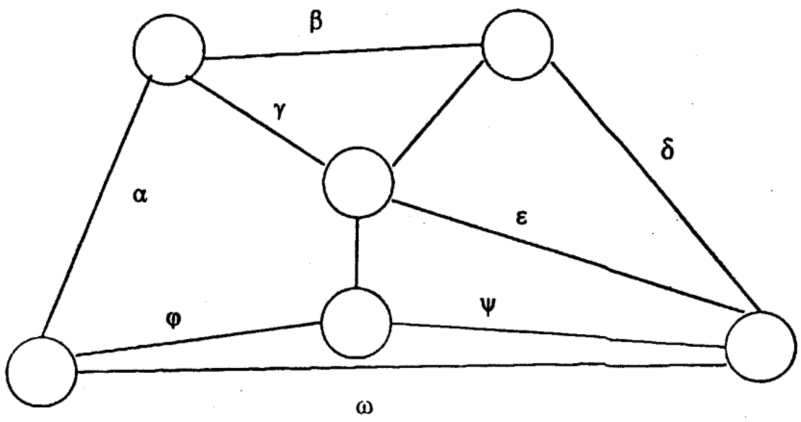


Рис. 1. Інтерпретація паросполучення

Кількість дуг, що містяться в паросполученні, називається його *потужністю.*

Паросполучення, що містить найбільшу кількість дуг, називається *максимальним.*

Загальна сума ваг усіх дуг, що входять у паросполучення називається його *вагою.*

Задача 5 (Підбір екіпажів літаків). Під час Другої світової війни багато льотчиків з інших країн тікали в Англію, щоб поступити на службу в Королівські військово-повітряні сили (ВПС). У той час для виконання польотів потрібно було два пілоти. Основною вимогою до екіпажу була наявність спільної мови. Задача полягала в тому, щоб у повітрі одночасно перебувала максимальна кількість літаків.

Вказівка. Побудувати граф, кожна вершина якого відповідає пілоту ВПС. Дуги з'єднують тих пілотів, що можуть літати разом, тобто знають одну й ту ж мову. Будь-яке паросполучення цього графа являє собою припустиму множину літаків, що одночасно можуть перебувати у повітрі. Задача зводиться до знаходження максимального паросполучення.

Задача 6 (Розміщення в готелі). У готель, в якому вільні лише двомісні номери, прибула велика група туристів. Задача адміністрації – розмістити їх таким чином, щоб в одному номері поселити або тільки родичів, або осіб однієї статі. Як це зробити, щоб зайнятими виявилися мінімум номерів?

Вказівка. Побудувати граф, кожна вершина якого відповідає туристу. Дугами з'єднати осіб, що можуть проживати в одному номері. Задача зводиться до знаходження для отриманого графа паросполучення мінімальної потужності.

Задача 7 (Обслуговування перекладачами). Конференцію, на яку прибула велика кількість різномовних учасників, обслуговує обмежена кількість перекладачів. Кожен перекладач володіє кількома мовами. Потрібно так скомплектувати групи з учасників конференції, щоб задіяти мінімальну кількість перекладачів.

Вказівка. Побудувати граф, вершини якого відповідають учасникам конференції і перекладачам. Дугами попарно з'єднати учасників і перекладачів, прийнятних один для одного. Задача звелася до побудови на отриманому графі паросполучення мінімальної потужності.

Пошук розв'язання і розв'язки всіх цих задач зручно оформляти у вигляді дводольних графів, у яких один вид вершин має першу ознаку, інший – другу. Отримуємо структуру, схематично зображену на рис. 2. У даному випадку показано сполучення кожної вершини з кожною іншою тільки у вигляді однієї пари. Залежно від постановки задачі їх може бути більше. Наприклад, перекладач може обслуговувати кілька груп, бюро знайомств може запропонувати кожному клієнту кілька варіантів зустрічей.

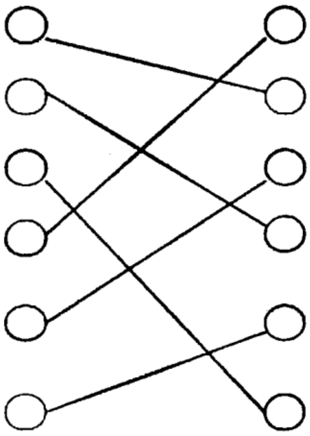


Рис. 2. Вигляд дводольного графа

Покриття

Покриттям графа називається деяка множина його дуг, така, що кожна його вершина інцидентна (зв'язана) принаймні одній дузі цієї множини. Як і назва "паросполучення", поняття покриття також є наочним. Воно означає якусь частину графа, що "покриває" вихідний граф, але певним чином: у кожну вершину має заходити чи виходити дуга.

Наприклад, для графа на рис. 3 множини дуг $\{\alpha,\gamma\}$, $\{\beta,\varepsilon\}$, $\{\delta,\beta\}$ утворюють паросполучення графа, а кожна з множин $\{\alpha,\varepsilon,\gamma\}$, $\{\alpha,\beta,\delta,\varepsilon\}$, $\{\beta,\delta,\varepsilon\}$ – покриття.

Кількість дуг, що містяться в покритті, називається його *потужністю.*

Загальна сума ваг усіх дуг, що входять у покриття, називається його *вагою.*

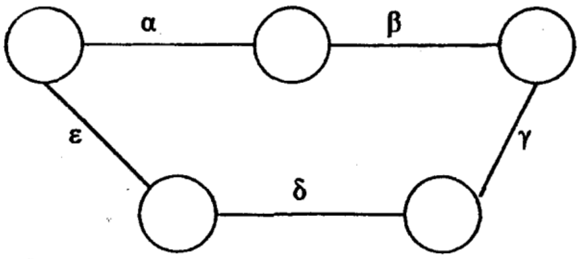


Рис. 3. Інтерпретація понять "паросполучення" і "покриття"

Серед практичних задач зустрічаються задачі на визначення покриття мінімальної чи максимальної потужності, з мінімальною і максимальною вагою.

Задача 8 (Організація комітету). Деякий загальнонаціональний комітет має бути сформований таким чином, щоб до нього входило принаймні по одному представнику від кожної з наявних у країні

15 областей і принаймні по одному представнику кожної із зареєстрованих у країні 40 партій. Свої послуги для участі в роботі комітету запропонувало 60 чоловік, для кожного з яких відомо, яку область він представляє і від імені якої партії виступає. Необхідно визначити склад комітету, що включає найменшу кількість претендентів і задовольняє перелічені вище вимоги.

Вказівка. Побудувати граф, у якому кожна область або край і кожна партія представлені однією вершиною. Таким чином, граф матиме кількість вершин, що дорівнює сумі числа партій і областей. Нехай кожна людина, котра виявила бажання брати участь у роботі комітету, зображається дугою графа, яка з'єднує вершини, що відповідають області, у якій вона проживає, і партії, до якої вона належить. Будь-який комітет, що задовольняє всі географічні і партійні вимоги, відповідає якому-небудь покриттю цього графа. Шуканий комітет з найменшим числом членів відповідає покриттю графа з найменшою кількістю дуг, тобто покриттю мінімальної потужності.

Задача 9 (Проблема посередницького агентства). Посередницьке агентство створює можливість зустрічі кожному клієнту, що звернувся до нього, принаймні з одним прийнятним для нього виробником товару. Розмір витрат агентства на організацію кожної зустрічі є різним залежно від конкретних заходів, необхідних для організації (витрати часу агента, витрати коштів агентства, час і місце зустрічі клієнтів, система уподобань учасників зустрічі тощо).

Як агентство може з мінімальними витратами виконати всі зобов'язання перед клієнтами?

Вказівка. Для розв'язання задачі слід побудувати граф, у якому кожному клієнту відповідає вершина і кожній сумісній парі – дуга. Кожній дузі потрібно приписати вагу, що дорівнює витратам на відповідну зустріч.

Кожне покриття цього графа являє собою спосіб організації принаймні однієї прийнятної зустрічі для кожного клієнта.

У задачі потрібно знайти для агентства покриття з найменшими загальними витратами, інакше кажучи – покриття з мінімальною вагою.

Задача 10 (Задача бюро знайомств). Останнім часом значного поширення набули організації, що підшукують пари людей, які задовольняють один одного відповідно до певних вимог. Задача бюро – надати кожному клієнту якнайбільше варіантів вибору. Яку стратегію роботи йому слід вибрати?

Вказівка. Це один з різновидів попередньої задачі. Слід побудувати граф, у якому кожному клієнту відповідає вершина і кожній сумісній парі – дуга. Кожній дузі потрібно приписати вагу, що дорівнює витратам на відповідну зустріч.

Як і в попередній задачі, будь-яке покриття цього графа являє собою спосіб організації принаймні однієї зустрічі, прийнятної для кожного клієнта. Бюро потрібно знайти покриття з мінімальною вагою.

На завершення наведемо дві історичні задачі, що поклали основу теорії мереж.

Задача листоноші

Задача листоноші полягає в тому, щоб пройти всі вулиці свого маршруту і повернутися в початкову точку, мінімізуючи при цьому довжину пройденого шляху.

Задача листоноші має два різновиди:

- перший пов'язаний з вимогою обходу кожної ділянки шляху тільки один раз;
- другий – з відсутністю цієї вимоги.

Для розв'язання задачі треба побудувати граф, у якому кожне ребро відповідає відрізку шляху, а кожна вершина – точці перетину цих шляхів. Тоді задача листоноші являє собою задачу пошуку найкоротшого маршруту (див. розділ 5), у якому вершини початку і кінця руху збігаються і який включає кожне ребро або лише один раз, або принаймні один раз.

Задачу можна розв'язувати як для орієнтованих, так і для неорієнтованих графів.

Для неорієнтованих графів проблема вирішується набагато простіше. Найкоротший маршрут для них можна знайти щоразу, якщо не потрібно, щоб кожне ребро проходилося рівно один раз. Для орієнтованих графів навіть за цієї умови розв'язок існує не завжди.

До подібних формулювань зводяться задачі.

Задача 11. Виконання кур'єрських завдань.

Задача 12. Патрулювання вулиць.

Задача 13. Вибір найкращого маршруту руху сільськогосподарських машин під час сівби.

Задача 14. Вибір оптимального маршруту обслуговування клієнтів телемайстром.

Задача 15. Визначення оптимальної стратегії доставки товарів.

Задача 16. Огляд експонатів виставки.

Задача 17. Відвідування визначних пам'яток туристичною групою.

Задача 18. Перевірка готовності великого об'єкта до пуску.

Задача 19. Оптимальне проходження стадій незалежних операцій.

З орієнтованими графами справа набагато складніша. У застосуванні до практичних задач орієнтований граф для задачі листоноші відповідає таким ситуаціям, коли, наприклад, весь шлях чи якась його ділянка допускають тільки односторонній рух. На відміну від неорієнтованих графів, для яких розв'язок задачі існує завжди, задача листоноші на орієнтованому графі може не мати розв'язку.

Цю ситуацію підтверджує рис. 4. На ньому видно, що якщо листоноша досягає вершини *a,* він не зможе повернутися у вершину *c,* тому що з вершини *a* не виходить жодної дуги до вершини с. Відповідна ситуація і для вершини *b*.

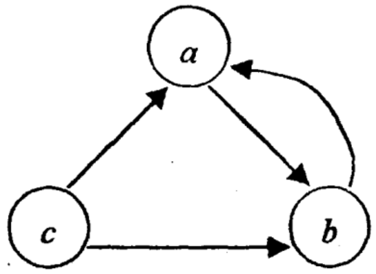


Рис. 4. Граф, для якого маршруту листоноші не існує

Задача комівояжера

Загальною задачею комівояжера називається задача пошуку маршруту найменшої довжини, що включає усі вершини графа (не обов'язково один раз).

Задачею комівояжера називається задача пошуку гамільтонового контуру (тобто маршруту, в який кожна вершина входить рівно один раз).

Оптимальний маршрут комівояжера не обов'язково є гамільтоновим контуром. Для прикладу розглянемо граф, зображений на рис. 5.

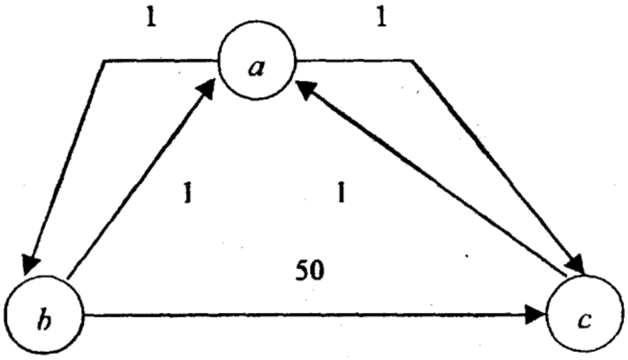


Рис. 5. Інтерпретація гамільтонового контуру

У цьому графі є один гамільтонів контур: (a, *b*), (*b,* c), (c, a), загальна довжина якого 1+50+1 = 52.

Оптимальний маршрут комівояжера не обов'язково є гамільтоновим контуром. Наприклад, для розглянутого контуру оптимальний маршрут проходить через кожну вершину двічі: (*a*, *b*), *(b, a)*, *(a, c)*, (*c*, *a*) і має довжину 1+1+1+1 =4.

Гамільтонів контур існує не для кожного графа. Загального методу пошуку гамільтонового контуру не знайдено. Не існує також і загального методу розв'язання задачі комівояжера. Взагалі методів досить багато (Беллемора, Немхаузена, Хелдома і Карна, метод гілок і меж, метод послідовного поліпшення розв'язання та інші). Усі ці методи належать до одного з двох класів:

- методи, що завжди приводять до знаходження оптимального розв'язку, але вимагають у найгіршому разі неприпустимо великої кількості операцій;
- методи, що не завжди приводять до оптимального розв'язку, але вимагають для здійснення прийнятної кількості операцій.

Практичні задачі, що формулюються як задачі комівояжера, часто мають цікаві і несподівані на перший погляд формулювання. Наведемо деякі з них.

Задача 20 (Вибір оптимального туристичного маршруту). Турист збирається на автомобілі відвідати ряд цікавих з його погляду місць, проїхавши при цьому мінімальну відстань. Який маршрут він має вибрати?

Ця задача може розв'язуватися в різних різновидах. Наприклад, з погляду туристичної фірми можна розв'язувати задачу розробки різних екскурсійних маршрутів, що відповідають певним вимогам, пов'язаним з їх довжиною або часом обходу запланованих місць.

Вказівка. Як вершини графа слід узяти пункти всіх вибраних визначних пам'яток, як ваги – довжини ділянок доріг. Тоді сформульована задача відповідає загальній задачі комівояжера на отриманому графі.

Задача 21 (Обробка виробу на кількох верстатах). У цеху з п різними верстатами виріб має бути оброблений на кожному з верстатів у деякій довільній послідовності. При цьому час переналагоджування, необхідний для передачі виробу від верстата х до верстата у, дорівнює а(х, у). Яку послідовність обробки на кожному з верстатів вибрати, щоб мінімізувати загальний час обробки виробу?

Вказівка. Ця задача розв'язується як задача комівояжера на графі, кожна вершина якого відповідає верстату, а кожна дуга (*x, y)* має вагу *a(x, y)*.

Задача 22 (Тактика кур'єра). Кур'єр банку щодня має доставляти листи з центрального офісу в усі інші відділення банку. Зазвичай, коли він приносить листи в яке-небудь відділення, йому вручають додаткову кореспонденцію, що має бути доставлена в наступне на його шляху відділення. Негативно ставлячись до цієї роботи, кур'єр хотів би довідатися, у якому порядку йому потрібно відвідувати пункти призначення, щоб мінімізувати загальну кількість листів, що розносяться додатково.

Вказівка. Кур'єр має розв'язати свою задачу шляхом розв'язання загальної задачі комівояжера на графі, вершини якого відповідають відділенням банку, а дуги – можливим переїздам між ними.

Дуги при цьому слід інтерпретувати в такий спосіб. Нехай у цьому графі довжина дуги (x, y) дорівнює прогнозованій кількості додаткових листів, що кур'єру довелося б перевезти з відділення x у відділення y.

Припустимо, що кур'єр хоче дістати підвищення на службі і для цього має максимально збільшити загальну кількість листів, що доставляються додатково. У цьому випадку він міг би, звичайно, безупинно переміщатися між усіма відділеннями банку й у підсумку доставляти практично необмежену кількість листів. Однак припустимо, що йому дозволяється кожне відділення відвідувати рівно один раз. Нехай М дорівнює деякому дуже великому числу і нехай довжина кожної дуги тепер дорівнює М мінус її початкова довжина. Якщо є n відділень банку, то кожен гамільтонів контур графа включає n дуг.

Тепер задача кур'єра може розв'язуватися шляхом пошуку мінімального за довжиною гамільтонового контуру з n дуг.

§ 6. Мережне планування

Ця задача називається також задачею управління проектом. Полягає вона в наступному. Багато проектів, такі як:

- велике будівництво;
- наукова робота;
- складання бухгалтерського звіту;
- розробка програмного комплексу для комп'ютера;
- навчання в інституті;
- прибирання приміщення;
- написання дипломної роботи

і цілий ряд інших можна розбити на велику кількість операцій. Деякі з цих операцій можуть виконуватися одночасно, інші – тільки послідовно.

Задача управління проектом полягає в тому, щоб забезпечити його своєчасне завершення з урахуванням часу, необхідного для виконання кожної операції, і дотриманням певної послідовності робіт. Дуже часто при цьому вимагається завершення за мінімальний час.

З теорією графів ця задача пов'язана безпосередньо, тому що кожен проект можна зобразити у вигляді деякого графа, що має назву *мережного графіка.*

Уперше системи, що використовують мережні графіки, були застосовані в США наприкінці 50-х років і одержали назву СРМ (метод критичного шляху) і PERT (метод оцінювання й огляду програми). У Росії роботи із мережного планування почалися в 60-х роках, спочатку в будівництві і наукових розробках. Згодом мережні методи почали широко застосовуватися і в інших галузях.

Для того, щоб побудувати мережний графік, необхідно задати:

- перелік всіх операцій проекту;
- час, необхідний для виконання кожної операції;
- переліки операцій, що безпосередньо передують кожній операції.

Після цього кожну операцію слід зобразити у вигляді дуги орієнтованого графа, вершинам поставити у відповідність моменти закінчення конкретних робіт.

Розглянемо кілька типових задач зі складання мережних графіків.

Задача 23 (Спорудження будинку). Спорудження будинку передбачає тривалу послідовність робіт, основні з яких: розчищення ділянки, закладка фундаменту, зведення стін, штукатурні роботи, благоустрій території, оздоблювальні роботи, настил даху тощо. Потрібно скласти графік проведення робіт, що охоплює мінімально можливий час.

Вказівка. Для розв'язання задачі слід скласти послідовність всіх операцій проекту і для отриманої мережі знайти критичний шлях.

Задача 24 (Виконання наукового дослідження). Велике наукове дослідження завжди передбачає тривалу послідовність проміжних робіт, деякі з яких можна виконувати паралельно, а деякі тільки послідовно, після закінчення попередніх. Необхідно знайти таку послідовність цих робіт, виконання якої вимагає мінімального часу.

Вказівка. Для розв'язання задачі слід скласти послідовність усіх робіт, з яких складається наукове дослідження, і для отриманої мережі знайти критичний шлях.

Очевидно, що час закінчення проекту не може бути меншим ніж сума тривалості всіх операцій, взята протягом найбільш несприятливого, тобто найдовшого шляху, що з'єднує першу й останню роботи на мережному графіку.

Алгоритм, що дає можливість знайти цей шлях, називається методом критичного шляху. Метод його знаходження, а також інші алгоритми, пов'язані з цією задачею, описані в [10, 5, 20, 21, 16].

Для урахування безперервності виконання кожної операції розроблено спеціальний метод PERT [16].

ВАРІАНТИ ЗАВДАНЬ ДЛЯ СЕМІНАРСЬКОГО ЗАНЯТТЯ ЗА ТЕМОЮ “МЕРЕЖНЕ ПЛАНУВАННЯ”

Варіант 1

Завдання 1

Побудувати фрагмент графіка мережі, якщо:

1. початок роботи **e** залежить від закінчення роботи **c**; початок роботи **d** – тільки від закінчення робіт **a** і **c**; початок роботи **h** – від закінчення робіт **b** і **e;**
2. початок роботи **e** залежить тільки від закінчення робіт **a** і **c**; початок роботи **d** – тільки від закінчення роботи **c**; початок роботи **h** – тільки від закінчення робіт **b** і **c;**
3. початок роботи **e** залежить від закінчення роботи **b**; початок роботи **d** – від закінчення робіт **a**, **c** і **e**; початок роботи **h** – від закінчення робіт **a** і c.

Завдання 2

Визначити критичний шлях і найбільш ранні із можливих термінів настання завершеної події та події під номерами 3, 5, 6, якщо графік мережі дано:

а) на рисунку 1;
б) на рисунку 2.

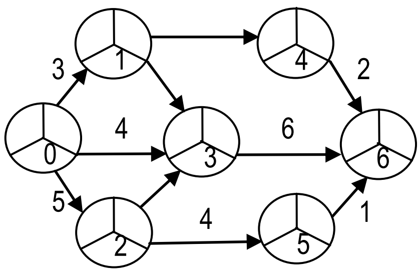


Рис.1

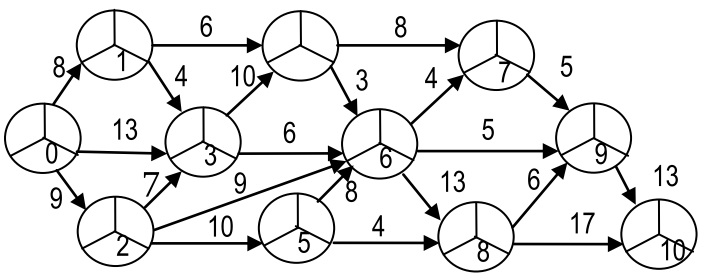
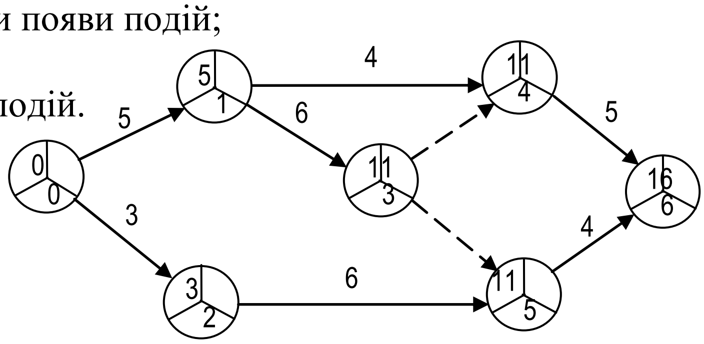


Рис. 2

Завдання 3

За графіком мережі визначити:

а) найбільш ранні терміни появи подій;
б) критичний шлях;
в) резерви часу кожної з подій.



Варіант 2

Завдання 1

Побудувати фрагменти графіків мережі, якщо:

а) роботу **e** можна почати після закінчення робіт **a** і **c**; роботу **h** – після закінчення робіт **b** і **c**;

б) роботи **a** і **c** мають початком одну і ту саму подію; робота **e** може початися після закінчення робіт **a** і **c**; робота **d** – після закінчення робіт **b** і **c**; робота **h** – після закінчення роботи **b**.

Завдання 2

Визначити критичний шлях і найбільш ранні із можливих термінів настання завершеної події та події під номерами 3, 5, 6, якщо графік мережі дано:

а) на рисунку 1;

б) на рисунку 2.

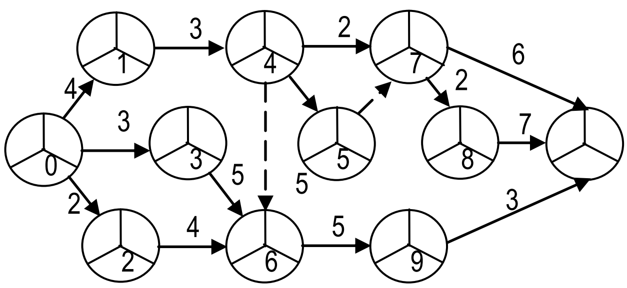


Рис. 1

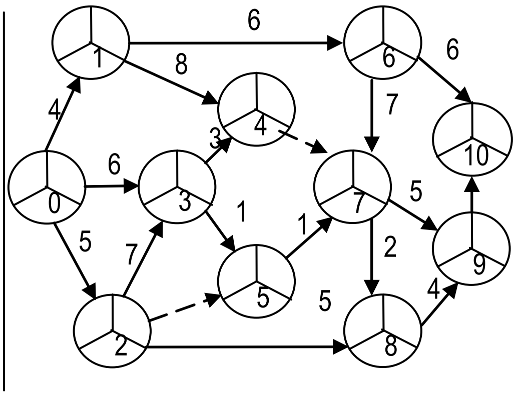


Рис. 2

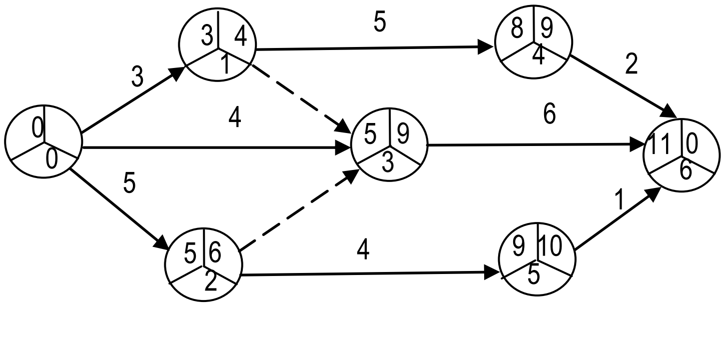
Завдання 3

За графіком мережі визначити:

а) найбільш ранні терміни появи подій;

б) критичний шлях;

в) резерви часу кожної з подій.



Варіант 3

Завдання 1

Побудувати фрагмент графіка мережі, якщо:

а) початок роботи **е** залежить тільки від закінчення робіт **а**, **б** і **с**; початок роботи **d** – тільки від закінчення робіт **с** і **б;**

б) початок роботи **е** залежить тільки від закінчення робіт **а** і **с**; початок роботи **d** – тільки від закінчення робіт **б** і **с;**

в) роботи **б** і **с** мають початком одну й ту саму подію; початок роботи **е** залежить тільки від закінчення робіт **а**, **б** і **с**; початок роботи **d** – тільки від закінчення роботи **б**.

Завдання 2

Визначити критичний шлях і найбільш ранні з можливих термінів настання завершеної події та події під номерами 3, 5, 6, якщо графік мережі дано:

а) на рисунку 1;

б) на рисунку 2.

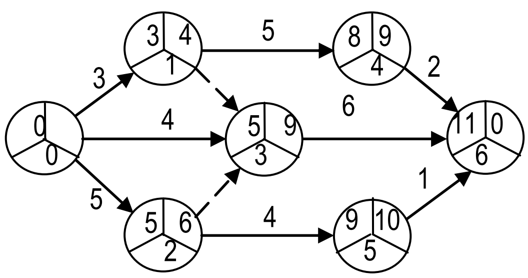


Рис. 1

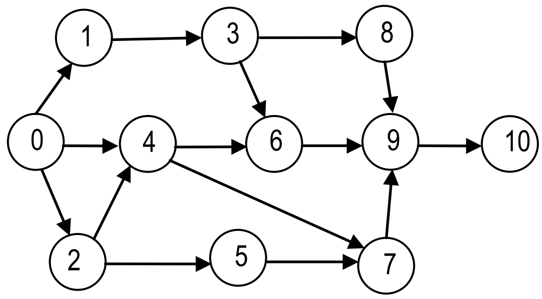


Рис. 2

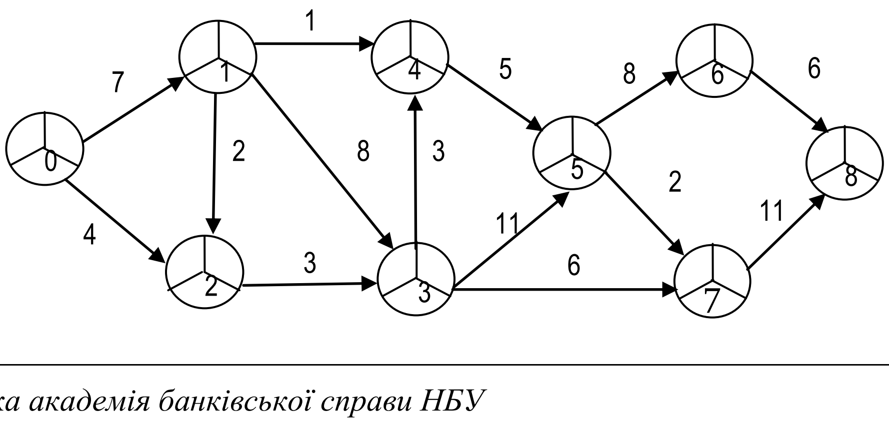
Завдання 3

За графіком мережі визначити:

а) ранішні терміни появи подій;

б) критичний шлях;

в) резерви часу кожної з подій.



СПИСОК ЛІТЕРАТУРИ

1. Авондо-Бодино Дж. Применение в экономике теории графов. – М.: Прогресс, 1966.
2. Бардачов Ю.М., Соколова Н.А., Ходаков В.Э. Дискретна математика. – К.: Вища школа, 2002.
3. Басакер Р., Саати Т. Конечные графы и сети. – М.: Наука, 1974.
4. Белоусов А.И., Ткачев С.Б. Дискретная математика. – М.: Изд-во МГТУ им. Баумана, 2001.
5. Березина Л.Ю. Графы и их применение. – М.: Просвещение, 1979.
6. Берж К. Теория графов и ее применение. – М.: ИМ, 1962.
7. Венцель Е.С. Исследование операций: задачи, принципы, методология. – М.: Наука, ГРФМЛ, 1980.
8. Дискретная математика и математические вопросы кібернетики / Под ред.: С.В. Яблонского и О.Б. Лупанова. – М.: Наука, 1974.
9. Емеличев В.А., Мельников О.И., Сарванов В.И., Тышкевич Р.И. Лекции по теории графов. – М.: Наука, 1990.
10. Исследование операций в экономике: Учебное пособие для вузов / Н.Ш. Кремер, Б.А. Путко, И.М. Трешкин, М.Н. Фридман / Под ред. Н.Ш. Кремера. – М.: Банки и биржи, ЮНИТИ, 1997.
11. Кофман А., Дебазей Г. Сетевые методы планирования и их применение. – М.: Прогресс, 1968.
12. Кривцов А.М., Шеховцов В.В. Сетевое планирование и управление. – М.: Экономика, 1969.
13. Кристофидес Н. Теория графов: алгоритмический подход. – М.: Мир, 1978.
14. Логистика: Учебное пособие / Под ред. Б.А. Аникина. – М.: ИНФРА-М, 2000.
15. Логистика: Учебник / Под ред. Б.А. Аникина. – М.: ИНФРА-М, 2000.
16. Майника Э. Алгоритмы оптимизации на сетях и графах. – М.: Мир, 1981.
17. Машина Н.І. Математичні методи в економіці. – К.: ЦНЛ, 2003.
18. Новиков Ф.А. Дискретная математика для программистов. – СПб.: Питер, 2003.
19. Оре О. Графы и их применение. – М.: Мир, 2002.
20. Параубек Г.Э. Сетевое планирование и управление. – М.: Экономика, 1967.
21. Разумов И.М., Белова Л.Д. и др. Сетевые графики в планировании. – М.: Высшая школа, 1975.
22. Уилсон Р. Введение в теорию графов. – М.: Мир, 1977.
23. Форд Л.Р., Фалкерсон Д.Р. Потоки в сетях. – М.: Мир, 1966.
24. Харари Ф. Теория графов. – М.: Мир, 1973.
25. Ху Т. Целочисленное программирование и потоки в сетях. – М.: Мир, 1974.
26. Яблонский С.В. Введение в дискретную математику. – М.: Наука, 1986.
27. Ядренко М.Й. Дискретна математика. – К.: КНУ, 2002.

ЗМІСТ

Навчально-методичне видання

Ніколаєва Катерина Володимирівна
Койбічук Віталія Василівна

ДИСКРЕТНИЙ АНАЛІЗ
Графи та їх застосування в економіці

Навчально-методичний посібник

Редактори
Н.І. Одарченко
І.О. Кругляк
Комп'ютерна верстка
В.А. Івакін

Підписано до друку 25.01.2007. Формат 60x90/16. Гарнітура Times.
Обл.-вид. арк. 2,5. Умов. друк. арк. 5,2. Тираж 56 пр. Зам. № 696.

Інформаційно-видавничий відділ
Української академії банківської справи Національного банку України
Адреса: 40030, м. Суми, вул. Петропавлівська, 57.

Надруковано на обладнанні Української академії банківської справи
Національного банку України