

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ СЕМЕНА КУЗНЕЦЯ

ДИСКРЕТНА МАТЕМАТИКА

Методичні рекомендації
до самостійної роботи з теми "Теорія графів"
для студентів галузі знань 12 "Інформаційні технології"
першого (бакалаврського) рівня

Харків
ХНЕУ ім. С. Кузнеця
2020

УДК 519.1(07.034)

Д48

Укладачі: Т. В. Денисова
В. Ф. Сенчуков

Затверджено на засіданні кафедри вищої математики й економіко-математичних методів.

Протокол № 6 від 04.12.2019 р.

Самостійне електронне текстове мережеве видання

Дискретна математика [Електронний ресурс] : методичні рекомендації до самостійної роботи з теми "Теорія графів" для студентів галузі знань 12 "Інформаційні технології" першого (бакалаврського) рівня / уклад. Т. В. Денисова, В. Ф. Сенчуков. – Харків : ХНЕУ ім. С. Кузнеця, 2020. – 100 с.

Подано теоретичний матеріал із теорії графів, запитання для самоконтролю його засвоєння, варіанти задач самостійної контрольної роботи та зразки їхнього розв'язання.

Рекомендовано для студентів галузі знань 12 "Інформаційні технології" першого (бакалаврського) рівня всіх форм навчання.

УДК 519.1(07.034)

© Харківський національний економічний університет імені Семена Кузнеця, 2020

*Для сприйняття чужої мудрості
потрібна, насамперед, самотійна робота
(Л. М. Толстой).*

Вступ

Тема "Теорія графів" є важливою складовою частиною базової (нормативної) навчальної дисципліни "Дискретна математика", яку широко застосовують у математичній кібернетиці, комп'ютерній математиці, програмуванні, а також під час створення засобів передавання й оброблення інформації, автоматизованих систем управління та проєктування.

Спеціальності, які потребують знань із "Дискретної математики", такі: 121 "Інженерія програмного забезпечення", 122 "Комп'ютерні науки", 124 "Системний аналіз", 126 "Інформаційні системи та технології".

Згідно з робочою програмою навчальної дисципліни, від загального обсягу – 120 год – на самотійну роботу студентів відведено 50 %. Саме такими обставинами обумовлено необхідність розроблення методичних рекомендацій, що пропонують, та їхню актуальність. Головною *метою* їхнього складання є ознайомлення студентів із фундаментальними поняттями, ідеями та методами теорії графів, сприяння розвитку їхнього логічного й аналітичного мислення, надання допомоги щодо використання здобутих знань та набутих умінь під час розв'язування конкретних практичних задач фахової спрямованості, підготовка до вивчення спеціальних дисциплін і самотійного опрацювання математичної та науково-технічної літератури.

Бурхливий розвиток та поширення електронних обчислювальних машин викликав значне зростання ролі задач дискретного характеру. На сьогодні, завдяки наочному зображенню, граф став однією з найпоширеніших і найпопулярніших математичних моделей у багатьох сферах науки й техніки для вивчення різноманітних об'єктів, явищ та процесів. Спостерігають проникнення відповідних методів у різні сфери людської діяльності, одним із потужних інструментів якого є саме граф. Один із прикладів такого проникнення пов'язаний із розділом "Алгоритми на графах" у прикладній теорії алгоритмів. Ця обставина визначає вагому роль графів у підготовці фахівців і місце "Дискретної математики" серед інших навчальних дисциплін.

У вигляді графа може бути зображено: електричні та транспортні мережі; інформаційні й комп'ютерні мережі; карти газонафтопроводів, автомобільних, залізничних і повітряних шляхів; моделі кристалів; структури молекул хімічних речовин; моделі ігор; різні математичні об'єкти (відношення, частково впорядковані множини, решітки, автомати, ланцюги Маркова, алгоритми та програми тощо); лабіринти; плани діяльності або плани виконання певних робіт (розклади); генеалогічні дерева тощо.

Теорію графів тісно пов'язано з такими розділами математики, як теорія множин, теорія матриць, математична логіка та теорія ймовірностей. В усіх цих розділах графи застосовують для подання (у відповідному вигляді) різноманітних математичних об'єктів. Водночас сама теорія графів широко використовує апарат споріднених розділів математики.

Попередні знання, що є основою для вивчення теорії графів, студент здобуває, вивчаючи навчальну дисципліну "Вища математика", а також теми "Дискретної математики", які їй передують, а саме: теорію множин і відношень, комбінаторний аналіз.

Пропоновані методичні рекомендації містять чотири розділи: перший присвячено неорієнтованим графам, другий – орієнтованим, у третьому розділі подано варіанти самостійної контрольної роботи, кожний із яких містить набір задач, що охоплює основні теоретичні положення теорії графів та їхнє застосування, а в четвертому розділі наведено зразок виконання типового варіанта контрольної роботи з коментарем усіх кроків розв'язання.

Формули, рисунки, теореми тощо нумерують парами, перший елемент пари є номером розділу, а другий (після крапки) указує на порядковий номер (у межах розділу).

Рекомендована література має допомогти студенту поглибити та поширити власну поінформованість із питань, що його зацікавили.

1. Неорієнтовані графи

1.1. Означення основних понять, способи задання

Неорієнтованим графом (НГ), або просто **графом** (від грец. *grapho* – пишу), називають упорядковану пару $G = (X, U)$, де $X = \{x_i\}_1^n$ – непорожня скінченна множина деяких об'єктів (елементів), $U = \{u_j\}_1^m$ – множина невпорядкованих пар, складених з елементів множини X : $u_j = \{x_i, x_k\}$, $i, k \in I_x = \{1, 2, \dots, n\}$, $j \in I_u = \{1, 2, \dots, m\}$; I_x, I_u – множини індексів.

Елементи множини X (U) називають **вершинами (ребрами)** графа. Кількість вершин графа називають його **порядком**.

Граф $G = (X, U)$, який містить n вершин і m ребер, називають **(n, m) -графом**.

Вершини x_i, x_k , які визначають ребро $u_j = \{x_i, x_k\}$, називають **граничними вершинами**, або **кінцями**, цього ребра. Якщо вершина x_i є кінцем ребра u_j , то кажуть, що вони **інцидентні**: вершина x_i інцидентна ребру u_j , а ребро u_j інцидентне вершині x_i .

Два ребра називають **суміжними (сусідніми)**, якщо вони інцидентні одній і тій самій вершині. Дві вершини називають **суміжними (сусідніми)**, якщо вони інцидентні одному й тому самому ребру.

Вершину, не інцидентну жодному ребру, називають **ізолюваною**, а вершину, інцидентну тільки одному ребру, – **кінцевою**. Ребро, кінцями якого є одна й та сама вершина, називають **петлею** (множина U в цьому разі містить пару вигляду $\{x_i, x_i\}$).

Ребра з однаковими граничними вершинами, називають **кратними**, або **паралельними**.

Залежно від того, у якому вигляді подають пару (X, U) , є різні *способи задання* графів.

Теоретико-множинний спосіб – це задання графа переліком елементів множини вершин $X = \{x_i\}_1^n$ і множини ребер $U = \{u_j\}_1^m$. Такий

метод, на жаль, не є наочним із погляду встановлення тих чи тих його властивостей.

Геометричне задання графів: кожній вершині графа ставлять у відповідність точку з \mathbf{R}^2 чи \mathbf{R}^3 , а кожному ребру – прямолінійний або криволінійний відрізок, що з'єднує інцидентні ребру вершини. Водночас намагаються зобразити граф так, щоб його ребра перетиналися тільки у їхніх граничних вершинах.

Матричне задання графів – це задання графа матрицею інцидентностей і/або матрицею суміжності вершин. **Матрицею інцидентностей**, або **інциденцій**, (n, m) -графу G називають матрицю $A = [a_{ij}]_{n \times m}$ розміру $n \times m$, елементи якої a_{ij} ($i = \overline{1, n}, j = \overline{1, m}$) визначають таким чином:

$$a_{ij} = \begin{cases} 0, \text{ якщо } x_i \notin u_j, \\ 1, \text{ якщо } x_i \in u_j, \\ 2, \text{ якщо } u_j \text{ – петля при вершині } x_i. \end{cases} \quad (1.1)$$

Кожний стовпець такої матриці містить два одиничні елементи (якщо ребро не є петлею), а нульовий рядок відповідає ізольованій вершині.

Матрицею суміжності вершин (n, m) -графу G називають квадратну матрицю $B = [b_{ij}]_{n \times n}$ n -го порядку, елементи якої b_{ij} дорівнюють кількості ребер із граничними вершинами x_i, x_j ($i = \overline{1, n}, j = \overline{1, n}$); за домовленістю петлю враховують двічі.

Матриця B неорієнтованого графу завжди є симетричною: $b_{ij} = b_{ji}$ ($\forall i = \overline{1, n}, \forall j = \overline{1, n}$). А рядки і стовпці, що відповідають ізольованим вершинам, містять лише нулі.

Зауважимо, що:

1) і матриця інцидентностей, і матриця суміжності вершин однозначно описує неорієнтований граф;

2) перевагою матричного задання графу є зручність опрацювання матриць на комп'ютері. Однак за великої кількості вершин і ребер обидві матриці стають занадто громіздкими, тому часто для задання графів складають так звані *списки суміжності*, коли кожній вершині графу

відповідає свій список, у якому послідовно наведено всі суміжні їй вершини (біля позначення певної вершини ставлять двокрапку (:), після якої через кому наводять перелік суміжних вершин);

3) вибір та зручність того чи того способу залежать від особливостей задачі, що розв'язують.

Приклад задання (5,6)-графу G різними способами:

а) теоретико-множинне задання:

$$G = (X, U) = (\{x_1, x_2, x_3, x_4, x_5\}, \{u_1 = \{x_1, x_2\}, u_2 = \{x_2, x_2\}, u_3 = \{x_2, x_3\}, u_4 = \{x_3, x_4\}, u_5 = \{x_3, x_4\}, u_6 = \{x_1, x_3\}\});$$

б) геометричне зображення цього ж графу подано на рис. 1.1:

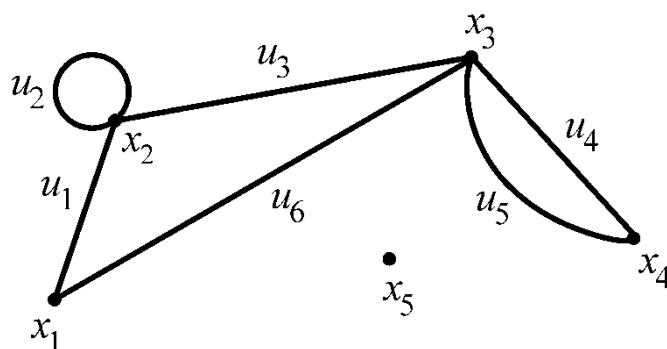


Рис. 1.1. Геометричне зображення графу

в) матриці інцидентностей A і суміжності вершин B мають такий вигляд:

$$A = \begin{matrix} & u_1 & u_2 & u_3 & u_4 & u_5 & u_6 \\ \begin{matrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{matrix} & \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 2 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{matrix}, \quad B = \begin{matrix} & x_1 & x_2 & x_3 & x_4 & x_5 \\ \begin{matrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{matrix} & \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 2 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{matrix}.$$

1.2. Підграф. Маршрут, ланцюг, цикл

Якщо у графі $G=(X,U)$ виділити підмножину вершин $X' \subset X$ разом із підмножиною ребер $U' \subset U$, що інцидентні вершинам із X' , то дістанемо граф $G'=(X',U')$, який називають **підграфом** графа G .

Наприклад, у розглянутому на рис. 1.1 графі можна виділити підграф $G' = (\{x_3, x_4, x_5\}, \{u_4, u_5\}) \subset G$.

Нехай $G=(X,U)$ – неорієнтований граф; x_i, x_j – його довільні вершини. **Маршрутом** у графі G , що з'єднує вершини x_i, x_j , називають таку скінченну послідовність суміжних ребер (не обов'язково різних), що кожна гранична вершина (можливо, крім x_i, x_j) будь-якого ребра інцидентна, принаймні, двом ребрам.

Наприклад, для графа на рис. 1.1 можна вказати такий маршрут, що з'єднує вершини x_1 і x_2 :

$$\mu_{12} = (u_6, u_4, u_5, u_6, u_1) = (\{x_1, x_3\}, \{x_3, x_4\}, \{x_4, x_3\}, \{x_1, x_3\}, \{x_1, x_2\}).$$

Маршрут також можна записати як послідовність вершин, через які він проходить, відділяючи позначення вершин рисками:

$$\mu_{12} = (x_1 - x_3 - x_4 - x_3 - x_1 - x_2).$$

Довжиною маршруту називають кількість ребер у ньому; однакові ребра лічать стільки разів, скільки разів вони входять до маршруту. Довжину маршруту μ позначають через $l(\mu)$. У нашому прикладі $l(\mu_{12}) = 5$.

Маршрут, що з'єднує вершини x_i, x_j , називають **замкненим (незамкненим)**, якщо вершини x_i, x_j збігаються (не збігаються).

Незамкнений (замкнений) маршрут, усі ребра у якому різні, називають **ланцюгом (циклом)**. **Ребро (петля)** – це ланцюг (цикл) одиничної довжини. Ланцюг (цикл), у якому всі вершини різні (крім першої та останньої), називають **простим**, або **елементарним**.

Так для графа на рис. 1.1, *наприклад*, маємо:

$\mu_{14} = (u_1, u_3, u_4) = (x_1 - x_2 - x_3 - x_4)$ – простий ланцюг;

$\mu_{11} = (u_1, u_3, u_6) = (x_1 - x_2 - x_3 - x_1)$ – простий цикл;

$\mu_{13} = (u_1, u_2, u_3) = (x_1 - x_2 - x_2 - x_3)$ – ланцюг (непростий).

Граф без циклів називають **ациклічним**, інакше граф називають **циклічним**.

За допомогою понять ланцюга й циклу здійснюють різноманітні класифікації графів, формулюють і доводять їхні властивості.

Щодо маршрутів мають місце *твердження*:

- 1) будь-який незамкнений маршрут містить простий ланцюг із тими самими кінцями;
- 2) будь-який замкнений маршрут містить простий цикл;
- 3) якщо ланцюг не є простим, то він містить простий цикл.

1.3. Відшукування ланцюга найменшої довжини

У практичних застосуваннях велике значення має задача знаходження найкоротшого ланцюга між двома вершинами зв'язного графа. До такої задачі зводять численні задачі вибору найбільш економічного (із погляду відстані, часу або вартості) маршруту між двома об'єктами.

Задачу знаходження на графі ланцюга найменшої довжини в загальному вигляді можна сформулювати так: задано неорієнтований граф $G = (X, U)$, кожному ребру $\{x_i, x_j\}$ якого приписано деяке дійсне число $d(x_i, x_j) = d(i, j) = d_{ij} \geq 0$, що називають **довжиною** ребра (залежно від конкретного застосування це число може бути мірою фізичної відстані, ваги, вартості або іншого важливого параметра). Треба для двох довільних вершин x_α, x_β знайти у графі G ланцюг $L^* = L^*(\alpha, \beta)$ такий, щоб сумарна довжина його ребер d_{L^*} була мінімальною:

$$d_{L^*} = \sum_{(x_i, x_j) \in L^*} d_{ij} \rightarrow \min . \quad (1.2)$$

Надалі (для визначеності) одну із граничних вершин x_α , x_β ланцюга $L(\alpha, \beta)$ будемо називати **початковою**, другу – **кінцевою**. Отже, нехай $x_\alpha \in G$ – початкова вершина. У разі геометричного зображення, переважно, вершину x_α розташовують лівіше від вершини x_β .

Розглянемо розв'язання сформульованої задачі **методом індексації (зважування) вершин**, який полягає в такому:

1. *Початкова індексація вершин.* Кожній вершині x_i графа приписують певне число p_i , яке називають **індексом (вагою)** вершини. Спочатку кінцевій вершині x_β приписують вагу $p_\beta = 0$, а решті вершин – вагу $p_i = \infty$ ($i \neq \beta$).

2. *Поновлення індексів (переважування) вершин.* На кожному наступному (після початкового зважування) кроці шукають ребро $\{x_i, x_j\}$, для якого $p_j - p_i > d_{ij}$, і замінюють індекс p_j новим індексом: $p_j^H = p_i + d_{ij}$ ($p_j^H < p_j$). Продовжують процес заміни індексів доти, поки не залишиться хоча б одна вершина, індекс якої можна зменшити.

3. *Знаходження власне найкоротшого ланцюга між x_α і x_β .* Відштовхуючись від початкової вершини x_α з індексом p_α , знаходять вершину x_{i_1} , для якої $p_\alpha - p_{i_1} = d(\alpha, i_1)$ (така вершина завжди є, якщо процес поновлення індексів завершено). Далі знаходять вершину x_{i_2} , для якої $p_{i_1} - p_{i_2} = d(i_1, i_2)$, і так далі, поки не дійдуть до кінцевої вершини $x_{i_{k+1}} = x_\beta$.

Ланцюг $L(\alpha, \beta) = (x_\alpha - x_{i_1} - x_{i_2} - \dots - x_{i_k} - x_\beta)$, довжина якого дорівнює вазі початкової вершини $d_L = p_\alpha$, є найкоротшим, тобто $L(\alpha, \beta) = L^*$.

Зауважимо, що проведена індексація вершин графа дає змогу знайти найкоротший ланцюг між двома будь-якими вершинами графа, а не тільки $L(\alpha, \beta)$.

Приклад. Розглянемо (7,12)-граф (рис. 1.2), беручи $x_\alpha = x_1$, $x_\beta = x_7$. Після проведення процесу зважування вершин дістають індекси, указані у кружечках.

Найкоротший ланцюг $L_1^*(1,7) = (x_1 - x_2 - x_4 - x_6 - x_7)$ на рис. 1.2 позначено пунктирною ламаною. Зазначимо, що оптимальним також є ланцюг $L_2^*(1,7) = (x_1 - x_2 - x_5 - x_6 - x_7)$. Довжини обох ланцюгів однакові: $d_{L_1^*} = d_{L_2^*} = p_1 = 8$.

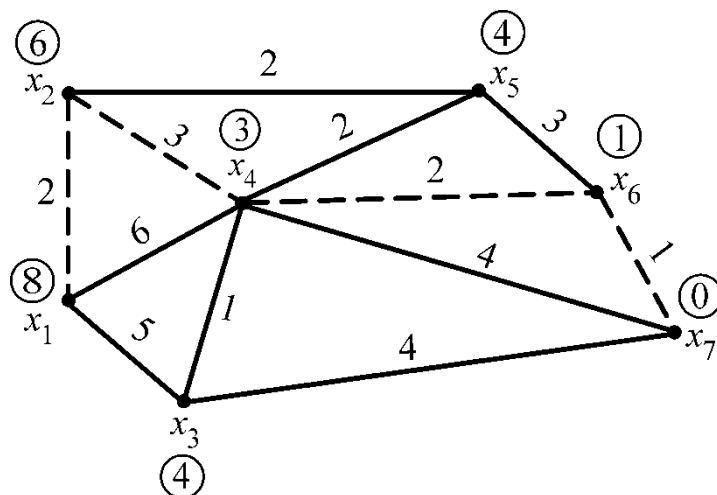


Рис. 1.2. Граф із проіндексованими вершинами та найкоротшим ланцюгом на ньому

1.4. Матриця відстаней графа

Нехай задано зв'язний граф $G = (X, U)$. Відстанню між двома вершинами x_i та x_j графа G називають довжину найкоротшого ланцюга, що з'єднує ці вершини та позначають $v(x_i, x_j)$ або v_{ij} ($x_i, x_j \in X$, $i, j \in I_x = \{1, 2, \dots, n\}$).

Відстань $v(x_i, x_j)$ має такі властивості:

- 1) $(v(x_i, x_j) > 0, i \neq j) \wedge (v(x_i, x_j) = 0, i = j)$;
- 2) $v(x_i, x_j) = v(x_j, x_i)$;
- 3) $\forall x_i, x_j, x_k \in X: v(x_i, x_k) = v(x_i, x_j) + v(x_j, x_k)$.

Матрицею відстаней (n, m) -графу G називають квадратну матрицю $V = [v_{ij}]_{n \times n}$ n -го порядку, кожний елемент якої v_{ij} дорівнює відстані між вершинами x_i, x_j ($i = \overline{1, n}, j = \overline{1, n}$).

Ексцентриситетом вершини $x_i \in X$ у графі G називають відстань від x_i до найвіддаленішої від неї вершини графа і позначають $e(x_i)$:

$$e(x_i) = \max_{\forall x_j \in X} v(x_i, x_j).$$

Радіусом $r(G)$ графа G називають найменший з ексцентриситетів його вершин:

$$r(G) = \min_{\forall x_i \in X} e(x_i). \quad (1.3)$$

Діаметром $diam(G)$ графа G називають найбільший з ексцентриситетів його вершин, тобто найбільшу з відстаней між будь-якими двома вершинами графа:

$$diam(G) = \max_{\forall x_i \in X} e(x_i) = \max_{x_i} \max_{x_j} v(x_i, x_j). \quad (1.4)$$

Вершину $x_i \in X$ ($i = \overline{1, n}$), для якої $e(x_i) = r(G)$ ($e(x_i) = diam(G)$), називають **центральною (периферійною) вершиною** графа G .

Центром графа G називають множину $C(G)$ усіх його центральних вершин:

$$C(G) = \{x_i \in X \mid e(x_i) = r(G), i = \overline{1, n}\}.$$

Центр графа може містити як одну, так і дві або більше вершин.

1.5. Зв'язність графів, дерева та ліс

Нехай $G = (X, U)$ – неорієнтований граф; x_i, x_j – його довільні вершини ($x_i, x_j \in X$). Вершину x_i називають **зв'язаною** з вершиною x_j , якщо є ланцюг, який з'єднує (сполучає) ці вершини. У цьому разі також кажуть, що вершина x_j **досяжна** з вершини x_i .

Граф називають **зв'язним**, якщо будь-які дві його вершини є зв'язаними. Інакше кажуть, що граф **незв'язний**.

Незв'язний граф можна розбити на частини (підграфи), які є зв'язними графами та називають **зв'язними компонентами**, або **компонентами зв'язності** графа G (ізолювана вершина – компонента). Так граф на рис. 1.1 має дві компоненти зв'язності, одна з яких – вершина x_5 .

Зв'язний граф може бути поділено на незв'язані між собою підграфи вилученням із нього певних вершин і/або ребер. У разі *вилучення вершин* вилучають і всі інцидентні до них ребра, а в разі *вилучення ребер* інцидентні до них вершини зберігають.

Вершину, вилучення якої перетворює зв'язний граф на незв'язний, називають **точкою зчленування** (рис. 1.3). Ребро, вилучення якого перетворює зв'язний граф на незв'язний, називають **мостом** (рис. 1.4). За наявності моста граф має принаймні дві точки зчленування.

Граф називають **нероздільним**, якщо він є зв'язним і не має точок зчленування. Граф, який має принаймні одну точку зчленування, є роздільним і його називають **сепарабельним**.

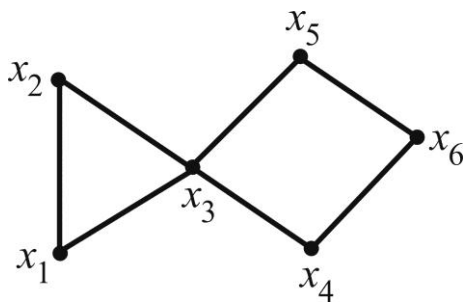


Рис. 1.3. Точка зчленування – x_3

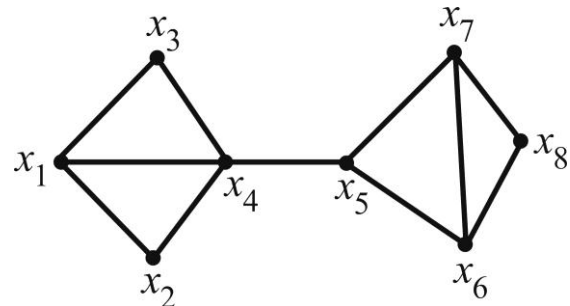


Рис. 1.4. Міст – ребро $\{x_4, x_5\}$

Зв'язний граф, який не містить циклів (зокрема, петель і кратних ребер) називають **деревом**.

Якщо задано множину вершин $X = \{x_1, x_2, \dots, x_n\} \subset \mathbf{R}^2$, то дерево можна побудувати так: одну з вершин (наприклад, x_1) беруть за початкову і називають **коренем дерева**, або **вершиною нульового рівня**. Із цієї вершини проводять ребра в найближчі вершини (нехай це будуть x_2, x_3) – **вершини першого рівня**. На кожному наступному кроці (для знаходження вершин чергового рівня) додають ребра, одна із граничних вершин яких є вершиною попереднього рівня, а друга – ближчою до неї з тих, що залишилися ще ізолюваними.

Наприклад, для $n = 8$ одне з дерев зображено на рис. 1.5а. Якщо вибрати корінь дерева і вершини одного й того самого рівня розташувати на одній горизонталі, то будемо мати інше зображення того самого дерева (рис. 1.5б), яке здебільшого буває зручнішим. Зокрема, воно показує, що в дереві завжди є кінцеві вершини (наприклад, вершини останнього рівня). Дерево з визначеними рівнями вершин (див. рис. 1.5б) називають **упорядкованим за рівнями вершин**.

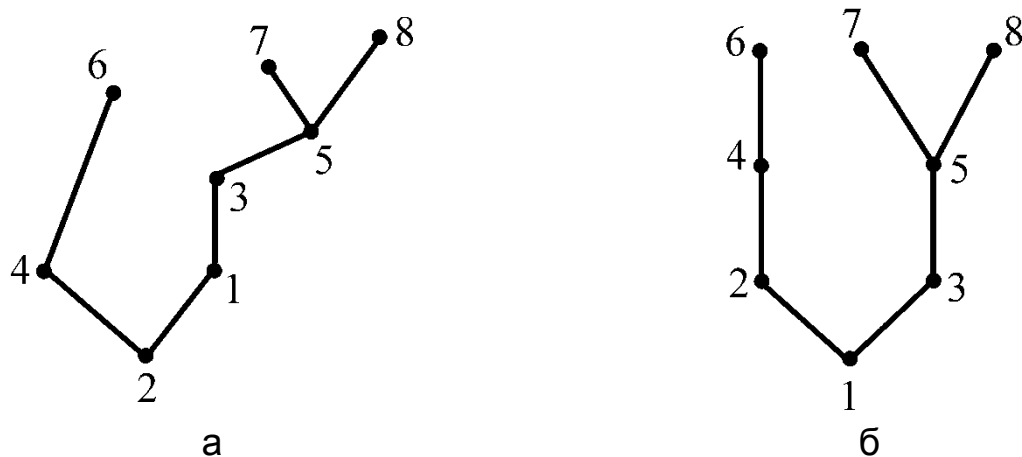


Рис. 1.5. **Дерево:**
а) із невизначеними рівнями вершин;
б) упорядковане за рівнями вершин

Щодо дерев мають місце *твердження*: 1) кожне дерево, яке містить (покриває) n вершин, має $(n - 1)$ -не ребро; 2) кількість усіх дерев, які покривають n заданих вершин, дорівнює n^{n-2} .

У разі додавання в дерево ребра утворюється цикл, а в разі видалення принаймні одного ребра дерево розпадається на компоненти, кожна з яких є деревом або ізольованою вершиною. Кожний ланцюг у дереві є простим, через те, що інакше дерево містило б цикл, що неможливо.

Часто дерева розглядають не самі по собі, а у взаємозв'язку із графами складнішої структури. Частиною графа G , яка містить усі його вершини та є деревом, називають **кістяком**, або **покривним деревом**, графа G .

Число $\gamma = m - n + p$, де p – кількість компонент зв'язності, називають **цикломатичним числом** графа. Зрозуміло, що для зв'язного графа $\gamma = m - n + 1$. Доведено, що для будь-якого графа його цикломатичне

число є невід'ємним: $\gamma \geq 0$; так, якщо $\gamma > 0$, то це свідчить про наявність у графі принаймні одного циклу. Щоб дістати кістяк графа G з n вершинами і m ребрами (який містить цикли), треба вилучити з нього γ ребер. Ребра графа, які не увійшли до кістяка, називають **хордами**.

Наприклад, покривне дерево у зв'язному підграфі G' графа G ($G' \subset G$) (див. рис. 1.1) із чотирма вершинами (x_1, x_2, x_3, x_4) і шістьма ребрами ($u_j, j = \overline{1, 6}$) здобудемо вилученням із нього трьох ребер u_2, u_5, u_6 , оскільки $\gamma = m - n + 1 = 6 - 4 + 1 = 3$.

Незв'язний граф, який не містить циклів (зокрема, петель і кратних ребер) називають **лісом**. Інакше кажучи, ліс – це граф, усі компоненти зв'язності якого є деревами.

1.6. Побудова економічного дерева

У багатьох задачах практичного характеру потрібно зв'язати деякі об'єкти (міста, села, будинки, підприємства й т. ін.) найбільш економічним чином із погляду відстані, часу, вартості тощо. Це стосується, зокрема, побудови ліній зв'язку, автомобільних шляхів, електромереж.

Мовою теорії графів цю **задачу** формулюють у загальному вигляді так: нехай $G = (X, U)$ – зв'язний граф, кожному ребру $\{x_i, x_j\}$ якого поставлено у відповідність дійсне число $d_{ij} \geq 0$, що називають **довжиною ребра**. Треба побудувати покривне дерево T графа G так, щоб сумарна довжина його ребер d_T була мінімальною:

$$d_T = \sum_{(x_i, x_j) \in T} d_{ij} \rightarrow \min. \quad (1.5)$$

Кожне дерево, яке задовольняє умову (1.5), називають **економічним**.

Є дуже простий *спосіб побудови економічного дерева*, який ґрунтується на послідовному введенні в нього ребер із пріоритетом за мінімумом їхніх довжин:

1) вибираємо ребро найменшої довжини (якщо таких ребер декілька, то беремо будь-яке з них);

2) на кожному наступному кроці додаємо (приєднуємо) найкоротше з ребер, у разі долучення якого до вже вибраних ребер не утворюється жодного циклу;

3) процес закінчуємо, щойно буде вибрано $(n - 1)$ ребер.

Приклад. Розглянемо п'ятивершинний повний граф (рис. 1.6а). Кількість його ребер дорівнює $C_5^2 = 10$, а покривних дерев на ньому можна побудувати $5^3 = 125$. Певна річ, переглядати й порівнювати всі кістяки недоцільно.

Шукане економічне дерево здобудемо за кроками 1 – 3 наведеного алгоритму; на рис. 1.6а вибрані ребра позначено поперечними рисками. На рис. 1.6б зображено знайдене покривне дерево за умови, що його коренем є вершина x_3 .

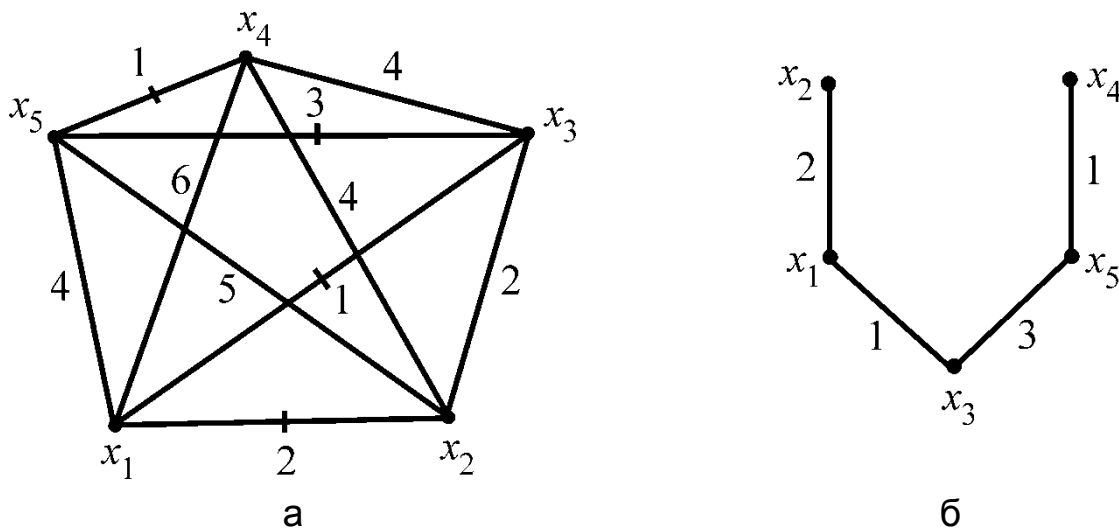


Рис. 1.6. Побудова економічного дерева:

а) заданий граф та економічне дерево на ньому;

б) економічне дерево, упорядковане за рівнями вершин

Можна вказати ще один кістяк мінімальної довжини, якщо у знайденому дереві замінити ребро $\{x_1, x_2\}$ на ребро $\{x_2, x_3\}$.

Для реалізації процедури формування дерев (зокрема економічних) на ЕОМ розроблено спеціальні програми. Своєю чергою, графи допомагають спілкуванню з ЕОМ і їх називають інструментом програміста.

1.7. Операції над графами

Об'єднанням графів $G_1=(X_1,U_1)$ і $G_2=(X_2,U_2)$ називають граф G із множиною вершин $X=X_1\cup X_2$ і множиною ребер $U=U_1\cup U_2$ (рис. 1.7) і позначають $G_1\cup G_2$, отже:

$$G = G_1 \cup G_2 = (X_1, U_1) \cup (X_2, U_2) = (X, U) = (X_1 \cup X_2, U_1 \cup U_2).$$

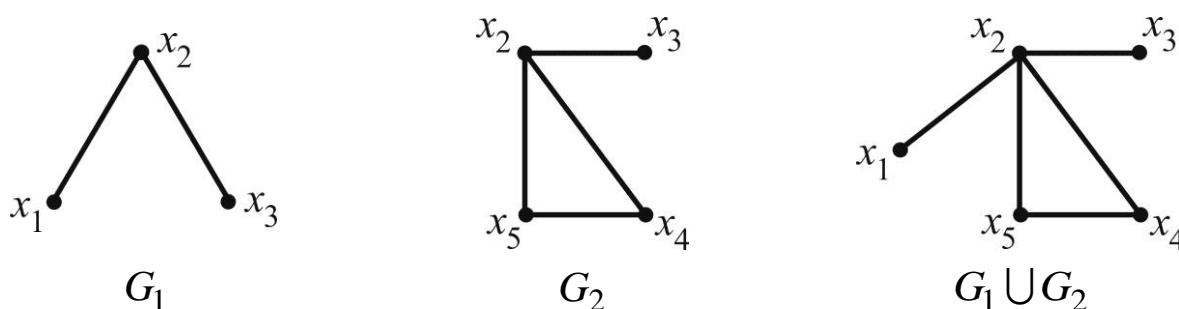


Рис. 1.7. Графи G_1 , G_2 та об'єднання $G_1 \cup G_2$

Об'єднання $G_1 \cup G_2$ називають **прямою сумою** графів G_1 і G_2 , якщо множини вершин цих графів не перетинаються: $X_1 \cap X_2 = \emptyset$.

Перетином графів $G_1=(X,U_1)$ і $G_2=(X,U_2)$ з однаковими множинами вершин називають граф G із множиною вершин X і множиною ребер $U=U_1 \cap U_2$ і позначають $G_1 \cap G_2$ (рис. 1.8):

$$G = G_1 \cap G_2 = (X, U_1) \cap (X, U_2) = (X, U) = (X, U_1 \cap U_2).$$

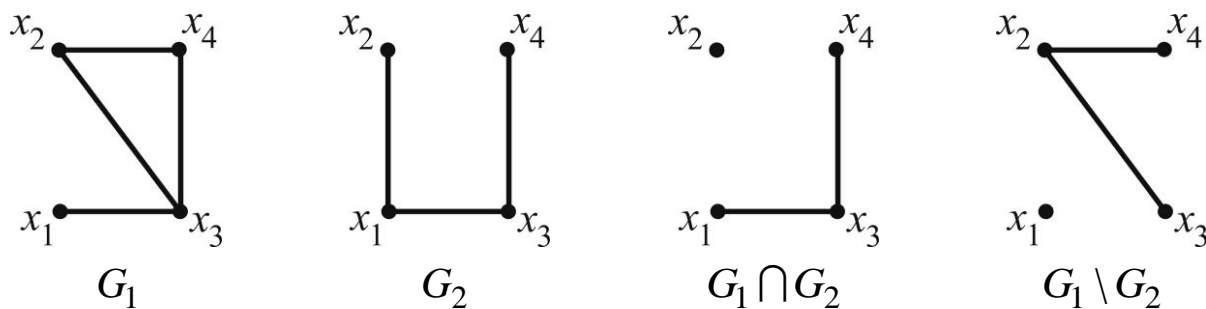


Рис. 1.8. Графи G_1 , G_2 , перетин $G_1 \cap G_2$ і різниця $G_1 \setminus G_2$

Різницею графів $G_1 = (X, U_1)$ і $G_2 = (X, U_2)$ з однаковими множинами вершин називають граф G із множиною вершин X і множиною ребер $U = U_1 \setminus U_2$ і позначають $G_1 \setminus G_2$ (див. рис. 1.8):

$$G = G_1 \setminus G_2 = (X, U_1) \setminus (X, U_2) = (X, U) = (X, U_1 \setminus U_2).$$

Доповненням графа $G = (X, U)$ називають граф G^* , який має ту саму множину вершин X , що й граф G , а будь-які дві вершини у G^* суміжні тоді й тільки тоді, коли вони несуміжні у G (рис. 1.9), тобто:

$$G^* = (X, X^{(2)} \setminus U),$$

де $X^{(2)} = \{\{x_i, x_j\} | x_i \in X, x_j \in X, i \neq j\}$ – множина всіх двоелементних підмножин множини X .



Рис. 1.9. Граф G та його доповнення G^*

1.8. Класифікація графів (за структурою). Ізоморфні та гомеоморфні графи

Залежно від наявності чи відсутності кратних ребер і петель, розрізняють:

мультиграф – граф із кратними ребрами, але без петель (рис. 1.10а);

псевдограф – граф із кратними ребрами й петлями (рис. 1.10б);

простий, або звичайний, граф – граф без петель і кратних ребер (рис. 1.10в).

Надалі будемо розглядати, переважно, прості графи (які мають широке застосування на практиці). Частинними випадками серед них є:

порожній, або нуль-граф – граф, усі вершини якого ізольовані (множина ребер такого графа є порожньою: $U = \emptyset$) (рис. 1.10г);

тривіальний граф – граф, що має лише одну вершину ($n=1$) і не має жодного ребра, (рис. 1.10д);

повний граф – граф, у якому кожна вершина суміжна з усіма іншими (рис. 1.10е);

біхроматичний, або двочастковий, граф – граф, множину вершин якого можна розбити на дві підмножини X_1, X_2 , які не перетинаються ($X_1 \cap X_2 = \emptyset$), так, що кожне ребро з'єднує деяку вершину з X_1 із деякою вершиною з X_2 (рис. 1.10є).

Установлено, що граф є біхроматичним тоді й тільки тоді, коли він не має циклів, складених із непарної кількості ребер, тобто циклів непарної довжини.

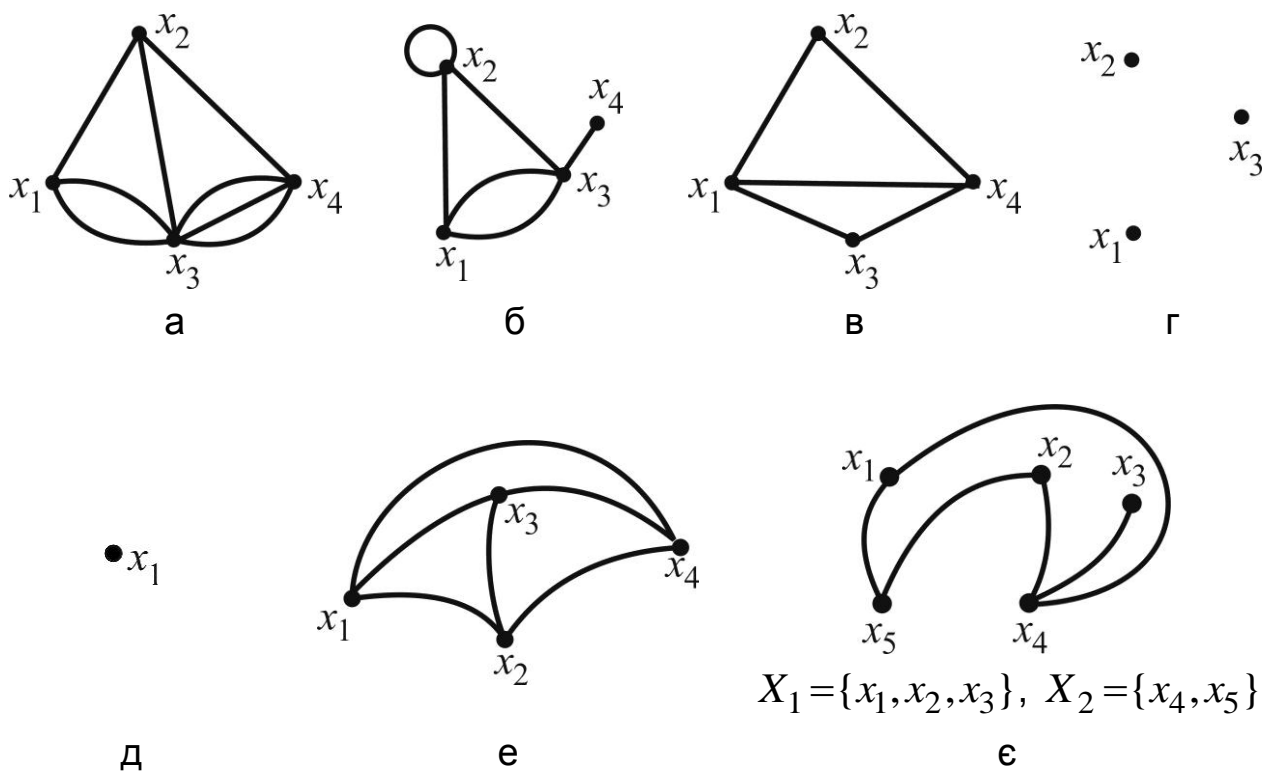


Рис. 1.10. Різновиди графів:

- а) мультиграф; б) псевдограф; в) простий; г) порожній;
 д) тривіальний; е) повний; є) біхроматичний

Кількість ребер, інцидентні певній вершині графа x_i ($i = \overline{1, n}$), називають **степенем** цієї вершини й позначають через s_i . Якщо s_i – парне (непарне) число, то вершину x_i називають **парною (непарною)**. Степень

ізольованої вершини дорівнює нулю, а висячої – одиниці. Так для графа на рис. 1.10є маємо: $s_1 = s_2 = s_5 = 2$, $s_3 = 1$, $s_4 = 3$.

Граф, усі вершини якого мають степінь s , називають **регулярним (однорідним) графом степеня S** , або **графом S -го степеня**. Так граф на рис. 1.10е є однорідним графом третього степеня, або степеня 3.

Наведемо деякі властивості графів, пов'язані з поняттям "степінь", доведення яких пропонуємо знайти самостійно:

1. Сума степенів вершин будь-якого графа дорівнює подвоєній кількості ребер, тобто:

$$\sum_{i=1}^n s_i = 2m, \quad (1.6)$$

де s_i – степінь i -ї вершини, $i = \overline{1, n}$;

m – кількість ребер графа.

Зокрема, якщо граф однорідний (повний), то з (1.6) будемо мати:

$$m = \frac{1}{2} n s \left(m = \frac{n(n-1)}{2} = C_n^2 \right), \quad (1.7)$$

де C_n^2 – кількість комбінацій без повторень із n елементів по 2.

2. Кількість непарних вершин будь-якого графа парна.

Ця властивість має цікаві ілюстрації, а саме:

у будь-якому гурті (зібранні) кількість осіб із непарною кількістю знайомих завжди парна;

кількість вершин многогранника, інцидентних непарній кількості ребер, парна;

кількість усіх людей, які будь-коли потисли руку іншим людям непарну кількість разів, парна.

У задачах застосовного характеру важливими є поняття "ізоморфізм" і "гомеоморфізм" графів.

Графи $G_1 = (X_1, U_1)$ і $G_2 = (X_2, U_2)$ називають **ізоморфними** (однаковими за формою, ідентичними), якщо між множинами X_1 і X_2 , U_1 і U_2 можна встановити бієкцію (взаємно однозначну відповідність), яка

зберігає інцидентність відповідних вершин і ребер (рис. 1.11), і позначають: $G_1 \cong G_2$ (на рис. 1.11 вершини графів зображено жирними точками, а точки перетину ребер не є вершинами графів). Бієктивна відповідність між вершинами графів така: $n \leftrightarrow n'$, $n = \overline{0,7}$ (\leftrightarrow – символ бієкції); неважко простежити збереження інцидентності вершин і ребер.

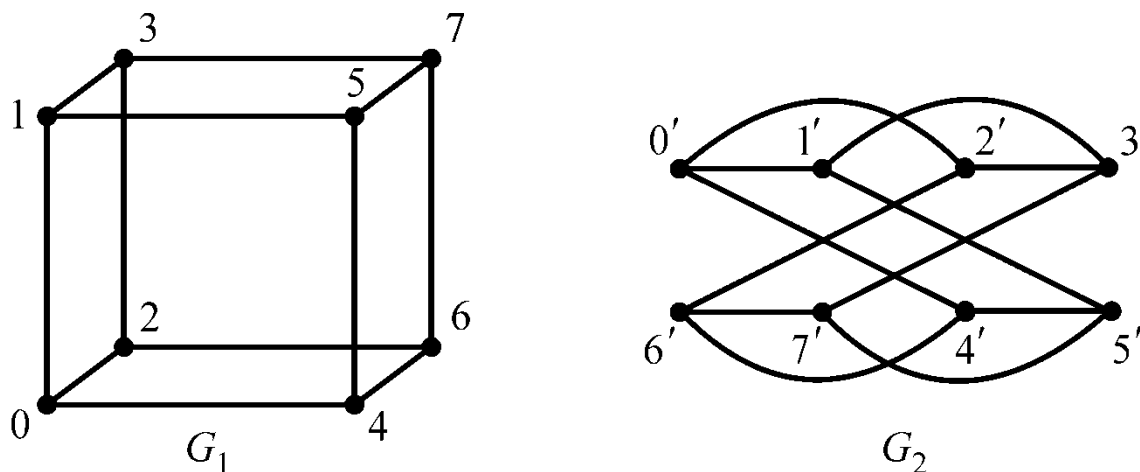


Рис. 1.11. Ізоморфні графи ($G_1 \cong G_2$)

Зауважимо, що графи, які мають однакове зображення і відрізняються лише нумерацією вершин і ребер, є ізоморфними.

Два графи G_1 і G_2 називають **гомеоморфними** (подібними, схожими), якщо після вилучення з них вершин другого степеня й об'єднання кожної пари інцидентних цим вершинам ребер в одне, вони виявляються ізоморфними (рис. 1.12), і позначають: $G_1 \sim G_2$.

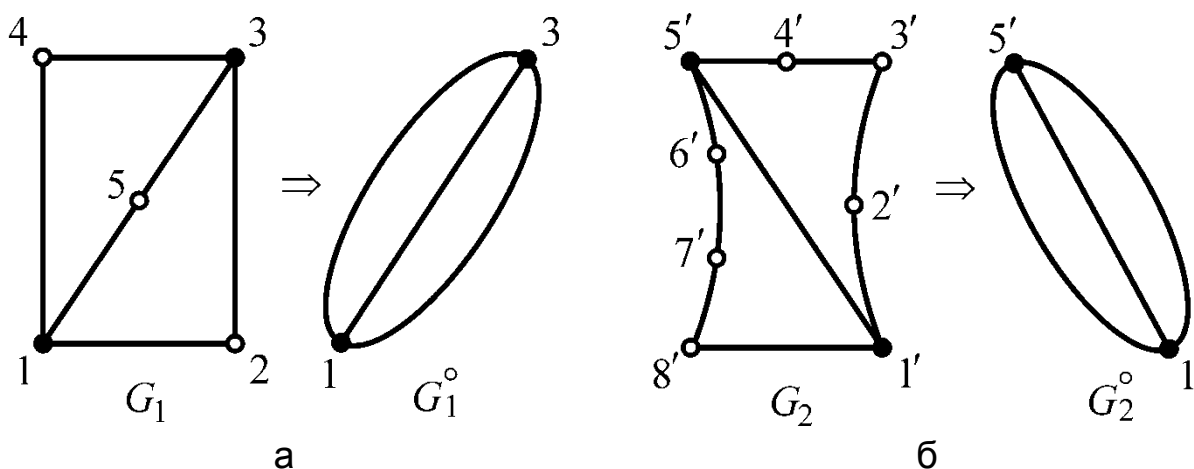


Рис. 1.12. Гомеоморфні графи ($G_1 \sim G_2$):

а) п'ятивершинний граф G_1 ; б) восьмивершинний граф G_2

На рис. 1.12а (1.12б) зображено граф G_1 (G_2), вершини 2-го степеня якого позначено кружками, і граф G_1° (G_2°), що здобувають із нього вилученням (згідно з означенням) вершин 2-го степеня. Як бачимо, $G_1^\circ \cong G_2^\circ$, а значить $G_1 \sim G_2$.

Гомеоморфні графи також називають **ізоморфними графами з точністю до вершин 2-го степеня**.

1.9. Планарність графів

Один і той самий граф G , заданий теоретико-множинним способом (переліком елементів множини вершин і множини ребер) чи матричним способом (за допомогою матриць інцидентностей або суміжності вершин), у разі геометричного задання можна подати по-різному. Проте всі геометричні зображення графа (площинні чи просторові) будуть ізоморфними між собою графами. Будь-який граф із сукупності ізоморфних графів, які є геометричним зображенням графа G , називають його **геометричною реалізацією**. Геометричну реалізацію графа на площині називають **плоскою реалізацією**, якщо вершинам відповідають різні точки площини, а ребра не мають спільних точок, окрім тих, які є вершинами графа. Зображення ізоморфних графів G_1 і G_2 (див. рис. 1.11) не є плоскими реалізаціями деякого графа G , а геометричні реалізації гомеоморфних графів G_1 і G_2 (див. рис. 1.12) плоскі.

Граф G , для якого є плоска реалізація, називають **плоским**, або **планарним графом** (ПГ). *Наприклад*, плоским графом є всяке дерево або граф, що містить цикл, із вершин якого "ростуть" дерева. Плоскі реалізації графів G_1 і G_2 (див. рис. 1.11) наведено на рис. 1.13, де $G'_1 \cong G_1$, $G'_2 \cong G_2$.

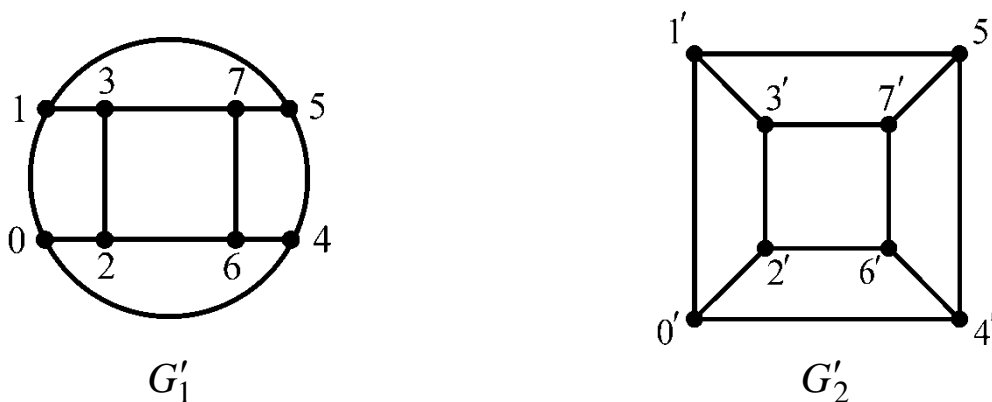


Рис. 1.13. Плоскі реалізації ПГ

Зрозуміло, що за сутністю графи G'_1 і G'_2 – це різні плоскі реалізації одного й того самого (8,12)-графа ($n=8$, $m=12$).

Із плоскою реалізацією графа пов'язують поняття "грань", на якому ґрунтуються важливі висновки щодо ПГ. **Гранню плоскої реалізації** ПГ називають частину площини, яка обмежена простим циклом і не містить інших циклів. Частину площини, розташовану зовні плоского зображення ПГ, називають **нескінченною**, або **зовнішньою, гранню**. Плоскі реалізації G'_1 і G'_2 (див. рис. 1.13) поділяють площину на шість граней, одна з яких, визначена контуром із вершинами 0, 1, 5, 4 ($0'$, $1'$, $5'$, $4'$), є нескінченною – це зовнішність відповідного кола (квадрата). Внутрішність кола (квадрата) не є гранню, бо вона містить цикл, визначений вершинами 2, 3, 7, 6 ($2'$, $3'$, $7'$, $6'$).

Для зв'язного ПГ справедлива **формула Ейлера**, яка описує залежність (зв'язок) між кількістю вершин n , кількістю ребер m і кількістю граней h , включаючи зовнішню грань:

$$n - m + h = 2. \quad (1.8)$$

Поділ площини на грані, який здійснюють плоскою реалізацією зв'язного ПГ, називають **плоскою картою**. До вивчення плоских карт зводять задачі, які виникають під час проектування комунікацій (доріг, водо-, газо-, нафтопроводів, ліній електромереж тощо) або планування експерименту, у радіоелектроніці та інших галузях.

Із плоскими картами пов'язано, зокрема, відому *проблему чотирьох фарб*: чи завжди можна пофарбувати плоску карту так, щоб ніякі дві суміжні грані не було пофарбовано в один колір і щоб водночас було використано не більше ніж чотири кольори? Зразу було доведено, що для такого пофарбування достатньо п'яти фарб (кольорів), але не було показано необхідність цієї умови. Відповідь на запитання, причому *позитивну*, було знайдено лише через півтора століття після його постановки – 1976 р. – за допомогою ЕОМ.

Загалом планарність, тобто властивість графа бути плоским, є істотною та важливою властивістю графів, коли йдеться про моделювання ними задач застосовного характеру.

Але чи для кожного графа можна знайти його плоску реалізацію? Відповідь негативна. До неї (незалежно один від одного) дійшли російський

математик Л. С. Понтрягін та польський математик К. Куратовський. Вони показали, що плоску реалізацію не допускають:

- 1) повний граф із п'яти вершин (K_5);
- 2) повний біхроматичний граф із шести вершин ($K_{3,3}$), який має по три вершини в кожній частці (рис. 1.14).

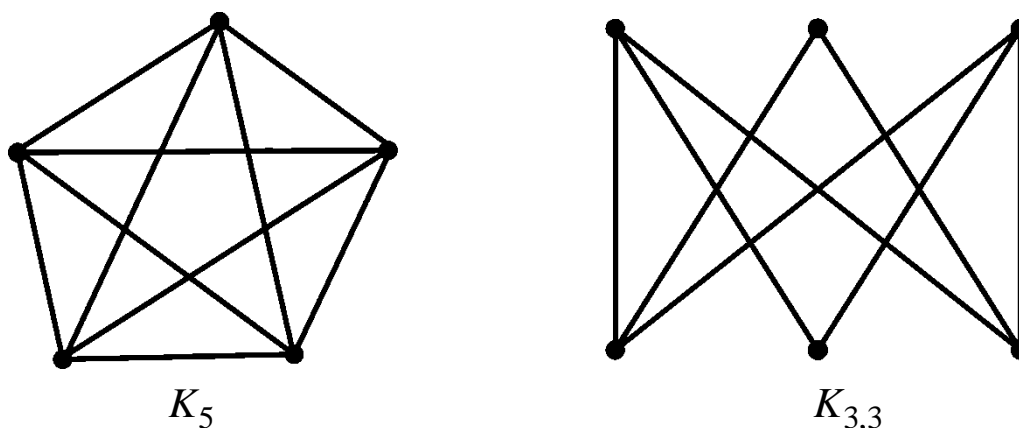


Рис. 1.14. Графи Понтрягіна – Куратовського

Теорема Понтрягіна – Куратовського (*критерій існування ПГ*): граф є планарним тоді й тільки тоді, коли він не містить підграфа, гомеоморфного графу K_5 або $K_{3,3}$.

Є й інші критерії планарності графів, але їх у пропонованих методичних рекомендаціях не будемо розглядати.

1.10. Ейлерові та гамільтонові графи

Цікавими та важливими із практичного погляду є задачі на ПГ пов'язані з так званими **обходами графа** – маршрутами, які містять усі вершини чи ребра графа й мають певні властивості. Найбільш відомими обходами графа є ейлерові та гамільтонові ланцюги та цикли.

Маршрут (замкнений маршрут) називають **ейлеровим ланцюгом (циклом)**, якщо він містить усі ребра графа та проходить через кожне ребро тільки один раз.

Установлено ефективний критерій існування ейлерових циклів для зв'язних графів – **теорема Ейлера**: зв'язний граф містить ейлерів цикл тоді й тільки тоді, коли кожна його вершина парна.

Отже, якщо виконано умови теореми Ейлера, то має розв'язок **задача 1.1**: починаючи з деякої вершини та закінчуючи нею, зобразити заданий планарний граф, не відриваючи олівця від паперу, не обводячи двічі одне й те саме ребро.

Розглянемо один зі способів розв'язання цієї задачі – спосіб "**штрихування граней і роз'єднання графа**" – на прикладі ПГ, зображеного на рис. 1.15.

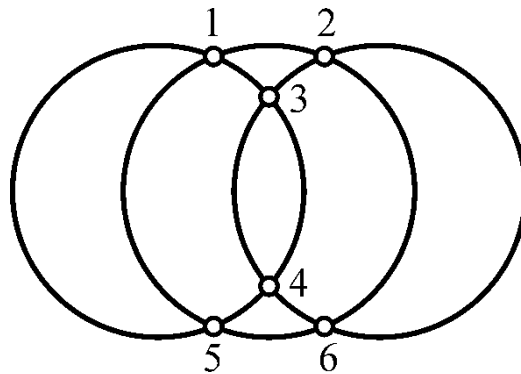


Рис. 1.15. Однорідний граф 4-го степеня

Розв'язання. Заданий 6-вершинний ($n = 6$) 12-реберний ($m = 12$) граф поділяє площину на 8 граней ($h = 8$), включаючи зовнішню грань (що, без сумніву, погоджено з формулою Ейлера (1.8)): $n - m + h = 6 - 12 + 8 = 2$.

Залишаючи зовнішню грань без змін, вибираємо одну із суміжних із нею (по ребру) граней і заштриховуємо її; наприклад, ту, що визначається вершинами 1, 2, 3 (рис. 1.16а). Грань, яка має спільне ребро із цією (заштрихованою) гранню, треба проминути, а ту, що має лише спільну вершину, теж заштрихувати. Так діємо до тих пір, поки всі можливі грані не буде заштриховано (див. рис. 1.16а).

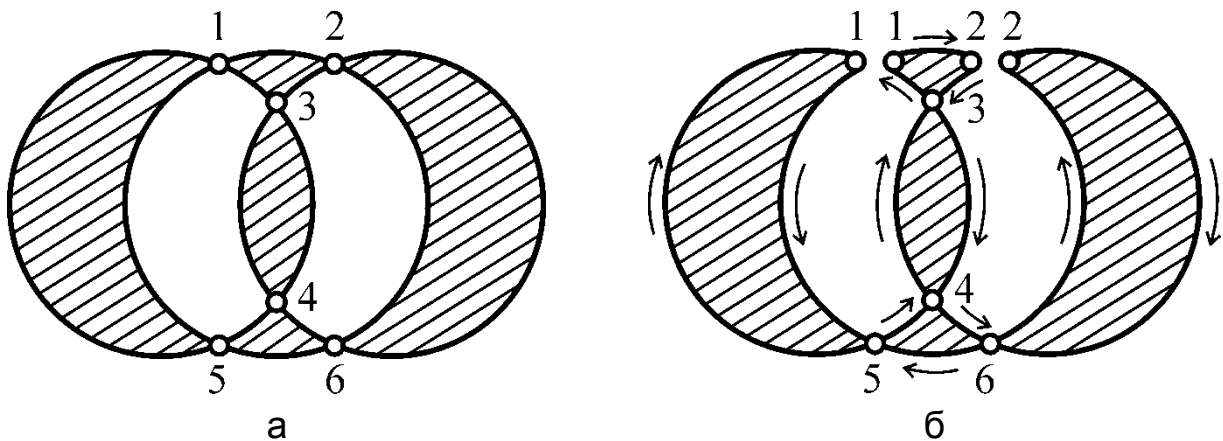


Рис. 1.16. Штрихування граней (а) і роз'єднання графа (б)

Далі заштрихований граф слід роз'єднати в одній або декількох вершинах так, щоб утворилася суцільно заштрихована область (без "дірок"), як це показано на рис. 1.16б.

Видно, що знайдену таким чином область (роз'єднанням графа у вершинах 1, 2) можна охопити (окреслити) суцільною лінією, яка складається з усіх ребер графа, не обходячи двічі одне й те саме ребро. Один із циклів, що охоплює наш своєрідний "бутон", такий:

$$C = (1-2-3-4-6-2-6-5-1-5-4-3-1).$$

Спробуйте самостійно знайти інші ейлерові цикли, запропонуйте інше роз'єднання графа й укажіть хоча б один відповідний обхід графа.

Задачі розглянутого типу виникають під час проектування самописних приладів; наприклад, графопобудовників для ЕОМ.

Граф називають **ейлеровим**, якщо він містить ейлерів цикл.

Ейлер також показав:

1) якщо дві вершини графа непарні, то, вийшовши з однієї непарної вершини, можна досягти іншої, не проходячи ні по одному з ребер двічі;

2) якщо непарних вершин $2k$, то існує k ланцюгів, які починаються і закінчуються в непарних вершинах.

Застосування цих висновків продемонстровано під час розв'язання нескладної практичної задачі.

Задача 1.2. Ділянку трамвайної мережі схематично подано графом із вершинами-зупинками M , N , P , Q , R , S (рис. 1.17). Треба встановити мінімальну кількість маршрутів для забезпечення руху пасажирів від однієї будь-якої зупинки до всякої іншої та указати самі маршрути; водночас кожен трамвай має рухатися лише своїм маршрутом, тобто відповідні ланцюги на графі не повинні мати спільних ребер.

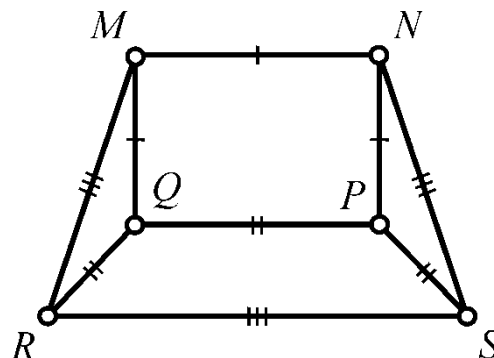


Рис. 1.17. Граф ділянки трамвайної мережі

Розв'язання. Аналізуючи граф, робимо висновок, що він не є ейлеровим (чому?). У нього шість ($6 = 2 \cdot 3$) непарних вершин, а значить, існує 3 ланцюги-маршрути, які починаються та закінчуються в непарних вершинах. Ці маршрути позначено (див. рис. 1.17), відповідно, однією, двома та трьома рисками: $QMNP$, $RQPS$, $MRSN$.

Перейдемо тепер до ознайомлення з маршрутами, які містять усі вершини графа. Маршрут (замкнений маршрут) називають **гамільтоновим ланцюгом (циклом)**, якщо він містить усі вершини графа й через кожну з них проходить лише один раз.

На перший погляд, за аналогією з відшукуванням ейлерових циклів, задача про побудову гамільтонових циклів здається розв'язною. Але це не так: необхідні й достатні умови – критерій – існування гамільтонового циклу у графі поки що не відомі.

Установлено ряд *достатніх умов*, наприклад:

1. Якщо для будь-яких двох несуміжних вершин зв'язного графа сума степенів не менша за кількість вершин, то в такому графі існує гамільтонів цикл C_G :

$$\forall x_i, x_j : \bar{\exists} \{x_i, x_j\} \wedge s_i + s_j \geq n \Rightarrow \exists C_G, \quad (1.9)$$

де x_i, x_j – вершини графа, $i \in \{1, 2, \dots, n\}$, $j \in \{1, 2, \dots, n\}$;

n – кількість вершин;

s_i, s_j – степені вершин;

\forall – квантор загальності; який читають: "будь-який", "для всіх", "усі";

\Rightarrow – символ логічного наслідку, який читають: "якщо ..., то ..." або "із ... випливає ...", або "... тягне за собою ...";

\wedge – символ, який читають: "і".

\exists – квантор існування, який читають: "існує"; $\bar{\exists}$ – "не існує".

2. Якщо граф не має петель і кратних ребер, а кількість n його вершин і кількість m його ребер задовольняють умови:

$$n \geq 3, \quad m \geq C_{n-1}^2 + 2 = \frac{1}{2}(n^2 - 3n + 6), \quad (1.10)$$

то в ньому існує гамільтонів цикл C_G .

Перевірте, чи виконуються наведені достатні умови для графа, зображеного на рис. 1.17, і зробіть відповідні висновки.

Графи, у яких існує гамільтонів цикл, називають **гамільтоновими**.

Задача про відшукування гамільтонових циклів має різні узагальнення. Одне з них – широко відома *задача комівояжера* (посильного): є n міст, відстані між якими відомі. Треба знайти найкоротший маршрут, який забезпечив би відвідування всіх міст (лише по одному разу) і повернення у вихідний пункт.

Принципово розв'язати цю задачу дуже просто: перебрати всі можливі маршрути на відповідному графі й порівняти їхні довжини. Але з технічного погляду здійснити це важко навіть за допомогою ЕОМ (бо кількість маршрутів, яка дорівнює $(n-1)!$, швидко зростає зі збільшенням n). На практиці цю (й інші) оптимізаційні задачі на графах розв'язують здебільшого **методом гілок і меж**, який вивчають у навчальній дисципліні "Дослідження операцій та методи оптимізації".

Запитання для самоконтролю за засвоєнням матеріалу

1. Що називають неорієнтованим графом, або просто графом?
2. Що розуміють під ребрами, вершинами, граничними вершинами ребра графа?
3. У якому разі ребро графа називають інцидентним вершині графа, і навпаки?
4. Які ребра (вершини) графа називають суміжними, або сусідніми?
5. Як називають вершину графа, не інцидентну жодному ребру (інцидентну тільки одному ребру)?
6. Яке ребро графа називають петлею?
7. Які ребра графа називають кратними, або паралельними?
8. У чому полягає кожний зі способів задання НГ: а) теоретико-множинний; б) геометричний; в) матричний?
9. Що розуміють під маршрутом на графі, його довжиною, незамкненим (замкненим) маршрутом?
10. Як називають незамкнений (замкнений) маршрут, усі ребра якого різні?
11. Який ланцюг (цикл) називають простим, або елементарним?
12. Який граф називають зв'язним (незв'язним)?
13. Що розуміють під компонентами зв'язності незв'язного графа?

14. Як формулюють постановку задачі про знаходження найкоротшого ланцюга між двома вершинами графа?
15. Як розв'язують задачу про знаходження найкоротшого ланцюга методом індексації (зважування) вершин графа?
16. Який зв'язний граф називають деревом (лісом)?
17. Що розуміють під коренем дерева, рівнем вершин дерева?
18. Скільки ребер містить дерево, яке покриває n вершин? Доведіть відповідне твердження.
19. Скільки існує різних дерев, які покривають n заданих вершин? Доведіть відповідне твердження.
20. Що називають кістяком (покривним дерево) графа?
21. Як дістати кістяк графа з n вершинами та m ребрами, якщо він містить цикли?
22. Як формулюють постановку задачі про знаходження на графі економічного дерева? Як розв'язують цю задачу?
23. Який граф називають простим (або звичайним), мультиграфом, псевдографом? Наведіть відповідні геометричні зображення.
24. Який граф називають тривіальним, нуль-графом, повним? Наведіть відповідні геометричні зображення.
25. Що називають степенем вершини графа?
26. Яку умову щодо степенів вершин має задовольняти однорідний граф?
27. Який зв'язок є між сумою степенів вершин графа і кількістю його ребер?
28. Який граф називають біхроматичним (біграфом)? Наведіть геометричне зображення.
29. Що називають планарним (плоским) графом?
30. Які графи називають ізоморфними?
31. Якщо заданий граф планарний, чи буде ізоморфний йому граф теж планарним?
32. Чи будуть ізоморфними графи, які мають однакові зображення та відрізняються лише нумерацією вершин і ребер?
33. Що називають гранню планарного графа, його плоскою картою?
34. Який зв'язок є між кількістю вершин, ребер і граней плоского графа?
35. Який вигляд мають графи Понтрягіна – Куратовського: повний граф із п'ятьма вершинами та повний біхроматичний граф із шістьма вершинами?

36. Які два графи називають гомеоморфними (ізоморфними з точністю до вершин 2-го степеня)?
37. Як формулюють критерій існування плоского графа (теорема Понтрягіна – Куратовського)?
38. Для яких практичних задач під час математичного моделювання планарність графа є істотною властивістю?
39. Який маршрут (замкнений маршрут) на графі називають ейлеровим ланцюгом (циклом)?
40. Який маршрут (замкнений маршрут) на графі називають гамільтоновим ланцюгом (циклом)?
41. Які графи називають ейлеровими, гамільтоновими?
42. Які критерії (чи тільки достатні умови) існування ейлерових та гамільтонових графів вам відомі?
43. Яку умову мають задовольняти дві вершини зв'язного графа, якщо між цими вершинами є ланцюг, який містить усі ребра графа?
44. За яких умов на зв'язному графі існує k ($k > 1$) ланцюгів, які починаються і закінчуються в непарних вершинах?
45. У чому полягає постановка задачі про комівояжера та які підходи до її розв'язання вам відомі?

2. Орієнтовані графи (орграфи)

2.1. Означення основних понять, способи задання

Орієнтованим графом (ОГ), або **орграфом**, називають упорядковану пару $\bar{G} = (X, U)$, де $X = \{x_i\}_1^n$ – непорожня скінченна множина деяких об'єктів (елементів), $U = \{u_j\}_1^m$ – множина упорядкованих пар, складених з елементів множини X : $u_j = (x_i, x_k)$, $i, k \in I_x = \{1, 2, \dots, n\}$, $j \in I_u = \{1, 2, \dots, m\}$; I_x, I_u – множини індексів.

Як бачимо, означення неорієнтованого й орієнтованого графів відрізняються лише тим, що у графі розглядають невпорядковані пари $\{x_i, x_k\} = \{x_k, x_i\}$, а в орграфі – упорядковані пари (x_i, x_k) , тобто $(x_i, x_k) \neq (x_k, x_i)$. Ця (єдина) структурна відміна дозволяє (іноді з деякими

змiнами) перенести на орграфи поняття й термини, якi використовують пiд час вивчення неорiєнтованих графiв.

Наведемо деякi вiдповiднi термини:

1) ребро $\{x_i, x_k\} \leftrightarrow$ **орiєнтоване ребро (орребро)**, або **дуга** (x_i, x_k) , що виходить iз вершини x_i i заходить у вершину x_k ; x_i (x_k) – **початок (кiнець) дуги**;

2) маршрут $(x_1 - x_2 - \dots - x_k) \leftrightarrow$ **ормаршрут** $(x_1 \rightarrow x_2 \rightarrow \dots \rightarrow x_k)$;

3) ланцюг \leftrightarrow **орланцюг**, або **шлях**;

4) цикл \leftrightarrow **орцикл**, або **контур**.

Щодо орграфiв розрiзняють два види iнцидентностi: дуга **додатно (вiд'ємно) iнцидентна** її початковiй (кiнцевiй) вершинi, а також два види паралельностi: двi дуги з однаковими граничними точками називають **строго (не строго) паралельними**, якщо їхнi початковi вершини й кiнцевi вершини збiгаються (не збiгаються).

Основою орграфа \overline{G} називають неорiєнтований граф, який дiстають iз \overline{G} у результатi зняття орiєнтацiї з його дуг.

Способи задання орiєнтованих графiв такi самi, як i неорiєнтованих, але з певними змiнами.

У разi **теоретико-множинного** способу наводять перелiк елементiв множини вершин $X = \{x_i\}_1^n$ i множини дуг $U = \{u_j\}_1^m$ орграфа \overline{G} .

У разi **геометричного** способу вершини орграфа зображають у \mathbf{R}^2 чи \mathbf{R}^3 точками, а дуги $u_j = (x_i, x_k)$ – стрiлками, напрямленими вiд вершини x_i до вершини x_k ; i лише цим вiдрiзняються зображення графiв та орграфiв.

У разi **матричного** задання орграф задають матрицею iнцидентностей i/або матрицею сумiжностi вершин. **Матрицею iнцидентностей** графа \overline{G} iз n вершинами та m дугами називають матрицю $A = [a_{ij}]_{n \times m}$ розмiру $n \times m$ з елементами a_{ij} ($i = \overline{1, n}$, $j = \overline{1, m}$), такими що:

$$a_{ij} = \begin{cases} +1, & \text{якщо } x_i \text{ – початок } u_j; \\ -1, & \text{якщо } x_i \text{ – кiнець } u_j; \\ 0, & \text{якщо } x_i \text{ не iнцидентна } u_j; \\ \pm 1, & \text{якщо } u_j \text{ – петля при вершинi } x_i. \end{cases} \quad (2.1)$$

Матрицею суміжності вершин графа \overline{G} називають квадратну матрицю $B = [b_{ij}]_{n \times n}$ n -го порядку, елементи якої b_{ij} дорівнюють кількості дуг, що йдуть від вершини x_i до вершини x_j ($i = \overline{1, n}, j = \overline{1, n}$). На відміну від матриці B неорієнтованого графа, матриця B орграфа в загальному випадку не є симетричною.

Приклад задання (6,10)-орграфа \overline{G} різними способами:

а) теоретико-множинне задання:

$$G = (X, U) = (\{x_1, x_2, x_3, x_4, x_5, x_6\}, \{u_1 = (x_1, x_1), u_2 = (x_1, x_3), u_3 = (x_1, x_2), u_4 = (x_3, x_2), u_5 = (x_2, x_4), u_6 = (x_4, x_3), u_7 = (x_4, x_3), u_8 = (x_3, x_5), u_9 = (x_4, x_5), u_{10} = (x_5, x_4)\});$$

б) геометричне зображення цього ж графа подано на рис. 2.1:

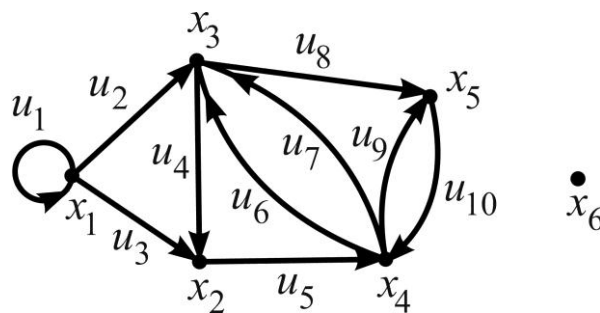


Рис. 2.1. Геометричне зображення орграфа

в) матриці інцидентностей A і суміжності вершин B мають такий вигляд:

$$A = \begin{matrix} & \begin{matrix} u_1 & u_2 & u_3 & u_4 & u_5 & u_6 & u_7 & u_8 & u_9 & u_{10} \end{matrix} \\ \begin{matrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{matrix} & \begin{bmatrix} \pm 1 & +1 & +1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & -1 & +1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & +1 & 0 & -1 & -1 & +1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & +1 & +1 & 0 & +1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & +1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{matrix},$$

$$B = \begin{matrix} & x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \\ \begin{matrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{matrix} & \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 2 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{matrix}.$$

Нульовий рядок у матриці A відповідає ізольованій вершині, а нульовий стовпець – петлі; у матриці B рядки та стовпці, що відповідають ізольованій вершині, містять лише нулі.

В орграфі розрізняють *додатний півстепінь* вершини (*півстепінь виходу*) $s_+(x_i)$ та *від'ємний півстепінь* (*півстепінь входу*) $s_-(x_i)$: $s_+(x_i)$ – кількість дуг, що виходять із вершини x_i , а $s_-(x_i)$ – кількість дуг, що заходять у вершину x_i ($i = \overline{1, n}$). Наприклад, для вершини x_4 орграфа на рис. 2.1 маємо: $s_+(x_4) = 3$, $s_-(x_4) = 2$.

Зрозуміло, що сума півстепенів виходу вершин орграфа дорівнює сумі півстепенів входу і дорівнює кількості всіх дуг:

$$\sum_{i=1}^n s_+(x_i) = \sum_{i=1}^n s_-(x_i) = m.$$

Орграфи, як і графи, мають численні застосування в задачах практичного характеру. Це, зокрема, задачі соціології, економіки, кібернетики, техніки, електроніки, лінгвістики тощо. Орграфи широко застосовують у програмуванні як спосіб опису систем зі складними зв'язками.

2.2. Досяжність вершин, відстань між вершинами, типи зв'язності орграфа

Нехай $\overline{G} = (X, U)$ – орієнтований граф; x_i, x_j – його довільні вершини ($x_i, x_j \in X$). Кажуть, що вершина x_j *досяжна з вершини* x_i , якщо існує шлях (орланцюг) μ_{ij} , що веде із x_i до x_j ; інакше кажуть, що вершина x_j *недосяжна з вершини* x_i . Певна річ, кожна вершина досяжна

із самої себе. Вершини x_i і x_j називають **взаємно досяжними**, якщо x_j досяжна з x_i , а x_i досяжна з x_j .

Вершину, із якої недосяжна жодна інша вершина орграфа, називають **тупиковою**. Вершину, яка не є досяжною з будь-якої іншої вершини орграфа, називають **недосяжною**. Згідно з наведеними означеннями, справедливими є такі *твердження*: вершина орграфа є тупиковою (недосяжною) тоді й тільки тоді, коли її додатний (від'ємний) степінь дорівнює нулеві.

Вершину, із якої досяжна будь-яка інша вершина орграфа, називають **джерелом**, а вершину, яка досяжна з будь-якої іншої вершини орграфа, називають **стоком**.

Відстанню від вершини x_i до вершини x_j називають довжину найкоротшого шляху з x_i до x_j і позначають $l(x_i, x_j)$. Якщо вершина x_j недосяжна з вершини x_i , то пишуть: $l(x_i, x_j) = \infty$.

Наприклад, для орграфа на рис. 2.1 відстань від вершини x_1 до вершини x_5 дорівнює 2: $l(x_1, x_5) = 2$, оскільки шлях $\mu_{17} = (x_1 \rightarrow x_3 \rightarrow x_5)$ є найкоротшим із шляхів, що ведуть із x_1 до x_5 . Але вершина x_1 не є досяжною з вершини x_5 , оскільки жоден шлях не веде з x_5 до x_1 , тому $l(x_5, x_1) = \infty$. Більш того, $l(x_3, x_2) = 1$, а $l(x_2, x_3) = 2$. Звідси зрозуміло, що відношення на множині вершин графа "бути досяжною" в загальному випадку не є симетричним.

Матрицею досяжності орграфа \overline{G} називають квадратну матрицю $C = [c_{ik}]_{n \times n}$ n -го порядку (n – кількість вершин \overline{G}) з елементами $c_{ik} = 1$, якщо є шлях від вершини x_i до вершини x_k , і $c_{ik} = 0$, а інакше ($i, k \in I_x$). Матрицю Q , транспоновану щодо матриці C , називають **матрицею контрдосяжності**: $Q = C^T$.

Зв'язність для орієнтованих графів визначають так само, як і для неорієнтованих, тобто без урахування напрямків дуг. Розрізняють кілька типів зв'язності. Орграф \overline{G} називають:

слабкозв'язним, якщо його основа G зв'язна (рис. 2.2а);

сильнозв'язним, якщо для будь-яких двох його вершин x_i, x_k існує орієнтований шлях із x_i в x_k , і навпаки (рис. 2.2б);

односторонньо зв'язним, якщо для будь-яких двох його вершин x_i і x_k принаймні одна є досяжною з іншої, тобто існує хоча б один із маршрутів від x_k до x_i чи від x_i до x_k (рис. 2.2в).

Орграф називають **незв'язним**, якщо він не є слабкозв'язним.

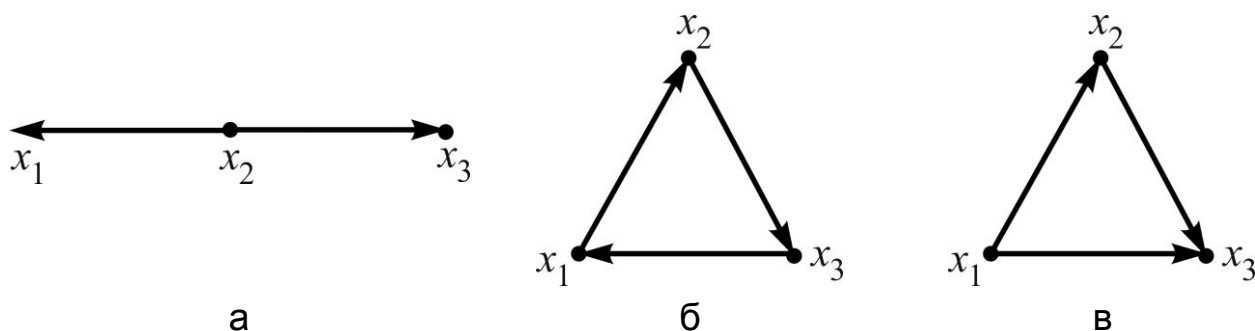


Рис. 2.2. Орграф:

а) слабкозв'язний; б) сильнозв'язний; в) односторонньо зв'язний

Компоненти зв'язності, згідно з означеннями, називають, відповідно, **слабкою (сильною, односторонньою) компонентою зв'язності**.

Одним із важливих понять у задачах прикладного характеру є поняття "графа конденсації". Нехай орграф \bar{G} містить контури. Орграф, який дістають об'єднанням усіх вершин кожного з них в одну, називають **графом конденсації**. Вислідний граф є ациклічним, тобто не містить циклів.

Задача 2.1. Для орграфа \bar{G} на рис. 2.3 необхідно: 1) установити, до якого типу графів він належить; 2) визначити компоненти зв'язності у графі; 3) побудувати граф конденсації \bar{G}^* ; 4) побудувати матриці досяжності та контрдосяжності.

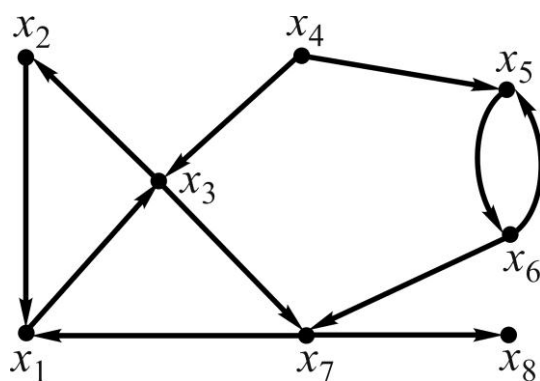


Рис. 2.3. Вихідний орграф

Розв'язання:

1. Здійснюємо візуальний аналіз заданого графа, із метою встановлення того, до якого типу графів він належить. Отже, граф \overline{G} – *орієнтований мультиграф*, оскільки всі його ребра орієнтовані та є кратні дуги; *слабкозв'язний*, адже відповідний неорієнтований граф є зв'язним.

2. Визначаємо компоненти зв'язності: граф \overline{G} містить компоненту сильної зв'язності, яку визначають вершини x_1, x_2, x_3 (усі її вершини взаємно досяжні), та односторонню компоненту зв'язності, її визначають вершини $x_4 - x_8$ (для будь-яких двох вершин принаймні одна досяжна з іншої) (рис. 2.4).

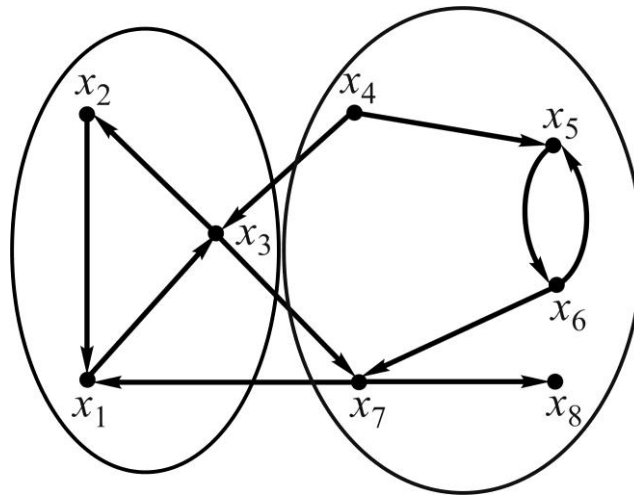


Рис. 2.4. Компоненти зв'язності вихідного орграфа

3. Будуємо граф конденсації \overline{G}^* , для чого в заданому орграфі \overline{G} знаходимо контур та об'єднуємо всі його вершини в одну. Ці дії повторюємо доти, поки в орграфі \overline{G} не залишиться контурів. На рис. 2.3 орцикли визначають вершини: 1) x_1, x_3, x_2 ; 2) x_1, x_3, x_7 ; 3) x_5, x_6 . Перший із них об'єднуємо (стягнемо) у вершину x_2 ; другий – у вершину x_3 ; третій – у вершину x_5 ; граф \overline{G}^c зображено на рис. 2.5. Згідно з означенням графа конденсації, як результат, дістаємо ациклічний орграф (він не містить контурів).

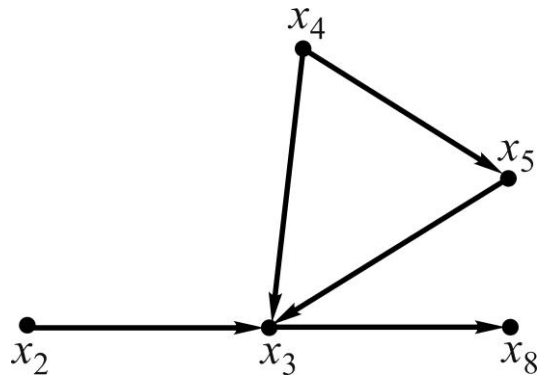


Рис. 2.5. Граф конденсації

4. Будуємо матрицю досяжності C і матрицю контрдосяжності Q :

$$C = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \quad Q = \begin{pmatrix} 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}.$$

2.3. Сіткові графіки

2.3.1. Сіткові графіки: основні означення, правила побудови

Практична реалізація будь-якого проєкту в різних галузях людської діяльності (створення інформаційних систем, виготовлення технічних виробів, підготовка та проведення експериментів, будівництво споруд і т. ін.) потребує виконання значної кількості робіт.

Під **роботою** (операцією) розуміють будь-який процес, сукупність дій, за допомогою яких досягають певних результатів. Результати виконання однієї або кількох робіт називають **подіями** (етапами, стадіями). Безумовно, усі роботи мають бути погоджені в часі їхнього виконання і враховувати певні (додаткові) умови щодо ресурсів (людських, технічних, енергетичних, фінансових тощо), строків виконання і т. ін.

Припустимо, що визначено комплекс робіт $U = \{u_1, u_2, \dots, u_m\}$, необхідних для реалізації певного проєкту, і множину подій $X = \{x_0, x_1, \dots, x_n\}$

, де x_0 – початок, x_n – кінець, x_i ($i \neq 0, n$) – проміжні події (залежно від взаємозв'язку робіт).

Задачі, у яких ставлять питання про знаходження мінімального часу виконання всіх робіт комплексу U , називають **оптимізаційними задачами за критерієм часу**. Для розв'язання таких задач на основі теорії графів розроблено спеціальні математичні методи так званого *сіткового планування й управління* (СПУ).

Орієнтований зв'язний граф $\bar{G} = (X, U)$ без петель і контурів, який відображає природний порядок виконання робіт комплексу U в часі, у поєднанні з певними числовими даними – характеристиками проекту, – називають **сітковим графіком (СГ)** комплексу робіт (цього проекту).

Перш ніж будувати СГ проекту, треба скласти відповідну **структурно-часову таблицю** – докладний перелік подій і робіт комплексу, місце роботи в комплексі, її технологічні зв'язки з іншими роботами, строки виконання робіт, інші числові характеристики (витрати ресурсів, кількість виконавців і т. ін.).

Слід навести найважливіші правила побудови СГ:

1. Скласти структурно-часову таблицю.

2. Зобразити в \mathbf{R}^2 чи \mathbf{R}^3 роботи дугами, над (під) якими вказують потрібний для їхнього виконання час (іноді й інші числові характеристики), а події (граничні точки дуг) – кружечками, у яких записують порядковий номер або шифр події. (Подія як результат виконання попередніх робіт є передумовою виконання наступних операцій.)

3. На СГ не має бути (згідно з означенням) контурів і петель, а також **тупикових** подій (окрім кінцевої x_n), тобто таких проміжних подій, які не дають початку жодній роботі, і **хвостових** подій (окрім початкової x_0), тобто таких проміжних подій, яким не передує жодна робота.

4. На СГ уникають наявності паралельних дуг, що відповідає одночасно виконуваним роботам між двома подіями. Якщо, *наприклад*, u_1, u_2 – операції (дуги), що визначають одні й ті самі події (вершини) x_1, x_2 , то беруть на розгляд **допоміжну** подію x'_2 і роботу $u'_2 = (x'_2, x_2)$ із нульовою тривалістю, яку називають **фіктивною**, тобто кінцева вершина x_2 немовби "роздвоюється" (рис. 2.6а). Якщо виконують r паралельних операцій, то вводять $(r - 1)$ допоміжних подій і фіктивних робіт.

5. Якщо не всі роботи, що починаються подією $x \in X$, спираються на всі роботи, що закінчуються подією x , то для відповідного розподілу запроваджують додаткові (допоміжні) вершини та фіктивні дуги. Нехай, *наприклад*, роботи u_1, u_2, u_3 передують події x , а u_4, u_5 – наступні роботи, причому u_4 спирається на u_1, u_2, u_3 , а робота u_5 іде лише за u_3 . Тоді відповідний розподіл здійснюють уведенням додаткової події x' і фіктивної роботи u' (рис. 2.6б).

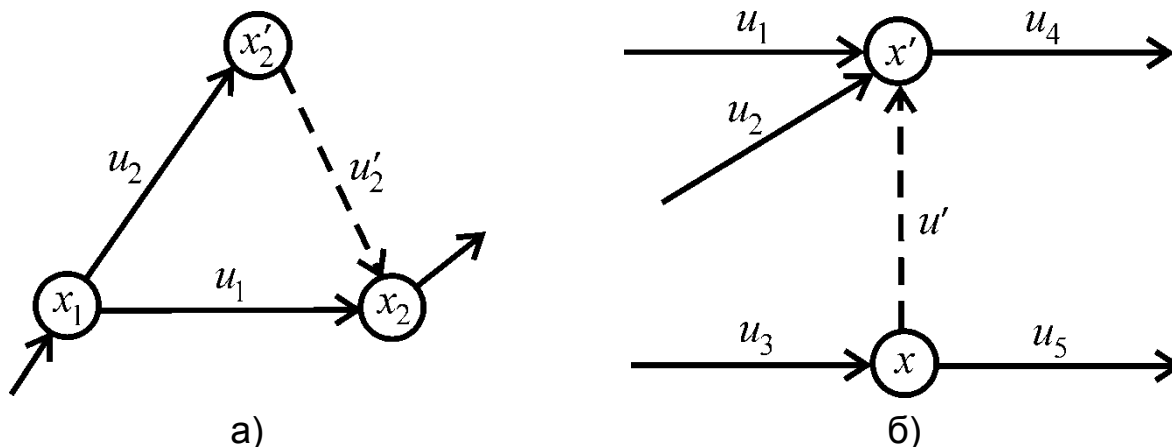


Рис. 2.6. Уведення на СГ:
а) допоміжних вершин; б) фіктивних робіт

2.3.2. Задача знаходження критичного часу і критичного шляху на сітковому графіку

Для спрощення формального опису розрахунків на СГ $\bar{G} = (X, U)$ вершини (події) $x_i, x_k \in X$ будемо називати за їхніми номерами i, k ($\forall i, k \in I_x = \{0, 1, \dots, n\}$), а дуги (роботи) розглядати як пари (i, k) .

Уведемо позначення:

$U_{i \rightarrow k}$ ($U_{\rightarrow k}$) – множина операцій, які починаються (закінчуються) подією i (k), тобто множина дуг, додатно (від'ємно) інцидентних вершині i (k);

$t(i)$ – **строк звершення події i** – час, який пройшов від нульової події до моменту, коли завершилася (насталла, відбулася) подія i ; аналогічно $t(k)$ – строк звершення події k ;

$t(i, k)$ – **тривалість роботи** (i, k) , яка починається подією i та закінчується подією k .

Зрозуміло, що тривалість $t(i, k)$ не може перевищувати різницю в часі між моментами настання подій i, k . Отже, між величинами $t(i), t(k), t(i, k)$ мають місце співвідношення:

$$t(i, k) \leq t(k) - t(i); \quad (2.2)$$

$$t(k) \geq t(i) + t(i, k); \quad (2.3)$$

$$t(i) \leq t(k) - t(i, k), \quad (2.4)$$

які дають змогу обчислити ранній і пізній строки настання події на СГ.

Раннім строком $t_p(k)$ **звершення події** k називають найменший можливий час настання події k за умови, що всі попередні події відбуваються без затримки.

Згідно з нерівністю (2.3), маємо:

1) якщо подією k закінчується одна робота (i, k) , то

$$t(k) \geq t(i) + t(i, k) \Rightarrow t_p(k) = t_p(i) + t(i, k);$$

2) якщо подією k закінчується декілька робіт, то за кожною роботою знаходять свій ранній строк, а шуканим $t_p(k)$ буде максимальний із них:

$$(\forall (i, k) \in U_{\rightarrow k} : t(k) \geq t(i) + t(i, k)) \Rightarrow t_p(k) = \max_{(i, k) \in U_{\rightarrow k}} \{t_p(i) + t(i, k)\}. \quad (2.5)$$

Пізнім строком $t_{\Pi}(i)$ **звершення події** i називають найбільший можливий час настання події i за умови, що всі наступні події відбуваються без затримки.

Згідно з нерівністю (2.4), маємо:

1) якщо подією i починається одна робота (i, k) , то

$$t(i) \leq t(k) - t(i, k) \Rightarrow t_{\Pi}(i) = t_{\Pi}(k) - t(i, k);$$

2) якщо ж подією i починається декілька робіт, то за кожною роботою знаходять свій пізній строк, а шуканим $t_{\Pi}(i)$ буде мінімальний із них:

$$(\forall (i,k) \in U : t(i) \leq t(k) - t(i,k)) \Rightarrow t_{\Pi}(i) = \min_{\substack{(i,k) \in U \\ i \rightarrow k}} \{t_{\Pi}(k) - t(i,k)\}. \quad (2.6)$$

Приклад. Нехай подією k закінчуються три роботи тривалістю $t(i,k)$, $i=1,2,3$, і відомі ранні строки $t_p(i)$ (рис. 2.7а); подією i починаються три роботи $t(i,k)$, $k=4,5,6$, і відомі пізні строки $t_{\Pi}(k)$ (рис. 2.7б). Тоді, згідно з (2.5) і (2.6), маємо:

$$\begin{aligned} t_p(k) &= \max\{t_p(1) + t(1,k), t_p(2) + t(2,k), t_p(3) + t(3,k)\} = \\ &= \max\{3 + 4, 2 + 3, 2 + 1\} = \max\{7, 5, 3\} = 7; \\ t_{\Pi}(i) &= \min\{t_{\Pi}(4) - t(i,4), t_{\Pi}(5) - t(i,5), t_{\Pi}(6) - t(i,6)\} = \\ &= \min\{6 - 2, 10 - 8, 15 - 10\} = \min\{4, 2, 5\} = 2. \end{aligned}$$

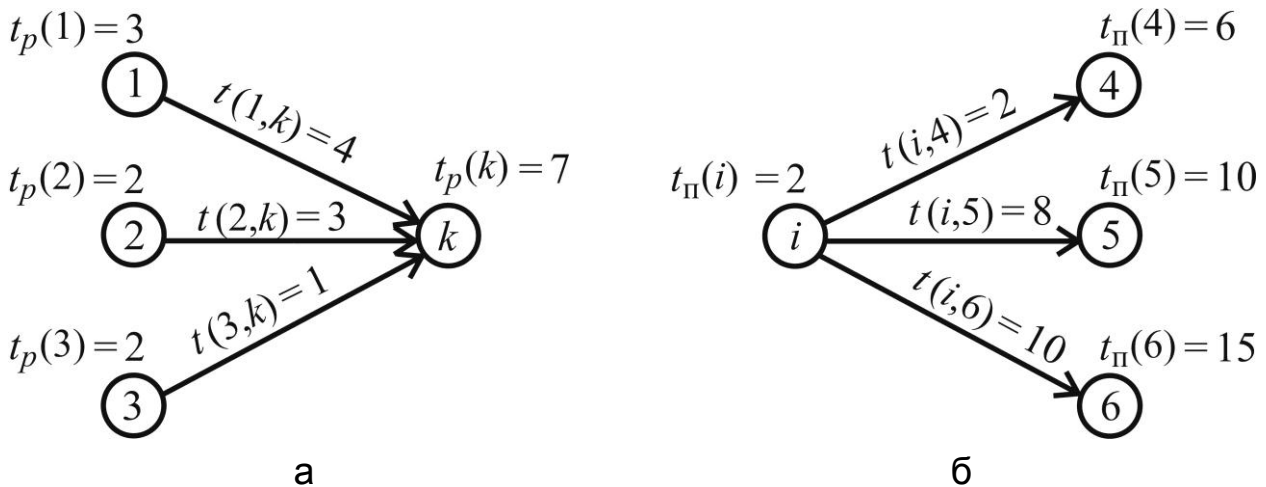


Рис. 2.7. Визначення строків звершення події:
а) раннього; б) пізнього

Резервом часу $R(i)$ події i називають різницю між пізнім і раннім строками звершення цієї події:

$$R(i) = t_{\Pi}(i) - t_p(i). \quad (2.7)$$

Резерв $R(i)$ показує, на який найбільший можливий час можна затримати звершення події i за умови, що це не спричинить затримки завершення всього проєкту.

Звичайно, настання початкової та кінцевої подій на СГ не може затримуватися, тому $R(0) = R(n) = 0$, тобто:

$$t_p(0) = t_{\Pi}(0), \quad t_p(n) = t_{\Pi}(n). \quad (2.8)$$

Із нульовим резервом часу можуть бути й проміжні події; саме вони визначають найбільш відповідальні події та роботи на СГ, стежачи за виконанням яких можна тримати в полі зору хід виконання всього проєкту.

Мінімальний час виконання всіх робіт проєкту називають **критичним строком (часом)** $t_{kp.}$. Певна річ, його визначає ранній строк $t_p(n)$ звершення кінцевої події (а згідно з (2.8) – і пізній строк $t_{\Pi}(n)$), тобто:

$$t_{kp.} = t_p(n) = t_{\Pi}(n). \quad (2.9)$$

Орланцюг (шлях) μ_{0n} довжиною $t_{kp.}$, який проходить через вершини з нульовим резервом часу, називають **критичним шляхом** $\mu_{kp.}$ на СГ:

$$\mu_{0n} = \mu_{kp.} \iff l(\mu_{0n}) = t_{kp.} \quad (2.10)$$

Отже, критичний шлях – це найдовший у часі орланцюг робіт, які виконують від початкової до завершальної події. Усі роботи й події, які лежать на критичному шляху, теж називають **критичними**, решту робіт і подій на СГ – **некритичними**.

Тепер, підсумовуючи розглянуте, розв'яжемо **задачу знаходження критичного часу та критичного шляху на СГ** $\bar{G} = (X, U)$ заданого комплексу робіт.

Розв'язують цю задачу в чотири етапи.

I етап. Обчислення ранніх строків звершення подій: рухаються по СГ від початкової події до наступних у порядку зростання їхніх номерів і за ф-лою (2.7) знаходять $t_p(k) \quad \forall k \in I_x = \{0, 1, \dots, n\}$, причому:

$$t_{kp.} = t_p(n).$$

II етап. *Обчислення пізніх строків звершення подій:* рухаються по СГ від кінцевої події до попередніх у порядку спадання їхніх номерів і за ф-лою (2.8) знаходять $t_{\Pi}(i) \forall i \in I_x$, з урахуванням, що $t_{\Pi}(n) = t_p(n)$.

III етап. *Підрахунок резервів часу подій:* знаходять різницю між пізніми та ранніми строками подій, згідно з ф-лою (2.7).

IV етап. *Визначення критичного шляху:* рухаючись від x_0 до x_n , знаходять орланцюг довжини $t_{кр.}$, події якого мають нульовий резерв часу (критичні події); це визначає відповідні критичні роботи та критичний шлях загалом.

Приклад. Нехай структурно-часову таблицю комплексу робіт наведено в табл. 2.1. Тоді відповідний СГ можна зобразити, як показано на рис. 2.8. Оскільки робота u_4 спирається тільки на роботу u_1 , а роботи u_5, u_6 – на роботи u_1, u_2 , то, згідно із правилами побудови СГ, слід увести фіктивну роботу (1, 2).

Таблиця 2.1

Структурно-часова таблиця комплексу робіт

Перелік робіт u_j	Роботи, на які спирається u_j	Тривалість роботи u_j	Перелік відповідних подій x_i
u_1	–	3	x_1
u_2	–	6	x_2
u_3	–	4	x_3
u_4	u_1	5	x_4
u_5	u_1, u_2	1	x_3
u_6	u_1, u_2	9	x_4
u_7	u_3, u_5	6	x_4
u_8	u_4, u_6, u_7	8	x_5
u_9	u_3, u_5	5	x_5

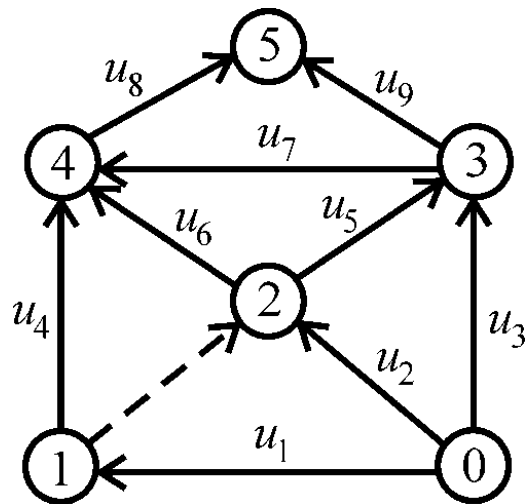


Рис. 2.8. СГ, побудований за структурно-часовою таблицею

Для зручності виконання етапів I – IV кожен подію зобразимо кружечком, поділеним двома діаметрами на чотири сектори. У верхньому секторі запишемо номер i події, у лівому в міру обчислення будемо записувати ранній строк $t_p(i)$ звершення події i , у правому – пізній строк $t_{\Pi}(i)$, у нижньому – резерв $R(i)$ часу події (рис. 2.9). Результати підрахунків для визначення $t_{кр.}$ і $\mu_{кр.}$ подано на СГ, зображеному на рис. 2.10.

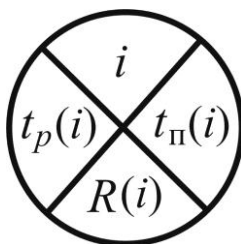


Рис. 2.9. Зображення вершин (подій) СГ

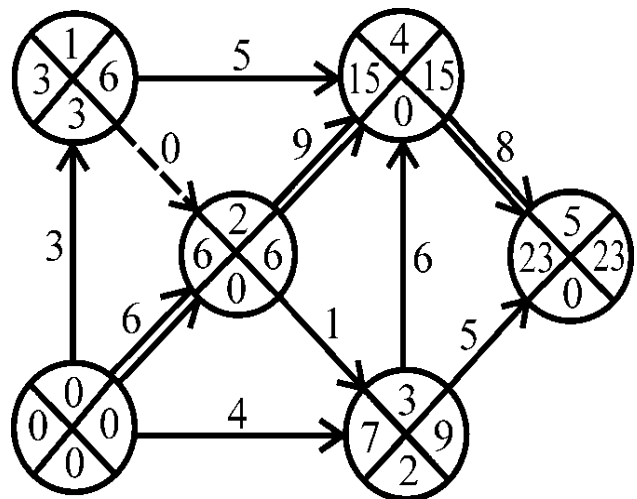


Рис. 2.10. Результати підрахунків для визначення $t_{кр.}$, $\mu_{кр.}$

Тривалості робіт візьмемо із табл. 2.1. За ф-лою (2.5) за вхідними дугами знаходимо ранні строки звершення подій:

$$\begin{aligned}
 t_p(1) &= t_p(0) + t(0,1) = 0 + 3 = 3; \\
 t_p(2) &= \max\{ t_p(0) + t(0,2), t_p(1) + t(1,2) \} = \max\{ 0 + 6, 3 + 0 \} = 6; \\
 t_p(3) &= \max\{ t_p(0) + t(0,3), t_p(2) + t(2,3) \} = \max\{ 0 + 4, 6 + 1 \} = 7; \\
 t_p(4) &= \max\{ t_p(1) + t(1,4), t_p(2) + t(2,4), t_p(3) + t(3,4) \} = \\
 &= \max\{ 3 + 5, 6 + 9, 7 + 6 \} = 15; \\
 t_p(5) &= \max\{ t_p(3) + t(3,5), t_p(4) + t(4,5) \} = \max\{ 7 + 5, 15 + 8 \} = 23 = t_{kp}.
 \end{aligned}$$

За ф-лою (2.6) за вихідними дугами обчислимо пізні строки:

$$\begin{aligned}
 t_{\Pi}(5) &= t_p(5) = 23; \\
 t_{\Pi}(4) &= t_{\Pi}(5) - t(4,5) = 23 - 8 = 15; \\
 t_{\Pi}(3) &= \min\{ t_{\Pi}(4) - t(3,4), t_{\Pi}(5) - t(3,5) \} = \min\{ 15 - 6, 23 - 8 \} = 9; \\
 t_{\Pi}(2) &= \min\{ t_{\Pi}(3) - t(2,3), t_{\Pi}(4) - t(2,4) \} = \min\{ 9 - 1, 15 - 9 \} = 6; \\
 t_{\Pi}(1) &= \min\{ t_{\Pi}(2) - t(1,2), t_{\Pi}(4) - t(1,4) \} = \min\{ 6 - 0, 15 - 5 \} = 6; \\
 t_{\Pi}(0) &= \min\{ t_{\Pi}(1) - t(0,1), t_{\Pi}(2) - t(0,2), t_{\Pi}(3) - t(0,3) \} = \\
 &= \min\{ 6 - 3, 6 - 6, 9 - 4 \} = 0.
 \end{aligned}$$

Знаходимо резерви часу подій:

$$R(0) = 0, R(1) = 3, R(2) = 0, R(3) = 2, R(4) = 0, R(5) = 0.$$

Критичному часу $t_{kp.} = 23$ відповідає критичний шлях, зображений подвійною ламаною: $\mu_{kp.} = (0 \rightarrow 2 \rightarrow 4 \rightarrow 5)$.

Зазначимо, що в усіх розрахунках фіктивну роботу враховують разом із реальними. У розглянутому прикладі критичний шлях виявився єдиним, але їх може бути декілька. Критичний шлях може містити й фіктивні роботи.

За умови невеликих комплексів робіт розрахунки виконують уручну, за значних (понад 500 подій) – на ЕОМ.

2.4. Транспортні мережі

2.4.1. Транспортні мережі: основні означення, знаходження повного потоку

Транспортною мережею (ТМ) називають оргграф $\bar{G} = (X, U)$ без контурів і петель – такий, у якому:

1) є єдина вершина $x_0 \in X$ тільки з додатно інцидентними дугами (x_0 – **вхід ТМ**);

2) є єдина вершина $x_n = z \in X$ тільки з від'ємно інцидентними дугами (z – **вихід ТМ**);

3) кожній дузі $u \in U$ відповідає ціле невід'ємне число $c(u)$, назване **пропускною спроможністю дуги**.

Якщо $x \in X$ – довільна вершина ТМ, то позначимо множину дуг, які заходять у x через U_x^- , а через U_x^+ – множину дуг, які виходять із x , тобто $U_x^-(U_x^+)$ – множина дуг від'ємно (додатно) інцидентних вершині x .

Потоком по дузі $u = (x, y)$, $x, y \in X$, або **дуговим потоком**, називають числову цілозначну функцію дуги $\varphi(u)$, яка задовольняє умови:

$$1) 0 \leq \varphi(u) \leq c(u) \quad \forall u \in U \text{ (умова обмеженості);} \quad (2.11)$$

$$2) \sum_{u \in U_x^-} \varphi(u) = \sum_{u \in U_x^+} \varphi(u), \quad x \neq x_0, \quad x \neq z \text{ (умова балансу).} \quad (2.12)$$

У фізичній інтерпретації $\varphi(u)$ – це сукупність однорідних об'єктів, які треба переслати з однієї вершини графа до іншої по його дузі, або кількість речовини, що проходить (за одиницю часу) по дузі $u = (x, y)$ від вершини x до вершини y . Ліва (права) частина у ф-лі (2.12) – це сума потоків до (від) вершини x .

Згідно з ф-лою (2.11), $\varphi(u)$ (кількість речовини, сукупність об'єктів) не може перевищувати пропускної спроможності дуги $c(u)$. Згідно з ф-лою (2.12), у кожній вершині x , відмінній від входу x_0 і виходу z , кількість речовини, що надходить, дорівнює кількості речовини, що виходить.

Отже, речовина не може накопичуватися в жодній із вершин ТМ, за винятком входу і виходу. Це означає, що сумарний потік на множині дуг U_z^- (**вхідний потік**) і сумарний потік на множині дуг $U_{x_0}^+$ (**вихідний потік**) дорівнюють один одному.

Потоком (на) ТМ φ_z називають суму дугових потоків від вершини x_0 , або, що те саме, суму дугових потоків до вершини z :

$$\varphi_z = \sum_{u \in U_{x_0}^+} \varphi(u) = \sum_{u \in U_z^-} \varphi(u). \quad (2.13)$$

Дугу u називають **насиченою**, якщо $\varphi(u) = c(u)$, тобто дуговий потік дорівнює пропускній спроможності цієї дуги. Дугу u називають **ненасиченою**, якщо $\varphi(u) < c(u)$. Якщо $\varphi(u) > 0$ ($\varphi(u) = 0$), то кажуть, що дуга u **навантажена** (**ненавантажена**).

Потік φ_z^Π називають **повним**, якщо кожний шлях μ від x_0 до z містить принаймні одну насичену дугу (такий шлях μ також називають **повним**); інакше φ_z – **неповний потік**, μ – **неповний шлях**.

Якщо на ТМ указано потік φ_z і дугові потоки $\varphi(u) \forall u \in U$, то кажуть, що *здійснено розподіл потоку на ТМ* (за дугами).

Тепер розв'яжемо **задачу побудови (знаходження) повного потоку** φ_z^Π на заданій ТМ із фіксованими пропускними спроможностями її дуг. Розв'язують цю задачу у два етапи.

I етап. Побудова довільного (початкового) потоку φ_z^0 : вибирають деякий (переважно, невеликий) потік i , згідно з умовами обмеженості та балансу, здійснюють його розподіл за дугами. Побудову початкового потоку регламентують тільки умовами (2.11), (2.12), жодних інших правил не використовують, а тому φ_z^0 визначають неоднозначно.

II етап. Побудова повних шляхів від x_0 до z : розглядають кожний неповний шлях μ від x_0 до z і для всіх дуг такого шляху підраховують різниці $c(u) - \varphi(u)$. Нехай r – найменша із цих різниць:

$$r = \min_{u \in \mu} \{ c(u) - \varphi(u) \},$$

а потім збільшують на r дугові потоки всіх дуг шляху μ , унаслідок чого кожний неповний шлях перетворюється на повний, а φ_z^0 – на повний потік φ_z^{Π} . Як і φ_z^0 , повний потік визначають неоднозначно.

Приклад. Розглянемо п'ятивершинну ТМ $\bar{G} = (X, U)$, x_0 – вхід, $x_4 = z$ – вихід (рис. 2.11а).

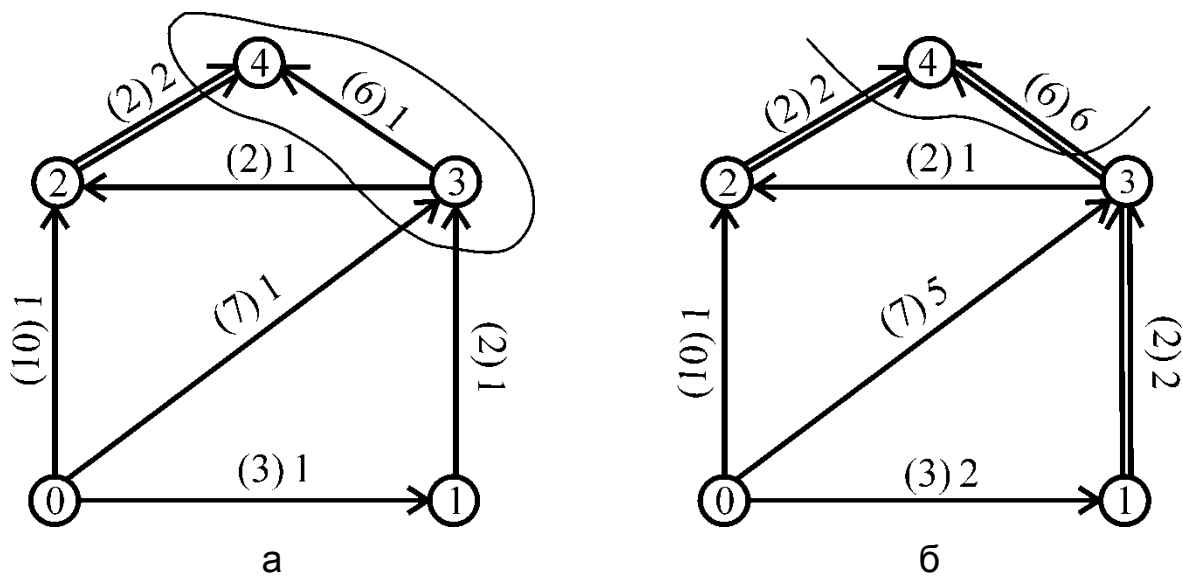


Рис. 2.11. Транспортна мережа:

- а) розподіл початкового потоку та розріз на ТМ;
- б) побудова повного потоку та мінімальний розріз на ТМ

Пропускні спроможності $c(u)$ вкажемо у круглих дужках приблизно посередині дуг ТМ, а дугові потоки $\varphi(u)$ будемо записувати праворуч від них; насичені дуги позначати подвійними лініями.

Візьмемо початковий потік $\varphi_z^0 = 3$ і здійснимо розподіл його за дугами. Покладемо, наприклад, $\varphi(0,1) = \varphi(0,2) = \varphi(0,3) = 1$.

Далі, ураховуючи умови обмеженості й балансу, дістанемо решту дугових потоків. Унаслідок розподілу виявилось, що дуга (2,4) насичена ($\varphi(2,4) = c(2,4) = 2$), але потік φ_z^0 загалом не буде повним: шляхи $\mu_1 = (0 \rightarrow 2 \rightarrow 4)$, $\mu_2 = (0 \rightarrow 3 \rightarrow 2 \rightarrow 4)$, $\mu_3 = (0 \rightarrow 1 \rightarrow 3 \rightarrow 2 \rightarrow 4)$ – повні, а шляхи $\mu_4 = (0 \rightarrow 1 \rightarrow 3 \rightarrow 4)$, $\mu_5 = (0 \rightarrow 3 \rightarrow 4)$ – неповні.

Знайдемо на шляху μ_4 величину:

$$r = \min_{u \in \mu_4} \{ c(u) - \varphi(u) \} = \min\{ 3 - 1, 2 - 1, 6 - 1 \} = 1,$$

і збільшимо на $r = 1$ всі дугові потоки на цьому шляху, тоді дуга (1,3) стане насиченою (рис. 2.11б): $\varphi(1,3) = c(1,3) = 2$.

Після цього на шляху μ_5 обчислюємо r :

$$r = \min_{u \in \mu_5} \{ c(u) - \varphi(u) \} = \min\{ 7 - 1, 6 - 2 \} = 4,$$

і збільшуємо на $r = 4$ всі дугові потоки на μ_5 , тоді дуга (3,4) стане насиченою: $\varphi(3,4) = c(3,4) = 6$.

Побудований потік $\varphi_z^{\Pi} = 2 + 1 + 5 = 2 + 6 = 8$ є повним, оскільки кожний шлях від x_0 до z містить хоча б одну насичену дугу.

2.4.2. Розрізи на ТМ

Поділимо множину вершин X на дві підмножини A і $\bar{A} = X \setminus A$ так, що вхід і вихід ТМ будуть належати різним множинам: $x_0 \notin A$, $z \in A$. Сукупність дуг U_A , які з'єднують (сполучають) вершини з A з вершинами з \bar{A} або навпаки, називають **розрізом на ТМ**.

Якщо позначити через U_A^- (U_A^+) множину дуг від вершин з \bar{A} (A) до вершин з A (\bar{A}), то можна записати:

$$U_A = U_A^- \cup U_A^+. \quad (2.14)$$

Розріз на ТМ виділяють розімкненою лінією, яка перетинає всі дуги розрізу U_A , або зімкненою лінією, яка (до того ж) охоплює всі вершини множини A .

Наприклад, для ТМ на рис. 2.11а розріз U_A визначають множини: $A = \{3, 4\}$, $\bar{A} = \{0, 1, 2\}$, де $U_A^+ = \{(3, 2)\}$, $U_A^- = \{(0, 3), (1, 3), (2, 4)\}$, тоді $U_A = U_A^+ \cup U_A^- = \{(0, 3), (1, 3), (2, 4), (3, 2)\}$; на рис. 2.11б маємо: $A = \{4\}$, $\bar{A} = \{0, 1, 2, 3\}$, $U_A^+ = \emptyset$, $U_A^- = \{(2, 4), (3, 4)\}$, отже, $U_A = U_A^-$.

Дуги множини U_A^- (U_A^+) розрізу U_A називають **вхідними (вихідними) дугами розрізу**.

Величини

$$\varphi_A^- = \sum_{u \in U_A^-} \varphi(u), \quad \varphi_A^+ = \sum_{u \in U_A^+} \varphi(u)$$

називають, відповідно, **вхідним і вихідним потоками розрізу U_A** , а їхню різницю – **загальним потоком φ_A через розріз U_A** :

$$\varphi_A = \varphi_A^- - \varphi_A^+ = \sum_{u \in U_A^-} \varphi(u) - \sum_{u \in U_A^+} \varphi(u). \quad (2.15)$$

З означень потоку φ_z на ТМ і загального потоку φ_A через розріз U_A випливає (факт не дуже дивний, але важливий), що вони дорівнюють один одному:

$$\varphi_z = \varphi_A = \varphi_A^- - \varphi_A^+ \quad \forall A \subset X \quad (z \in A). \quad (2.16)$$

Дійсно, адже кожен об'єкт чи частка речовини, які рухаються від x_0 до z , обов'язково пройдуть по якій-небудь дузі розрізу, не накопичуючись (за умовою балансу) у жодній із проміжних вершин ТМ.

Пропускною спроможністю $c(U_A)$ розрізу U_A називають суму пропускних спроможностей його вхідних дуг:

$$c(U_A) = \sum_{u \in U_A^-} c(u). \quad (2.17)$$

Легко встановити зв'язок між $c(U_A)$ і φ_z , а саме: пропускна спроможність будь-якого розрізу не менша від потоку на ТМ, тобто:

$$c(U_A) \geq \varphi_z. \quad (2.18)$$

Дійсно, на підставі ф-ли (2.16) маємо:

$$\varphi_A^- = \sum_{u \in U_A^-} \varphi(u) = \varphi_z + \varphi_A^+.$$

Замінімо $\varphi(u)$ на $c(u)$, тоді й дістанемо ф-лу (2.18), оскільки $\varphi(u) \leq c(u)$.

Розріз, який має найменшу пропускну спроможність серед усіх розрізів заданої ТМ, називають **мінімальним розрізом**.

Теорема 2.1 (про мінімальні розрізи та максимальні потоки): якщо для деякого потоку φ_z на ТМ і деякого розрізу U_B має місце рівність $\varphi_z = c(U_B)$, то: 1) розріз U_B мінімальний; 2) потік φ_z максимальний.

Доведення ґрунтується на зв'язку між пропускну спроможністю будь-якого розрізу і потоком на ТМ (ф-ла (2.18)), а саме: $\varphi_z \leq c(U_A) \forall A \subset X$ ($z \in A$).

1. За умовою теореми $\varphi_z = c(U_B)$, тоді

$$c(U_B) \leq c(U_A) \Rightarrow c(U_B) = \min_A c(U_A),$$

де мінімум беруть за всіма множинами A , які визначають усі можливі розрізи на ТМ.

2. Оскільки φ_z не перевищує пропускну спроможності будь-якого розрізу U_A , то він не може бути й більшим від мінімального розрізу:

$$\varphi_z \leq c(U_A) \Rightarrow \varphi_z \leq \min_A c(U_A) = c(U_B).$$

Отже, для максимального потоку φ_z^* маємо:

$$\varphi_z^* = c(U_B) = \min_A c(U_A). \quad (2.19)$$

Наслідок. Максимальний потік φ_z^* на ТМ визначають розрізом U_B , у якого: 1) усі вхідні дуги насичені: $\varphi(u) = c(u) \forall u \in U_B^-$; 2) вихідних дуг немає: $U_B^+ = \emptyset$.

Доведення. На підставі ф-ли (2.19), означення пропускної спроможності розрізу (ф-ла (2.17)) та означення загального потоку через розріз (ф-ла (2.15)) маємо:

$$\varphi_z^* = c(U_B) = \sum_{u \in U_B^-} c(u) = \sum_{u \in U_B^-} \varphi(u) - \sum_{u \in U_B^+} \varphi(u) = \varphi_B,$$

де $\varphi(u) = c(u) \quad \forall u \in U_B^-, U_B^+ = \emptyset$.

Так, *наприклад*, на рис. 2.11б розріз $U_A = \{(2,4), (3,4)\}$ визначає максимальний потік $\varphi_z^* = 6$, оскільки він містить тільки вхідні дуги (2,4), (3,4) та обидві вони – насичені.

Доведена теорема й наслідок із неї є базовими положеннями для розв'язання задачі про знаходження максимального потоку на ТМ.

2.4.3. Задача знаходження максимального потоку на ТМ.

Алгоритм Форда – Фалкерсона

Постановка задачі: за заданої конфігурації ТМ і відомих пропускних спроможностей її дуг знайдіть максимальний потік на ТМ, а також розподіл цього потоку за дугами.

Розглянемо доволі поширений метод розв'язання цієї задачі, основними складовими частинами якого є дві процедури, а саме: індексація вершин і зміна потоку. Їх виконують одна за одною доти, поки не дістануть максимальний потік.

1. Індиксація вершин. Нехай здійснено розподіл деякого потоку φ_z за дугами ТМ, серед яких можуть бути як навантажені (зокрема насичені), так і ненавантажені (порожні).

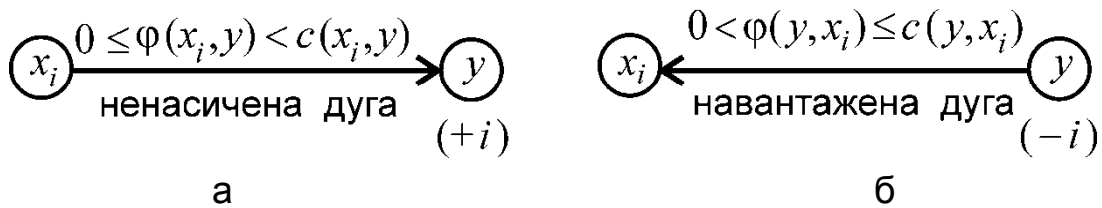
Під **індексацією (зважуванням)** вершин ТМ розуміють процес приписування їм вершинам $x_0, x_1, \dots, x_n = z$ чисел таким чином:

вершині x_0 приписують індекс (вагу) 0, або кажуть, що *вершину* x_0 *позначено* індексом 0;

якщо вершину з номером i , тобто x_i , уже зважено, то:

а) усі непозначені вершини y , до яких із x_i ідуть *ненасичені* дуги, позначають індексом $(+i)$ (рис. 2.12а);

б) усі непозначені вершини y , із яких до x_i ідуть *навантажені* дуги, позначають індексом $(-i)$ (рис. 2.12б).



**Рис. 2.12. Зважування вершин ТМ:
а) до яких входять дуги; б) із яких виходять дуги**

Якщо з непозначеної вершини y виходять дві або більше дуг до позначених вершин, то вершині y можна приписати індекс за будь-якою із цих позначених вершин (на результат розв'язання задачі це не вплине). Зазначимо також, що потік нульової величини ($\varphi(x, y) = 0$) можна вважати спрямованим як від x до y , так і від y до x (під час індексації вершин необхідно розглядати обидва можливі напрямки).

Із процедури зважування вершин ТМ випливає, що вершині y (за умови вже зваженої вершини x_i) індексу приписати не можна, якщо дуга (x_i, y) *насичена* або дуга (y, x_i) *ненавантажена* (порожня).

Припустимо, що, унаслідок процесу індексації вершин, вихід ТМ z разом із деякими іншими вершинами (або без них) виявився непроіндексованим. Згідно зі сказаним у попередньому абзаці, це означає, що:

1) розріз, який визначає множина непозначених вершин, має тільки насичені вхідні дуги;

2) вихідних дуг немає, оскільки інакше початкові вершини навантажених дуг розглянутого розрізу було б позначено від'ємними індексами.

Отже, згідно з наслідком із теореми 2.1, розріз, який визначає множина непозначених вершин ТМ, дає максимальний потік φ_z^* . Якщо ж вершину z у результаті індексації буде позначено, то потік на ТМ не є максимальним; отже, його можна збільшити.

2. Зміна потоку. Нехай вихід ТМ z у процесі зважування вершин набув якогось індексу. Тоді є шлях $\mu = \mu_{x_0 z}$ від x_0 до z , усі вершини

якого різні та, починаючи із другої, позначені (без урахування знака) номерами попередніх вершин.

Наприклад, якщо $\mu = (x_0 \rightarrow x_2 \rightarrow x_4 \rightarrow x_5 = z)$, або (за номерами вершин) $\mu = (0 \rightarrow 2 \rightarrow 4 \rightarrow 5)$, то кортеж I_μ індексів вершин (із точністю до знака) на цьому шляху буде мати такий вигляд: $I_\mu = (0, 0, 2, 4)$.

На шляху μ можуть бути дуги, орієнтація яких збігається або не збігається з напрямком руху по μ (від x_0 до z). Дугу $u \in U$, орієнтовану в напрямку (проти напрямку) руху по μ , називають **попутною (зустрічною)**, відповідну обставину будемо позначати символом $u \uparrow \uparrow \mu$ ($u \uparrow \downarrow \mu$). Потіки на цих дугах теж називати **попутними, або зустрічними**.

Згідно із процедурою індексації, кінцеві вершини попутних ненасичених дуг набувають додатних індексів, а початкові вершини навантажених зустрічних дуг – від'ємних індексів (див. рис. 2.12).

Зміну потоку на ТМ у бік його збільшення виконують так: знаходять шлях μ від x_0 до z (із проіндексованими, як показано раніше, вершинами) та дугові потоки $\varphi(u)$ на ньому замінюють новими:

$$\varphi'(u) = \begin{cases} \varphi(u) + 1, & \text{якщо } u \uparrow \uparrow \mu, \\ \varphi(u) - 1, & \text{якщо } u \uparrow \downarrow \mu. \end{cases} \quad (2.20)$$

Тобто, попутні (зустрічні) дугові потоки збільшують (зменшують) на одиницю, унаслідок чого дістають новий потік на ТМ: $\varphi'_z = \varphi_z + 1$. Після кожної зміни потоку повертаються до індексації вершин, аж поки вихід z неможливо буде позначити.

Потік φ_z можна (за певних умов) одразу збільшити не на одиницю, а на величину $r > 1$, яку визначають потоками на дугах та їхніми пропускними спроможностями. Так, згідно з ф-лою (2.20), зустрічний дуговий потік залишається зустрічним, якщо $\varphi(u) > 1$, або стає нульовим, якщо $\varphi(u) = 1$. В останньому випадку за наступної зміни потоку зустрічну раніше дугу можна переорієнтувати на попутну, тобто зміниться напрямок

її дугового потоку й по ній надалі можна пропустити попутний потік, не менший від її пропускної спроможності.

Зважаючи на це, якщо для попутних дуг обчислити мінімальну різницю між $c(u)$ і $\varphi(u)$:

$$r^+ = \min_{u \uparrow \uparrow \mu} \{c(u) - \varphi(u)\}, \quad (2.21)$$

а для зустрічних дуг знайти мінімальний потік:

$$r^- = \min_{u \uparrow \downarrow \mu} \{\varphi(u)\}, \quad (2.22)$$

то попутні (зустрічні) дугові потоки можна збільшити (зменшити) на величину r : $r = \min\{r^+, r^-\}$, тобто:

$$\varphi'(u) = \begin{cases} \varphi(u) + r, & \text{якщо } u \uparrow \uparrow \mu, \\ \varphi(u) - r, & \text{якщо } u \uparrow \downarrow \mu. \end{cases}$$

Унаслідок цього одна із зустрічних дуг може стати порожньою. Зрозуміло, що за відсутності зустрічних дуг $r = r^+$.

Такий підхід до зміни потоку сприяє значному зменшенню кількості індексацій, що скорочує розв'язання задачі загалом. Зазначимо також, що індексацію вершин і зміну потоку можна виконувати, відштовхуючись від розподілу довільного потоку φ_z (навіть нульового), але краще застосувати їх до повного потоку φ_z^{Π} , щоб зменшити кількість обчислень.

Підсумовуючи розглянуте, наведемо у вигляді орграфа (для наочності) **алгоритм Форда – Фалкерсона** – послідовність операцій, які треба виконати для знаходження максимального потоку на ТМ із розподілом його за дугами (рис. 2.13).

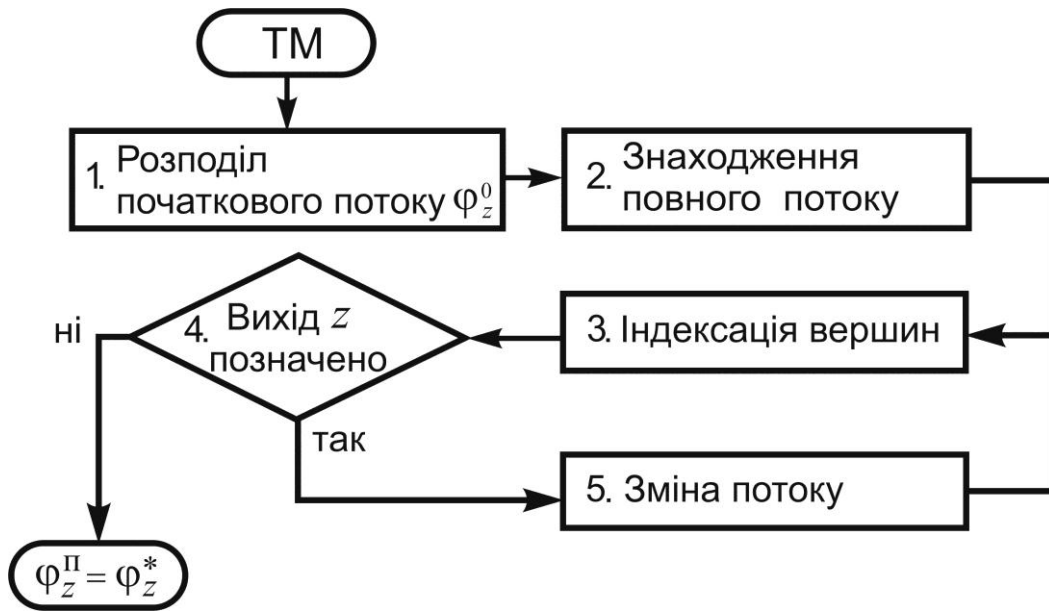


Рис. 2.13. Структурна схема алгоритму Форда – Фалкерсона

Насамкінець розглянемо реалізацію алгоритму на конкретній ТМ.

Задача 2.2. Знайдіть максимальний потік на транспортній мережі, зображеній на рис. 2.14а.

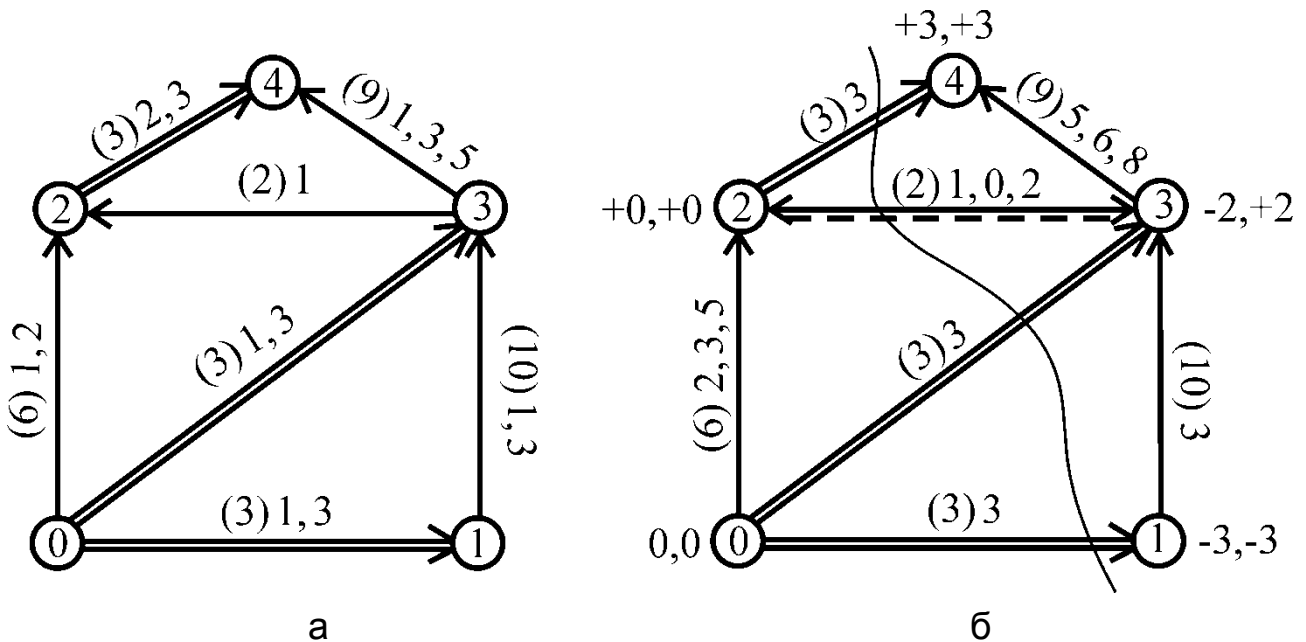


Рис. 2.14. Транспортна мережа:
а) побудова повного потоку; б) індексація вершин та знаходження максимального потоку на ТМ

Розв'язання. Будемо діяти, згідно зі схемою на рис. 2.13.

1. Виберемо початковий потік $\varphi_z^0 = 3$ (за кількістю дуг, що виходять з x_0) і візьмемо, що: $\varphi(0,1) = \varphi(0,2) = \varphi(0,3) = 1$.

Потім, урахувуючи умову балансу (2.12), маємо $\varphi(1,3) = 1$, візьмемо $\varphi(3,2) = \varphi(3,4) = 1$ (адже до вершини x_3 надходить дві одиниці потоку), тоді $\varphi(2,4) = 2$. Усі дуги виявилися навантаженими, але сформований потік не є повним (насичені дуги зовсім відсутні).

2. Для знаходження повного потоку перетворюємо кожний неповний шлях від x_0 до $x_4 = z$ на повний.

Розглядаємо шлях $\mu_1 = (0 \rightarrow 2 \rightarrow 4)$ і підраховуємо для його дуг величину $r = r^+$ (адже всі дуги на шляху μ_1 попутні):

$$\begin{aligned} r = r^+ &= \min_{u \uparrow \mu_1} \{c(u) - \varphi(u)\} = \min \{c(0,2) - \varphi(0,2), c(2,4) - \varphi(2,4)\} = \\ &= \min\{6 - 1, 3 - 2\} = 1. \end{aligned}$$

Збільшуємо дугові потоки всіх дуг шляху μ_1 на $r = 1$; нові дугові потоки записано через кому з попередніми: $\varphi(0,2) = 2$, $\varphi(2,4) = 3 = c(2,4)$, тобто дуга $(2,4)$ стала насиченою.

Аналогічно для шляху $\mu_2 = (0 \rightarrow 3 \rightarrow 4)$ дістанемо:

$$\begin{aligned} r = r^+ &= \min\{c(0,3) - \varphi(0,3), c(3,4) - \varphi(3,4)\} = \min\{3 - 1, 9 - 1\} = 2, \\ \varphi(0,3) &= 1 + 2 = 3 = c(0,3), \quad \varphi(3,4) = 1 + 2 = 3. \end{aligned}$$

Нарешті, для $\mu_3 = (0 \rightarrow 1 \rightarrow 3 \rightarrow 4)$ маємо:

$$\begin{aligned} r = r^+ &= \min\{3 - 1, 10 - 1, 9 - 3\} = 2, \\ \varphi(0,1) &= 1 + 2 = 3 = c(0,1), \quad \varphi(1,3) = 1 + 2 = 3, \quad \varphi(3,4) = 3 + 2 = 5. \end{aligned}$$

Тепер кожен шлях від входу до виходу ТМ є повним. Отже, побудовано повний потік: $\varphi_z^{\Pi} = 3 + 2 + 3 = 3 + 5 = 8$.

3. Індекссацію вершин (і подальші кроки) будемо виконувати, виходячи з повного потоку (див. рис. 2.146).

Припишемо початковій вершині індекс 0. Тоді можна проіндексувати вершину x_2 (надаємо їй індекс $(+0)$), адже дуга $(0,2)$ ненасичена; вершини x_1, x_3 , відштовхуючись від x_0 , проіндексувати не можна, оскільки дуги $(0,1), (0,3)$ насичені. Дуга $(3,2)$ навантажена й орієнтована в бік позначеної вершини x_2 , отже, вершина x_3 набуває індексу (-2) . На тій самій підставі вершині x_1 приписуємо індекс (-3) ; вершині ж $x_4 = z$ приписуємо індекс $(+3)$, оскільки дуга $(3,4)$ ненасичена та виходить із проіндексованої вершини x_3 .

4. Вихід ТМ виявився позначеним. Отже, потік $\varphi_z = 8$ хоча й повний, але не максимальний.

5. Для зміни потоку ТМ знаходимо шлях μ від x_0 до $z = x_4$, вершини якого позначено (із точністю до знака) номерами попередніх вершин:

$$\mu = (0 \rightarrow 2 \rightarrow 3 \rightarrow 4), \quad I_\mu = (0, 0, -2, +3).$$

На цьому шляху дві попутні дуги: $(0,2), (3,4)$, та одна дуга зустрічна – $(3,2)$. Підрахуємо, на скільки одиниць можна збільшити дугові потоки попутних дуг і зменшити зустрічної:

$$r^+ = \min \{6-2, 9-5\} = 4, \quad r^- = 1, \quad r = \min \{r^+, r^-\} = \min \{4, 1\} = 1.$$

Тоді нові дугові потоки будуть такими: $\varphi(0,2) = 2+1=3$, $\varphi(3,2) = 0$, $\varphi(3,4) = 5+1=6$, а потік на ТМ: $\varphi_z = 9$.

3'. Повертаємося до індекссації вершин. Дуга $(3,2)$ ненавантажена й за наступної індекссації вершина x_3 вже не набуде від'ємного індексу; проте як ненасичена (орієнтована від x_2 до x_3) дуга вона дозволяє проіндексувати вершину x_3 додатним індексом $(+2)$. Індеси вершин x_1, x_2, x_4 не зміняться (див. рис. 2.146).

4'. Вихід ТМ знову проіндексовано. Отже, потік $\varphi_z = 9$ хоча й повний, але не максимальний.

5'. Знову аналізуємо шлях $\mu = (0 \rightarrow 2 \rightarrow 3 \rightarrow 4)$ із кортежем індексів $I_\mu = (0, 0, +2, +3)$. Усі дуги на цьому шляху попутні, тому:

$$r = r^+ = \min\{6 - 3, 2 - 0, 9 - 6\} = 2.$$

Отже, збільшуємо дугові потоки на дві одиниці: $\varphi(0, 2) = 3 + 2 = 5$, $\varphi(2, 3) = 0 + 2 = 2 = c(2, 3)$, $\varphi(3, 4) = 6 + 2 = 8$, а $\varphi_z = 11$.

Дуга $(2, 3)$ стала насиченою (на рис. 2.14б це позначено двома лініями, одна з яких пунктирна, що відтворює переорієнтацію дуги $(3, 2)$). Отже, за наступної індексації проіндексувати можна лише вершини x_0 і x_2 .

Множина непроіндексованих вершин $B = \{x_1, x_3, x_4 = z\}$ визначає розріз на ТМ: $U_B = \{(0, 1), (0, 3), (2, 3), (2, 4)\}$, який не містить вихідних дуг, а всі вхідні дуги якого насичені. Він і дає максимальний потік φ_z^* із відповідним розподілом за дугами ТМ:

$$\varphi_z^* = 3 + 3 + 2 + 3 = 3 + 3 + 5 = 3 + 8 = 11.$$

2.5. Упорядкування вершин і дуг плоского орграфа

Реалізація алгоритмів розв'язання задач на орграфах і відповідні розрахунки помітно (і навіть значно) спрощуються, якщо їхні *елементи* – вершини, дуги – *упорядковано*, тобто на множинах вершин і дуг установлено певні відношення порядку.

Упорядкування вершин (дуг) орграфа $\overline{G} = (X, U)$ можна здійснити, спираючись на поняття додатної та від'ємної інцидентності вершин і дуг (див. п. 2.1). Водночас із двох суміжних вершин (рис. 2.15а) додатно (від'ємно) інцидентну дузі u вершину x_i (x_j) називають **попередньою** (**наступною**).

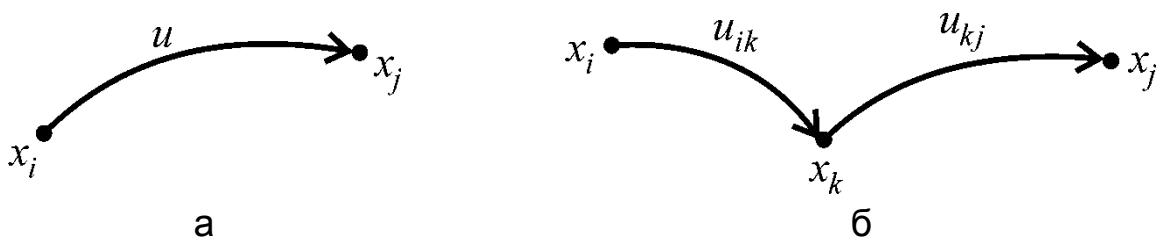


Рис. 2.15. Попередня і наступна:
а) вершини (x_i, x_j) ; **б) дуги** (u_{ik}, u_{kj})

Суміжні дуги орграфу називають **послідовними** (див. рис. 2.15б), якщо кінець однієї з них є початком другої. Серед двох послідовних дуг орграфу теж розрізняють **попередню** і **наступну**: попередня (наступна) дуга від'ємно (додатно) інцидентна їхній спільній вершині x_k . (Спробуйте дати означення попередньої та наступної вершин (дуг), не залучаючи поняття інцидентності.)

Надалі, для стислості записів і зручності формулювань у задачі упорядкування, замість слів "вершина" або "дуга" будемо вживати "елемент" (за принципом "два в одному").

Задача упорядкування елементів орграфу: для заданого орграфу \bar{G} здійснити поділ множин (X, U) його елементів на групи-підмножини, які задовольняють умови:

- 1) *перша група* містить елементи, які не мають попередніх;
- 2) *кожну чергову за номером групу* утворено з елементів, які є наступними щодо елементів з уже створених груп (згідно зі структурою заданого графа);
- 3) *остання група* містить елементи, які не ввійшли до створених груп і не мають наступних.

Аналізуючи другу умову, робимо висновок, що: 1) попередній і наступний елементи належать різним групам; 2) елементи кожної групи не мають попередніх у групах із більшим номером; 3) вершини (дуги) однієї й тієї самої групи не є суміжними (послідовними).

Установлено, що сформульована задача завжди має єдиний (із точністю до ізоморфізму) розв'язок для зв'язних орграфів без контурів і петель.

Є **графічний** і **матричний** способи упорядкування елементів орграфу. Розглянемо реалізацію графічного способу упорядкування вершин, відомого під назвою **алгоритм Фалкерсона**, на прикладі орграфу \bar{G} ,

зображеного на рис. 2.16а. На рис. 2.16б наведено граф \bar{G}' , ізоморфний (на якій підставі?) графові \bar{G} , з упорядкованими вершинами.

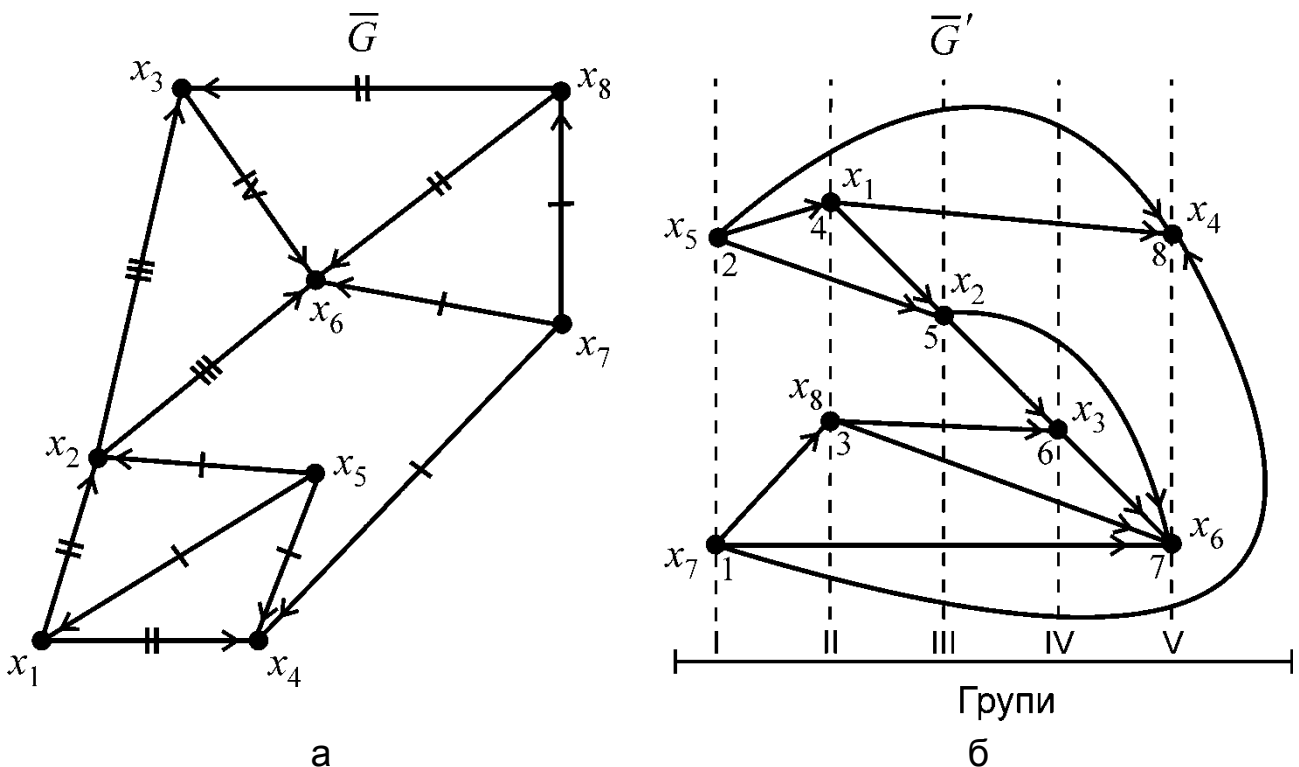


Рис. 2.16. Орграфи:
а) вихідний \bar{G} ; б) ізоморфний граф \bar{G}' ,
упорядкований за вершинами

Алгоритм Фалкерсона. Робимо візуальний аналіз заданого орграфу \bar{G} та:

1) знаходимо вершини графа, які не мають попередніх вершин, – у нас це вершини x_5, x_7 (простежте!); вони утворюють першу групу (група I на рис. 2.16б);

2) вилучаємо із графа (подумки) вершини групи I і дуги, що з них виходять, позначаючи їх рисками. Здобутий граф розглядаємо як початковий (вихідний), і в ньому також знаходимо вершини, які не мають попередніх, – у нас це вершини x_1, x_8 ; вони утворюють другу групу (група II на рис. 2.16б);

3) здійснюємо на кожному наступному кроці вилучення вершин попередньої групи і дуг, що з них виходять, позначаючи їх відповідною

кількістю рисок; *аналізуємо* здобутий граф на наявність вершин, до яких не входить жодна дуга, вони утворюють наступну групу вершин. Продовжуємо поділ множини вершин на групи – *групування вершин* – доти, поки не буде вичерпано всі вершини. У нас це x_2 (група III), x_3 (група IV). Залишилися ізольовані вершини x_4 , x_6 , які й становлять останню, п'яту групу (адже вони не мають наступних!);

4) *з'єднуємо* точки-вершини груп дугами, відповідно до заданого орграфу, і *нумеруємо* їх у натуральному порядку (див. рис. 2.16б).

Зауваження:

1) останню групу (x_4, x_6) можна було б сформувати зразу після першої, помітивши, що вершини x_4 , x_6 не мають наступних;

2) зображення дуг на орграфі \overline{G}' не обов'язково здійснювати після остаточного групування вершин: це можна робити між іншим у процесі створення груп;

3) якщо заданий орграф має порівняно багато вершин, то ті вершини, які послідовно вилучають у процесі групування, позначають (щоб не плутатися) галочками.

Алгоритм розв'язання задачі групування дуг принципово нічим не відрізняється від того, який було застосовано під час групування вершин, – достатньо лише замінити (погоджуючи з контекстом) термін "вершина" терміном "дуга". Не вдаючись до детального опису, на прикладі того самого графа \overline{G} , упорядкованого за вершинами, виконаємо групування дуг (рис. 2.17), а саме:

група I – (x_5, x_1) , (x_5, x_2) , (x_5, x_4) , (x_7, x_4) , (x_7, x_6) , (x_7, x_8) ;

група II – (x_1, x_2) , (x_1, x_4) , (x_8, x_3) , (x_8, x_6) ;

група III – (x_2, x_3) , (x_2, x_6) ;

група IV – (x_6) .

Можна помітити, що дуги, позначені рисками на графі \overline{G} в разі групування вершин, відповідають групам дуг на графі \overline{G}'' . А чи може бути інакше?

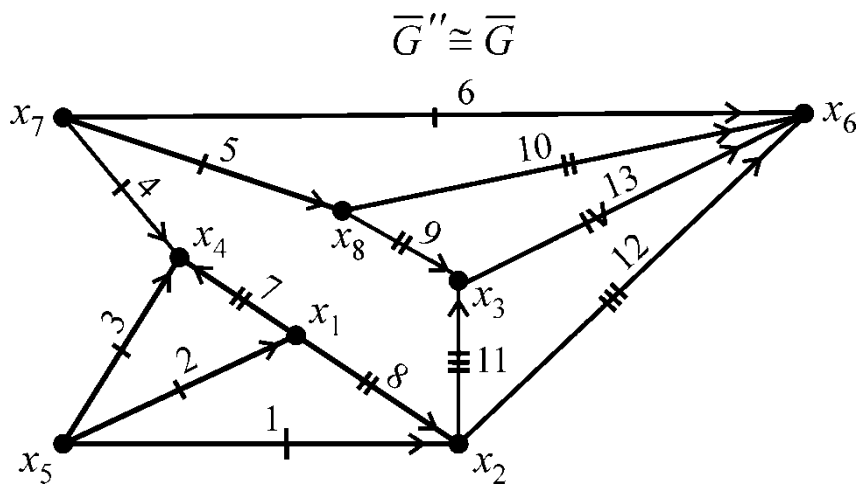


Рис. 2.17. Граф $\overline{G}'' \cong \overline{G}$, упорядкований за дугами

Упорядковані дуги нумерують у натуральному порядку (як на рис. 2.17) або позначають буквами з відповідними індексами.

У застосовному плані розглянутий граф \overline{G} можна тлумачити як транспортну мережу із двома входами (x_5, x_7) і двома виходами (x_4, x_6).

Наприкінці зазначимо, що роль графів в інформаційних технологіях важко переоцінити. Файлові структури на носіях пам'яті комп'ютерів, коди комп'ютерних програм, гіпертексти, інформаційні системи та бази даних – усім їм відповідають певні графи. Роботу із цими структурами зведено до побудови ефективних алгоритмів пошуку на графах або встановлення їхнього ізоморфізму.

Запитання для самоконтролю за засвоєнням матеріалу

1. Що називають орграфом, дугою, шляхом, контуром ОГ?
2. Які два види інцидентності між вершиною та дугою (і навпаки) розрізняють на ОГ?
3. Які два види паралельності дуг розрізняють на ОГ?
4. Які є способи задання ОГ і в чому вони полягають?
5. Що таке "основа" ОГ і який ОГ називають сильнозв'язним (слабкозв'язним, односторонньо зв'язним, порожнім, тривіальним)?
6. Як визначають матрицю інцидентностей (суміжності вершин) ОГ?
7. Що називають матрицею досяжності та контрдосяжності ОГ?
8. Що називають графом конденсації ОГ і як його побудувати?
9. Що називають сітковим графіком комплексу робіт (проєкту)?

10. Що називають раннім (пізнім) строком звершення події, резервом часу, критичним часом, критичним шляхом на СГ?
11. На які три категорії поділяють множину всіх стадій комплексу робіт будь-якого проєкту?
12. Як називають задачі, у яких ставлять питання про знаходження мінімального часу виконання всіх робіт комплексу (проєкту)?
13. Що має передувати (щодо відомостей) побудові СГ?
14. У чому полягають найважливіші правила побудови СГ?
15. Як формулюють постановку задачі про знаходження критичного часу та критичного шляху на СГ?
16. Із яких етапів складається розв'язання задачі про критичний час і критичний шлях на СГ?
17. Чи є такі СГ, на яких можна вказати (знайти) декілька:
а) значень критичного часу; б) критичних шляхів?
18. Чи є такі СГ, у яких: а) усі роботи (події) є критичними; б) резерви часу всіх подій однакові та відмінні від нуля; в) резерви часу всіх подій однакові й дорівнюють нулю?
19. Що називають транспортною мережею?
20. Які умови (згідно з означенням) мають задовольняти дуговий потік, насичені (ненасичені, навантажені, ненавантажені) дуги на ТМ?
21. Що називають потоком і повним потоком на ТМ та у якому разі кажуть, що здійснено їхній розподіл за дугами?
22. Як розв'язують задачу про знаходження повного потоку на ТМ?
23. Що розуміють під розрізом на ТМ і яку умову має задовольняти мінімальний розріз?
24. Які умови щодо дуг розрізу ТМ визначають максимальний потік?
25. Що розуміють під індексацією (зважуванням) вершин ТМ і як її виконують?
26. Який висновок слід зробити в разі, коли вихід ТМ виявився непроіндексованим?
27. Як здійснюють зміну потоку (у бік його збільшення) на ТМ зі зваженими вершинами?
28. У чому полягає алгоритм Форда – Фалкерсона знаходження максимального потоку на ТМ?
29. Що розуміють під упорядкуванням вершин орграфа, упорядкуванням дуг орграфа?
30. У чому полягає алгоритм Фалкерсона упорядкування вершин і дуг орграфа? Опишіть і проілюструйте його застосування на прикладі.

3. Варіанти задач самостійної контрольної роботи

Задача 1. Для графа $G = (X, U)$, заданого теоретико-множинним способом: 1) надайте його геометричне зображення; 2) побудуйте матриці інцидентностей та суміжності вершин.

$$1.1. G = (\{x_1, x_2, x_3, x_4, x_5, x_6\}, \{\{x_1, x_2\}, \{x_1, x_5\}, \{x_2, x_4\}, \{x_3, x_5\}, \{x_3, x_6\}, \{x_4, x_5\}, \{x_5, x_6\}\}).$$

$$1.2. G = (\{x_1, x_2, x_3, x_4, x_5, x_6\}, \{\{x_2, x_1\}, \{x_1, x_4\}, \{x_2, x_3\}, \{x_2, x_4\}, \{x_2, x_5\}, \{x_3, x_6\}, \{x_5, x_6\}\}).$$

$$1.3. G = (\{x_1, x_2, x_3, x_4, x_5\}, \{\{x_1, x_2\}, \{x_4, x_1\}, \{x_2, x_4\}, \{x_3, x_4\}, \{x_3, x_5\}, \{x_4, x_5\}, \{x_1, x_3\}\}).$$

$$1.4. G = (\{x_1, x_2, x_3, x_4, x_5\}, \{\{x_4, x_1\}, \{x_1, x_5\}, \{x_2, x_3\}, \{x_2, x_5\}, \{x_3, x_5\}, \{x_4, x_5\}, \{x_1, x_2\}\}).$$

$$1.5. G = (\{x_1, x_2, x_3, x_4, x_5\}, \{\{x_1, x_2\}, \{x_1, x_4\}, \{x_2, x_3\}, \{x_2, x_4\}, \{x_2, x_5\}, \{x_3, x_5\}, \{x_3, x_4\}\}).$$

$$1.6. G = (\{x_1, x_2, x_3, x_4, x_5, x_6\}, \{\{x_1, x_5\}, \{x_1, x_4\}, \{x_2, x_3\}, \{x_2, x_5\}, \{x_3, x_6\}, \{x_4, x_5\}, \{x_5, x_6\}\}).$$

$$1.7. G = (\{x_1, x_2, x_3, x_4, x_5, x_6\}, \{\{x_4, x_1\}, \{x_1, x_6\}, \{x_2, x_3\}, \{x_2, x_5\}, \{x_3, x_5\}, \{x_4, x_5\}, \{x_5, x_6\}\}).$$

$$1.8. G = (\{x_1, x_2, x_3, x_4, x_5, x_6\}, \{\{x_1, x_2\}, \{x_2, x_3\}, \{x_2, x_4\}, \{x_1, x_4\}, \{x_2, x_5\}, \{x_3, x_4\}, \{x_4, x_5\}\}).$$

$$1.9. G = (\{x_1, x_2, x_3, x_4, x_5\}, \{\{x_1, x_5\}, \{x_1, x_3\}, \{x_1, x_4\}, \{x_2, x_1\}, \{x_2, x_3\}, \{x_2, x_4\}, \{x_2, x_5\}\}).$$

$$1.10. G = (\{x_1, x_2, x_3, x_4, x_5, x_6\}, \{\{x_3, x_4\}, \{x_1, x_5\}, \{x_2, x_4\}, \{x_2, x_5\}, \{x_1, x_4\}, \{x_3, x_5\}, \{x_4, x_5\}\}).$$

$$1.11. G = (\{x_1, x_2, x_3, x_4, x_5\}, \{\{x_4, x_2\}, \{x_1, x_3\}, \{x_1, x_4\}, \{x_1, x_5\}, \{x_1, x_4\}, \{x_3, x_4\}, \{x_4, x_5\}\}).$$

$$1.12. G = (\{x_1, x_2, x_3, x_4, x_5\}, \{\{x_5, x_2\}, \{x_1, x_5\}, \{x_2, x_3\}, \{x_2, x_4\}, \{x_2, x_1\}, \{x_3, x_4\}, \{x_1, x_3\}\}).$$

$$1.13. G = (\{x_1, x_2, x_3, x_4, x_5, x_6\}, \{\{x_1, x_2\}, \{x_1, x_4\}, \{x_2, x_3\}, \{x_2, x_5\}, \{x_2, x_6\}, \{x_3, x_6\}, \{x_4, x_5\}\}) .$$

$$1.14. G = (\{x_1, x_2, x_3, x_4, x_5\}, \{\{x_1, x_2\}, \{x_1, x_4\}, \{x_2, x_3\}, \{x_2, x_4\}, \{x_2, x_5\}, \{x_4, x_5\}, \{x_1, x_3\}\}) .$$

$$1.15. G = (\{x_1, x_2, x_3, x_4, x_5\}, \{\{x_1, x_2\}, \{x_1, x_4\}, \{x_2, x_3\}, \{x_2, x_4\}, \{x_2, x_5\}, \{x_3, x_4\}, \{x_4, x_5\}\}) .$$

$$1.16. G = (\{x_1, x_2, x_3, x_4, x_5, x_6\}, \{\{x_1, x_2\}, \{x_1, x_5\}, \{x_2, x_4\}, \{x_3, x_5\}, \{x_3, x_6\}, \{x_4, x_5\}, \{x_5, x_6\}\}) .$$

$$1.17. G = (\{x_1, x_2, x_3, x_4, x_5, x_6\}, \{\{x_1, x_2\}, \{x_1, x_4\}, \{x_2, x_3\}, \{x_2, x_4\}, \{x_2, x_5\}, \{x_3, x_6\}, \{x_5, x_6\}\}) .$$

$$1.18. G = (\{x_1, x_2, x_3, x_4, x_5\}, \{\{x_1, x_2\}, \{x_1, x_4\}, \{x_2, x_4\}, \{x_3, x_4\}, \{x_3, x_5\}, \{x_2, x_3\}, \{x_4, x_5\}\}) .$$

$$1.19. G = (\{x_1, x_2, x_3, x_4, x_5\}, \{\{x_1, x_4\}, \{x_1, x_5\}, \{x_2, x_3\}, \{x_2, x_5\}, \{x_2, x_1\}, \{x_3, x_5\}, \{x_4, x_5\}\}) .$$

$$1.20. G = (\{x_1, x_2, x_3, x_4, x_5\}, \{\{x_1, x_2\}, \{x_1, x_4\}, \{x_2, x_3\}, \{x_2, x_4\}, \{x_2, x_5\}, \{x_3, x_5\}, \{x_1, x_3\}\}) .$$

$$1.21. G = (\{x_1, x_2, x_3, x_4, x_5, x_6\}, \{\{x_5, x_6\}, \{x_4, x_5\}, \{x_3, x_6\}, \{x_2, x_5\}, \{x_2, x_3\}, \{x_1, x_5\}, \{x_1, x_4\}\}) .$$

$$1.22. G = (\{x_1, x_2, x_3, x_4, x_5, x_6\}, \{\{x_1, x_4\}, \{x_1, x_6\}, \{x_2, x_3\}, \{x_2, x_5\}, \{x_3, x_5\}, \{x_4, x_5\}, \{x_5, x_6\}\}) .$$

$$1.23. G = (\{x_1, x_2, x_3, x_4, x_5\}, \{\{x_1, x_2\}, \{x_2, x_3\}, \{x_1, x_4\}, \{x_2, x_4\}, \{x_2, x_5\}, \{x_1, x_3\}, \{x_4, x_5\}\}) .$$

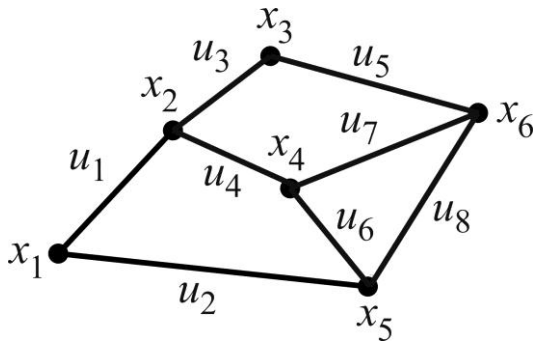
$$1.24. G = (\{x_1, x_2, x_3, x_4, x_5\}, \{\{x_1, x_3\}, \{x_1, x_2\}, \{x_1, x_4\}, \{x_1, x_5\}, \{x_2, x_3\}, \{x_2, x_4\}, \{x_2, x_5\}\}) .$$

$$1.25. G = (\{x_1, x_2, x_3, x_4, x_5\}, \{\{x_1, x_4\}, \{x_1, x_5\}, \{x_2, x_4\}, \{x_2, x_5\}, \{x_3, x_4\}, \{x_3, x_5\}, \{x_1, x_2\}, \{x_4, x_5\}\}) .$$

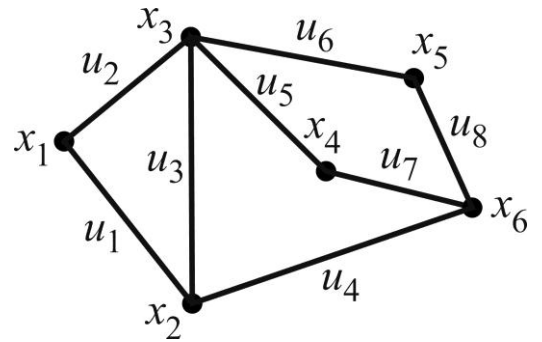
$$1.26. G = (\{x_1, x_2, x_3, x_4, x_5, x_6\}, \{\{x_1, x_2\}, \{x_1, x_4\}, \{x_2, x_3\}, \{x_2, x_4\}, \{x_2, x_5\}, \{x_3, x_4\}, \{x_4, x_5\}\}) .$$

Задача 2. За геометричним зображенням графа G необхідно:
 1) знайти його теоретико-множинне подання; 2) побудувати матриці інцидентностей та суміжності вершин; 3) знайти матрицю відстаней, ексцентриситети вершин, діаметр, радіус, центр та цикломатичне число графа;
 4) указати будь-який ланцюг, простий ланцюг, цикл, простий цикл і їхні довжини; 5) визначити степені всіх вершин і переконатися, що їхня сума дорівнює подвоєній кількості ребер; 6) з'ясувати, чи є граф ейлеровим.

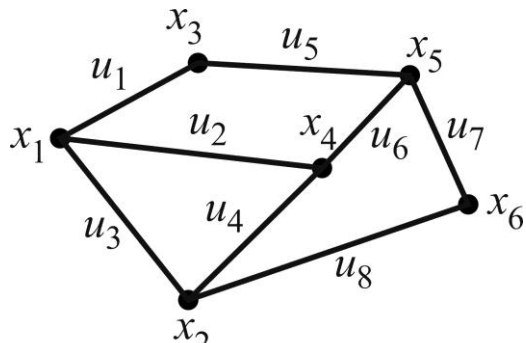
2.1.



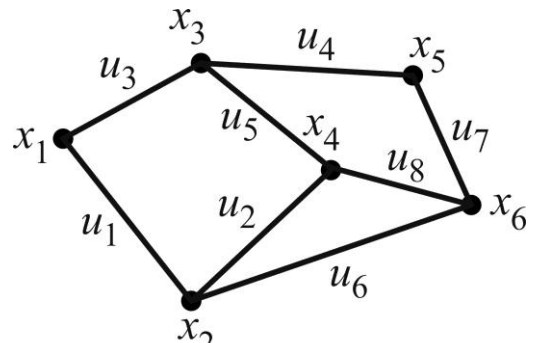
2.2.



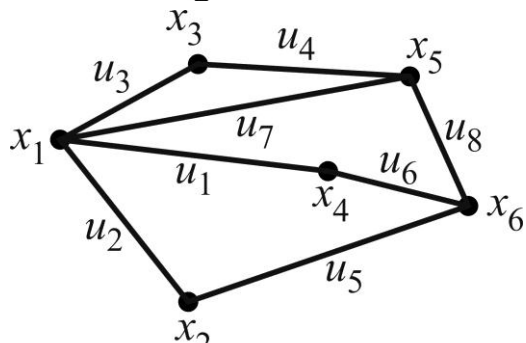
2.3.



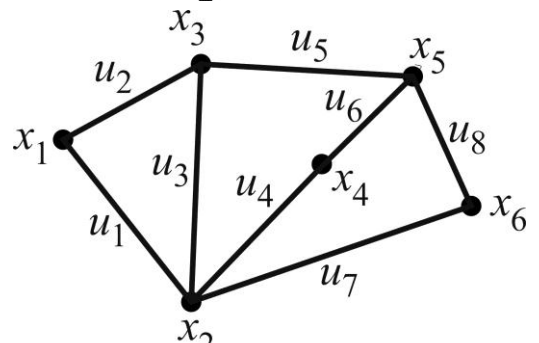
2.4.



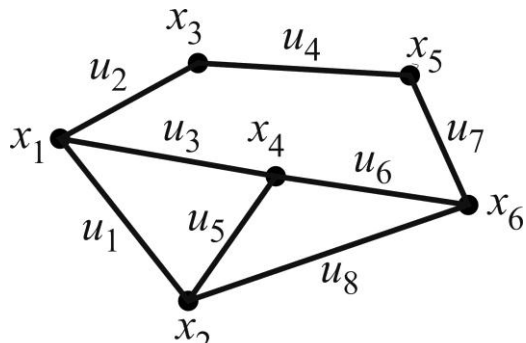
2.5.



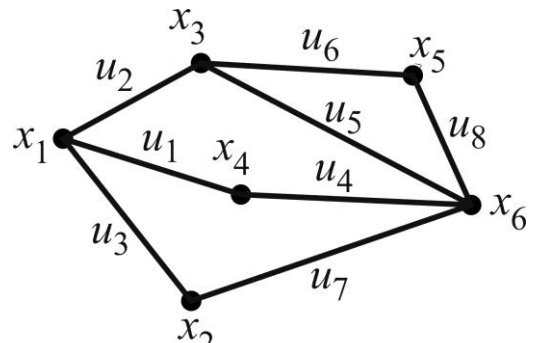
2.6.



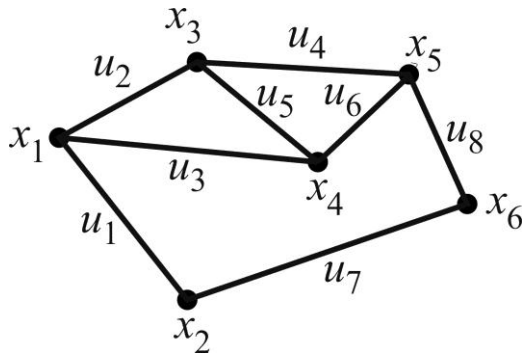
2.7.



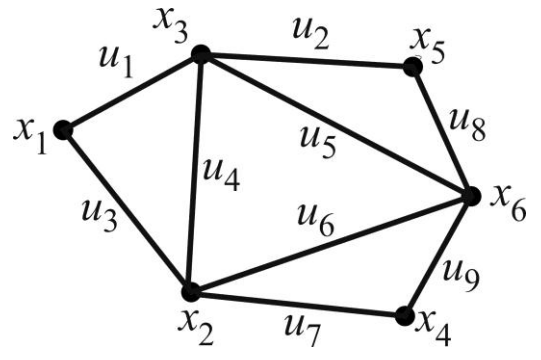
2.8.



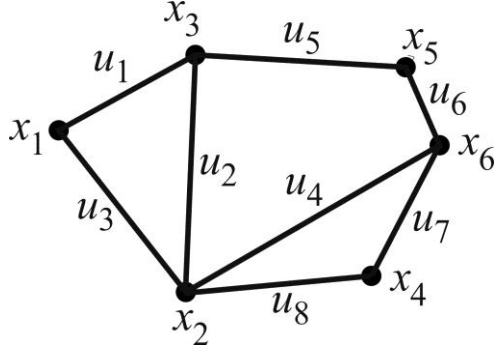
2.9.



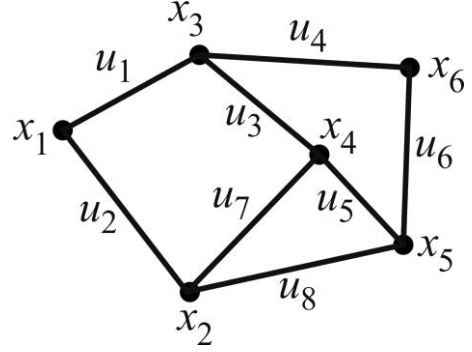
2.10.



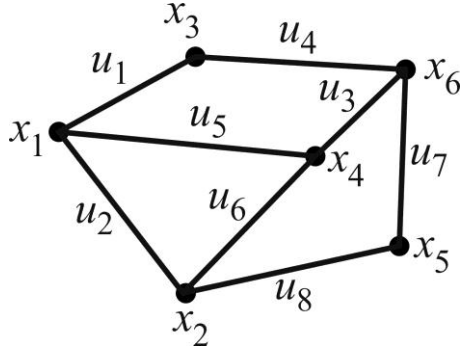
2.11.



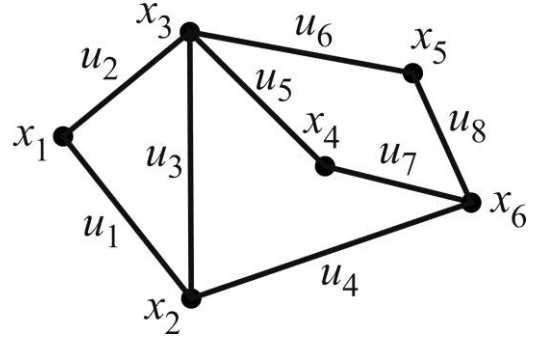
2.12.



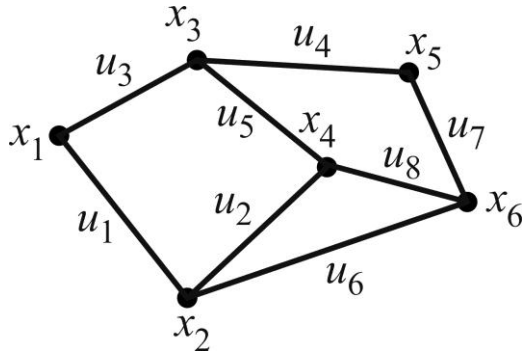
2.13.



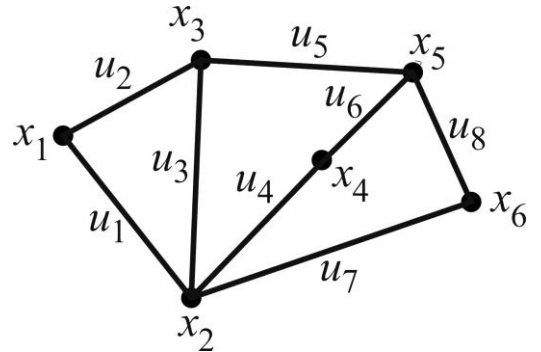
2.14.



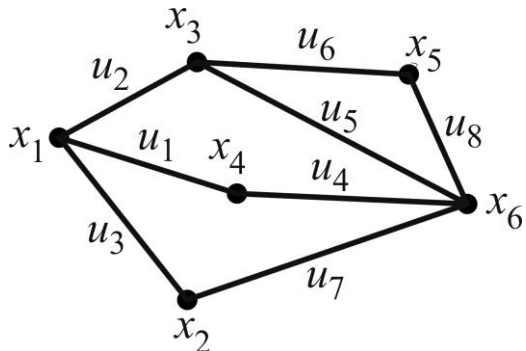
2.15.



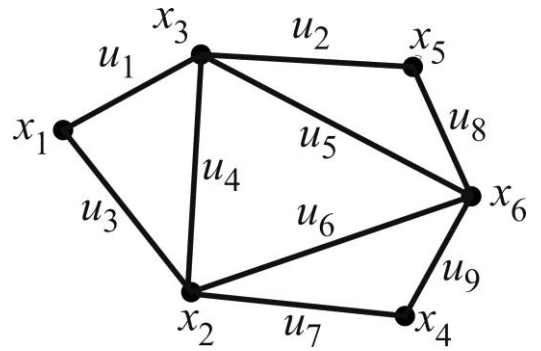
2.16.



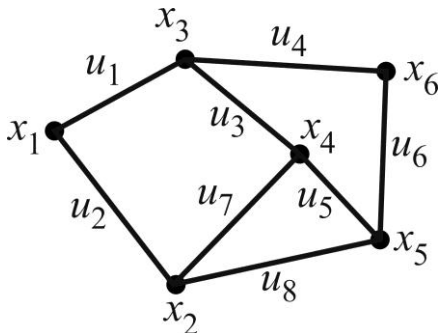
2.17.



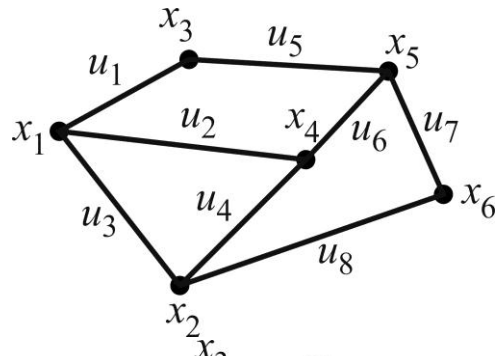
2.18.



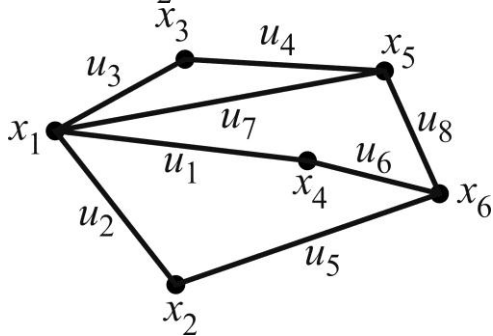
2.19.



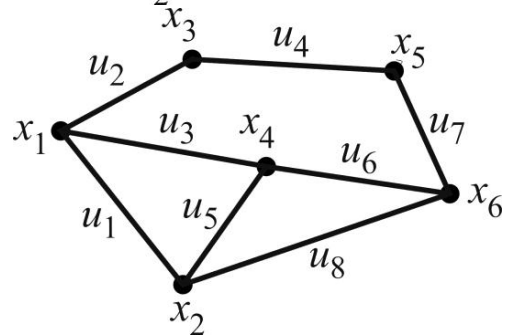
2.20.



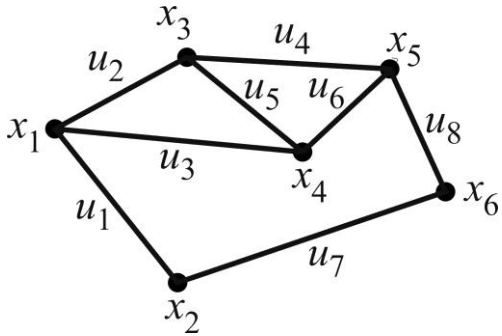
2.21.



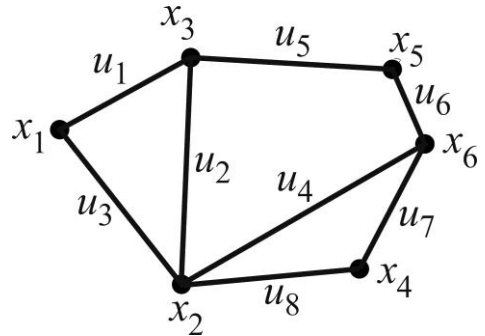
2.22.



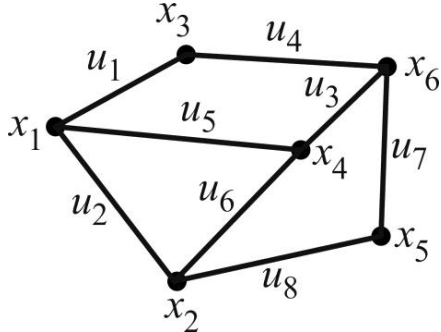
2.23.



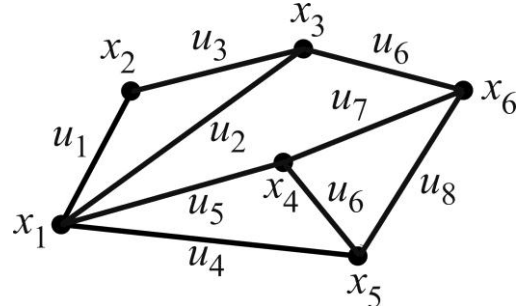
2.24.



2.25.

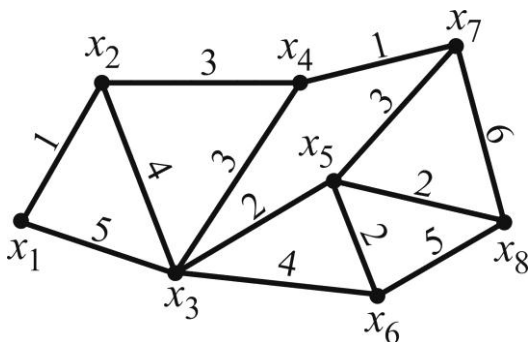


2.26.

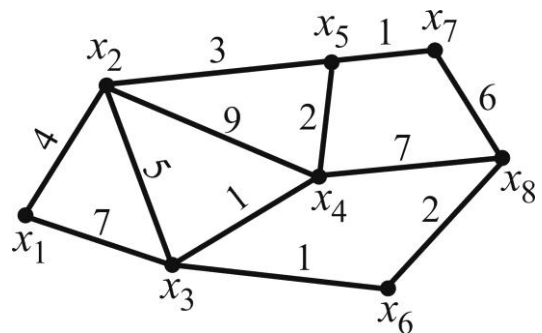


Задача 3. На заданому навантаженому зв'язному неорієнтованому графі G побудуйте економічне дерево, знайдіть його довжину і зобразіть упорядкованим за рівнями вершин.

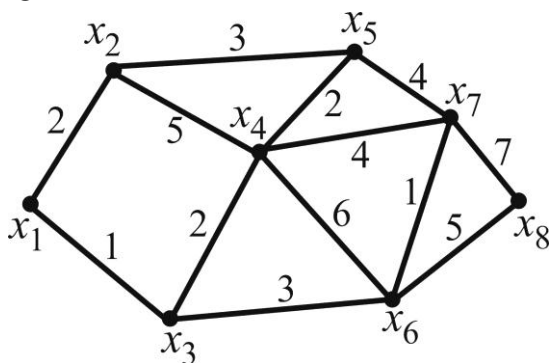
3.1.



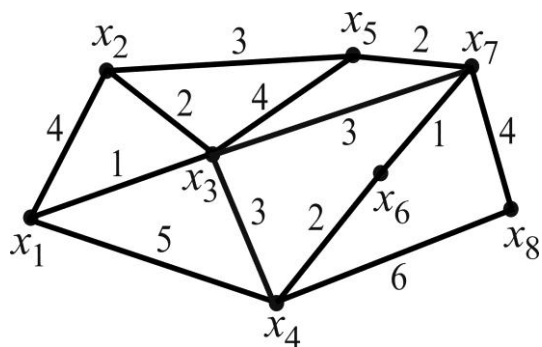
3.2.



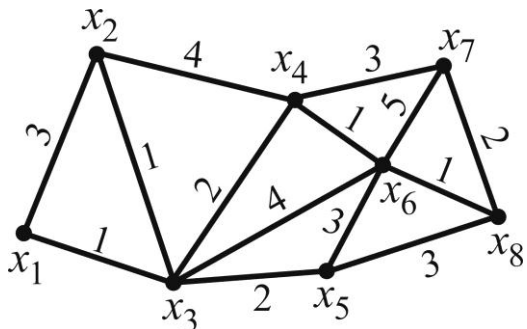
3.3.



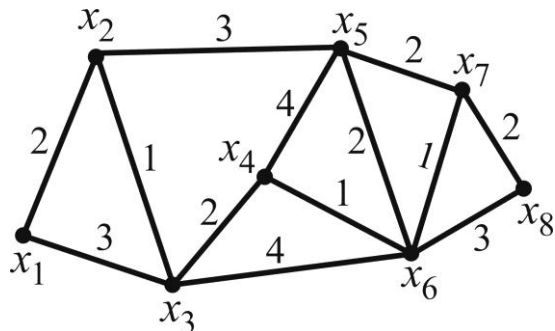
3.4.



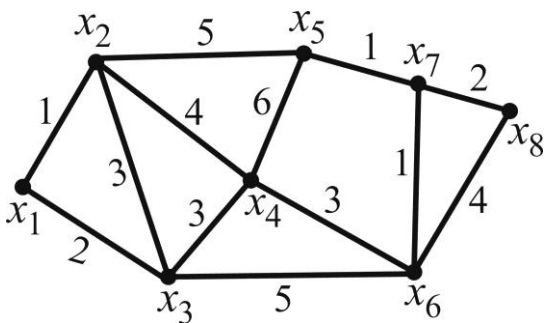
3.5.



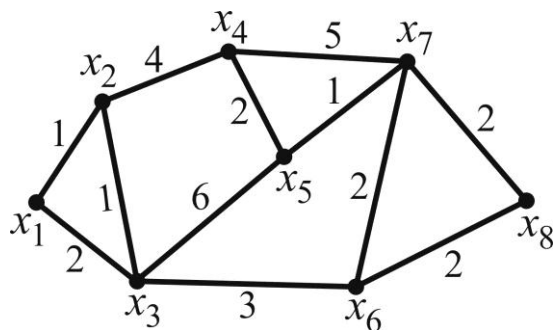
3.6.



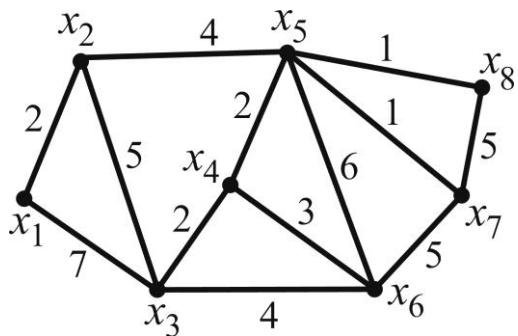
3.7.



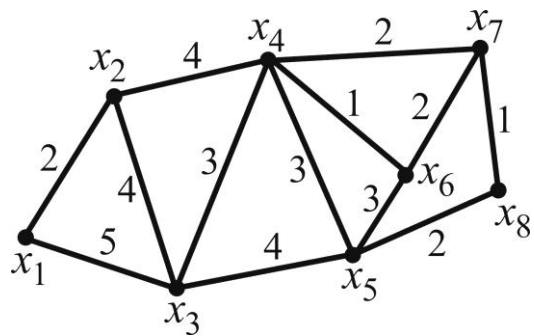
3.8.



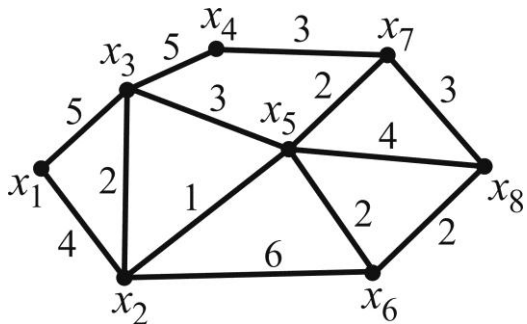
3.9.



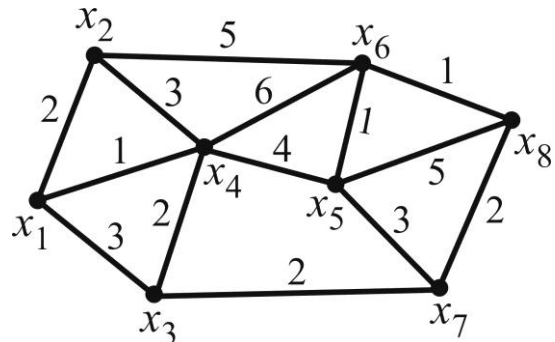
3.10.



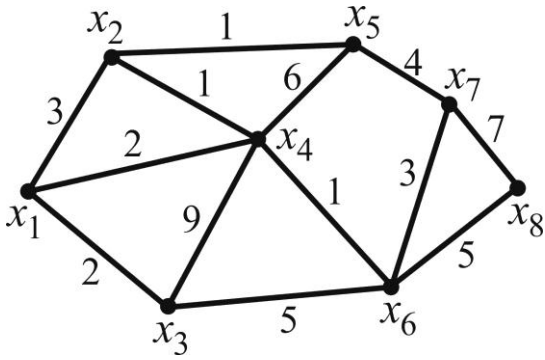
3.11.



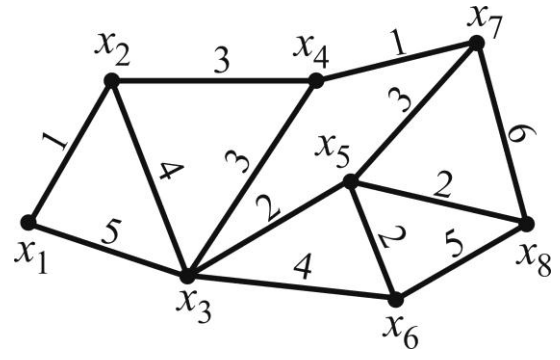
3.12.



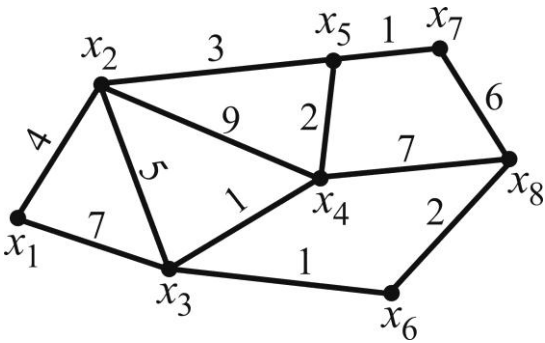
3.13.



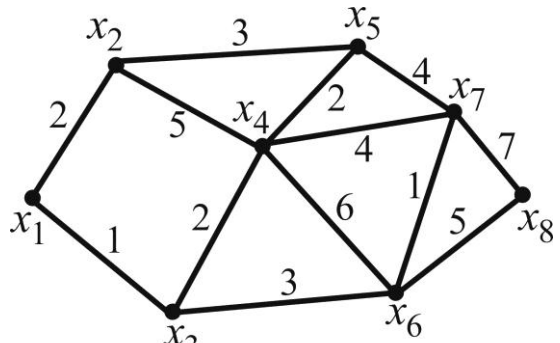
3.14.



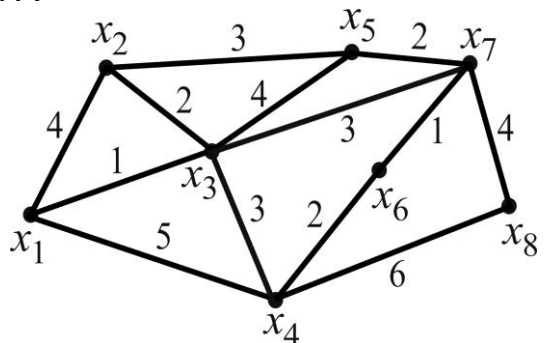
3.15.



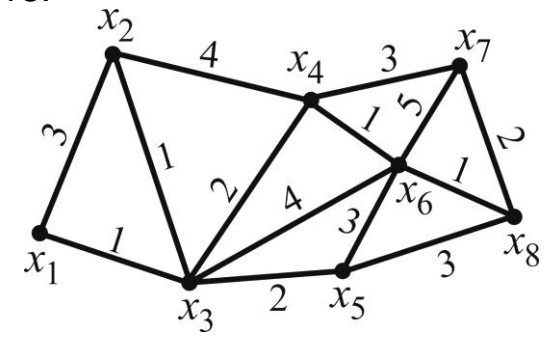
3.16.



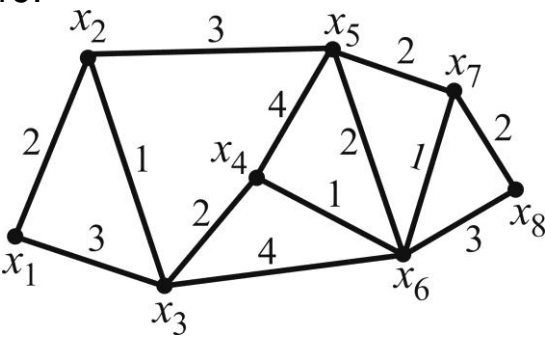
3.17.



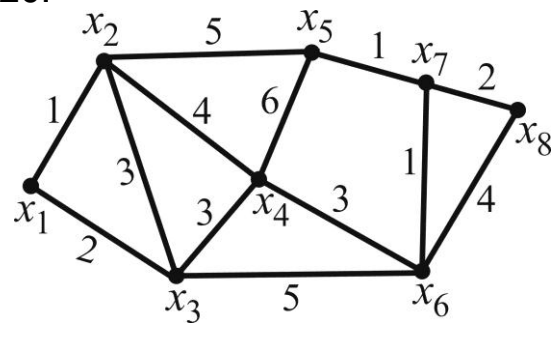
3.18.



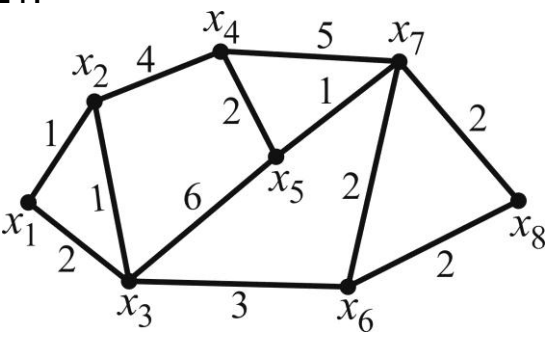
3.19.



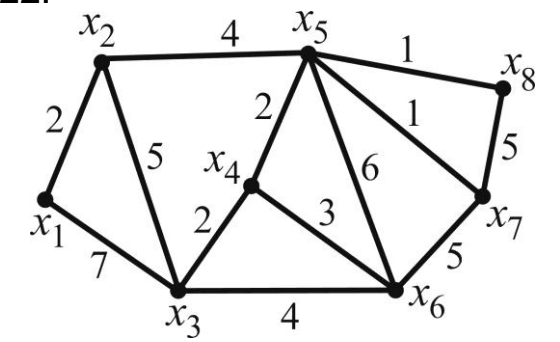
3.20.



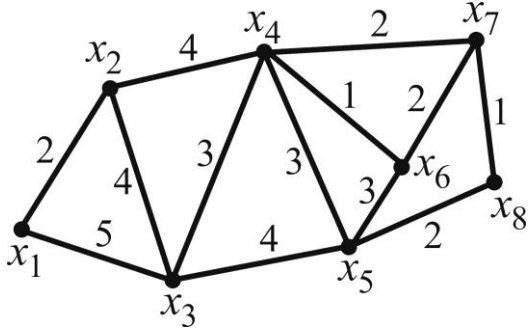
3.21.



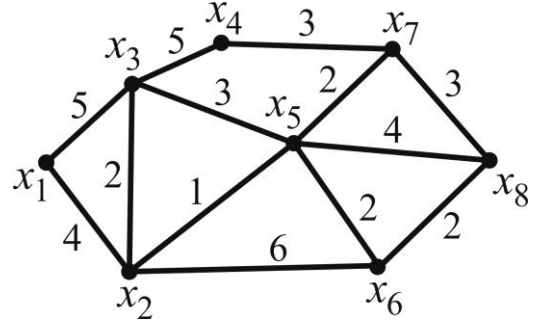
3.22.



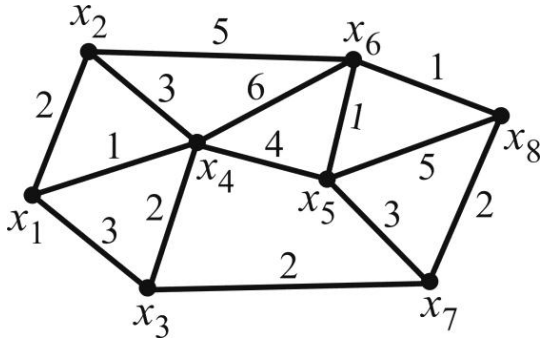
3.23.



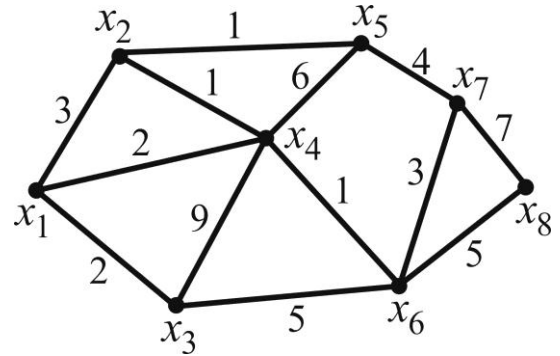
3.24.



3.25.



3.26.



Задача 4. На вихідному графі задачі 3 методом індексації вершин знайдіть найкоротший ланцюг між вершинами x_1 і x_8 та його довжину.

Задача 5. Для орієнтованого графа $\bar{G} = (X, U)$, заданого теоретико-множинним способом: 1) надайте його геометричне зображення; 2) побудуйте матриці інцидентностей та суміжності вершин.

5.1. $\bar{G} = (\{x_1, x_2, x_3, x_4\}, \{(x_1, x_1), (x_1, x_2), (x_1, x_4), (x_2, x_1), (x_2, x_2), (x_3, x_3), (x_4, x_1), (x_4, x_3)\})$.

5.2. $\bar{G} = (\{x_1, x_2, x_3, x_4\}, \{(x_1, x_1), (x_2, x_1), (x_2, x_2), (x_2, x_3), (x_2, x_4), (x_3, x_3), (x_3, x_4), (x_4, x_4)\})$.

5.3. $\bar{G} = (\{x_1, x_2, x_3, x_4\}, \{(x_1, x_1), (x_1, x_2), (x_2, x_2), (x_3, x_2), (x_3, x_4), (x_3, x_3), (x_4, x_1), (x_4, x_2)\})$.

5.4. $\bar{G} = (\{x_1, x_2, x_3, x_4\}, \{(x_1, x_4), (x_2, x_1), (x_2, x_4), (x_3, x_2), (x_3, x_3), (x_3, x_4), (x_4, x_4), (x_4, x_3)\})$.

5.5. $\bar{G} = (\{x_1, x_2, x_3, x_4\}, \{(x_1, x_2), (x_2, x_1), (x_3, x_1), (x_3, x_2), (x_3, x_4), (x_4, x_1), (x_4, x_3), (x_4, x_4)\})$.

5.6. $\bar{G} = (\{x_1, x_2, x_3, x_4\}, \{(x_1, x_1), (x_1, x_2), (x_1, x_4), (x_2, x_2), (x_2, x_4), (x_3, x_3), (x_3, x_2), (x_4, x_3)\})$.

$$5.7. \overline{G} = (\{x_1, x_2, x_3, x_4\}, \{(x_1, x_1), (x_1, x_2), (x_1, x_4), (x_2, x_2), (x_2, x_3), (x_3, x_3), (x_4, x_3), (x_4, x_4)\}).$$

$$5.8. \overline{G} = (\{x_1, x_2, x_3, x_4\}, \{(x_1, x_1), (x_2, x_1), (x_2, x_2), (x_2, x_3), (x_1, x_4), (x_3, x_4), (x_3, x_3), (x_2, x_4)\}).$$

$$5.9. \overline{G} = (\{x_1, x_2, x_3, x_4\}, \{(x_1, x_1), (x_1, x_4), (x_2, x_1), (x_2, x_2), (x_3, x_1), (x_3, x_4), (x_4, x_1), (x_4, x_3)\}).$$

$$5.10. \overline{G} = (\{x_1, x_2, x_3, x_4\}, \{(x_1, x_1), (x_1, x_4), (x_2, x_3), (x_2, x_2), (x_3, x_2), (x_3, x_3), (x_4, x_1), (x_4, x_4)\}).$$

$$5.11. \overline{G} = (\{x_1, x_2, x_3, x_4\}, \{(x_1, x_1), (x_2, x_4), (x_2, x_1), (x_3, x_3), (x_4, x_1), (x_4, x_2), (x_1, x_3), (x_2, x_3)\}).$$

$$5.12. \overline{G} = (\{x_1, x_2, x_3, x_4\}, \{(x_1, x_1), (x_1, x_2), (x_1, x_3), (x_2, x_1), (x_2, x_3), (x_3, x_2), (x_3, x_1), (x_4, x_4)\}).$$

$$5.13. \overline{G} = (\{x_1, x_2, x_3, x_4\}, \{(x_1, x_3), (x_1, x_4), (x_2, x_1), (x_2, x_2), (x_2, x_3), (x_3, x_3), (x_4, x_3), (x_4, x_4)\}).$$

$$5.14. \overline{G} = (\{x_1, x_2, x_3, x_4\}, \{(x_1, x_1), (x_1, x_2), (x_2, x_2), (x_2, x_3), (x_3, x_2), (x_3, x_3), (x_4, x_3), (x_4, x_4)\}).$$

$$5.15. \overline{G} = (\{x_1, x_2, x_3, x_4\}, \{(x_1, x_4), (x_1, x_1), (x_2, x_1), (x_2, x_3), (x_3, x_3), (x_3, x_4), (x_4, x_1), (x_4, x_3)\}).$$

$$5.16. \overline{G} = (\{x_1, x_2, x_3, x_4\}, \{(x_1, x_1), (x_2, x_2), (x_2, x_3), (x_3, x_2), (x_3, x_3), (x_3, x_4), (x_4, x_3), (x_4, x_4)\}).$$

$$5.17. \overline{G} = (\{x_1, x_2, x_3, x_4\}, \{(x_1, x_3), (x_1, x_2), (x_2, x_2), (x_2, x_3), (x_3, x_2), (x_1, x_1), (x_4, x_1), (x_4, x_2)\}).$$

$$5.18. \overline{G} = (\{x_1, x_2, x_3, x_4\}, \{(x_1, x_2), (x_1, x_3), (x_1, x_4), (x_2, x_3), (x_3, x_2), (x_4, x_2), (x_4, x_3), (x_4, x_4)\}).$$

$$5.19. \overline{G} = (\{x_1, x_2, x_3, x_4\}, \{(x_1, x_1), (x_1, x_2), (x_2, x_2), (x_2, x_4), (x_3, x_3), (x_3, x_2), (x_4, x_1), (x_4, x_4)\}).$$

$$5.20. \overline{G} = (\{x_1, x_2, x_3, x_4\}, \{(x_1, x_1), (x_1, x_3), (x_2, x_2), (x_2, x_3), (x_3, x_3), (x_3, x_4), (x_4, x_1), (x_4, x_4)\}).$$

5.21. $\bar{G} = (\{x_1, x_2, x_3, x_4\}, \{(x_1, x_1), (x_1, x_3), (x_2, x_4), (x_4, x_4), (x_3, x_3), (x_3, x_1), (x_4, x_2), (x_4, x_3)\})$.

5.22. $\bar{G} = (\{x_1, x_2, x_3, x_4\}, \{(x_1, x_1), (x_1, x_4), (x_2, x_3), (x_4, x_1), (x_2, x_2), (x_4, x_3), (x_3, x_4), (x_3, x_3)\})$.

5.23. $\bar{G} = (\{x_1, x_2, x_3, x_4\}, \{(x_1, x_1), (x_2, x_4), (x_2, x_1), (x_2, x_2), (x_3, x_3), (x_3, x_4), (x_4, x_1), (x_4, x_3)\})$.

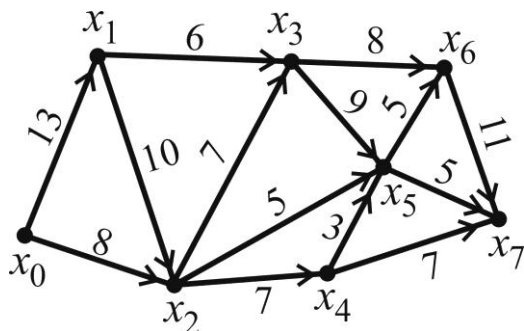
5.24. $\bar{G} = (\{x_1, x_2, x_3, x_4\}, \{(x_1, x_1), (x_2, x_2), (x_1, x_2), (x_3, x_3), (x_4, x_4), (x_4, x_1), (x_3, x_2), (x_1, x_3)\})$.

5.25. $\bar{G} = (\{x_1, x_2, x_3, x_4\}, \{(x_1, x_1), (x_1, x_2), (x_2, x_1), (x_2, x_2), (x_2, x_4), (x_3, x_3), (x_4, x_4), (x_4, x_2)\})$.

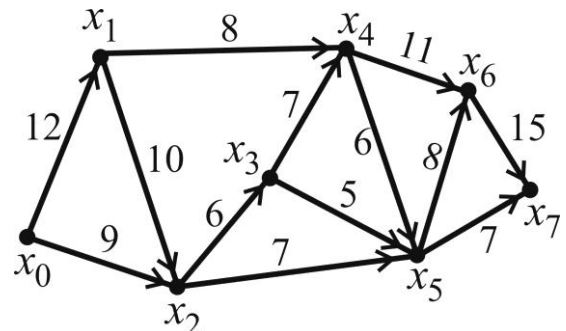
5.26. $\bar{G} = (\{x_1, x_2, x_3, x_4\}, \{(x_1, x_1), (x_2, x_2), (x_2, x_4), (x_2, x_3), (x_4, x_4), (x_4, x_2), (x_3, x_3), (x_3, x_4)\})$.

Задача 6. За геометричним зображенням орграфа необхідно:
 1) знайти його теоретико-множинне подання; 2) записати матриці інцидентностей та суміжності вершин; 3) визначити півстепінь входу та півстепінь виходу кожної вершини.

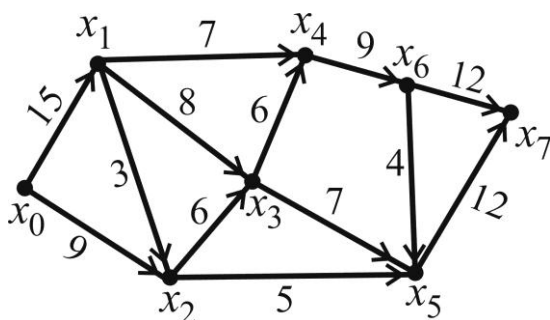
6.1.



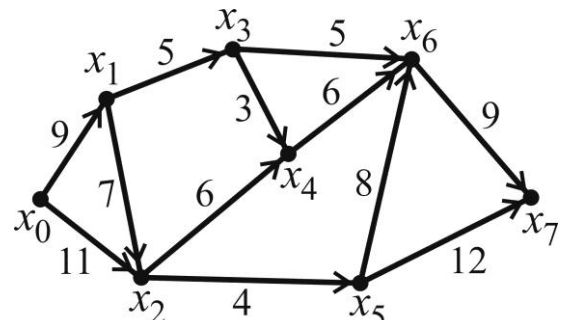
6.2.



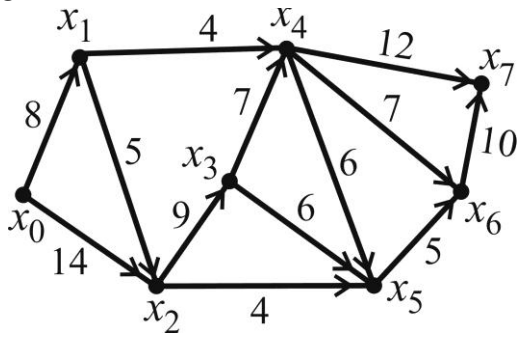
6.3.



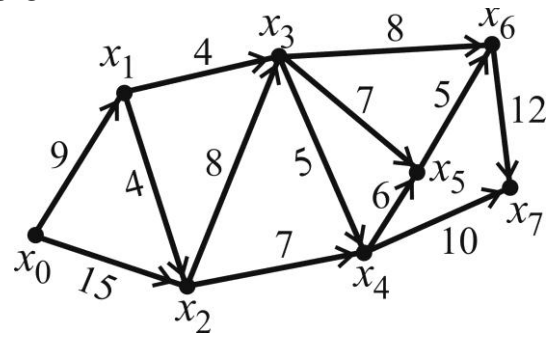
6.4.



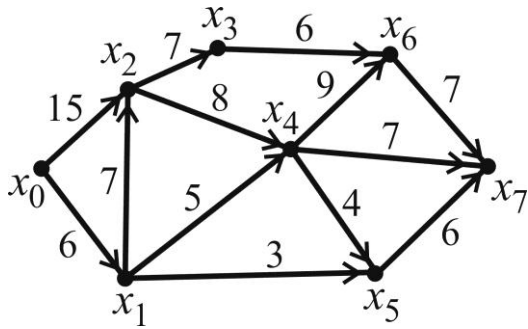
6.5.



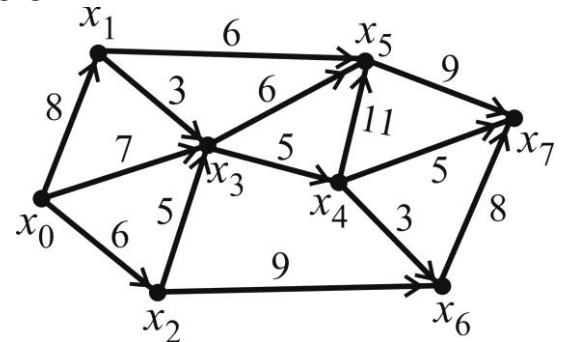
6.6.



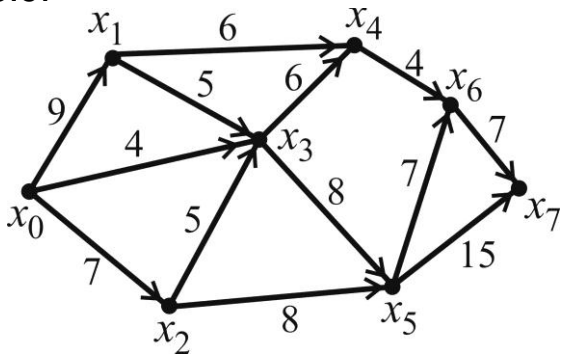
6.7.



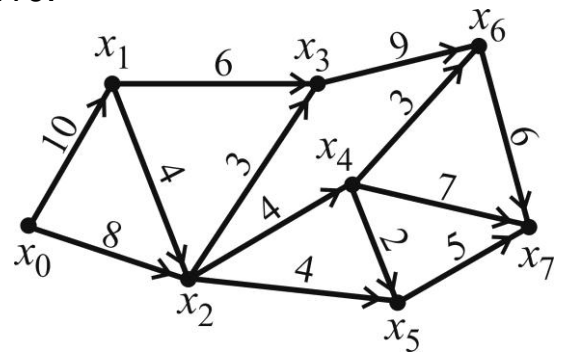
6.8.



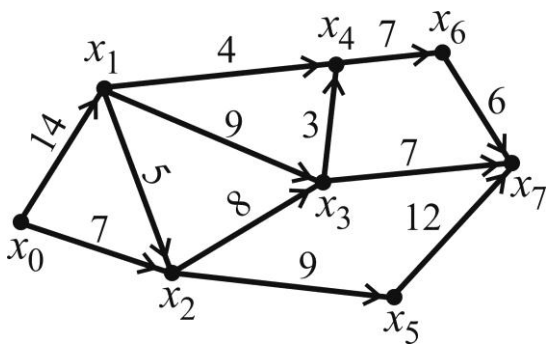
6.9.



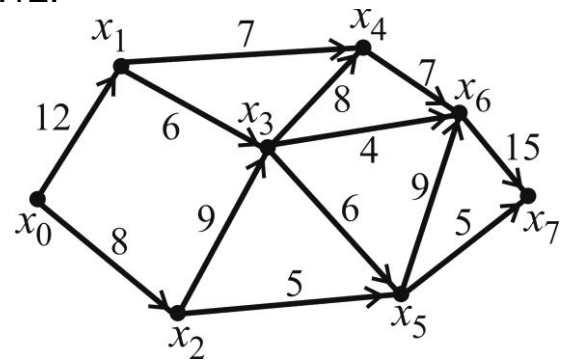
6.10.



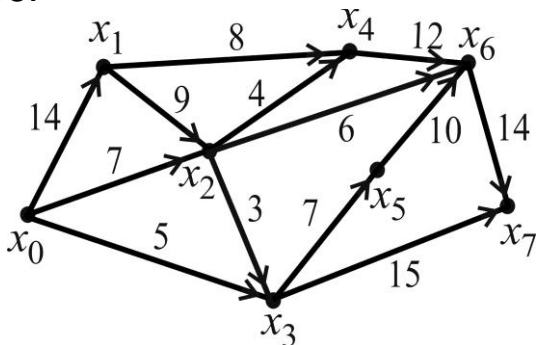
6.11.



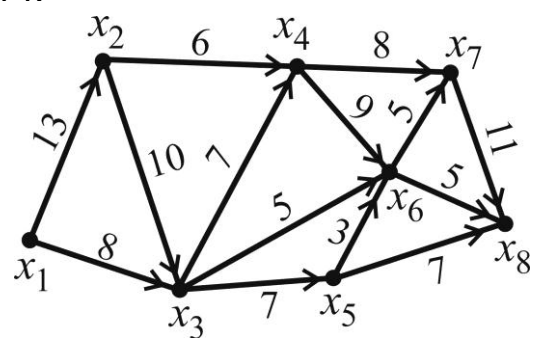
6.12.



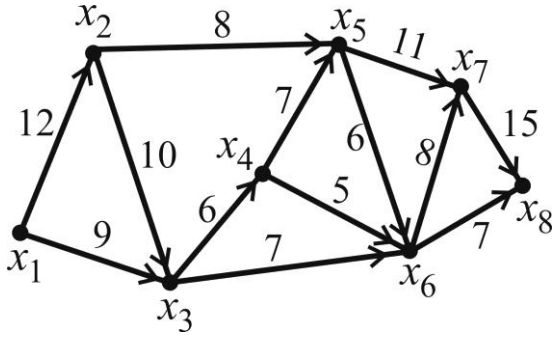
6.13.



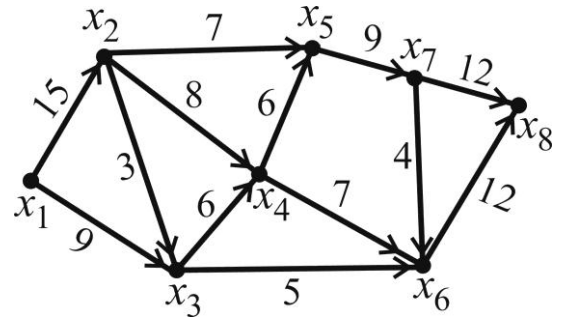
6.14.



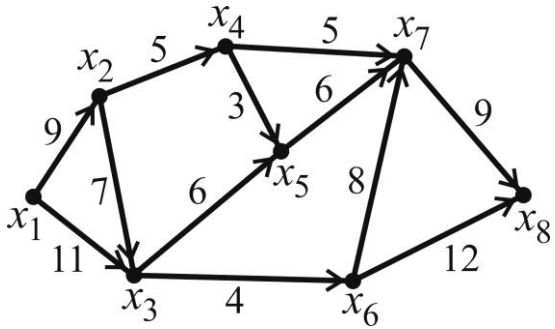
6.15.



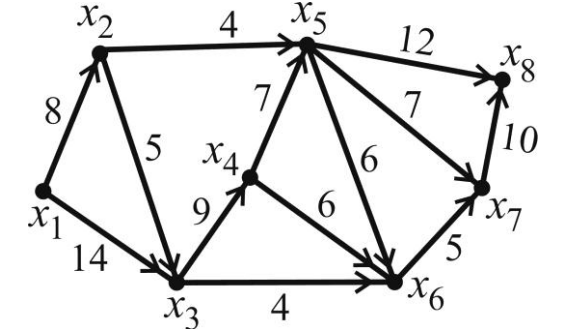
6.16.



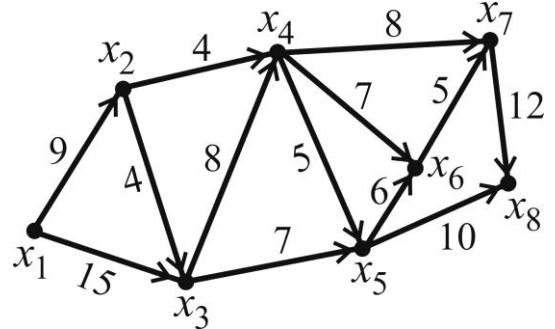
6.17.



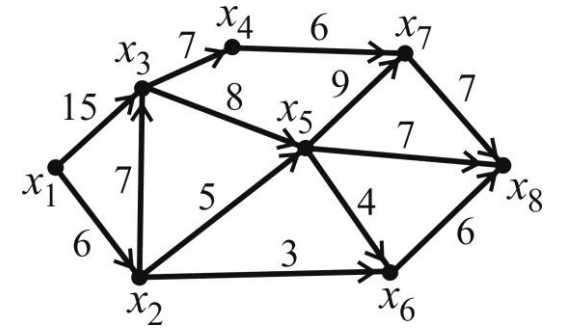
6.18.



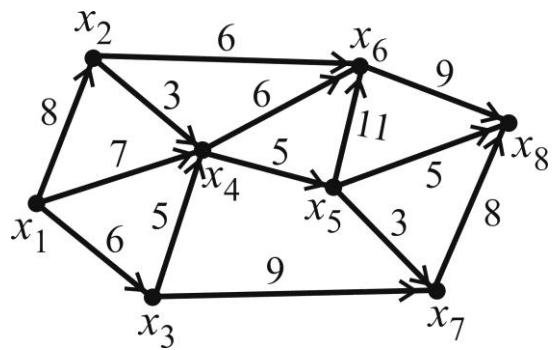
6.19.



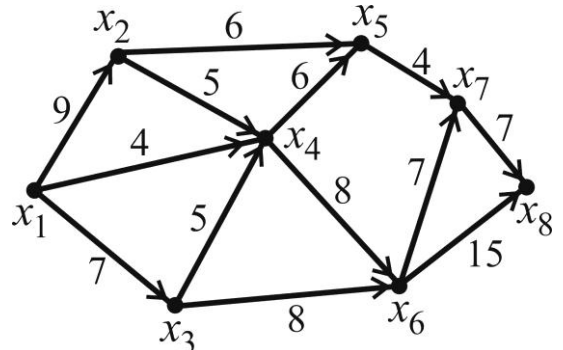
6.20.



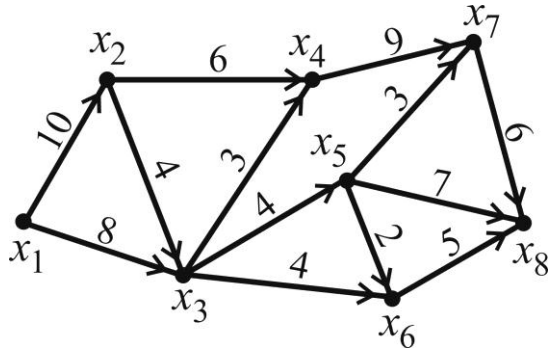
6.21.



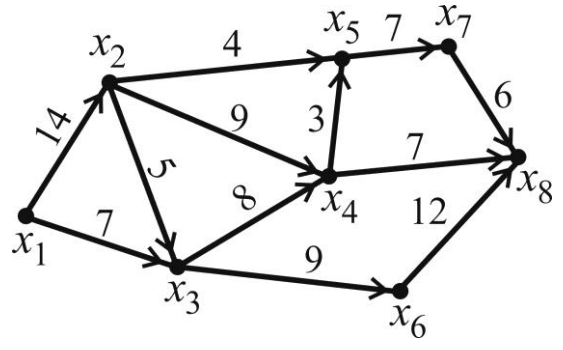
6.22.



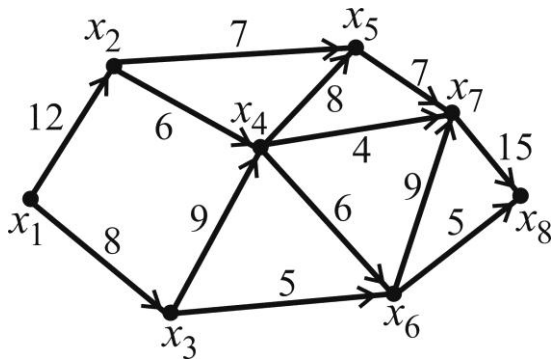
6.23.



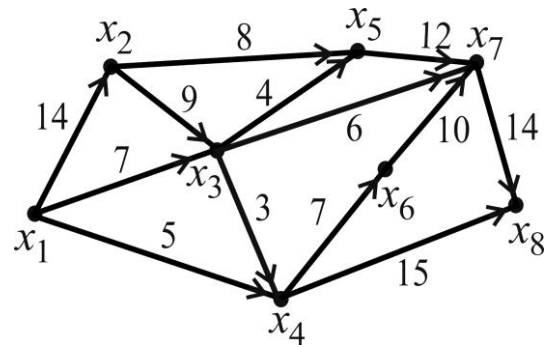
6.24.



6.25.



6.26.



Задача 7. Розглядаючи орграф із задачі 6 як сітковий графік із відомими тривалостями робіт, знайдіть критичний час $t_{кр.}$ і критичний шлях $\mu_{кр.}$ на ньому.

Задача 8. Розглядаючи орграф із задачі 6 як транспортну мережу із заданими пропускними спроможностями її дуг, знайдіть максимальний потік φ_z^* на ній та його розподіл за дугами.

4. Зразки розв'язання задач самостійної контрольної роботи

Задача 1. Для графа $G = (X, U)$, заданого теоретико-множинним способом: 1) дайте його геометричне зображення; 2) побудуйте матриці інцидентностей та суміжності вершин.

$$G = (\{x_1, x_2, x_3, x_4, x_5, x_6\}, \{\{x_1, x_2\}, \{x_1, x_3\}, \{x_1, x_5\}, \{x_3, x_5\}, \{x_3, x_6\}, \{x_3, x_4\}, \{x_4, x_6\}\}).$$

Розв'язання:

1. За теоретико-множинним поданням графа G встановлюємо, що він має шість вершин ($n = 6$) і сім ребер ($m = 7$). Для геометричного зображення графа кожній його вершині ставимо у відповідність точку площини, а кожному ребру – прямолінійний (або криволінійний) відрізок, що з'єднує інцидентні цьому ребру вершини (рис. 4.1).

Зазначимо, що вершини бажано розташовувати в лексикографічному порядку, тобто зліва направо і знизу вгору, а ребра проводити так, щоб вони перетиналися тільки у їхніх граничних вершинах.

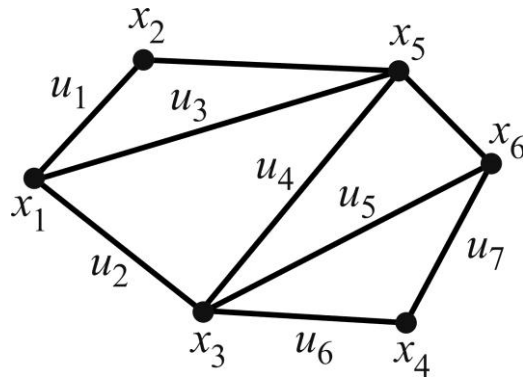


Рис. 4.1. Геометричне зображення графа задачі 1

2. Згідно з означенням, матрицею інцидентностей вихідного $(6,7)$ -графу G є прямокутна матриця $A = [a_{ij}]_{6 \times 7}$ з елементами: $a_{ij} = 0$, якщо вершина x_i не інцидентна ребру u_j ; $a_{ij} = 1$, якщо вершина x_i інцидентна ребру u_j ; $a_{ij} = 2$, якщо u_j – петля при вершині x_i (рис. 4.2а).

Свою чергою, матрицею суміжності вершин графа G є квадратна матриця $B = [b_{ij}]_{6 \times 6}$, порядок якої дорівнює кількості вершин графа, а її елементи b_{ij} указують на кількість ребер із граничними вершинами $x_i, x_j, i, j = \overline{1,6}$; за домовленістю петлю враховують двічі (рис. 4.2б).

$$A = \begin{matrix} & \begin{matrix} u_1 & u_2 & u_3 & u_4 & u_5 & u_6 & u_7 \end{matrix} \\ \begin{matrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{matrix} & \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 \end{bmatrix} \end{matrix}, \quad B = \begin{matrix} & \begin{matrix} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \end{matrix} \\ \begin{matrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{matrix} & \begin{bmatrix} 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix} \end{matrix}.$$

а б

Рис. 4.2. Матриці:
а) інцидентностей; б) суміжності вершин

Задача 2. За геометричним зображенням графа G (рис. 4.3) необхідно: 1) знайти його теоретико-множинне подання; 2) побудувати матриці

інцидентностей та суміжності вершин; 3) знайти матрицю відстаней, ексцентриситети вершин, діаметр, радіус, центр та цикломатичне число графа; 4) указати будь-який ланцюг, простий ланцюг, цикл, простий цикл і їхні довжини; 5) визначити степені всіх вершин і переконатися, що їхня сума дорівнює подвоєній кількості ребер; 6) з'ясувати, чи є граф ейлеровим.

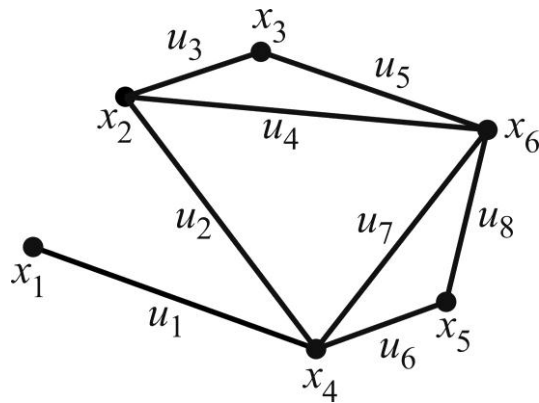


Рис. 4.3. Вихідний граф задачі 2

Розв'язання:

1. Теоретико-множинне подання вихідного графа G із шістьма вершинами та вісьмома ребрами дістанемо переліком елементів множини його вершин X і множини ребер U :

$$G=(X,U)=(\{x_1,x_2,x_3,x_4,x_5,x_6\}, \{u_1=\{x_1,x_4\}, u_2=\{x_2,x_4\}, u_3=\{x_2,x_3\}, u_4=\{x_2,x_6\}, u_5=\{x_3,x_6\}, u_6=\{x_4,x_5\}, u_7=\{x_4,x_6\}, u_8=\{x_5,x_6\}\}).$$

2. Побудуємо матриці інцидентностей A і суміжності вершин B :

$$A = \begin{matrix} & u_1 & u_2 & u_3 & u_4 & u_5 & u_6 & u_7 & u_8 \\ \begin{matrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{matrix} & \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \end{bmatrix} \end{matrix}, \quad B = \begin{matrix} & x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \\ \begin{matrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{matrix} & \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 \end{bmatrix} \end{matrix}.$$

3. Знайдемо матрицю відстаней заданого графа G – квадратну матрицю $V = [v_{ij}]_{n \times n}$ n -го порядку (n – кількість вершин графа), кожний елемент якої v_{ij} дорівнює відстані між вершинами x_i, x_j , тобто довжині найкоротшого ланцюга, що з'єднує ці вершини $i = \overline{1, n}, j = \overline{1, n}$. Переконуємося, що матриця V є симетричною матрицею: $v_{ij} = v_{ji}$ ($i, j = \overline{1, n}$).

$$V = \begin{matrix} & \begin{matrix} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \end{matrix} \\ \begin{matrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{matrix} & \begin{bmatrix} 0 & 2 & 3 & 1 & 2 & 2 \\ 2 & 0 & 1 & 1 & 2 & 1 \\ 3 & 1 & 0 & 2 & 2 & 1 \\ 1 & 1 & 2 & 0 & 1 & 1 \\ 2 & 2 & 2 & 1 & 0 & 1 \\ 2 & 1 & 1 & 1 & 1 & 0 \end{bmatrix} \end{matrix}.$$

Ексцентриситетом вершини x_i (позначають $e(x_i)$) називають відстань від x_i до найвіддаленішої від неї вершини графа, а значить, $e(x_i)$ дорівнює найбільшому елементу i -го рядка матриці відстаней V . Таким чином, маємо:

$$e(x_1) = 3, e(x_2) = 2, e(x_3) = 3, e(x_4) = 2, e(x_5) = 2, e(x_6) = 2.$$

Діаметр $diam(G)$ графа G – це найбільший з ексцентриситетів його вершин, тобто найбільша відстань між будь-якими двома вершинами графа. Радіус $r(G)$ графа G – це найменший з ексцентриситетів його вершин. У нас: $diam(G) = 3, r(G) = 2$. Неважко перевірити співвідношення, яким пов'язано радіус і діаметр будь-якого графа:

$$r(G) \leq diam(G) \leq 2r(G).$$

Центральною називають вершину x_i , ексцентриситет якої дорівнює радіусу графа: $e(x_i) = r(G), i = \overline{1, n}$. Отже, для вихідного графа центральними є відразу чотири вершини: x_2, x_4, x_5, x_6 , оскільки $e(x_2) = e(x_4) =$

$= e(x_5) = e(x_6) = r(G) = 2$. Центр графа G – це множина всіх його центральних вершин:

$$C(G) = \{x_2, x_4, x_5, x_6\}.$$

Цикломатичне число γ графа G з n вершинами та m ребрами обчислюють за такою формулою:

$$\gamma = m - n + p,$$

де p – кількість компонент зв'язності.

Воно показує, скільки ребер треба вилучити із графа, щоб дістати його кістяк. Оскільки граф G зв'язний, то $p = 1$, отже: $\gamma = 8 - 6 + 1 = 3$.

4. Згідно з означенням, ланцюгом (циклом) називають незамкнений (замкнений) маршрут, усі ребра у якому різні. Ланцюг (цикл), у якому всі вершини різні (крім першої та останньої), називають простим.

Наведемо приклади таких маршрутів на вихідному графі G задачі:

$\mu_{15} = (u_1, u_7, u_4, u_2, u_6) = (x_1 - x_4 - x_6 - x_2 - x_4 - x_5)$ – ланцюг (непростий) довжини $l(\mu_{15}) = 5$ (довжина маршруту дорівнює кількості ребер у ньому);

$\mu_{21} = (u_4, u_7, u_1) = (x_2 - x_6 - x_4 - x_1)$ – простий ланцюг, $l(\mu_{21}) = 3$;

$\mu_{22} = (u_3, u_5, u_8, u_6, u_7, u_4) = (x_2 - x_3 - x_6 - x_5 - x_4 - x_6 - x_2)$ – цикл (непростий), $l(\mu_{22}) = 6$;

$\mu_{44} = (u_2, u_3, u_5, u_7) = (x_4 - x_2 - x_3 - x_6 - x_4)$ – простий цикл, $l(\mu_{22}) = 4$.

5. Визначимо степені s_i вершин графа, тобто кількість ребер, які інцидентні кожній вершині x_i ($i = \overline{1, n}$): $s_1 = 1$, $s_2 = 3$, $s_3 = 2$, $s_4 = 4$, $s_5 = 2$, $s_6 = 4$. Знаходимо суму степенів усіх вершин і переконуємося, що вона дорівнює подвоєній кількості ребер графа ($m = 8$):

$$\sum_{i=1}^6 s_i = 1 + 3 + 2 + 4 + 2 + 4 = 16 = 2m.$$

6. З'ясуємо, чи є граф ейлеровим за критерієм (теоремою) Ейлера: зв'язний граф містить ейлерів цикл тоді й тільки тоді, коли кожна його вершина має парний степінь. Аналізуючи, знайдені в п. 5 степені вершин, робимо висновок, що вихідний граф задачі не є ейлеровим, оскільки він має вершини непарного степеня.

Задача 3. На заданому навантаженому зв'язному неорієнтованому графі G (рис. 4.4) побудуйте економічне дерево, знайдіть його довжину і зобразіть упорядкованим за рівнями вершин.

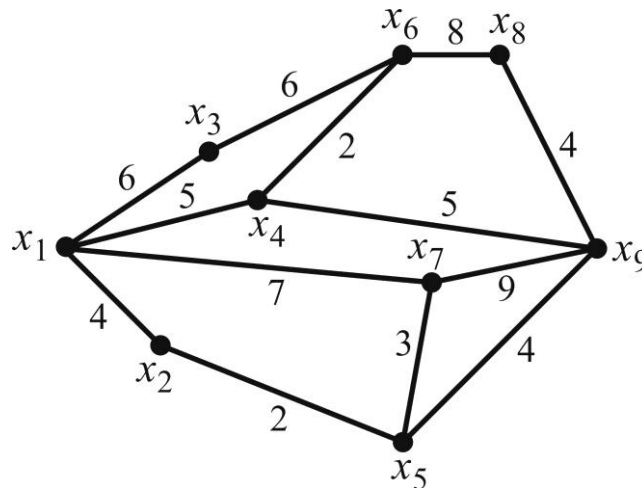


Рис. 4.4. Вихідний граф задачі 3

Розв'язання. Задачу побудови економічного дерева будемо розв'язувати за алгоритмом, який описано в п. 1.6. Довжину ребра $\{x_i, x_j\}$ позначимо через d_{ij} , де $i, j \in \{1, 2, \dots, 9\}$.

Вибираємо послідовно ребра із пріоритетом за мінімумом їхньої довжини й так, щоб не утворювалося циклів (рис. 4.5а). Крім того, слід пам'ятати, що кількість ребер покривного дерева має бути на одиницю меншою, ніж кількість вершин графа ($n = 9$), тобто вісім:

$$d_{25} = 2, d_{46} = 2, d_{57} = 3, d_{59} = 4, d_{12} = 4, \\ d_{89} = 4, d_{49} = 5, d_{36} = 6.$$

Ребра побудованого дерева на рис. 4.5а позначено пунктирними лініями, а їхня сумарна довжина визначає довжину економічного дерева:

$$l^* = 2 + 2 + 3 + 4 + 4 + 4 + 5 + 6 = 30.$$

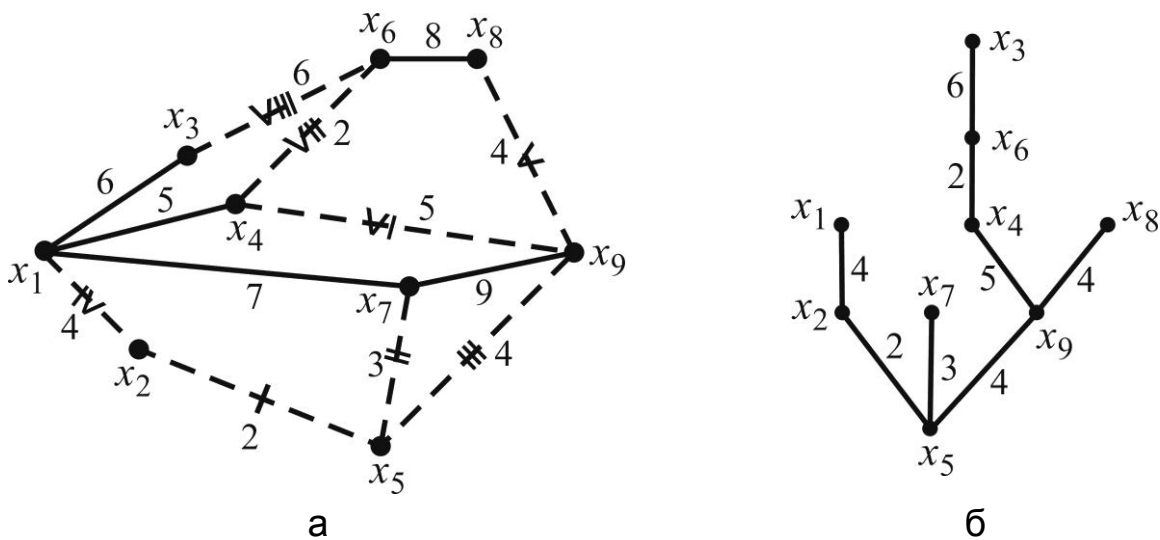


Рис. 4.5. Економічне дерево:
 а) на вихідному графі; б) упорядковане за рівнями вершин

Побудувати економічне дерево можна було б і так: після вибору ребра найменшої довжини на кожному наступному кроці додавати (приєднувати) суміжне (до вже вибраних ребер) ребро найменшої довжини з тих, що залишилися; у цьому разі ребра залучалися б у порядку, визначеному римськими цифрами.

На рис. 4.5б зображено знайдене економічне дерево, упорядковане за рівнями вершин за умови, що його корінь – вершина x_5 .

Зауважимо, що:

після вибору ребра $\{x_4, x_9\}$ довжини $d_{49} = 5$ ми не можемо взяти друге ребро такої самої довжини – $\{x_1, x_4\}$, оскільки утворився б цикл $(1-2-5-9-4-1)$;

інші економічні дерева можна дістати, якщо замість ребра $\{x_4, x_9\}$ узяти ребро $\{x_1, x_4\}$ і (або) замість $\{x_3, x_6\}$ – ребро $\{x_1, x_3\}$.

Задача 4. На вихідному графі задачі 3 методом індексації вершин знайдіть найкоротший ланцюг між вершинами x_1 і x_8 та його довжину.

Розв'язання. Знаходження ланцюга найменшої довжини між двома вершинами графа будемо здійснювати методом індексації (зважування) вершин, згідно з яким кінцевій вершині x_9 надають індекс 0 ($p_9 = 0$), а решті вершин – індекс $+\infty$ ($p_j = +\infty, j \neq 9$); на рис. 4.6 їх не зображено. Індекси будемо надписувати над або під позначенням вершин.

Далі для кожної пари суміжних вершин x_i, x_j перевіряємо, чи буде мати місце нерівність $p_j - p_i > d_{ij}$, де d_{ij} – довжина ребра $\{x_i, x_j\}$. Якщо "так", то вершинам x_j приписуємо нові індекси замість $+\infty$: $p_j^1 = p_i + d_{ij}$.

Для $i = 9, j = 8, 7, 5, 4$ маємо: $p_j - p_9 > d_{9j}$, отже, нові індекси вершин x_8, x_7, x_5, x_4 будуть такими: $p_8^1 = p_9 + d_{89} = 4, p_7^1 = p_9 + d_{79} = 9, p_5^1 = p_9 + d_{59} = 4, p_4^1 = p_9 + d_{49} = 5$ (див. рис. 4.6).

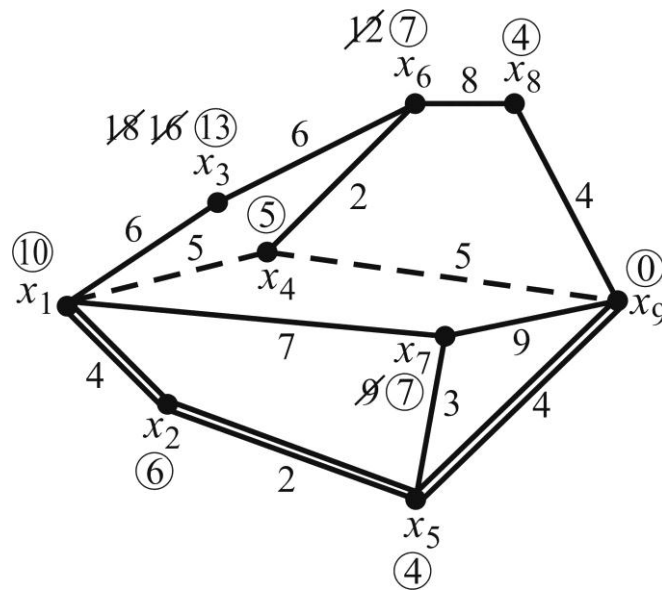


Рис. 4.6. Знаходження ланцюга найменшої довжини

Аналогічно для вершин x_6, x_3, x_2 маємо нові індекси:

$$p_6^1 = p_8 + d_{68} = 4 + 8 = 12, \quad p_3^1 = p_6 + d_{36} = 12 + 6 = 18,$$

$$p_2^1 = p_5 + d_{25} = 4 + 2 = 6.$$

Переіндексацію вершини x_1 можна здійснити неоднозначно, відштовхуючись від однієї з вершин x_2, x_3, x_4 . Щоб уникнути багаторазового переважування вершин, доцільно з усіх можливих варіантів індексації вибирати той, який дає *найменшу вагу*. У нашому разі для x_1 маємо:

$$p_1^1 = p_4 + d_{14} = 5 + 5 = 10 \text{ (або } p_1^1 = p_2 + d_{12} = 6 + 4 = 10).$$

Після "зняття" індексів $p_j = +\infty$ знову перевіряємо виконання умови $p_j - p_i > d_{ij}$ для кожної пари суміжних вершин; для виявлених таких пар знаходимо нові індекси відповідних вершин (на рис. 4.6. "старі" індекси закреслено):

$$\begin{aligned} p_7 - p_5 > d_{57} &\Rightarrow p_7^1 = p_5 + d_{57} = 4 + 3 = 7, \\ p_6 - p_4 > d_{46} &\Rightarrow p_6^1 = p_4 + d_{46} = 5 + 2 = 7, \\ p_3 - p_1 > d_{13} &\Rightarrow p_3^1 = p_1 + d_{13} = 10 + 6 = 16. \end{aligned}$$

Індекс вершини x_3 ще можна поновити:

$$p_3 - p_6 > d_{36} \Rightarrow p_3^1 = p_6 + d_{36} = 7 + 6 = 13.$$

Тепер процес індексації закінчено, оскільки немає таких пар суміжних вершин x_i, x_j , для яких $p_j - p_i > d_{ij}$.

Ланцюг $L^*(1,9)$ найменшої довжини, що з'єднує вершини x_1, x_9 знаходимо, відштовхуючись від вершини x_1 , індекс якої $p_1 = 10$ вказує на довжину цього ланцюга: на кожному кроці беремо таке ребро, різниця індексів інцидентних вершин якого дорівнює довжині цього ребра.

У нашому разі є два оптимальні ланцюги між x_1 і x_9 : ланцюг $L_1^*(1,9) = (x_1 - x_4 - x_9)$, ребра якого на рис. 4.6 зображено однією пунктирною лінією, і ланцюг $L_2^*(1,9) = (x_1 - x_2 - x_5 - x_9)$, ребра якого зображено двома суцільними лініями.

Зазначимо, що вага кожної вершини вказує на довжину найкоротшого ланцюга, що з'єднує розглядувану вершину з кінцевою вершиною x_9 .

Задача 5. Для орієнтованого графа $\bar{G} = (X, U)$, заданого теоретико-множинним способом: 1) надайте його геометричне зображення; 2) побудуйте матрицю інцидентностей та матрицю суміжності вершин.

$$\begin{aligned} \bar{G} = (\{x_1, x_2, x_3, x_4, x_5\}, \{ & (x_1, x_3), (x_1, x_2), (x_2, x_2), (x_4, x_2), \\ & (x_4, x_3), (x_3, x_4)\}). \end{aligned}$$

Розв'язання:

1. За теоретико-множинним поданням орграфа \overline{G} встановлюємо, що він має п'ять вершин ($n = 5$) і шість орребер – дуг ($m = 6$). Для геометричного зображення \overline{G} кожній його вершині x_i ставимо у відповідність точку площини, а кожній дузі (x_i, x_k) – прямолінійний (або криволінійний) відрізок зі стрілкою на кінці, який виходить із вершини x_i і заходить у вершину x_k $i = \overline{1,5}$, $j = \overline{1,5}$ (рис. 4.7).

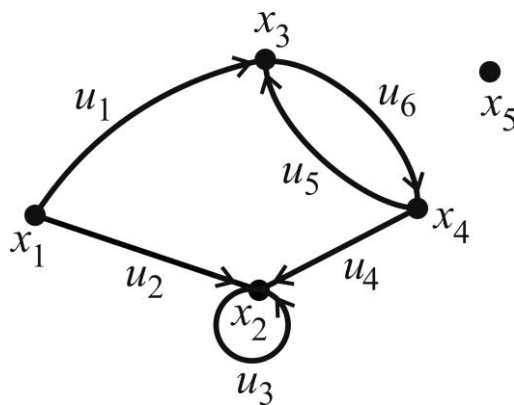


Рис. 4.7. Геометричне зображення графа задачі 5

2. За означенням матрицею інцидентностей орграфа \overline{G} із п'ятьма вершинами та шістьма дугами є прямокутна матриця $A = [a_{ij}]_{5 \times 6}$ з елементами: $a_{ij} = +1$, якщо вершина x_i є початком дуги u_j ; $a_{ij} = -1$, якщо вершина x_i є кінцем дуги u_j ; $a_{ij} = 0$, якщо вершина x_i не інцидентна дузі u_j ; $a_{ij} = \pm 1$, якщо u_j – петля при вершині x_i (рис. 4.8а).

Своєю чергою, матрицею суміжності вершин орграфа \overline{G} є квадратна матриця $B = [b_{ij}]_{5 \times 5}$, порядок якої дорівнює кількості вершин орграфа, а її елементи b_{ij} указують на кількість дуг, що йдуть від вершини x_i до вершини x_j , $i, j = \overline{1,5}$ (рис. 4.8б).

$$A = \begin{matrix} & u_1 & u_2 & u_3 & u_4 & u_5 & u_6 \\ \begin{matrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{matrix} & \begin{bmatrix} +1 & +1 & 0 & 0 & 0 & 0 \\ 0 & -1 & \pm 1 & -1 & 0 & 0 \\ -1 & 0 & 0 & 0 & -1 & +1 \\ 0 & 0 & 0 & +1 & +1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{matrix} \quad \text{а}$$

$$B = \begin{matrix} & x_1 & x_2 & x_3 & x_4 & x_5 \\ \begin{matrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{matrix} & \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{matrix} \quad \text{б}$$

Рис. 4.8. Матриці:
а) інцидентностей; б) суміжності вершин

Задача 6. За геометричним зображенням орграфа (рис. 4.9) необхідно: 1) знайти його теоретико-множинне подання; 2) записати матриці інцидентностей та суміжності вершин; 3) визначити півстепінь входу та півстепінь виходу кожної вершини.

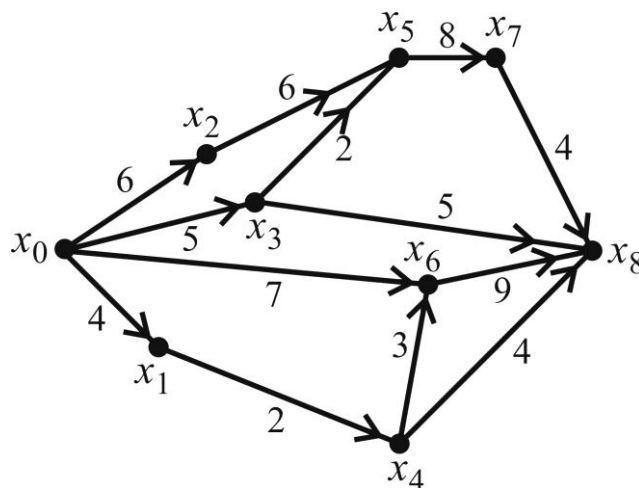


Рис. 4.9. Вихідний граф задачі 6

Розв'язання:

1. Теоретико-множинне подання орграфа \bar{G} має такий вигляд:

$$\bar{G} = (X, U) = (\{x_0, x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8\}, \{(x_0, x_1), (x_0, x_2), (x_0, x_3), (x_0, x_6), (x_2, x_5), (x_3, x_5), (x_3, x_8), (x_1, x_4), (x_4, x_6), (x_4, x_8), (x_6, x_8), (x_5, x_7), (x_7, x_8)\}).$$

2. Запишемо матриці інцидентностей A і суміжності вершин B :

$$A = \begin{matrix} & u_1 & u_2 & u_3 & u_4 & u_5 & u_6 & u_7 & u_8 & u_9 & u_{10} & u_{11} & u_{12} & u_{13} \\ \begin{matrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \end{matrix} & \begin{bmatrix} +1 & +1 & +1 & +1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & +1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & +1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & +1 & +1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & +1 & +1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & +1 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & -1 & 0 & +1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & +1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & -1 & 0 & 0 & -1 \end{bmatrix} \end{matrix},$$

$$B = \begin{matrix} & x_0 & x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_8 \\ \begin{matrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \end{matrix} & \begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{matrix}.$$

3. Визначимо півстепінь входу $s_+(x_i)$ та півстепінь виходу $s_-(x_i)$ кожної вершини x_i , $i = \overline{0,8}$. Для цього скористаємося побудованою матрицею інцидентностей A , а саме: $s_+(x_i)$ – кількість дуг, що виходять із вершини x_i , дорівнює сумі додатних елементів i -го рядка матриці A , а $s_-(x_i)$ – кількість дуг, що заходять у вершину x_i , – сумі від'ємних елементів цього самого рядка, отже:

$$\begin{aligned} s_+(x_0) &= 4, s_-(x_0) = 0; s_+(x_1) = 1, s_-(x_1) = 1; \\ s_+(x_2) &= 1, s_-(x_2) = 1; s_+(x_3) = 2, s_-(x_3) = 1; \\ s_+(x_4) &= 2, s_-(x_4) = 1; s_+(x_5) = 1, s_-(x_5) = 2; \\ s_+(x_6) &= 1, s_-(x_6) = 2; s_+(x_7) = 1, s_-(x_7) = 1; \\ s_+(x_8) &= 0, s_-(x_8) = 4. \end{aligned}$$

Неважко перевірити, що сума півстепенів виходу вершин орграфу дорівнює сумі півстепенів входу і дорівнює кількості всіх дуг:

$$\sum_{i=1}^n s_+(x_i) = \sum_{i=1}^n s_-(x_i) = m = 13.$$

Задача 7. Розглядаючи орграф із задачі 6 (див. рис. 4.9) як сітковий графік (СГ) із відомими тривалостями робіт, знайдіть критичний час $t_{кр.}$ і критичний шлях $\mu_{кр.}$ на ньому.

Розв'язання. За смислом задачі довжина орієнтованого ребра (x_i, x_k) на ньому (x_i, x_k – вершини-події СГ, $i, k \in \{0, 1, 2, \dots, 8\}$, $i < k$) указує на тривалість роботи $t(x_i, x_k)$, яка починається подією x_i і закінчується подією x_k .

Зобразимо кожен вершину-подію x_i СГ кружком, у сектори якого занесемо номер події (i), ранній ($t_p(i)$) і пізній ($t_{п}(i)$) строки її звершення та резерв часу ($R(i)$). Надалі вершини СГ будемо позначати їхніми номерами, тобто замість x_i, x_k писати i, k .

Згідно з алгоритмом знаходження критичного часу і критичного шляху (див. п. 2.3.2), розрахунки на СГ будемо виконувати в чотири етапи.

Спочатку обчислюємо ранній строк звершення кожної події: якщо k -ю подією закінчується одна робота (i, k) , то: $t_p(k) = t_p(i) + t(i, k)$, і якщо k -ю подією закінчується кілька робіт – множина $U_{\rightarrow k}$, то за кожною роботою знаходять свій ранній строк, а шуканим $t_p(k)$ буде найбільший із них:

$$t_p(k) = \max_{(i, k) \in U_{\rightarrow k}} \{t_p(i) + t(i, k)\}.$$

Отже, за вхідними щодо кожної події дугами маємо: $t_p(0) = 0$, оскільки початковій події не передуює ніяка робота;

кожною з подій 1, 2, 3, 4 закінчується одна робота, отже:

$$t_p(1) = t_p(0) + t(0, 1) = 0 + 4 = 4, \quad t_p(2) = t_p(0) + t(0, 2) = 0 + 6 = 6,$$

$$t_p(3) = t_p(0) + t(0, 3) = 0 + 5 = 5, \quad t_p(4) = t_p(1) + t(1, 4) = 4 + 2 = 6;$$

подією 5 закінчуються дві роботи, а саме: $U_{\rightarrow 5} = \{(2,5), (3,5)\}$, тому:

$$t_p(5) = \max\{t_p(2) + t(2,5), t_p(3) + t(3,5)\} = \max\{6 + 6, 5 + 2\} = 12;$$

подією 6 закінчуються дві роботи: $U_{\rightarrow 6} = \{(0,6), (4,6)\}$, через те:

$$t_p(6) = \max\{t_p(0) + t(0,6), t_p(4) + t(4,6)\} = \max\{0 + 7, 6 + 3\} = 9;$$

подією 7 закінчується одна робота, отже:

$$t_p(7) = t_p(5) + t(5,7) = 12 + 8 = 20;$$

подією 8 закінчуються роботи: $U_{\rightarrow 8} = \{(3,8), (4,8), (6,8), (7,8)\}$, тому:

$$t_p(8) = \max\{t_p(3) + t(3,8), t_p(4) + t(4,8), t_p(6) + t(6,8), t_p(7) + t(7,8)\} = \\ = \max\{5 + 5, 6 + 4, 9 + 9, 20 + 4\} = 24.$$

Усі здобуті $t_p(k)$, $k = \overline{0,8}$, уписуємо у ліві сектори вершин-подій СГ (рис. 4.10), причому $t_p(8)$ дає водночас пізній строк звершення кінцевої події $t_{\Pi}(8)$ і критичний час $t_{кр}$: $t_p(8) = t_{\Pi}(8) = t_{кр} = 24$.

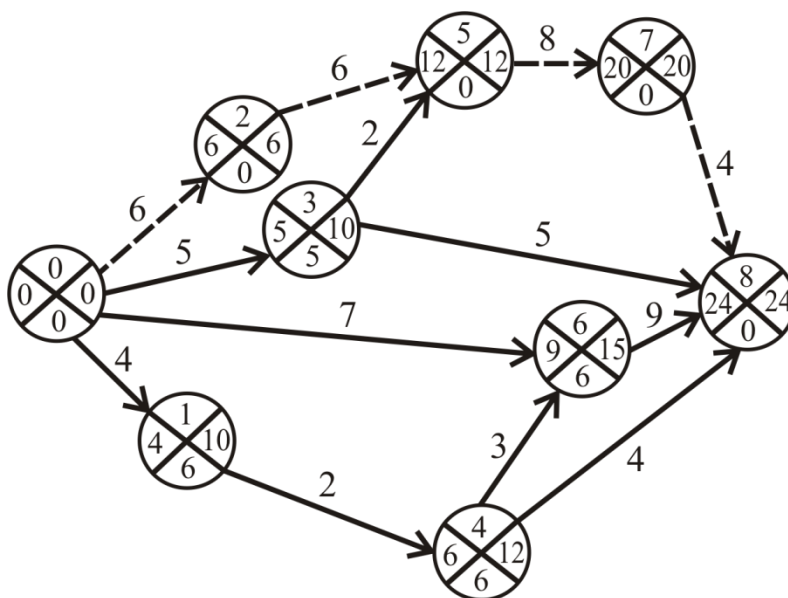


Рис. 4.10. Знаходження критичного часу і критичного шляху на СГ

Далі, відштовхуючись від кінцевої події x_8 , знаходимо пізні строки звершення подій: якщо подією i починається одна робота (i, k) , то: $t_{\Pi}(i) = t_{\Pi}(k) - t(i, k)$, і якщо i -ю подією починається кілька робіт (множина U), то за кожною роботою знаходять свій пізній строк, а шуканим $t_{\Pi}(i)$ буде мінімальний із них:

$$t_{\Pi}(i) = \min_{\substack{(i,k) \in U \\ i \rightarrow}} \{t_{\Pi}(k) - t(i, k)\}.$$

Отже, за вихідними щодо кожної події дугами маємо: кожною з подій 7, 6, 5 починається одна робота, отже:

$$t_{\Pi}(7) = t_{\Pi}(8) - t(7, 8) = 24 - 4 = 20,$$

$$t_{\Pi}(6) = t_{\Pi}(8) - t(6, 8) = 24 - 9 = 15,$$

$$t_{\Pi}(5) = t_{\Pi}(7) - t(5, 7) = 20 - 8 = 12;$$

подією 4 починаються дві роботи а саме: $U = \{(4, 6), (4, 8)\}$, тому:

$$t_{\Pi}(4) = \min\{t_{\Pi}(6) - t(4, 6), t_{\Pi}(8) - t(4, 8)\} = \min\{15 - 3, 24 - 4\} = 12;$$

подією 3 починаються дві роботи $U = \{(3, 5), (3, 8)\}$, через те:

$$t_{\Pi}(3) = \min\{t_{\Pi}(5) - t(3, 5), t_{\Pi}(8) - t(3, 8)\} = \min\{12 - 2, 24 - 5\} = 10;$$

кожною з подій 2, 1 починається одна робота, отже:

$$t_{\Pi}(2) = t_{\Pi}(5) - t(2, 5) = 12 - 6 = 6;$$

$$t_{\Pi}(1) = t_{\Pi}(4) - t(1, 4) = 12 - 2 = 10;$$

подією 0 починаються роботи: $U = \{(0, 1), (0, 2), (0, 3), (0, 6)\}$, тому:

$$\begin{aligned} t_{\Pi}(0) &= \min\{t_{\Pi}(1) - t(0, 1), t_{\Pi}(2) - t(0, 2), t_{\Pi}(3) - t(0, 3), t_{\Pi}(4) - t(0, 4)\} = \\ &= \min\{10 - 4, 6 - 6, 10 - 5, 15 - 7\} = 0. \end{aligned}$$

Усі здобуті $t_{\Pi}(i)$, $i = \overline{0,8}$, уписуємо у праві сектори вершин-подій СГ (див. рис. 4.10).

На наступному кроці підраховуємо резерв часу кожної події за формулою: $R(i) = t_{\Pi}(i) - t_p(i)$, $i = \overline{0,8}$, і вписуємо ці числа в нижні сектори вершин графа (див. рис. 4.10):

$$\begin{aligned} R(0) &= 0, & R(1) &= 10 - 4 = 6, & R(2) &= 6 - 6 = 0, \\ R(3) &= 10 - 5 = 5, & R(4) &= 12 - 6 = 6, & R(5) &= 12 - 12 = 0, \\ R(6) &= 15 - 9 = 6, & R(7) &= 20 - 20 = 0, & R(8) &= 24 - 24 = 0. \end{aligned}$$

Завершальний етап потребує, крім арифметики, візуального аналізу СГ: рухаючись від x_0 до x_8 , знаходимо орланцюг довжини $t_{кр.} = 24$, події якого мають нульовий резерв часу; він і визначає критичні (найбільш відповідальні) роботи комплексу і загалом критичний шлях на СГ (на рис. 4.10 його зображено пунктирною ламаною):

$$\mu_{кр.} = (0 \rightarrow 2 \rightarrow 5 \rightarrow 7 \rightarrow 8).$$

Для своєчасного завершення всіх робіт комплексу критичні роботи $(0,2)$, $(2,5)$, $(5,7)$, $(7,8)$ обов'язково має бути виконано без затримки в часі; виконання роботи $(0,5)$ можна затримати на 5 одиниць часу, а робіт $(0,1)$, $(1,4)$, $(4,6)$ – на 6 одиниць часу, і це не вплине на строк реалізації проєкту загалом без запізнення (за час $t_{кр.} = 24$).

Зазначимо, що СГ може бути таким, що на ньому є декілька критичних (найдовших) шляхів, довжина кожного з яких дорівнює $t_{кр.}$.

Задача 8. Розглядаючи оргграф із задачі 8 (див. рис. 4.9) як транспортну мережу (ТМ) із заданими пропускними спроможностями її дуг, знайдіть максимальний потік φ_z^* на ній та його розподіл за дугами.

Розв'язання. Задачу про максимальний потік на ТМ і його розподіл за дугами будемо розв'язувати на оргграфі, зображеному на рис. 4.11; x_0 – вхід, x_8 – вихід, x_i ($i = \overline{1,7}$) – проміжні вузли ТМ.

За смислом задачі приписані раніше довжини дуг тепер будуть означати їхні пропускні спроможності (їх указано у круглих дужках приблизно посередині орребер), а поруч будемо записувати дугові потоки.

За алгоритмом Форда – Фалкерсона (див. п. 2.4.3), спочатку виберемо якийсь початковий потік $\varphi_{x_0}^0$ із вершини x_0 і здійснимо його розподіл за дугами, урахувавши умову обмеженості (2.11) й умову балансу (2.12) для проміжних вузлів ТМ.

Візьмемо $\varphi_{x_0}^0 = 15$ і здійснимо один із можливих його розподілів за всіма дугами ТМ:

$$\varphi_{x_0}^0 = \varphi(0,1) + \varphi(0,2) + \varphi(0,3) + \varphi(0,6) = 2 + 3 + 5 + 5 = 15;$$

$$\varphi(0,1) = \varphi(1,4) = 2 = c(1,4);$$

$$\varphi(0,2) = \varphi(2,5) = 3;$$

$$\varphi(0,3) = c(0,3) = 5 = \varphi(3,5) + \varphi(3,5) = 1 + 4;$$

$$\varphi(2,5) + \varphi(3,5) = 3 + 1 = \varphi(5,7) = \varphi(7,8) = c(7,8) = 4;$$

$$\varphi(0,6) = 5 = \varphi(6,4) + \varphi(6,8) = 2 + 3;$$

$$\varphi(1,4) + \varphi(6,4) = 2 + 2 = \varphi(4,8) = c(4,8) = 4.$$

На рис. 4.11 відповідні дугові потоки вказано поруч із пропускними спроможностями дуг.

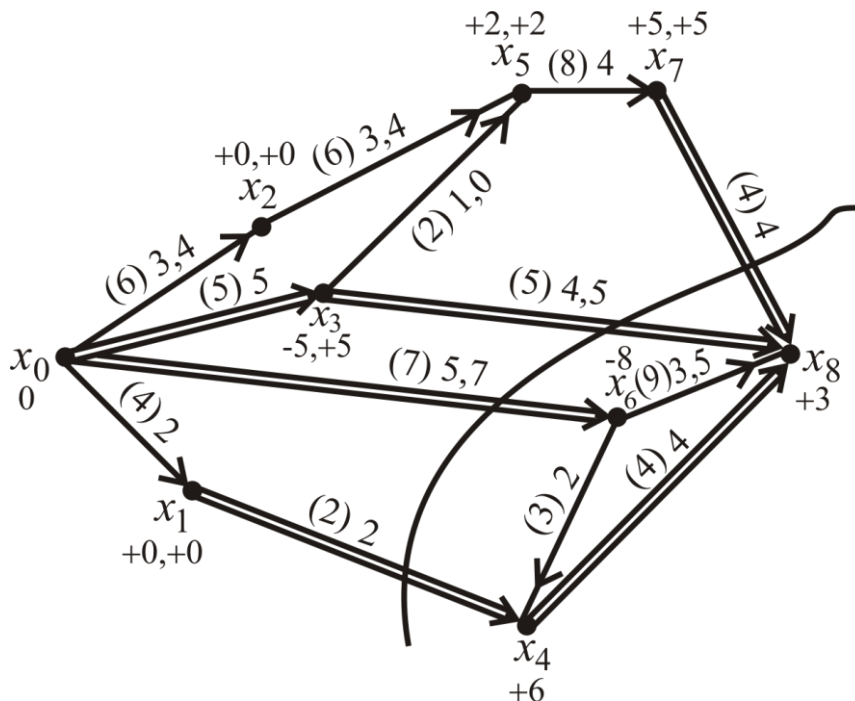


Рис. 4.11. Знаходження максимального потоку на ТМ

Підрахуємо (для контролю) вхідний потік у вершину $x_8 = z$:

$$\varphi_z^0 = \varphi(7,8) + \varphi(3,8) + \varphi(6,8) + \varphi(4,8) = 4 + 4 + 3 + 4 = 15 = \varphi_{x_0}^0.$$

Далі аналізуємо всі шляхи від x_0 до z , із метою встановлення того, повні вони чи неповні:

шляхи (за номерами вузлів) $(0 \rightarrow 3 \rightarrow 8)$, $(0 \rightarrow 3 \rightarrow 5 \rightarrow 7 \rightarrow 8)$, $(0 \rightarrow 2 \rightarrow 5 \rightarrow 7 \rightarrow 8)$, $(0 \rightarrow 1 \rightarrow 4 \rightarrow 8)$, $(0 \rightarrow 6 \rightarrow 4 \rightarrow 8)$ є повними, оскільки містять насичені дуги $(0,3)$, $(7,8)$, $(4,8)$, відповідно;

шлях $(0 \rightarrow 6 \rightarrow 8)$ – неповний, адже не містить жодної насиченої дуги. Пропускні спроможності його дуг $(0,6)$, $(6,8)$ дають можливість збільшити потік на ньому на дві одиниці, після чого дуга $(0,6)$ стане насиченою: $\varphi(0,6) = c(0,6) = 7$, а потік $\varphi_z = \varphi_z^0 + 2$ – повним: $\varphi_z^{\Pi} = 15 + 2 = 17$. Нові дугові потоки $\varphi'(0,6) = 7$ і $\varphi'(6,8) = 5$ указано через кому після попередніх 5, 3.

Переходимо до процедури індексації вершин. Вершині x_0 приписуємо індекс $p_0 = 0$, після чого індексуємо вершини x_1 , x_2 , оскільки дуги $(0,1)$, $(0,2)$ ненасичені: $p_1 = p_2 = +0$. Проіндексована вершина x_0 не дає змоги зважити вершини x_3 і x_6 , адже до них ідуть насичені дуги. Так само не можна зважити x_4 , хоча вершину x_1 проіндексовано.

Далі послідовно дістаємо:

$$p_5 = +2, \text{ бо } \varphi(2,5) = 3 < c(2,5) = 6 \text{ і } x_2 \text{ позначено;}$$

$$p_7 = +5, \text{ адже } \varphi(5,7) = 4 < c(5,7) = 8 \text{ і } x_5 \text{ зважено;}$$

$p_3 = -5$, оскільки дуга $(3,5)$ навантажена ($\varphi(3,5) = 1 > 0$) і x_5 проіндексовано;

$$p_8 = +3, \text{ адже дуга } (3,8) \text{ ненасичена і } x_3 \text{ позначено;}$$

$$p_6 = -8, \text{ оскільки } \varphi(6,8) = 5 > 0 \text{ і } x_8 \text{ зважено;}$$

$p_4 = +6$, зважаючи на те, що $\varphi(6,4) = 2 < c(6,4) = 3$ і вершину x_6 проіндексовано.

На рис. 4.11 індекси вказано над або під позначенням вершин.

Отже, усі вершини виявилися проіндексованими, а це означає, що розподілений повний потік $\varphi_z^{\Pi} = 17$ не є максимальним.

Звертаємося тепер до процедури зміни потоку. За допомогою візуального аналізу графа знаходимо шлях від x_0 до z , усі вершини якого різні і, починаючи із другої, позначені (із точністю до знака) номерами попередніх вершин. У нас шлях μ_{0z} і відповідний кортеж індексів J_{μ} мають такий вигляд:

$$\begin{aligned}\mu_{0z} &= (x_0 \rightarrow x_2 \rightarrow x_5 \rightarrow x_3 \rightarrow x_8), \\ J_{\mu} &= (0, +0, +2, -5, +3).\end{aligned}$$

Зазначимо, що на цьому шляху всі дуги ТМ попутні, крім однієї – $(3,5)$. Попутні дугові потоки $\varphi(0,2)$, $\varphi(2,5)$, $\varphi(3,8)$ збільшуємо на одиницю, унаслідок чого дуга $(3,8)$ стане насиченою: $\varphi'(3,8) = c(3,8) = 5$. Зустрічний потік $\varphi(3,5) = 1$ зменшуємо на одиницю, і тоді дуга $(3,5)$ стане не навантаженою: $\varphi'(3,5) = 0$. Нові дугові потоки на шляху μ_{0z} указано через кому поруч із попередніми потоками. Загалом потік на ТМ збільшився на одиницю і став дорівнювати $\varphi_z^{\Pi} + 1 = 17 + 1 = 18$.

Повертаємося до індексації вершин:

беремо, як і раніше, $p_0 = 0$;

$p_1 = p_2 = +0$, оскільки дуги $(0,1)$, $(0,2)$ ненасичені;

$p_5 = +2$, адже вершину x_2 проіндексовано і дуга $(2,5)$ ненасичена ($\varphi(2,5) = 4 < c(2,5) = 6$);

$p_3 = +5$, оскільки вершину x_5 позначено і дуга $(3,5)$ не навантажена, а отже, і ненасичена;

$p_7 = +5$, адже вершину x_5 позначено і дуга $(5,7)$ ненасичена ($\varphi(5,7) = 4 < c(5,7) = 8$);

вершини x_4 , x_6 , $x_8 = z$ позначити неможливо, оскільки до них від проіндексованих вершин x_0 , x_1 , x_3 , x_7 ідуть тільки насичені дуги.

Таким чином, у процесі повторного зважування вершина z (разом із вузлами x_4 , x_6) виявилася не проіндексованою, а це означає, що потік,

розподілений на ТМ, є максимальним: $\varphi_z^* = 18$; розподіл φ_z^* за дугами ТМ зображено на рис. 4.11.

Множина вершин $B = \{x_4, x_6, z\}$ визначає мінімальний розріз:

$$U_B = \{(7, 8), (3, 8), (0, 6), (1, 4)\},$$

у якого всі вхідні дуги насичені, а вихідних дуг немає. На рис. 4.11. розріз U_B показано лінією, яка перетинає всі його дуги.

Наприкінці зазначимо, що в цих методичних рекомендаціях розглянуто лише деякі основні положення теорії графів, тим не менш, сподіваємося, що вони будуть корисні студентам як для засвоєння теоретичних питань, так і для розв'язання задач на практичних заняттях та під час самостійної роботи.

5. Рекомендована література

5.1. Основна

1. Белов В. В. Теория графов / В. В. Белов, Е. М. Воробьев, В. Е. Шаталов. – Москва : Высшая школа, 1976. – 392 с.
2. Берж К. Теория графов и ее применение / К. Берж. – Москва : Изд-во иностранной литературы, 1962 – 320 с.
3. Гладких О. Б. Основные понятия теории графов : учеб. пособ. / О. Б. Гладких. – Елец : ЕГУ им. И. А. Бунина, 2008. – 175 с.
4. Глушкин Л. М. Задачи и алгоритмы комбинаторики и теории графов / Л. М. Глушкин, Л. А. Шор, В. Я. Шварц. – Донецк : ДПИ, 1982. – 112 с.
5. Карнаух Т. О. Теорія графів у задачах : навч. посіб. / Т. О. Карнаух, А. Б. Ставровський. – Київ : ВПЦ "Київський університет", 2004. – 76 с.
6. Москинова Г.И. Дискретная математика / Г. И. Москинова. – Москва : Логос, 2000. – 240 с.
7. Нефедов В. Н. Курс дискретной математики / В. Н. Нефедов, В. А. Осипова. – Москва : Наука, 1992. – 268 с.
8. Ніколаєва К. В. Дискретний аналіз. Графи та їх застосування в економіці : навч.-метод. посіб. / К. В. Ніколаєва, В. В. Койбічук. – Суми : УАБС НБУ, 2007. – 84 с.
9. Новиков Ф. А. Дискретна математика для програмістів : учеб. для вузов / Ф. А. Новиков. – 2-е изд. – Санкт-Петербург : Питер, 2007. – 364 с.
10. Сенчуков В. Ф. Основи дискретної математики : навч. посіб. / В. Ф. Сенчуков, Т. В. Денисова. – Харків : Вид. ХНЕУ, 2007. – 344 с.
11. Сигорский В. П. Математический аппарат инженера / В. П. Сигорский. – Киев : Техника, 1975. – 766 с.

5.2. Додаткова

12. Дискретна математика : методичні рекомендації до лабораторних робіт для студентів галузі знань 12 "Інформаційні технології" першого (бакалаврського) рівня / уклад. Т. В. Денисова, В. Ф. Сенчуков. – Харків : ХНЕУ ім. С. Кузнеця, 2018. – 114 с.
13. Евстигнеев В. А. Применение теории графов в программировании / В. А. Евстигнеев. – Москва : Наука, 1985. – 352 с.

14. Емеличев В. А. Лекции по теории графов / В. А. Емеличев, Д. И. Мельников. – Москва : Наука, 1990. – 384 с.
15. Зыков А. А. Основы теории графов / А. А. Зыков. – Москва : Наука, 1987. – 381 с.
16. Кристофидес Н. Теория графов. Алгоритмический подход / Н. Кристофидес. – Москва : Мир, 1978. – 432 с.
17. Мальцев Ю. Н. Введение в дискретную математику (элементы комбинаторики, теории графов и теории кодирования) : учеб. пособ. / Ю. Н. Мальцев, Е. П. Петров. – Барнаул : Изд. Алт. ун-та, 1997. – 138 с.
18. Оре О. Теория графов / О. Оре. – Москва : Мир, 1979. – 320 с.
19. Трохимчук Р. М. Збірник задач із теорії графів : навч. посіб. / Р. М. Трохимчук. – Київ : РВЦ "Київський університет", 1998. – 57 с.
20. Харрари Ф. Теория графов / Ф. Харрари. – Москва : Мир, 1973. – 208 с.
21. Яблонский В. Н. Введение в дискретную математику / В. Н. Яблонский. – Москва : Наука, 1991. – 312 с.

5.3. Інформаційні ресурси

22. Бондаренко М. Ф. Комп'ютерна дискретна математика [Електронний ресурс] : підручник / М. Ф. Бондаренко, Н. В. Білоус, А. Г. Руткас. – Харків : Компанія СМІТ, 2004. – 480 с. – Режим доступу : <http://www.ex.ua/75299234>.
23. Боднарчук Ю. В. Основи дискретної математики [Електронний ресурс] : навч. посіб. / Ю. В. Боднарчук, Б. В. Олійник. – Київ : НаУКМА, 2007. – 138 с. – Режим доступу : <http://www.twirpx.com/file/589927>.
24. Стрелковська І. В. Дискретна математика [Електронний ресурс] : навч. посіб. / І. В. Стрелковська, А. Г. Буслаєв, О. М. Харсун. – Одеса : ОНАЗ ім. О. С. Попова, 2010. – 196 с. – Режим доступу : http://www.dut.edu.ua/uploads/l_373_44193539.pdf.
25. Стрелковська І. В. Практичні заняття з дискретної математики [Електронний ресурс] / І. В. Стрелковська, А. Г. Буслаєв, В. М. Вишневська. – Одеса : ОНАЗ ім. О. С. Попова, 2008. – 49 с. – Режим доступу : <https://metod.onat.edu.ua/metod/download/443/ua>.
26. Швачич Г. Г. Основи дискретної математики. Ч. 3. Основи теорії графів [Електронний ресурс] : навч. посіб. / Г. Г. Швачич, Г. М. Бартенев, О. В. Оніщенко. – Дніпропетровськ : НМетАУ, 2014. – 67 с. – Режим доступу : http://nmetau.edu.ua/file/shvachich_g.g._osnovy_diskretnoy_matematiki._chast_iii._osnovy_teorii_grafov_2014.pdf.

Зміст

Вступ.....	3
1. Неорієнтовані графи	5
1.1. Означення основних понять, способи задання	5
1.2. Підграф. Маршрут, ланцюг, цикл	8
1.3. Відшукання ланцюга найменшої довжини	9
1.4. Матриця відстаней графа	11
1.5. Зв'язність графів, дерева та ліс	12
1.6. Побудова економічного дерева.....	15
1.7. Операції над графами.....	17
1.8. Класифікація графів (за структурою).	
Ізоморфні та гомеоморфні графи	18
1.9. Планарність графів	22
1.10. Ейлерові та гамільтонові графи	24
Запитання для самоконтролю за засвоєнням матеріалу.....	28
2. Орієнтовані графи (орграфи)	30
2.1. Означення основних понять, способи задання	30
2.2. Досяжність вершин, відстань між вершинами, типи зв'язності орграфа	33
2.3. Сіткові графіки.....	37
2.3.1. Сіткові графіки: основні означення, правила побудови	37
2.3.2. Задача знаходження критичного часу і критичного шляху на сітковому графіку	39
2.4. Транспортні мережі.....	46
2.4.1. Транспортні мережі: основні означення, знаходження повного потоку	46
2.4.2. Розрізи на ТМ	49
2.4.3. Задача знаходження максимального потоку на ТМ.	
Алгоритм Форда – Фалкерсона	52
2.5. Упорядкування вершин і дуг плоского орграфа	59
Запитання для самоконтролю за засвоєнням матеріалу.....	63
3. Варіанти задач самостійної контрольної роботи.....	65
4. Зразки розв'язання задач самостійної контрольної роботи.....	77
5. Рекомендована література.....	97
5.1. Основна	97
5.2. Додаткова	97
5.3. Інформаційні ресурси.....	98

НАВЧАЛЬНЕ ВИДАННЯ

ДИСКРЕТНА МАТЕМАТИКА

**Методичні рекомендації
до самостійної роботи з теми "Теорія графів"
для студентів галузі знань 12 "Інформаційні технології"
першого (бакалаврського) рівня**

Самостійне електронне текстове мережеве видання

Укладачі: **Денисова** Тетяна Володимирівна
Сенчуков Віктор Федорович

Відповідальний за видання *Л. М. Малярець*

Редактор *О. Г. Доценко*

Коректор *О. Г. Доценко*

План 2020 р. Поз. № 24 ЕВ. Обсяг 100 с.

Видавець і виготовлювач – ХНЕУ ім. С. Кузнеця, 61166, м. Харків, просп. Науки, 9-А

*Свідоцтво про внесення суб'єкта видавничої справи до Державного реєстру
ДК № 4853 від 20.02.2015 р.*