6 ТЕОРІЯ ГРАФІВ

6.1 Графи та відношення

6.1.1 Основні відомості

Теорія графів – потужний апарат для розв’язування прикладних завдань найрізноманітніших галузей науки й техніки, до яких належать, наприклад: аналіз та синтез електричних кіл та систем, проектування мереж зв’язку та дослідження скінченних автоматів, мережеве планування й керування, вибір оптимальних маршрутів та потоків у мережах, моделювання життєдіяльності й нервової системи в живих організмах тощо.

Початок теорії графів як математичної дисципліни було покладено Ейлером у його відомих міркуваннях щодо кенігсберзьких мостів. Однак ця стаття Ейлера, опублікована 1736 року була єдиною упродовж майже 100 років. Інтерес до проблем теорії графів відродився близько середини XIX сторіччя і був зосереджений переважно в Англії. Існувало чимало причин для такого пожвавлення вивчення графів. Природничі науки вплинули на це завдяки дослідженням електричних мереж, моделей кристалів та структур молекул. Розвинення формальної логіки призвело до вивчення бінарних відношень у формі графів.

Величезна кількість популярних головоломок формулювалися безпосередньо в термінах графів. Найвідоміше з-посеред цих задач – проблема чотирьох фарб – уперше поставлена перед математиками де Морганом приблизно 1850 року (задача щодо визначення кількості припустимих фарб для розфарбування розбиття будь-якої площини так, щоб ніякі суміжні області не були однакового кольору). Жодна інша проблема теорії графів не породжувала стільки численних, часто дотепних робіт.

У разі потреби подавання в наочній формі системи взаємопов’язаних об’єктів звертаються до такої побудови: на площині чи у просторі обирають кілька точок і певні пари з цих точок поєднують лініями. Об’єкт, здобутий у наслідок такої побудови, називається *графом*.

За приклади графів можуть слугувати блок-схема алгоритму, з’єднання в електричній схемі, мережа шляхів поміж населеними пунктами.

Одну й ту саму систему об’єктів та зв’язків між ними можна відобразити по-різному, застосовуючи наведену вище побудову: у різні способи розміщувати точки, за їхні з’єднувальні лінії брати ті чи інші криві тощо.

Більш того, можна взагалі не зображати, а зазначити систему зв’язків об’єктів у якій завгодно іншій формі, наприклад у словесній. Це міркування засвідчує, що потрібне визначення графа як певного формального об’єкта, який можна подавати наочно у всілякі способи.

6.1.2 Визначення графа

Стверджуватимемо, що задано *скінченний неорієнтований граф*, якщо задано такі два об’єкти:

1) скінчена непорожня множина $X = \{x_1, x_2, ..., x_n\}$; елементи цієї множини називають *вершинами графа*;

2) деяка множина невпорядкованих пар елементів з *X*; ця множина позначається *U*, її елементи називають *ребрами*.

Той факт, що граф означується парою множин *X* та *U*, записують у вигляді $G = (X, U)$.

За наочного подавання графа вершини зображуються *точками*, ребра – *лініями*, які з’єднують точки.

Приклад 6.1 $G = (X, U)$, **де** $X = \{x_1, x_2, x_3, x_4, x_5\}$**;** $U = \{\{x_1, x_2\}, \{x_2, x_3\}, \{x_3, x_4\}, \{x_4, x_5\}\}$**. Наочно цей граф зображено на рис. 6.l.**

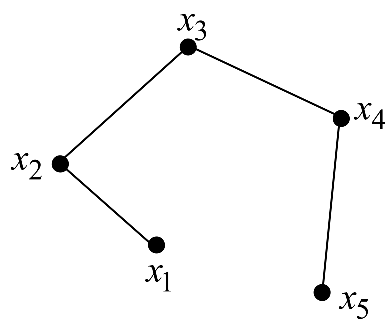


Рисунок 6.1 – Приклад графа

Зауваження. Якщо $u_1 = \{x_1, x_2\}$ – ребро графа, то стверджують, що ребро u_1 з’єднує вершини x_1 та x_2.

Поряд із наведеним визначенням графа можливі й інші визначення графа.

Іноді виникає потреба розглядати графи, в яких одну й ту саму пару вершин з’єднує кілька ребер. Такі графи називаються *мультиграфами* (рис. 6.2).

Можливі також графи, в яких певні ребра можуть мати збіжні кінці. Такі ребра називають *петлями* (рис. 6.3).

У більшості додатків теорії графів можна відкидати петлі й замінювати кратні ребра на одне ребром. Тому надалі подане вище визначення буде головним і словом «граф» позначатимемо *скінченний неорієнтований граф без петель і кратних ребер* (його ще називають *простим*, або *звичайним*) (рис. 6.4).

Граф з петлями і кратними ребрами називається *псевдографом* (рис. 6.5).

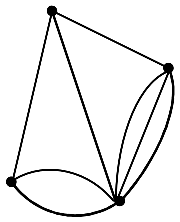


Рисунок 6.2 – Мультиграф

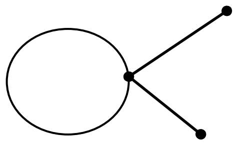


Рисунок 6.3 – Граф з петлею

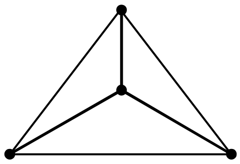


Рисунок 6.4 – Простий граф

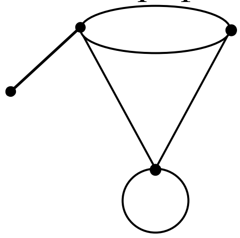


Рисунок 6.5 – Псевдограф

6.1.3 Орієнтовані графи

Поняття орієнтованого графа (орграфа) виникає, якщо ребрам графа надати напрямок (тобто орієнтацію) в такий спосіб, що один з кінців ребра буде початком, а інший – кінцем.

Стверджуватимемо, що задано *орієнтований граф*, якщо зазначено два об'єкти:

1) непорожня скінчена множина X – вершини графа;
2) множина U, утворена з упорядкованих пар вершин.

Елементи множини U називають *дугами*. Дуга орієнтованого графа зображується відрізком із зазначенням напрямку (стрілкою) (рис. 6.6).

Рисунок 6.6 – Зображення орієнтації дуг

Приклад 6.2 Орієнтований граф $G=(X,U)$, де $X=\{x_1, x_2, x_3, x_4, x_5\}$; $U=\{\{x_1, x_2\}, \{x_2, x_3\}, \{x_3, x_4\}, \{x_4, x_5\}\}$. Граф G зображено на рис. 6.7.

Зауваження. Якщо $u_1=(x_1, x_2)$ – дуга орграфа, то стверджують, що дуга u_1 виходить з вершини x_1 і закінчується у вершині x_2.

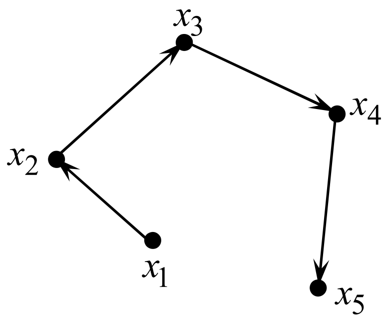


Рисунок 6.7 – Зображення орграфа G

Стверджуватимемо, що задано *орієнтований мультограф*, якщо зазначено два об'єкти:

1) непорожня множина X – вершини графа;
2) множина U, утворена з упорядкованих пар вершин.

Отже, елементи (дуги) в U в разі орієнтованого мультиграфа можуть повторюватись; такі дуги називають *кратними*. Зауважимо, що кратні дуги з'єднують одну пару вершин і однаково напрямлені.

Надалі ми будемо використовувати термін «граф» для опису довільних графів – орієнтованих і неорієнтованих, із петлями та кратними ребрами чи без них, а термін «неорієнтований граф» або «псевдо граф» – для довільного неорієнтованого графа, який може мати кратні ребра й петлі. Означення різних типів графів зведено в табл. 6.1.

Таблиця 6.1 – Означення різних типів графів

Тип графа	**Ребра**	**Чи дозволені кратні ребра?**	**Чи дозволені петлі?**
Простий граф	неорієнтовані	ні	ні
Мультиграф	неорієнтовані	так	ні
Псевдограф	неорієнтовані	так	так
Орієнтований граф	орієнтовані (дуги)	ні	так
Орієнтований мультиграф	орієнтовані (дуги)	так	так

6.1.4 Найпростіші поняття теорії графів

Нехай задано граф $G=(X,U)$.

Про ребро $u=\{x,\ y\}$ цього графа стверджують, що воно з'єднує вершини *x* та *y*.

Дві вершини, з'єднані ребром, називаються *суміжними*, якщо вони є кінцями одного ребра.

Про ребро $u=\{x,\ y\}$ та вершину *x* стверджують, що вони є *інцидентні*. Те ж саме можна сказати й про ребро $u=\{x,\ y\}$ та вершину *y*.

Далі позначатимемо *кількість вершин* графа – літерою *n*, а *кількість ребер* графа – літерою *m*: $|X|=n$, $|U|=m$. Це основні числові характеристики графа.

Кількість ребер, інцидентних до певної вершини *x*, називається *степенем* цієї вершини і позначається $\delta(x)$, або $\deg(x)$.

Вершина, в якої степінь дорівнює 0, називається *ізольованою* (вершина *x* рис. 6.8). Вершини, які мають степінь 1, називаються *висячими*, або *кінцевими* (вершина *x* рис. 6.9).

Рисунок 6.8 – Граф з ізольованою вершиною *x*

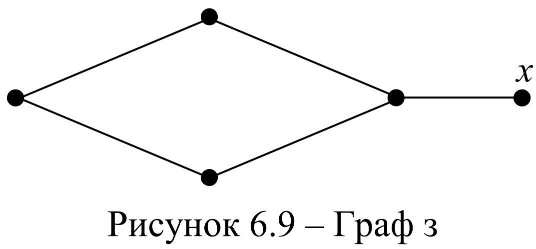


Рисунок 6.9 – Граф з висячою вершиною *x*

Справедливими є два такі простих твердження.

Теорема 6.1 Сума степенів усіх вершин графа дорівнює подвоєній кількості ребер.

Доведення. Кожне ребро двічі входить до суми, звідки й випливає твердження.

Теорема 6.2 У кожному графові число вершин, які мають непарний степінь, є парне.

Доведення. Нехай $X_1 \subseteq X$ – множина вершин непарного степеня; $X_2 \subseteq X$ – множина вершин парного степеня. Зазначимо, що

$$X = X_1 \cup X_2, \qquad X_1 \cap X_2 = \varnothing,$$

Отже

$$\sum_{x \in X} \delta(x) = \sum_{x \in X_1} \delta(x) + \sum_{x \in X_2} \delta(x).$$

Тоді

$$\sum_{x \in X_1} \delta(x) = \sum_{x \in X} \delta(x) - \sum_{x \in X_2} \delta(x).$$

Вочевидь, що $\sum_{x \in X_2} \delta(x)$ є парна як сума парних чисел: $\sum_{x \in X} \delta(x)$ – парна відповідно до теореми 1. Отже, $\sum_{x \in X_1} \delta(x)$ – парна, що й треба було довести.

Для орієнтованих графів замість степеня вершини x вводять поняття півстепенів: *додатні* $\delta_+(x)$ й *від'ємні* $\delta_-(x)$ *півстепені* вершини x:

$\delta_+(x)$ – число дуг, які входять до вершини x;

$\delta_-(x)$ – число дуг, які виходять з вершини x.

Граф, який не має ребер ($U = \varnothing$), називається *порожнім*. Усі вершини порожнього графа є *ізольовані*.

Граф, в якому кожна пара вершин з'єднана ребром, називається *повним*.

Повний n-вершинний граф позначається K_n; для кожної його вершини x маємо $\delta(x) = n - 1$ (рис. 6.10).

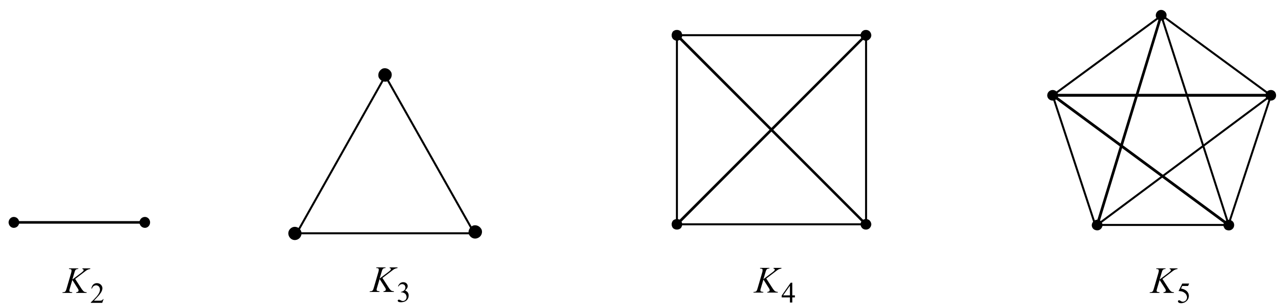


Рисунок 6.10 – Повні графи

6.1.5 Підграфи

Нехай задано граф $G = (X, U)$.

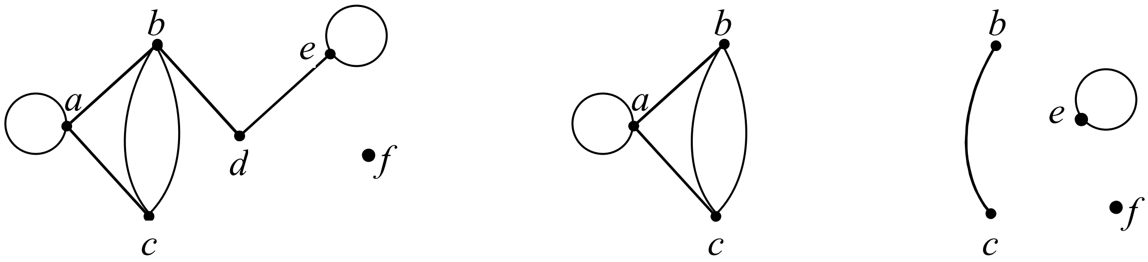
Граф $G_1 = (X_1, U_1)$ називається *підграфом* графа $G = (X, U)$, якщо $X_1 \subseteq X$ та $U_1 \subseteq U$.

Якщо вилучити з графа певні ребра та вершини, дістанемо підграфи вихідного графа (рис. 6.11).

Граф $G_1 = (X_1, U_1)$ ***називається*** **кістяковим підграфом** ***графа*** $G = (X, U)$***, якщо*** $X_1 = X$ ***та*** $U_1 \subseteq U$**.**

Кістяковий підграф здобудемо, якщо в графі *G* вилучимо частину ребер, не зачіпаючи вершин.

Відокремимо в графі *G* певну підмножину вершин $A \subseteq X$.



граф *G* підграф G_1 підграф G_2

Рисунок 6.11 – Підграфи графа *G*

Нехай U_A означує множину ребер графа *G*, обидва кінці яких належать до множини *A*. Підграф $G_A = (A, U_A)$ називають *підграфом, породженим множиною вершин A*.

6.1.6 Способи задання графів

1) Скінченний граф може бути задано ***переліком його елементів***, тобто за визначенням (елементи позначаються латинськими літерами з індексами або просто натуральними числами).

Приклад 6.3 **Граф** $G = (X, U)$**, де** $X = \{1, 2, 3, 4, 5, 6, 7, 8\}$**, можна задати переліком елементів наступним чином:**

$$U = \{\{1, 2\}, \{2, 3\}, \{2, 4\}, \{1, 4\}, \{3, 4\}, \{4, 5\}, \{6, 6\}, \{6, 7\}\}.$$

Такий метод не є наочний, що утруднює виявлення характеристик графа.

2) ***Геометричне задання графа***.

Кожен граф може бути задано геометрично у тривимірному просторі, але не завжди його можна зобразити на площині так, щоб ребра перетинались тільки в вершинах. Граф, який може бути зображено на площині, називається *планарним* (рис. 6.12). Не є планарним повний граф з п’ятьма вершинами (рис. 6.13).

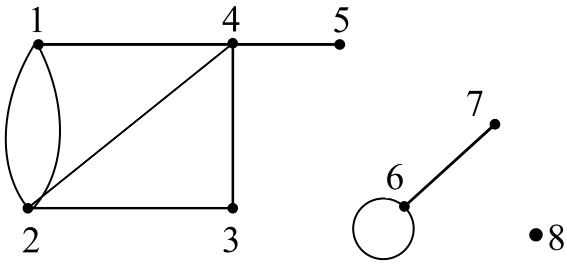


Рисунок 6.11 – Планарний граф

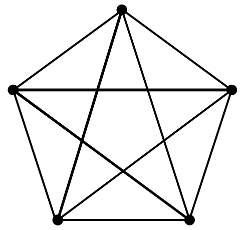


Рисунок 6.12 – Непланарний граф

3) ***Матричне задання графа***.

Не завжди зручно задавати граф у тому вигляді, як це зазначено вище. Наприклад, при опрацюванні графа на комп'ютері його зручно зображати в матричній формі.

1. Розглянемо $G=(X,U)$ – орграф, де

$$X=\{x_1, x_2, ..., x_n\},\ U=\{u_1, u_2, ..., u_m\}.$$

Скінченний орієнтований граф задається *матрицями суміжності* та *інцидентності*.

Матрицею суміжності орграфа G називається квадратна матриця $A(G)=\left(a_{ij}\right)_{i,\,j=\overline{1,n}}$, в якої:

$$a_{ij}=\begin{cases}1, \text{ якщо } \left(x_i, x_j\right)\in U,\\ 0, \text{ якщо } \left(x_i, x_j\right)\notin U.\end{cases}$$

Матрицею інцидентності (або *матрицею інциденцій*) орграфа G називається матриця $B(G)=\left(B_{ij}\right)_{i=\overline{1,n};\ j=\overline{1,m}}$, в якої елементи:

$$b_{ij}=\begin{cases}1, \text{ якщо вершина } x_i \text{ є кінцем дуги } u_j,\\ -1, \text{ якщо вершина } x_i \text{ є початком дуги } u_j,\\ 0, \text{ якщо вершина } x_i \text{ не є інциндентна до дуги } u_j.\end{cases}$$

Приклад 6.4 Розглянемо орграф G, який задано геометрично (рис. 6.14). Для нього матриця суміжності матиме вигляд

$$A(G)=\begin{array}{c} \\ x_1\\ x_2\\ x_3\end{array}\begin{array}{c}\begin{array}{ccc} x_1 & x_2 & x_3\end{array}\\ \begin{pmatrix}0 & 1 & 0\\ 1 & 0 & 1\\ 1 & 0 & 0\end{pmatrix}\end{array}.$$

Матриця інцидентності матиме вигляд

$$B(G)=\begin{array}{c} \\ x_1 \\ x_2 \\ x_3 \end{array}\!\!\begin{array}{c} \begin{array}{cccc} u_1 & u_2 & u_3 & u_4 \end{array} \\ \begin{pmatrix} -1 & 0 & 1 & 1 \\ 1 & -1 & 0 & -1 \\ 0 & 1 & -1 & 0 \end{pmatrix} \end{array}.$$

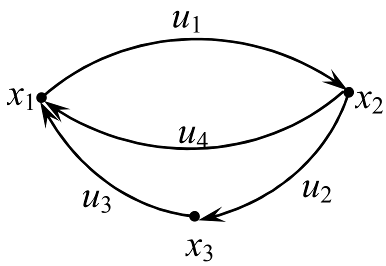


Рисунок 6.14 – Орграф G

2. Розглянемо $G=(X,U)$ – скінченний неорієнтований граф, де

$$X=\{x_1, x_2, ..., x_n\},\ U=\{u_1, u_2, ..., u_m\}.$$

Скінченний орієнтований граф задається *матрицями суміжності* та *інцидентності*.

Матрицею суміжності графа G називається квадратна матриця $A(G)=(a_{ij})_{i,\,j=\overline{1,n}}$, в якої:

$$a_{ij}=\begin{cases} 1, \text{ якщо } (x_i, x_j)\in U, \\ 0, \text{ якщо } (x_i, x_j)\notin U. \end{cases}$$

Матрицею інцидентності графа G називається матриця $B(G)=(B_{ij})_{i=\overline{1,n};\ j=\overline{1,m}}$, в якої елементи:

$$b_{ij}=\begin{cases} 1, \text{ якщо вершина } x_i \text{ є інциндентна до ребра } u_j, \\ 0, \text{ якщо вершина } x_i \text{ не є інциндентна до ребра } u_j. \end{cases}$$

Приклад 6.5 Розглянемо граф G_1, який задано геометрично (рис. 6.15). Для нього матриця суміжності $A(G_1)$ матиме вигляд

$$A(G_1)=\begin{array}{c} \\ x_1 \\ x_2 \\ x_3 \end{array}\!\!\begin{array}{c} \begin{array}{ccc} x_1 & x_2 & x_3 \end{array} \\ \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix} \end{array}.$$

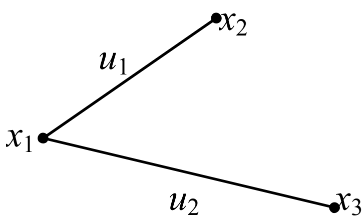


Рисунок 6.15 – Граф G_1

Матриця інцидентності матиме вигляд

$$B(G_1)=\begin{array}{c} \\ x_1 \\ x_2 \\ x_3 \end{array}\!\!\begin{array}{c} \begin{array}{cc} u_1 & u_2 \end{array} \\ \begin{pmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} \end{array}.$$

Зауваження. Матрицю суміжності можна визначити і для псевдографів. Тоді в разі орієнтованого (неорієнтованого) псевдографа $a_{ij}=k$, де k – кратність дуги (x_i, x_j) (ребра $\{x_i, x_j\}$) у цьому псевдографі.

Визначення матриці інцидентності без змін переносяться і на довільні мультиграфи (орієнтовані й неорієнтовані) і навіть на неорієнтовані псевдографи.

Приклад 6.6 Нехай задано геометрично орієнтований псевдограф G (рис. 6.16).

Тоді матриця суміжності $A(G)$ матиме вигляд

$$A(G)=\begin{array}{c} \\ x_1 \\ x_2 \\ x_3 \end{array}\begin{array}{c} \begin{array}{ccc} x_1 & x_2 & x_3 \end{array} \\ \begin{pmatrix} 1 & 2 & 0 \\ 0 & 0 & 0 \\ 2 & 3 & 2 \end{pmatrix} \end{array}.$$

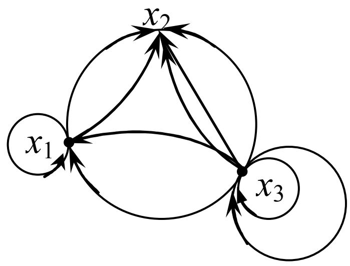


Рисунок 6.16 – Орієнтований псевдограф

Зауваження. Матриця суміжності для звичайних графів і матриця інцидентності для будь-яких графів задає граф однозначно.

Нескладно з'ясувати, що матриця $A(G)$ є симетричною для кожного неорієнтованого графа G. Матриця $A(G)$, де G – орграф, у загальному випадку не є симетричною.

За допомогою матриць зручно задавати графи (орграфи) для опрацювання на комп'ютері. Однак слід зазначити, що за великої кількості вершин матриця суміжності стає громіздкою. Те саме можна сказати і про матрицю інцидентності, причому її розміри залежать, окрім того, й від кількості ребер (дуг) графа.

6.1.7 Ізоморфізм графів

При визначенні будь-якого математичного поняття домовляються, які об'єкти вважати за однакові й які треба розрізнювати.

Ізоморфні об'єкти – це такі об'єкти, які в подальшій теорії не розрізнюються і розглядаються як один об'єкт.

Наприклад, два графи G_1 та G_2, зображені на рис. 6.17, відрізняються лише позначанням вершин і способом розміщення на площині.

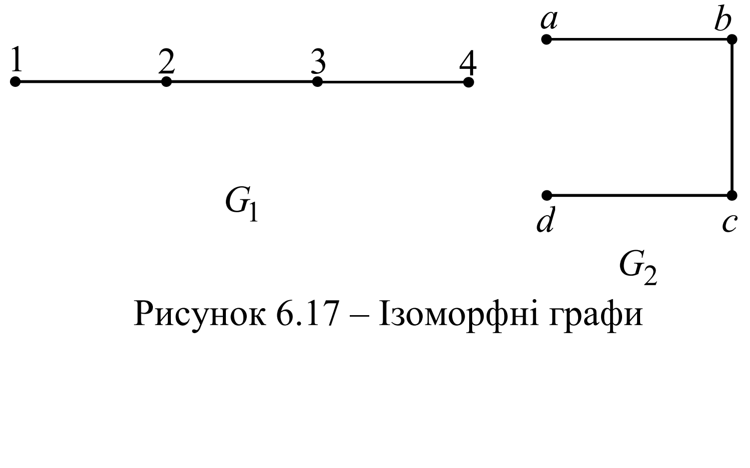


Рисунок 6.17 – Ізоморфні графи

Якщо в графі G_2 перепозначити вершини за схемою *a* – 1; *b* – 2; *c* – 3; *d* – 4, то множина вершин і ребер у графах G_1 та G_2 збігатиметься й матимемо той самий граф.

Графи $G_1 = (X_1, U_1)$ та $G_2 = (X_2, U_2)$ називаються *ізоморфними*, якщо поміж множинами їхніх вершин можна встановити взаємнооднозначну відповідність, за якої кожні дві вершини є суміжні в одному з графів тоді й лише тоді, коли відповідні їм вершини є суміжні в іншому графі.

Приклад 6.7 Графи G_1 та G_2, які зображені на рис. 6.18, є ізоморфні.

G_1 G_2

Рисунок 6.10 – Ізоморфні графи

6.1.8 Зв'язок графа з відношенням

Якщо на певній множині *M* задано бінарне відношення $R \subseteq M \times M$, то *R* можна зобразити як граф $G(R) = (M, R)$. Такий граф не має кратних ребер, тому що $R \subseteq M \times M$, а при перелічуванні елементів множини *M* кожний з них зазначається лише один раз.

Правильним є й обернене твердження: кожен граф без кратних ребер задає певне бінарне відношення на множині його вершин. Якщо відношення є рефлексивне, то граф має в кожній вершині петлю; якщо *R* – симетричне, то $G(R)$ – неорієнтований граф.

6.2 Елементи графів

6.2.1 Маршрути, ланцюги, шляхи та цикли

Нехай $G = (X, U)$ – скінченний неорієнтований граф. Скінчена послідовність вершин та ребер графа

$$x_0 u_1 x_1 u_2 x_2 ... x_{l-1} u_l x_l,$$

в якій кожне ребро u_i є ребро, яке з'єднує вершини x_{i-1} та x_i, називається *маршрутом* на графі *G*.

Говорять, що цей маршрут з'єднує вершини x_0 та x_l. Число l називають *довжиною маршруту*.

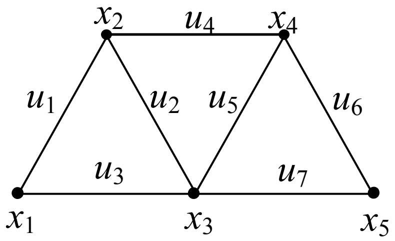
Отже, *довжина маршруту* – це кількість ребер, які входять до маршруту. Маршрут називають *замкненим*, якщо $x_0 = x_l$.

Маршрут, в якому всі ребра є різні, називають *ланцюгом*. Замкнений ланцюг називають *циклом*.

Ланцюг називають *простим*, якщо всі його вершини є різні.

Простий цикл – це цикл, у якому всі вершини, окрім першої та останньої, є різні.

Приклад 6.8 У графі, поданому на рисунку



$x_1u_3x_3u_3x_1u_3x_3u_2x_2u_4x_4u_5x_2u_3x_1$ – маршрут;

$x_1u_3x_3u_2x_2u_4x_4u_5x_3u_7x_5$ – ланцюг;

$x_1u_1x_2u_4x_4u_6x_5$ – простий ланцюг.

Орієнтовані маршрути на орграфі визначують в аналогічний спосіб, з тією різницею, що початкова вершина дуги маршруту має збігатися з кінцевою вершиною попередньої дуги.

Інакше кажучи, рух за маршрутом припускається лише в напрямках, зазначених стрілками.

Маршрут, який не містить повторних дуг, називається *шляхом*, а той, що не містить повторних вершин, – *простим шляхом*. Замкнений шлях називається *контуром*, а простий замкнений шлях – *простим контуром*.

Граф без циклів називається *ациклічним* (орграф – *безконтурним*), в іншому разі граф називається *циклічним* (орграф – *контурним*).

Умовимося вважати, що кожна вершина з'єднується сама з собою маршрутом довжини 0 і що цей маршрут є простим циклом. Такий цикл називають *нульовим* (якщо сказано просто цикл, то мається на увазі, що він не є нульовий).

Теорема 6.1 Нехай задано маршрут *S*. Тоді, якщо він не є замкнений, то містить простий ланцюг з одними й тими самими кінцями.

Доведення.

1) Нехай $S = S\left(a_0, a_n\right)$ – маршрут, який з'єднує вершини a_0 та a_n, де $a_0 \neq a_n$. Якщо він є простим ланцюгом, то твердження теореми доведено.

2) Припустимо, що маршрут *S* не є простим ланцюгом. Отже, знайдеться вершина a_i $(i = \overline{1,n})$, яка входить до цього маршрута, принаймні двічі:

$$S = a_0, a_1, ..., a_{i-1}, a_i, a_{i+1}, ..., a_{k-1}, a_i, a_{k+1}, ..., a_{n-1}, a_n.$$

Тоді $a_i, a_{i+1}, ..., a_{k-1}, a_i$ – замкнений підмаршрут маршрута *S*. Вилучимо його з маршрута *S*. Внаслідок цього дістанемо більш короткий маршрут з тими самими кінцями:

$$S_1 = a_0, a_1, ..., a_{i-1}, a_i, a_{k+1}, ..., a_{n-1}, a_n.$$

Якщо дістаний маршрут не є простим ланцюгом, повторивши попередні міркування, дістанемо новий маршрут S_2.

Оскільки S – скінченний маршрут, то через скінчену кількість кроків m дістанемось певного маршруту S_m, який не містить однакових вершин. Тоді S_m – простий ланцюг.

Теорема 6.2 Кожний замкнений маршрут C містить простий цикл.

Доведення.

Якщо замкнений маршрут $C = C(a_0)$ не є простим циклом, то знайдеться вершина $a_i \neq a_0$, яка входить до даного маршрута, принаймні двічі, чи вершина a_0, яка входить до маршрута принаймні тричі. Тоді з маршрута C можна виокремити більш короткий замкнений маршрут $C_1(a_i)$ або $C_1(a_0)$. Якщо цей маршрут не є простим циклом, повторимо попередні міркування. Оскільки вхідний маршрут є скінченний, то через певну кількість кроків m дістанемо маршрут C_m, який є простим циклом.

Наслідок. Якщо ланцюг не є простим, то він містить простий цикл.

6.2.2 Зв'язність. Компоненти зв'язності

Нехай граф $G = (X, U)$ – неорієнтований.

Вершина a називається *зв'язаною* з вершиною b, якщо існує маршрут, який з'єднує ці вершини. Стверджують, що вершина b *досяжна* з вершини a.

Граф, будь-яка пара вершин якого є зв'язана, називається *зв'язним*.

Якщо в довільному графі G вершина a зв'язана з b, а вершина b зв'язана з c, то, вочевидь, що a є зв'язана з c. Відношення зв'язності для вершин є *відношенням еквівалентності*. Отже, існує таке розкладання множини вершин X:

(1) $X = X_1 \cup X_2 \cup ... \cup X_p$, $X_i \cap X_j = \varnothing$ за $i \neq j$, що всі вершини b однієї множини X_i є зв'язані між собою, а вершини з різних множин X_i не є зв'язані.

(2) $U = U_1 \cup U_2 \cup ... \cup U_p$, $U_i \cap U_j = \varnothing$ за $i \neq j$.

Тоді, відповідно до (1) та (2) матиме пряме розкладання.

(3) $G = G_1 \cup G_2 \cup ... \cup G_p$,

де $G_1 = (X_1, U_1)$, $G_2 = (X_2, U_2)$, ... , $G_p = (X_p, U_p)$ – зв'язні підграфи, що не перетинаються.

***Ці підграфи називаються* зв'язними компонентами графа G, *або* компонентами зв'язності графа G.**

Число p – ще одна числова характеристика графа.

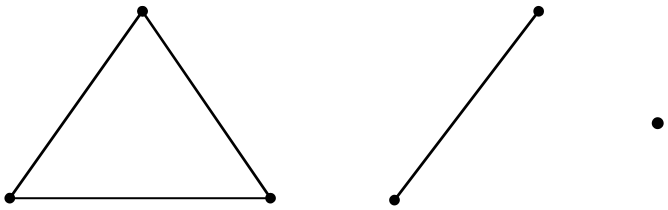
Для зв'язного графа $p = 1$; якщо граф є незв'язний, то $p \geq 2$.

Отже здобуто таке твердження:

Теорема 6.3 Кожен неорієнтований граф розкладається єдиним чином на пряму суму (3) власних зв'язних компонент.

Зауваження. Якщо певний граф не є зв’язним і розкладається на декілька компонентів, то вивчення цього незв’язного графа можна звести до досліджування окремих його компонентів, які є зв’язні. Тому у переважній більшості випадків має сенс припускати, що заданий граф є зв’язний.

Приклад 6.9 Граф, зображений на рисунку, має три компоненти зв’язності.



Через те що кількість компонентів зв’язності дорівнює кількості зв’язних підграфів графа, наведений граф – тризв’язний (число зв’язності $p = 3$).

Зв’язність для орієнтованих графів (орграфів) визначається так само, як і для неорієнтованих, тобто без урахування напрямків дуг. Специфічними для орграфів є поняття сильної, однобічної та слабкої зв’язності.

Орграф називається *сильно зв’язним*, якщо для кожної пари його вершин *a* та *b* існує шлях з вершини *a* до вершини *b*.

Орграф називається *однобічно зв’язним*, якщо для кожної пари його вершин принаймні одна є досяжна з іншої.

Орграф називається *слабко зв’язним*, якщо зв’язним є асоційований з ним псевдограф (рис. 6.19 та 6.20).

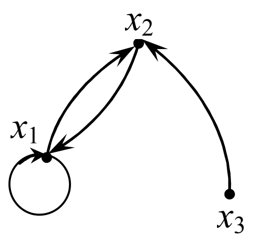


Рисунок 6.19 – Орієнтований псевдограф

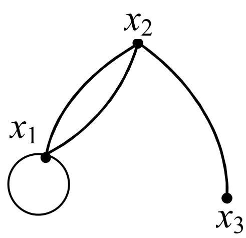
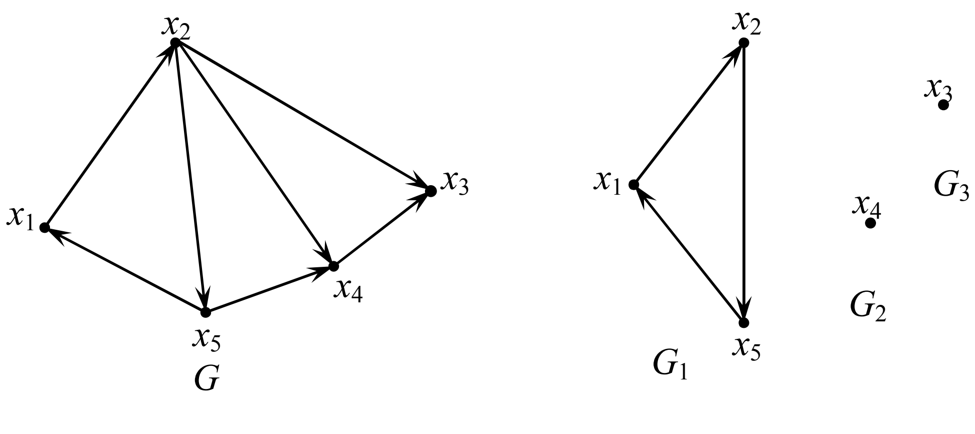


Рисунок 6.20 – Асоційований псевдограф до орієнтованого псевдографа

Приклад 6.10 ***Три компоненти*** G_1, G_2, G_3 ***сильної зв’язності орграфа* G *зображено на рисунку***



Теорема 6.4 Нехай A – матриця суміжності графа $G=(X,U)$. Тоді a_{ij}^{k} – число маршрутів довжини k від вершини x_i до вершини x_j.

6.2.3 Роздільність графа

Зв'язний граф може бути розділено на незв'язані поміж собою підграфи вилучанням з нього певних вершин і/або ребер. При вилученні вершин вилучаються і всі інцидентні до них ребра, а при вилученні ребер інцидентні до них вершини зберігаються.

Вершина, вилучення якої перетворює зв'язний граф на незв'язний, називається *точкою з'єднання* (рис. 6.21).

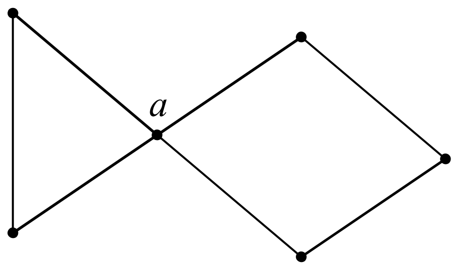


Рисунок 6.21 – Точка зчленування вершина a

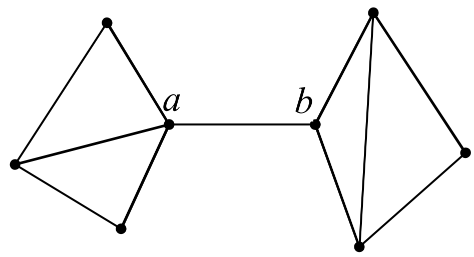


Рисунок 6.22 – Міст – ребро (a, b)

Ребро, вилучання якого перетворює зв'язний граф на незв'язний, називається *мостом* (рис. 3.22).

За наявності моста є хоча б дві точки з'єднання.

Граф називають *нероздільним*, якщо він є зв'язний і не має точок з'єднання.

Граф, який має хоча б одну точку з'єднання, є роздільним і називається *сепарабельним*. Він розбивається на блоки, кожний з яких являє собою максимально нероздільні підграфи.

6.2.4 Матриця відстаней графа

Нехай задано зв'язний граф $G=(X,U)$. *Відстанню поміж двома вершинами* x та y графа G називається довжина найкоротшого ланцюга, який зв'язує ці вершини.

Відстань між вершинами x та y графа G позначається через $d_G(x, y)$ або $d(x, y)$, якщо зрозуміло, про який саме граф йдеться.

Нескладно перевірити, що впроваджене в такий спосіб позначення відстані – задовольняє звичайним аксіомам метрики:

1) $d(x, y) > 0$; $d(x, y) = 0 \Leftrightarrow x = y$;
2) $d(x, y) = d(y, x)$;
3) $d(x, z) = d(x, y) + d(y, z)$.

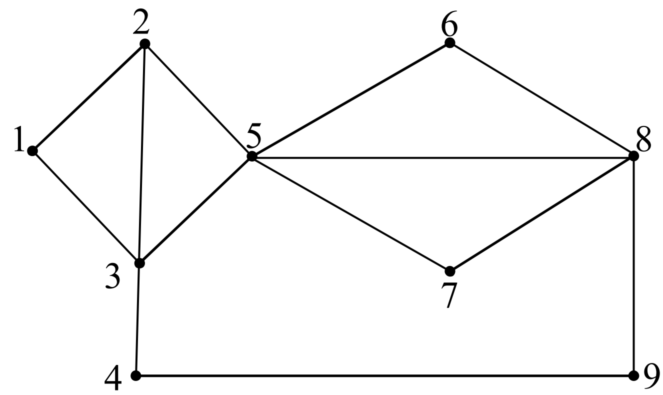
Діаметром графа називається величина $d(G)=\max\limits_{x,\,y} d(x,\,y)$, де максимум береться за всіма можливими парами вершин графа.

Визначимо для кожної вершини x графа G величину $r(x)=\max\limits_{y} d(x,\,y)$, тобто відстань від вершини x до найвіддаленішої від x вершини графа. Мінімум цієї величини за всіма вершинами графа називається *радіусом* графа G:

$$r(G)=\min_{x} r(x)=\min_{x}\max_{y} d(x,\,y).$$

Вершина x_0, в якій досягається цей мінімум, називається *центральною*.

Приклад 6.11 Знайти діаметр і радіус для графа, зображеного на рисунку:



Розв'язання.

Для розв'язання цієї задачі зручно попередньо обчислити так звану матрицю відстаней поміж вершинами графа. У даному разі це буде матриця розміром 9×9*, елемент якої, що стоїть на місці* $(i,\,j)$*, дорівнює відстані від вершини* i *до вершини* j*:*

$$\begin{array}{c c}
\begin{pmatrix}
0 & 1 & 1 & 2 & 2 & 3 & 3 & 3 & 3\\
1 & 0 & 1 & 2 & 1 & 2 & 2 & 2 & 3\\
1 & 1 & 0 & 1 & 1 & 2 & 2 & 2 & 2\\
2 & 2 & 1 & 0 & 2 & 3 & 3 & 2 & 1\\
2 & 1 & 1 & 2 & 0 & 1 & 1 & 1 & 2\\
3 & 2 & 2 & 3 & 1 & 0 & 2 & 1 & 2\\
3 & 2 & 2 & 3 & 1 & 2 & 0 & 1 & 2\\
3 & 2 & 2 & 2 & 1 & 1 & 1 & 0 & 1\\
3 & 3 & 2 & 1 & 2 & 2 & 2 & 1 & 0
\end{pmatrix} &
\begin{array}{c} 3\\ 3\\ 2\\ 3\\ 2.\\ 3\\ 3\\ 3\\ 3 \end{array}
\end{array}$$

За визначенням, діаметр графа дорівнює найбільшому елементові матриці відстаней. Отже, $d=3$.

Для знаходження радіуса відшукаємо в кожному рядку найбільше число; ці числа виписано праворуч від матриці відстаней. Найменші з них дають значення радіуса $r=2$.

Вершини 3-тя та 5-та є центральними.

6.2.5 Задача про найкоротший ланцюг

При обчислюванні відстаней поміж вершинами треба розв'язати таку задачу: у зв'язному графі G задано дві вершини – a та b; знайти ланцюг найменшої довжини, який зв'язує a з b.

Алгоритм розв'язання задачі полягає у послідовному присвоєнні вершинам графа позначок, які дорівнюють кількості ребер найкоротшого ланцюга поміж вершиною, що розглядається, і вершиною a.

Опис алгоритму.

Позначимо вершину a як 0. Усі вершини, суміжні з вершиною a, позначимо як 1. Непомічені вершини, суміжні з вершинами, які мають позначку 1, позначимо 2; суміжні з ними – 3 т. д., допоки не буде позначено вершину b.

Припустімо, що вершина b набула оцінки k. Повертаємося від b до a, відшукуючи послідовно: суміжну з b вершину x_{k-1}, яка має оцінку $k-1$, суміжну з x_{k-1} вершину x_{k-2}, яка має оцінку $k-2$, і т.д. доти, допоки з певної вершини x_1 з оцінкою 1 не дістанемось вершини a.

Ланцюг $a - x_1 - ... - x_{k-1} - b$ – шуканий, він має довжину k.

На прикладі графа, зображеного на рис. 6.22 покажемо оцінки, яких набувають вершини графа у перебігу роботи алгоритму.

Через те, що вершина b набула оцінки 7, довжина найкоротшого ланцюга від a до b дорівнює 7. Цей ланцюг виділений на рис. 6.23.

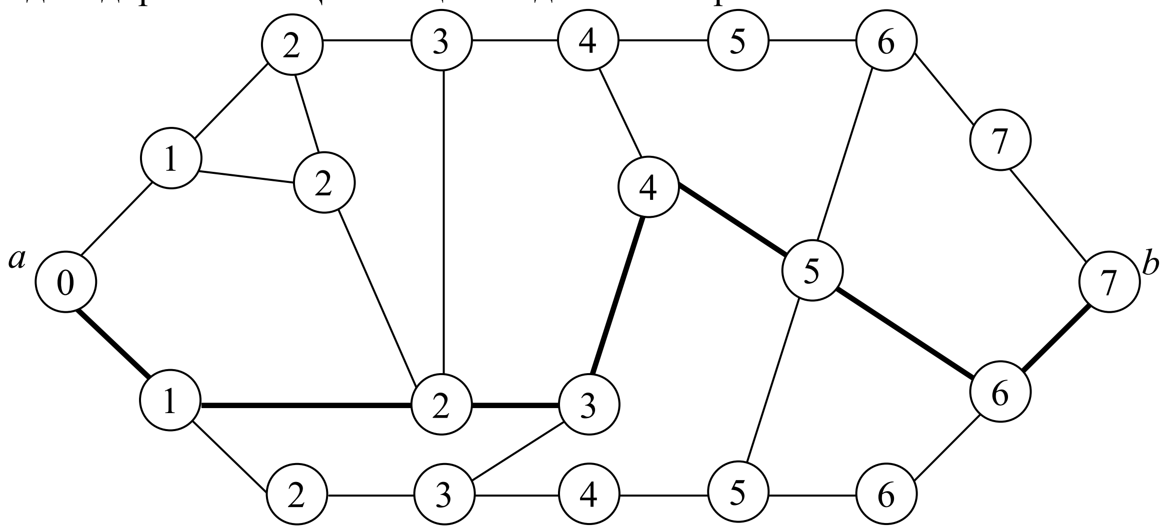


Рисунок 6.23 – Найкоротший ланцюг

Зауваження. Описаний алгоритм іноді називають *хвильовим*: процес розміщення оцінок нагадує поширення збурювання, яке виникає у вершині a і рухається зі швидкістю одне ребро за одиницю часу.

6.2.6 Ейлерові графи. Гамільтонові цикли

Якщо існує цикл у скінченному графі, в якому кожне ребро графа брало участь один раз, то такий цикл називається *ейлеровим циклом*, а граф, який містить такий цикл, – *ейлеровим графом*.

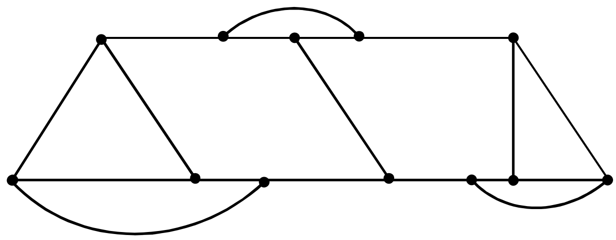
Теорема 6.5 Скінченний граф G є ейлеревим графом тоді й лише тоді, коли:

1) G – зв’язний;
2) усі його вершини мають парні степені.

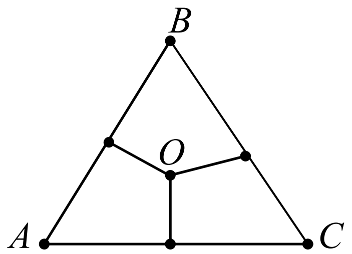
Гамільтоновим циклом називається простий цикл, що проходить через усі вершини графа. Інакше кажучи, гамільтонів цикл і гамільтонів шлях проходять через кожну вершину графа точно один раз (можливо, окрім першої й останньої вершин). Зазначимо, що гамільтонові цикл і шлях, узагалі кажучи, не містять усіх ребер графа.

Гамільтонів цикл названо іменем ірландського математика Вільямса Гамільтона, який 1859 року вперше почав вивчати ці питання.

Зауваження. Гамільтонів цикл існує далеко не в кожному графі. Більш того, через кожні дві вершини розглянутого графа може проходити простий цикл, а гамільтонів цикл при цьому може бути відсутній:



– граф з гамільтоновим циклом;



– граф, в якому немає гамільтонового циклу.

Щоб пройти через вершини A, B, C, гамільтонів цикл має містити всі ребра, які лежать на сторонах трикутника. Але тоді він не проходить через розміщену в центрі трикутника вершину O.

Іноді можна побудувати простий ланцюг, який проходить через усі вершини графа, з початком і кінцем у різних заданих вершинах x' і $x'' \in G$. Такий ланцюг теж називається *гамільтоновим*.

Розповсюджена інтерпретація задачі щодо гамільтонових циклів полягає в такому. Обід накрито на круглому столі. З-посеред гостей декотрі є друзями. За яких умов можна розсадити всіх в такий спосіб, щоби по обидва боки кожного з присутніх сиділи саме його друзі?

Так звана *задача про комівояжера* (її називають ще задачею про бродячого торговця), є задачею, яка належить до гамільтонових ланцюгів.

Район, який має відвідати комівояжер, має певну кількість міст. Відстані поміж ними є відомі. Треба відшукати найкоротший шлях, який проходив би через усі пункти і повертався до вхідного.

Ця задача має низку додатків у досліджуванні операцій, наприклад у питаннях щодо найбільш ефективного використовування рухомого складу чи обладнання.

У задачі про комівояжера міста можна подати як вершини графа *G*, в якому кожній парі вершин приписується відстань $\mu(a, b)$. Якщо дві вершини не є з'єднані, то можна покласти, що $\mu(a, b) = \infty$.

Завдання полягає в тому, щоб знайти такий гамільтонів цикл *C*, для якого сума $\mu(C) = \sum_{i=0}^{n-1} \mu(a_i, a_{i+1})$ – мінімальна.

Оскільки зазвичай йдеться лише про скінчену кількість вершин, задачу може бути розв'язано шляхом перебирання. Однак жодного ефективного алгоритму, окрім перебору, не відомо. Є тільки певні окремі схеми для окремих випадків. Один доволі важливий приклад визначення найкоротшої повітряної лінії, яка сполучає всі столиці штатів у США, обчислили до кінця Дагірі, Далкерсон та Джонсон.

Незважаючи на подібність у визначеннях для ейлерових та гамільтонових циклів, у цих поняттях є мало спільного.

Критерій існування ейлерових циклів було встановлено просто; для гамільтонових циклів жодне загальне правило невідоме. Більш того, іноді навіть для конкретних графів буває надто складно вирішити, чи можна знайти такий цикл.

6.3 Цикломатика графів. Дерева

Цикломатика – це вивчення циклів у графі.

З усієї сукупності циклів даного графа можна відокремити цілковито певну кількість незалежних (базисних) циклів, а решту здобути з базисних циклів за допомогою спеціальної операції додавання. Приклад такого додавання на рис. 6.24.

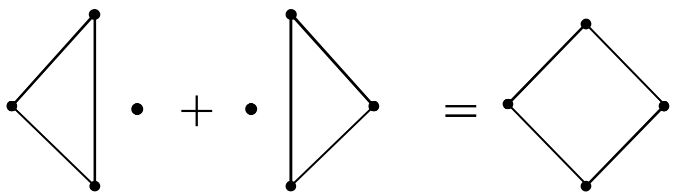


Рисунок 6.24 – Приклад додавання циклів

6.3.1 Циклові ребра та перешийки

Нехай задано граф $G = (X, U)$. Ребро графа, через яке проходить хоча б один цикл, назвемо *цикловим ребром*. Ребро, яке не входить до жодного циклу, називатимемо *перешийком*.

Приклад 6.12 У графі, зображеному на рис. 6.25, ребра u_1 та u_2 – перешийки, решта ребер – циклові.

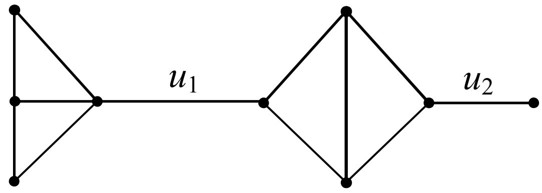


Рисунок 6.25 – Граф з перешийками та циклowими ребрами

Лема. При вилученні зі зв’язного графа циклового ребра він залишається зв’язним. При вилученні зі зв’язного графа перешийка граф розпадається на дві компоненти зв’язності.

Доведення.

Якщо ребро $\{a, b\}$, яке вилучається, є циклове, то після його вилучення вершини a та b, як і раніше, можна з’єднати в ланцюг – залишком циклу, до якого входило ребро $\{a, b\}$. Звідси випливає, що і кожні дві вершини графа після вилучення ребра $\{a, b\}$ залишаються зв’язаними ланцюгом.

Й навпаки, якщо $\{a, b\}$ – перешийок, то після його вилучення вершини a та b не можна зв’язати ланцюгом, інакше цей ланцюг разом з ребром $\{a, b\}$ утворить цикл у вихідному графі. З іншого боку, кожна вершина залишається зв’язаною чи то з вершиною a, чи то з вершиною b.

Наслідок. При вилученні з графа циклового ребра кількість зв’язних компонентів графа не змінюється. При вилученні перешийка кількість зв’язних компонентів графа збільшується на одиницю.

6.3.2 Цикломатичне число

Нехай задано граф $G=(X,U)$, $|X|=n$, $|U|=m$; p – кількість компонент зв’язності графа. Величина $\lambda = m - n + p$ називається *цикломатичним числом графа*.

Теорема 6.6 Для кожного графа цикломатичне число є невід’ємне, тобто $\lambda \geq 0$.

Доведення.

Вилучаємо з графа по одному ребру, і відстежуємо змінювання величини λ. Параметри вихідного графа позначимо m, n і p. Ті самі величини після вилучення ребра позначимо через m', n' та p'.

У процесі вилучення ребер можливі два випадки:

а) вилучуване ребро – циклове. Тоді $m' = m-1$, $n' = n$, $p' = p$;

$$\lambda' = m' - n' + p' = m - 1 - n + p = \lambda - 1.$$

б) вилучуване ребро – перешийок. У такому разі

$$m' = m-1,\ n' = n,\ p' = p+1.$$

Тоді $\lambda' = m' - n' + p' = m - 1 - n + p + 1 = \lambda$.

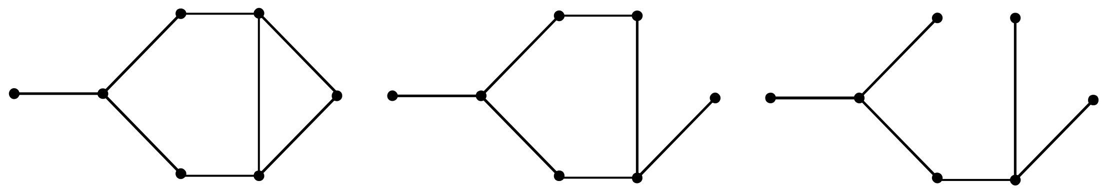
Отже, при вилученні ребра величина λ або не змінюється, або зменшується на одиницю. Після вилучення всіх ребер дістанемо порожній граф, у якому $m_0 = 0$, $n_0 = n$, $p_0 = n$, тобто $\lambda_0 = m_0 - n_0 + p_0 = n - n = 0$.

Отже, у вхідного графа $\lambda \geq 0$.

Зауваження. Доведення теореми свідчить про зв'язок цикломатичного числа з наявністю циклів у графі.

Якщо $\lambda > 0$, то у графі є принаймні один цикл. При вилученні циклового ребра деякі цикли розриваються – і це призводить до зменшення λ. Якщо продовжувати вилучення ребер, то, зрештою, розриваються всі цикли – і λ стає рівним 0. Після цього λ вже не змінюється, тому що всі ребра стали перешийками.

Приклад 6.13 Нехай треба зв'язати кілька населених пунктів мережею доріг або телефонною мережею; взагалі яким-небудь чином зв'язати один пункт з одним. Пропонований проект подано на рис. 6.26, а.



а б в

Рисунок 6.26 – Різні приклади мереж

Надалі виникла потреба здешевити проект, внаслідок чого певні зв'язки було вилучено (рис. 6.26 б, в).

У графі на рис. 6.26, в вже жодного зв'язку вилучати не можна, в противному разі граф перестане бути зв'язним і поставлене завдання не буде розв'язано.

Такого роду графи називаються *деревами*.

6.3.3 Дерева

Деревом називається зв'язний граф без циклів.

Дерево не може мати ані кратних ребер, ані петель, ані ізольованих вершин. Кожний ланцюг у дереві є простий, через те що в іншому разі дерево містило б цикл, що є неможливе.

При додаванні до дерева ребра утворюється цикл, а при вилученні хоча б одного ребра дерево розпадається на компоненти, кожна з яких є або деревом, або ізольованою вершиною.

Дерева називаються *істотно різними*, якщо вони не є ізоморфні.

На рис. 6.27 показано усі можливі дерева з шістьма вершинами – неізоморфні.

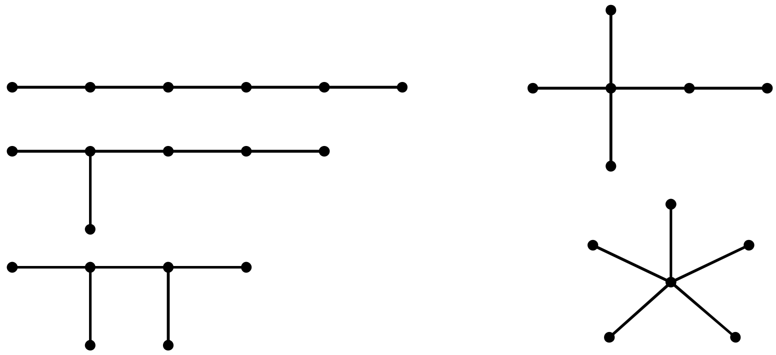


Рисунок 6.27 – Зображення неізоморфних дерев з шістьма вершинами

У поданій нижче теоремі перелічуються властивості дерев, кожна з яких повністю характеризує дерево.

Теорема 6.7 Еквівалентними є такі означення дерева:

а) *дерево* – це зв'язний граф без циклів;

б) *дерево* – це зв'язний граф, у якому кожне ребро є перешийком;

в) *дерево* – це зв'язний граф, цикломатичне число якого дорівнює нулю;

г) *дерево* – це граф, у якому для кожних двох вершин існує лише один з'єднувальний ланцюг.

Наведені твердження нескладно виводяться одне з іншого за схемою а) => б) => в) => г) => а).

У властивості в) маємо: $m - n + 1 = 0$ або $m = n - 1$, тобто у кожному дереві кількість ребер є на одиницю менша за кількість вершин.

Зауваження. Рис. 6.28 певним чином пояснює назву «дерево». Відокремимо в дереві певну вершину a (корінь). Оскільки ребра, які прилягають до неї, – перешийки, то після вилучання кожного з них від дерева відокремлюється компонента зв'язності графа. На рисунку вони позначені T_1, T_2, ..., T_k. Кожна з цих компонентів також є деревом.

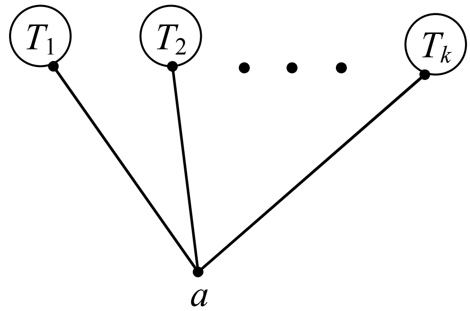


Рисунок 6.28 – Дерево

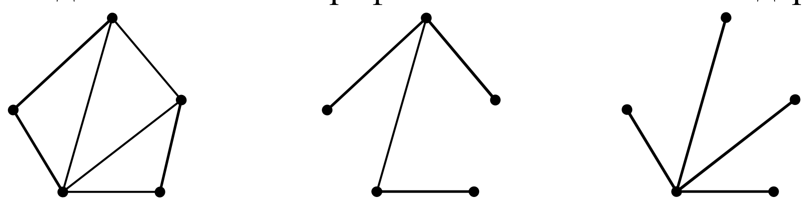
6.3.4 Кістякове дерево графа

Вилучання з довільного зв'язного графа всіх циклових ребер дає в наслідок дерево $T = \left(X', U'\right)$ таке, що:

1) $X' = X$, тобто множина вершин дерева T збігається з множиною вершин графа G;

2) $U' \subseteq U$, тобто кожне ребро дерева є водночас ребром графа G. Кожне дерево T, яке задовольняє умовам 1) та 2) називається кістяковим *деревом графа G*.

Зауваження. Через те, що вилучання циклових ребер можна провадити у різні способи, то один і той самий граф має багато кістякових дерев.



G T_1 T_2

а б в

Рисунок 6.29 – Кістякові дерева графа G

Нехай у графі G виокремлено кістякове дерево T. Ребра графа, які не ввійшли до T, називатимемо *хордами*.

Теорема 6.8 Якими б не були кістякове дерево і хорда u цього дерева, у графі G існує єдиний цикл, який має хорду u і не має інших хорд.

Доведення.

Нехай $u = \{a, b\}$. У дереві T є єдиний ланцюг, який з'єднує вершини a та b. Долучаючи до цього ланцюга ребро u, здобуваємо потрібний цикл.

Довільний незв'язний граф, який не містить циклів, називається *лісом*.

Еквівалентне визначення лісу: граф G, усі компоненти зв'язності якого є деревами, називається *лісом*.

Задача про найкоротшу з'єднувальну мережу.

Нехай $G = (X, U, l)$ – навантажений граф, у якого задано вагу l_{ij} кожного ребра $\{i, j\}$. Треба побудувати кістякове дерево T графа G, сума ваги ребер якого є мінімальна.

Цій задачі можна дати таку інтерпретацію: n пунктів на місцевості треба зв'язати мережею доріг чи то ліній телефонного зв'язку. Для кожної пари пунктів i та j задано вартість їхнього з'єднання – це і є «вага» l_{ij} ребра $\{i, j\}$ у даному разі.

Треба побудувати з'єднувальну мережу мінімальної вартості. Така мережа буде деревом, і при цьому серед усіх кістякових дерев вона матиме мінімальну суму довжин ребер, які входять до неї.

Алгоритм розв'язання поставленої задачі *є надто простий. Дерево будується покроково; на кожному кроці долучається одне ребро.*

Правило для вибору цього ребра є таке: серед ще не обраних ребер береться найкоротше, яке не утворює циклу з вже обраними ребрами.

Приклад 6.14 У матриці C елемент, що стоїть в i-му рядку та j-му стовпчику, зазначає в умовних одиницях витрати, потрібні для того, щоб зв'язати пункт i з пунктом j:

$$C = \begin{pmatrix} 0 & 5 & 6 & 2 & 1 & 7 & 5 & 4 & 6 \\ 5 & 0 & 7 & 7 & 5 & 1 & 4 & 6 & 5 \\ 6 & 7 & 0 & 5 & 6 & 6 & 2 & 5 & 4 \\ 2 & 7 & 5 & 0 & 1 & 7 & 6 & 4 & 5 \\ 1 & 5 & 6 & 1 & 0 & 5 & 6 & 3 & 7 \\ 7 & 1 & 6 & 7 & 5 & 0 & 4 & 5 & 5 \\ 5 & 4 & 2 & 6 & 6 & 4 & 0 & 6 & 2 \\ 4 & 6 & 5 & 4 & 3 & 5 & 6 & 0 & 7 \\ 6 & 5 & 4 & 5 & 7 & 5 & 2 & 7 & 0 \end{pmatrix}.$$

Задача полягає в тому, щоб з найменшими витратами з'єднати всі пункти один з одним.

Застосування сформульованого вище алгоритму реалізовується так.

У матриці C відшукується мінімальний ненульовий елемент. Він вилучається з матриці, а відповідне йому ребро долучається до мережі, якщо при цьому не утвориться циклів.

Ці дії повторюються.

Отже, на перших п'яти кроках роботи алгоритму буде обрано ребра $\{1, 5\}$, $\{2, 6\}$, $\{4, 5\}$, $\{3, 7\}$ та $\{7, 9\}$. Вони становитимуть частину мережі, подану на рис. 6.30, а. З ребер, що залишилися, мінімальну довжину має ребро $\{1, 4\}$, але до мережі воно не долучається, тому що утворює цикл з вже обраними ребрами.

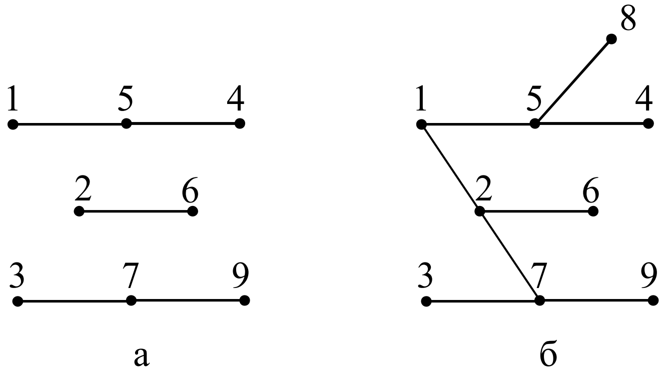


Рисунок 6.30

На наступних етапах алгоритму до дерева буде долучено ребра $\{5, 8\}$, $\{2, 7\}$ та $\{1, 2\}$. Побудовано дерево подано на рис. 6.30, б.