МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ОДЕСЬКИЙ НАЦІОНАЛЬНИЙ МОРСЬКИЙ УНІВЕРСИТЕТ

Кафедра вищої математики

ДИСКРЕТНА МАТЕМАТИКА
Частина 2

Навчальний посібник

Одеса — 2026

Навчальний посібник розроблено кандидатом фізико-математичних наук **Григор'євим Юрієм Олександровичем** – доцентом кафедри "Вища математика" Одеського національного морського університету.

Посібник схвалено Вченою радою Одеського національного морського університету 24 березня 2026 року (протокол № __11__).

Рецензенти – доктор технічних наук, професор, завідувач кафедри інформаційних технологій Національного університету «Одеська політехніка» **В.В. Вичужанін** (м. Одеса),

доктор технічних наук, професор **Л.І. Коростильов** (Національний університет кораблебудування, м. Миколаїв),

доктор педагогічних наук, кандидат технічних наук, професор **Л.Д. Герганов** (Дунайський інститут НУ «Одеська морська академія», м. Ізмаїл).

ЗМІСТ

РОЗДІЛ 5. БУЛЕВІ ФУНКЦІЇ

5.1. Основні булеві функції та їх задання таблицями істинностей

Булевою функцією називається функція виду

$$y = f(x_1, x_2, ..., x_n),$$

де кожен із аргументів x_i та функція y можуть приймати значення 0 або1. Множина, яка складається з цих елементів позначається $B = \{0,1\}$. Змінні, які приймають значення із множини B, називаються ***булевими змінними.*** Оскільки

$$x_1 \in B, \quad x_2 \in B, \quad ..., \quad x_n \in B,$$

то вектор

$$(x_1, x_2, ..., x_n) \in B \times B \times ... \times B = B^n.$$

Тому булеву функцію можна визначити так:

$$f : B^n \to B.$$

Спочатку розглянемо булеву функцію однієї змінної $y = f(x)$. Таких функцій чотири: константа 0, x, $\overline{x}$ та константа 1. Запишемо їх в таблицю.

x	0	x	$\overline{x}$	1
0	0	0	1	1
1	0	1	0	1

Тепер розглянемо функцію двох змінних $z = f(x, y)$. Підрахуємо кількість таких функцій. Кожний із аргументів x, y може приймати 2 значення. Тому числових векторів (x, y) буде $2 \cdot 2 = 4$. Оскільки на кожному числовому наборі (x, y) функція y може прийняти 2 значення, то кількість булевих функцій двох змінних дорівнює $2 \cdot 2 \cdot 2 \cdot 2 = 16$. Запишемо 7 основних функцій (Таблиця 5.1).

Таблиця 5.1

x	y	$x \wedge y$	$x \vee y$	$x \to y$	$x \leftrightarrow y$	$x + y$	$x \mid y$	$x \downarrow y$
0	0	0	0	1	1	0	1	1
0	1	0	1	1	0	1	1	0
1	0	0	1	0	0	1	1	0
1	1	1	1	1	1	0	0	0

Перші чотири функції (кон'юнкція, диз'юнкція, імплікація та еквіваленція) нам вже відомі. Далі кон'юнкцію будемо позначати як звичайне множення:

$$x \cdot y = x \wedge y.$$

Функція $x+y$ називається сумою за модулем 2, або – виключним або (x або y, але разом – ніколи).

Функція $x \mid y$ називається штрихом Шеффера (її можна називати просто штрихом). Із таблиці 1.1 видно, що ця функція є запереченням кон'юнкції:

$$x \mid y = \overline{x \wedge y}\,.$$

Функція $x \downarrow y$ називається стрілкою Пірса (її можна називати просто стрілкою). Із таблиці 1.1 видно, що ця функція є запереченням диз'юнкції:

$$x \downarrow y = \overline{x \vee y}\,.$$

5.2. Основні закони булевої алгебри

Булева алгебра складається із множини $B = \;\; 0{,}1$ разом із визначеними на ній операціями диз'юнкції ($\vee$), кон'юнкції ($\wedge$) та заперечення ($\neg$), тобто

$$A = \left\langle B, \vee, \wedge, \neg \right\rangle.$$

Ця алгебра ізоморфна алгебрі теорії множин з операціями об'єднання, перетину та доповненням. Це означає, що закони булевої алгебри можна отримати із законів теорії множин, замінивши множини на булеві змінні, об'єднання – на диз'юнкцію, перетин – на кон'юнкцію, доповнення – на заперечення. В результаті будемо мати

1. Комутативні закони

$$x \vee y = y \vee x\,, \qquad x \wedge y = y \wedge x$$

2. Асоціативні закони

$$x \vee y \;\; \vee z = x \vee \;\; y \vee z \;\; , \qquad x \vee y \;\; \vee z = x \vee \;\; y \vee z$$

3. Закони нуля та одиниці

$$x \vee 0 = x\,, \qquad x \wedge 1 = x$$
$$x \vee 1 = 1\,, \qquad x \wedge 0 = 0$$

4. Закони ідемпотентності

$$x \vee x = x\,, \qquad x \wedge x = x$$

5. Дистрибутивні закони

$$x \wedge \;\; y \vee z \;\; = \;\; x \wedge y \;\; \vee \;\; x \wedge z \;\; , \qquad x \vee \;\; y \wedge z \;\; = \;\; x \vee y \;\; \wedge \;\; x \vee z$$

6. Закони доповнення

$$x \vee \overline{x} = 1\,, \qquad x \wedge \overline{x} = 0$$
$$\overline{1} = 0, \qquad \overline{0} = 1$$
$$\overline{\overline{x}} = x\,, \qquad \overline{\overline{x}} = x$$

7. Закони де Моргана

$$\overline{x \vee y} = \overline{x} \wedge \overline{y}\,, \qquad \overline{x \wedge y} = \overline{x} \vee \overline{y}$$

Записані вище закони ми помістили у два стовпці. Кожну рівність з правого стовпця таблиці можна отримати з лівого стовпця заміною $\vee$ на $\wedge$, 0 на 1 та навпаки. Така властивість називається ***законом двоїстості.*** Отже, знаючи який-небудь закон булевої алгебри можна записати двоїстий закон і він буде правильним. Це твердження ми приймемо без доведення.

Будь-який закон булевої алгебри можна довести за допомогою таблиць істинності. Можна використовувати й інший метод, надаючи булевим змінним певні значення. Доведемо, наприклад, дистрибутивний закон

$$x \vee \ y \wedge z \ = \ x \vee y \ \wedge \ x \vee z \ .$$

При значенні $x = 0$ формула прийме вигляд $y \wedge z = y \wedge z$, а при значенні $x = 1$ буде $1 = 1$.

Ми упевнились у справедливості дистрибутивного закону при всіх значеннях x.

Надалі у записах булевих функцій будемо економити на дужках. При цьому старшою операцією будемо вважати заперечення, потім, за старшинством, – кон'юнкцію, диз'юнкцію, імплікацію та еквіваленцію. Враховуючи це правило, попередній дистрибутивний закон можна переписати так:

$$x \vee yz = \ x \vee y \quad x \vee z \ .$$

Наведемо ще один закон. Він випливає із наведених вище і часто використовується:

$$x \vee xy = x, \qquad x(x \vee y) = x.$$

Тут дві формули. Одна із них двоїста по відношенню до другої. Формулу зліва читають так: коротша кон'юнкція x поглинає довшу xy. Тому написані закони називаються ***законами поглинання***. Першу формулу можна довести наступним чином:

$$x \vee xy = x1 \vee xy = x \ 1 \vee y \ = x1 = x.$$

При доведенні ми скористались дистрибутивним законом і законом одиниці. Друга формула справедлива на підставі двоїстості.

5.3. Досконалі форми булевих функцій

В даному пункті ми доведемо, що будь-яку булеву функцію можна представити застосовуючи лише операції заперечення ($\neg$), кон'юнкції ($\wedge$) та диз'юнкції ($\vee$).

Спочатку розглянемо функцію трьох змінних $u = f\left(x, y, z\right)$, яка приймає значення $u = 1$ лише на наборі $\left(1,0,1\right)$, тобто $x = 1,\ y = 0,\ z = 1$, а на всіх інших наборах значення функції u дорівнюють 0. Таку функцію можна подати у вигляді кон'юнкції змінних x, y та z, знак заперечення потрібно поставити над y:

$$u = x\bar{y}z$$

бо тільки на цьому наборі $(1,0,1)$ значення функції u дорівнює 1, а на всіх інших наборах значення функції дорівнюють 0. Це так тому, що кон'юнкція дорівнює 1 лише тоді коли всі її компоненти дорівнюють 1.

А тепер припустимо, що наша функція дорівнює 1 на двох наборах: $(1,0,1)$ і $(0,1,1)$. На наборі $(0,1,1)$ кон'юнкція

$$\bar{x}yz$$

дорівнює 1 і лише на цьому наборі вона дорівнює 1, а на всіх інших наборах вона дорівнює 0. А на обох наборах одиниці дорівнює диз'юнкція отриманих раніше кон'юнкцій:

$$u = x\bar{y}z \vee \bar{x}yz. \qquad (5.1)$$

Зазначимо, що функція (5.1) дорівнює одиниці лише на наборах $(1,0,1)$ і $(0,0,1)$, на всіх інших наборах вона дорівнює нулю. Представлення функції u у вигляді (5.1) називається її досконалою диз'юнктивною нормальною формою.

Означення. Досконалою диз'юнктивною нормальною формою (ДДНФ) називається диз'юнкція кон'юнкцій. Кожна кон'юнкція повинна містити всі змінні функції безпосередньо або із запереченням.

Правило представлення функції у формі ДДНФ.

1. Запишемо функцію у табличному вигляді.
2. Виділимо набори, на яких значення функції дорівнює 1. Для кожного такого набору запишемо кон'юнкцію всіх змінних функції. Поставимо знак заперечення над тими змінними, значення яких в наборі дорівнюють нулю.
3. Отримані кон'юнкції з'єднаємо знаком диз'юнкції.

З цього правила витікає наступна теорема.

Теорема. Будь-яку булеву функцію, якщо вона не константа 0, можна представити у ДДНФ і це представлення єдине.

Якщо деякі кон'юнкції містять не всі змінні, то таке представлення функції називається ***диз'юнктивною нормальною формою (ДНФ).***

Наприклад, функція

$$f(x,y,z) = x\bar{y}z \vee \bar{x}y$$

представлена у формі ДНФ, а не у формі ДДНФ бо кон'юнкція $\bar{x}y$ не містить змінної z. Для представлення цієї функції у вигляді ДДНФ, скористаємось наступним прийомом

$$\bar{x}y = \bar{x}y1 = \bar{x}y(z \vee \bar{z}) = \bar{x}yz \vee \bar{x}y\bar{z}.$$

Тоді

$$f(x,y,z) = x\bar{y}z \vee \bar{x}yz \vee \bar{x}y\bar{z}.$$

Тепер вже функція $f(x,y,z)$ представлена у формі ДДНФ. Це представлення має ту перевагу, що воно є єдине, якщо не зважати на те, що компоненти кон'юнкції або диз'юнкції можна міняти місцями. У вигляді ж ДНФ функцію можна записати багатьма способами.

Запишемо основні булеві функції двох змінних у вигляді ДДНФ. Скориставшись таблицею 1.1 значень цих функцій, отримаємо:

$$x \leftrightarrow y = \overline{x}\,\overline{y} \vee xy\,, \tag{5.2}$$

$$x + y = \overline{x}y \vee x\overline{y}\,. \tag{5.3}$$

Якщо серед значень функції багато одиниць і мало нулів, то її запис буде компактнішим не у формі ДДНФ, а в іншій формі, яку ми розглянемо далі.

Означення. Досконалою кон'юнктивною нормальною формою (ДКНФ) називається кон'юнкція диз'юнкцій. Кожна диз'юнкція повинна містити всі змінні функції безпосередньо або із запереченням.

Якщо деякі диз'юнкції містять не всі змінні, то таке представлення функції називається ***кон'юнктивною нормальною формою (КНФ).***

Знову розглянемо функцію $x \longrightarrow y$. Звернемо увагу, що її значення дорівнює нулю лише на одному наборі $1,0$. Це означає, що її просто можна записати у вигляді наступної диз'юнкції змінних

$$x \to y = \overline{x} \vee y, \tag{5.4}$$

бо тільки на наборі $1,0$ диз'юнкція $\overline{x} \vee y$ дорівнює 0.

Формули (5.2), (5.3) і (5.4) бажано запам'ятати.

Правило представлення функції у формі ДКНФ.

1. Запишемо функцію у табличному вигляді.

2. Виділимо набори, на яких значення функції дорівнює 0. Для кожного такого набору запишемо диз'юнкцію всіх змінних функції. Поставимо знак заперечення над тими змінними, значення яких в наборі дорівнюють одиниці.

3. Отримані диз'юнкції з'єднаємо знаком кон'юнкції.

Теорема. Будь-яку булеву функцію, якщо вона не константа 1, можна представити у ДКНФ і це представлення єдине.

Приклад. Побудуйте ДДНФ і ДКНФ для наступної функції

$$f\ \ x,y,z\ \ = \overline{\ x \leftrightarrow y\ \ \to yz} \vee x.$$

Розв'язання. Побудуємо таблицю істинностей функції

x	y	z	$x \leftrightarrow y$	yz	$x \leftrightarrow y\ \ \to yz$	$\overline{x \leftrightarrow y\ \ \to yz}$	$f\ \ x,y,z$
0	0	0	1	0	0	1	1
0	0	1	1	0	0	1	1
0	1	0	0	0	1	0	0
0	1	1	0	1	1	0	0
1	0	0	0	0	1	0	1
1	0	1	0	0	1	0	1
1	1	0	1	0	0	1	1
1	1	1	1	1	1	0	1

На основі цієї таблиці запишемо ДДНФ і ДКНФ нашої функції:

$$f(x,y,z) = \overline{x}\overline{y}\overline{z} \vee \overline{x}\overline{y}z \vee x\overline{y}\overline{z} \vee x\overline{y}z \vee xy\overline{z} \vee xyz\,,$$

$$f(x,y,z) = (x \vee \overline{y} \vee z)(x \vee \overline{y} \vee \overline{z})\,.$$

Другий спосіб зведення функції до ДДНФ. Розв'яжемо задачу, користуючись законами булевої алгебри. За формулою (5.4), отримаємо

$$f(x,y,z) = \overline{\overline{(x \leftrightarrow y)} \vee yz} \vee x\,.$$

Тепер скористуємось законами де Моргана та формулою (5.2):

$$f(x,y,z) = (x \leftrightarrow y)\overline{yz} \vee x = (\overline{x}\overline{y} \vee xy)(\overline{y} \vee \overline{z}) \vee x\,.$$

За дистрибутивним законом відкриємо дужки

$$(\overline{x}\overline{y} \vee xy)(\overline{y} \vee \overline{z}) = \overline{x}\overline{y} \vee \overline{x}\overline{y}\overline{z} \vee xy\overline{z}\,.$$

Тепер нашу функцію можна записати у вигляді ДНФ:

$$f(x,y,z) = \overline{x}\overline{y} \vee \overline{x}\overline{y}\overline{z} \vee xy\overline{z} \vee x\,. \tag{5.5}$$

Для представлення функції у вигляді ДДНФ, запишемо кон'юнкції так, щоб вони містили всі змінні

$$\overline{x}\overline{y} = \overline{x}\overline{y}(z \vee \overline{z}) = \overline{x}\overline{y}z \vee \overline{x}\overline{y}\overline{z}\,,$$

$$x = x(y \vee \overline{y}) = xy \vee x\overline{y} = xy(z \vee \overline{z}) \vee x\overline{y}(z \vee \overline{z}) =$$
$$= xyz \vee xy\overline{z} \vee x\overline{y}z \vee x\overline{y}\overline{z}.$$

Підставляючи отримані вирази у нашу функцію і користуючись законом ідемпотентності, остаточно отримаємо

$$f(x,y,z) = \overline{x}\overline{y}z \vee \overline{x}\overline{y}\overline{z} \vee xy\overline{z} \vee xyz \vee x\overline{y}z \vee x\overline{y}\overline{z}\,.$$

Другий спосіб зведення функції до ДКНФ. Скористаємось записом функції у вигляді ДНФ (5.5). Спростимо цю функцію. За законом поглинання

$$\overline{x}\overline{y} \vee \overline{x}\overline{y}\overline{z} = \overline{x}\overline{y}, \quad xy\overline{z} \vee x = x\,.$$

Тому нашу функцію можна записати так:

$$f(x,y,z) = \overline{x}\overline{y} \vee x\,.$$

За дистрибутивним законом

$$f(x,y,z) = (\overline{x} \vee x)(\overline{y} \vee x) = x \vee \overline{y}\,.$$

Виходить, що функція $f(x,y,z)$ залежить лише від двох змінних: x та y. Щоб записати її у вигляді ДКНФ як функцію трьох змінних, ведемо змінну z і ще раз скористаємось дистрибутивним законом:

$$f(x,y,z) = x \vee \overline{y} = x \vee \overline{y} \vee z\overline{z} = (x \vee \overline{y} \vee z)(x \vee \overline{y} \vee \overline{z})\,.$$

5.4. Мінімізація булевих функцій

Під мінімізацією булевої функції розуміють її спрощення. Довільну булеву функцію можна єдиним способом представити у вигляді ДДНФ, але багатьма способами у вигляді ДНФ.

Означення. Мінімальною ДНФ розуміють таку ДНФ, яка містить найменшу кількість операцій диз'юнкцій і кон'юнкцій серед всіх ДНФ, які представляють дану функцію.

Мінімізуємо функцію, яка в попередньому розділі вже представлена у вигляді ДДНФ:

$$f\left(x,y,z\right)=\overline{x}\overline{y}\overline{z}\vee\overline{x}\overline{y}z\vee x\overline{y}\overline{z}\vee x\overline{y}z\vee xy\overline{z}\vee xyz.$$

Помітимо, що кон'юнкції $\overline{x}\overline{y}\overline{z}$ і $\overline{x}\overline{y}z$ відрізняються лише одним символом (z і $\overline{z}$). За дистрибутивним законом диз'юнкцію цих кон'юнкцій можна спростити

$$\overline{x}\overline{y}\overline{z}\vee\overline{x}\overline{y}z=\overline{x}\overline{y}\left(\overline{z}\vee z\right)=\overline{x}\overline{y}1=\overline{x}\overline{y}.$$

Помітимо, що результатом спрощення є кон’юнкція тих змінних, які не змінюються. Проведену операцію називають склеюванням кон’юнкцій. Аналогічно можна склеїти і наступні дві кон’юнкції: $x\overline{y}\overline{z}$ і $x\overline{y}z$:

$$x\overline{y}\overline{z}\vee x\overline{y}z=x\overline{y}.$$

Отримані в результаті склеювання кон’юнкції $\overline{x}\overline{y}$ і $x\overline{y}$ також відрізняються лише однією компонентою (y і $\overline{y}$), тому

$$\overline{x}\overline{y}\vee x\overline{y}=\left(\overline{x}\vee x\right)\overline{y}=\overline{y}.$$

Для того, щоб помітити всі пари кон'юнкцій, які відрізняються лише однією компонентою, їх записують у таблицю (кожну кон’юнкцію ДДНФ записують у окрему клітинку). Рядки і стовпці цієї таблиці позначають так, щоб сусідні клітинки (по горизонталі і по вертикалі) відрізнялись лише одним символом. Така таблиця називається ***картою Карно.*** В нашому прикладі карту Карно можна скласти наступним чином:

	xy	$\overline{x}y$	$\overline{x}\overline{y}$	$x\overline{y}$
z	1		1	1
$\overline{z}$	1		1	1

Тут символом 1 позначені ті клітинки, в яких повинні розміщуватися кон’юнкції нашої ДДНФ. Склеювати можна ті кон’юнкції, які розташовані в таблиці поруч (по горизонталі або по вертикалі). Звичайно їх об’єднують у групи. Для функції трьох змінних група може складатися з 2 або 4 кон’юнкцій. В нашому прикладі ми спочатку об’єднали у групу кон’юнкції, які розташовані у передостанньому стовпчику таблиці. В результаті склеювання отримали $\overline{x}\overline{y}$. Звернемо увагу на те, що $\overline{x}$ і $\overline{y}$ не змінювалися в клітинках передостаннього стовпця. Тому їх кон’юнкція стала результатом склеювання. Потім ми об'єднали у групу кон'юнкції останнього стовпця. В цьому стовпці не змінюються x і $\overline{y}$. В результаті склеювання ми і отримали їх кон'юнкцію $x\overline{y}$. Можна в одну групу об'єднати всі чотири кон'юнкції, що розташовані в передостанньому і останньому стовпцях. В клітинках цього стовпця не змінюється лише елемент $\overline{y}$. Тому ми його і отримали в результаті склеювання.

У нас залишились незадіяними ще дві кон'юнкції, що розташовані у першому стовпці. Їх можна об'єднати в одну групу і одержати при склеюванні xy, але краще помітити, що елементи першого стовпця таблиці і відповідні елементи останнього стовпця також відрізняються лише одним елементом. Це означає, що їх можна об'єднати в одну групу. В клітинках цієї групи не змінюється x. Значить x є результатом склеювання елементів утвореної групи. Запишемо відповідь мінімізації булевої функції:

$$f\;\; x,y,z\;\; = x \vee \overline{y}\,.$$

В результаті мінімізації з'ясувалося, що функція $f(x,y,z)$ суттєво залежить лише від двох змінних.

Зауважимо, що елементи останнього стовпця ми використовували двічі при склеюванні, хоча у нашу функцію вони входять лише один раз. Враховуючи закон ідемпотентності, їх можна записувати і використовувати скільки завгодно разів.

Далі розглянемо застосування карти Карно для функцій чотирьох змінних, наприклад,

$$f\left(x,y,z,t\right)=\overline{x}\overline{y}zt \vee \overline{x}y\overline{z}\overline{t} \vee \overline{x}\overline{y}\overline{z}t \vee \overline{x}yzt \vee x\overline{y}\overline{z}t \vee x\overline{y}zt \vee xy\overline{z}\overline{t} \vee xy\overline{z}t\,.$$

Перше питання потрібно вирішити: із скількох клітинок буде складатися карта Карно? Змінних чотири. Кожна змінна може увійти у кон'юнкцію у двох виглядах: із знаком заперечення чи без. За правилом множення комбінаторики, максимальна кількість кон'юнкцій дорівнює $2\cdot 2\cdot 2\cdot 2=16$. Отже, карта Карно повинна складатись з 16 клітинок: чотирьох рядків і чотирьох стовпців. Стовпці будемо позначати іксами та ігреками, а рядки – змінними z і t. В нашому прикладі карта Карно може мати наступний вигляд:

	xy	$x\overline{y}$	$\overline{x}\overline{y}$	$\overline{x}y$
zt		1	1	1
$\overline{z}t$	1	1		1
$\overline{z}\overline{t}$	1			1
$z\overline{t}$				

У групу A об'єднаємо чотири одинички, які стоять у першому і четвертому стовпцях, у другому і третьому рядках. В цій групі не змінюється y та $\overline{z}$. Значить $A=y\overline{z}$. У групу B об'єднаємо одинички, що стоять у другому стовпці. В цій групі не змінюється $x\overline{y}$, значить, $B=x\overline{y}$. Нарешті, дві одинички у першому рядку в останніх стовпцях об’єднаємо у групу C, $C = \bar{x}zt$. Тепер всі одинички охоплюються групами A, B і C:

$$f\left(x, y, z, t\right) = A \vee B \vee C,$$

тобто

$$f\left(x, y, z, t\right) = y\bar{z} \vee x\bar{y} \vee \bar{x}zt.$$

5.5. Поняття про повноту системи функцій

Із правил побудови ДДНФ та ДКНФ випливає, що довільну булеву функцію можна виразити через кон'юнкцію, диз'юнкцію та заперечення.

Означення. Множина функцій, через які можна виразити будь-яку булеву функцію, називається ***повною системою функцій.***

Отже, $\neg, \wedge, \vee$ – повна система функцій. Цю систему можна зменшити, якщо виразити, наприклад, диз'юнкцію через кон'юнкцію та заперечення. Користуючись законом де Моргана, отримаємо

$$\overline{x \vee y} = \bar{x} \wedge \bar{y}, \text{ або } x \vee y = \overline{\bar{x} \wedge \bar{y}}.$$

Звідси висновок: $\neg, \wedge$ – повна система функцій.

Можна навпаки, кон'юнкцію виразити через диз'юнкцію та заперечення:

$$\overline{x \wedge y} = \bar{x} \vee \bar{y}, \quad x \wedge y = \overline{\bar{x} \vee \bar{y}}.$$

Таким чином, $\neg, \vee$ – також повна система функцій.

Ми бачимо, що повна система функцій може складатися з трьох, двох і навіть однієї функції. Доведемо, що штрих Шеффера $|$ – повна система функції. Нагадаємо, що функція | визначається формулою

$$x \mid y = \overline{xy}.$$

Для доведення цього твердження достатньо, наприклад, заперечення та диз'юнкцію виразити через штрих:

$$\bar{x} = \overline{xx} = x \mid x,$$

$$x \vee y = \overline{\bar{x} \wedge \bar{y}} = \bar{x} \mid \bar{y} = \; x \mid x \;\; \mid \;\; y \mid y \;.$$

Можна, якщо цікаво, ще й кон'юнкцію виразити через штрих:

$$xy = \overline{\overline{xy}} = \overline{x \mid y} = \; x \mid y \;\; \mid \;\; x \mid y \;.$$

Ми довели, що штрих Шеффера $|$ – повна система функції. Аналогічно доводиться, що стрілка Пірса $\downarrow$ – також повна система функцій (завдання для самостійного розв'язання).

Розплатою за малу кількість операцій, якими записується функція, є громіздкість формул.

5.6. Детальніше про повноту системи функцій

В цьому розділі сформулюємо критерій повноти довільної системи функцій $f_1, f_2, \ldots, f_n$. Для цього потрібно ввести деякі означення.

Означення. Булева функція $f(x_1, x_2, \ldots, x_n)$ називається ***функцією, що зберігає константу 0,*** якщо $f(0,0,\ldots,0) = 0$.

Приклади функцій, які зберігають константу 0: 0, x, xy, $x \vee y$, $x + y$. Приклади функцій, які не зберігають константу 0: $\overline{x}$, $x \to y$.

Означення. Булева функція $f(x_1, x_2, \ldots, x_n)$ називається ***функцією, що зберігає константу 1,*** якщо $f(1,1,\ldots,1) = 1$.

Приклади функцій, які зберігають константу 1: 1, x, xy, $x \vee y$. Приклади функцій, які не зберігають константу 1: 0, $\overline{x}$, $x + y$.

Означення. Функція $f^*(x_1, x_2, \ldots, x_n)$ називається ***двоїстою*** до функції $f(x_1, x_2, \ldots, x_n)$, якщо $f^*(x_1, x_2, \ldots, x_n) = \overline{f}(\overline{x}_1, \overline{x}_2, \ldots, \overline{x}_n)$.

Приклад. Знайти функцію, двоїсту диз'юнкції $f(x, y) = x \vee y$.

Розв'язання. $f^*(x, y) = \overline{\overline{x} \vee \overline{y}} = xy$.

Відповідь. Кон'юнкція двоїста диз'юнкції.

Аналогічно доводиться, що диз'юнкція двоїста кон'юнкції.

Означення. Функція, двоїста сама собі, називається ***самодвоїстою.***

Приклад самодвоїстої функції:

$$f(x, y, z) = xy \vee xz \vee yz.$$

Дійсно, знайдемо двоїсту функцію:

$$f^*(x, y, z) = \overline{\overline{x}\,\overline{y} \vee \overline{x}\,\overline{z} \vee \overline{y}\,\overline{z}} = \overline{\overline{x}\,\overline{y}} \wedge \overline{\overline{x}\,\overline{z}} \wedge \overline{\overline{y}\,\overline{z}} =$$

$$= (x \vee y)(x \vee z)(y \vee z) = (x \vee xy \vee xz \vee yz)(y \vee z) =$$

$$= (x \vee yz)(y \vee z) = xy \vee yz \vee xz \vee yz = xy \vee yz \vee xz.$$

Як бачимо, $f(x, y, z) = f^*(x, y, z)$. Значить, функція $f(x, y, z)$ – самодвоїсна. Зауважимо, що самодвоїсних функцій двох змінних не існує.

Означення. Булева функція $f(x_1, x_2, \ldots, x_n)$ називається ***лінійною функцією,*** якщо її можна подати у вигляді

$$f(x_1, x_2, \ldots, x_n) = a_0 + a_1 x_1 + a_2 x_2 + \ldots + a_n x_n,$$

де коефіцієнти $a_i \in B$, $i = 0, 1, \ldots, n$.

Приклади лінійних функцій: 0, 1, x, $\overline{x} = x + 1$, $x + y$. Кон'юнкція та диз'юнкція – нелінійні функції. Припустимо супротивне, наприклад, що диз'юнкція – лінійна функція. Тоді її можна подати у вигляді

$$x \vee y = a + bx + cy.$$

Знайдемо коефіцієнти a, b і c, придаючи змінним x та y всі можливі значення:

При $x = 0, y = 0$, отримаємо $a = 0$.

При $x = 0, y = 1$, отримаємо $c = 1$.

При $x = 1, y = 0$, отримаємо $b = 1$.

При $x = 1, y = 1$, отримаємо $1 = 1 + 1$.

Прийшли до протиріччя. Отже, диз'юнкція – нелінійна функція. Аналогічно доводиться нелінійність кон'юнкції.

Введемо відношення часткового порядку $\leq$ для векторів. Будемо вважати, що

$$a_1', a_2', \dots, a_n' \;\; \leq \;\; a_1'', a_2'', \dots, a_n'' \;,$$

якщо $a_i' \leq a_i''$, $i = 1, 2, \dots, n$.

На рис. 5.1 та рис. 5.2 зображені діаграми Хассе відношення $\leq$ відповідно для двовимірних та тривимірних векторів. На цих діаграмах вектори, які можна порівнювати, з'єднані відрізками знизу уверх, а вектори, які не можна порівнювати, наприклад, (0,1) та (1,0) такими відрізками не з'єднані.

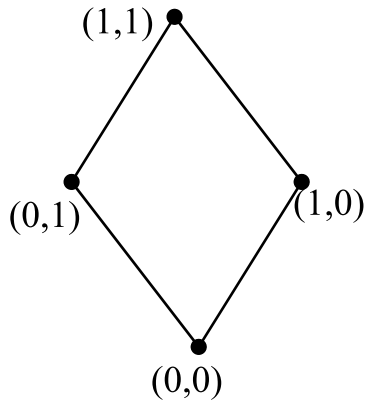


Рис. 5.1

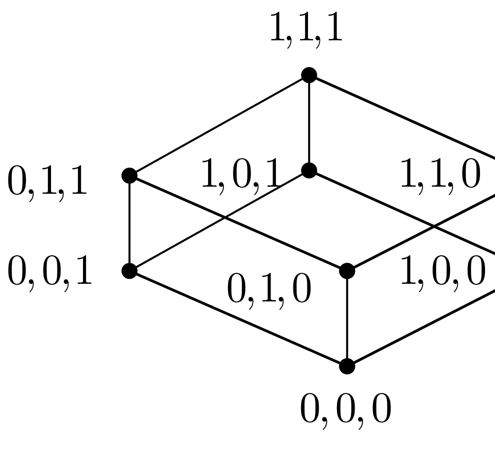


Рис. 5.2

Означення. Булева функція $f \;\; x_1, x_2, \dots, x_n$ називається ***монотонною,*** якщо для буд-яких векторів $a_1', a_2', \dots, a_n'$ та $a_1'', a_2'', \dots, a_n''$ таких, що

$$a_1', a_2', \dots, a_n' \;\; \leq \;\; a_1'', a_2'', \dots, a_n'' \;,$$

справедлива нерівність

$$f\left(a_1',a_2',\ldots,a_n'\right)\le f\left(a_1'',a_2'',\ldots,a_n''\right).$$

Для дослідження на монотонність функції двох чи трьох змінних, можна записати значення функції у вузлах її діаграмами Хассе. Якщо функція монотонна, то рухаючись від мінімального вектора до максимального знизу уверх по ребрах, значення функції повинно зростати (тобто, не спадати).

Приклади монотонних функцій: 0, 1, x, xy, $x \vee y$.

Приклади немонотонних функцій: $\overline{x}$, $x+y$, $x \to y$.

На рис. 5.3 у вузлах діаграми Хассе записані значення функції $f = x \vee y$, а на рис. 5.4 – значення функції $f = x + y$.

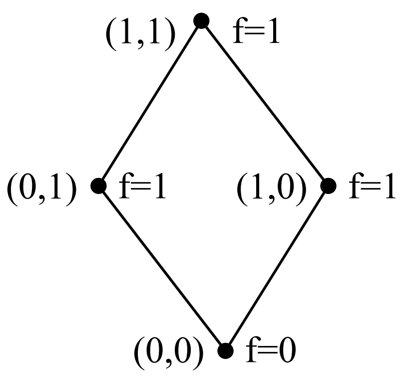


Рис. 5.3

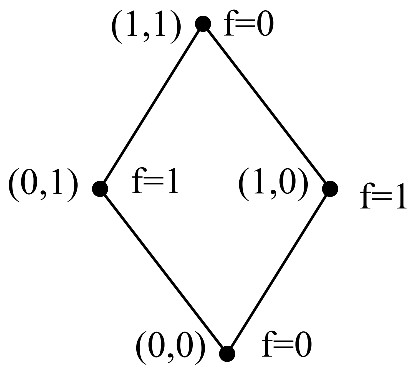


Рис. 5.4

Із діаграм видно, що функція $x \vee y$ монотонна, а функція $x + y$ – ні.

Не монотонність функцій пов'язана із наявністю заперечень. Можна довести, що функція монотонна тоді і тільки тоді, якщо існує ДНФ, в якій немає заперечень.

Теорема Поста (критерій повноти). Система функцій $f_1, f_2, \ldots, f_n$ є повною тоді і тільки тоді, якщо вона містить функцію, що не зберігає константу 0, містить функцію, що не зберігає константу 1, містить несамодвоїсту функцію, містить нелінійну функцію і містить немонотонну функцію.

5.7. Алгебра Жегалкіна

Алгебра Жегалкіна складається із операцій $\{+, \wedge, 1\}$. Ця алгебра більше за інші схожа на арифметичну алгебру. Запишемо її закони.

1. Комутативний закон

$$x + y = y + x.$$

2. Асоціативний закон

$$(x + y) + z = x + (y + z) = x + y + z.$$

3. Закони нуля

$$x + 0 = x, \qquad x + x = 0$$

4. Дистрибутивний закон

$$x(y + z) = xy + xz.$$

Тут кон'юнкція дистрибутивна відносно суми, а сума відносно кон'юнкції – ні.

Доведемо повноту алгебри Жегалкіна. Ми вже знаємо, що $\neg, \wedge$ – повна система функцій. Кон'юнкція в алгебрі Жегалкіна присутня. Нам лишилось заперечення виразити через $+, \wedge, 1$. Це здійснюється формулою

$$\overline{x} = x + 1. \tag{5.6}$$

Виразимо ще кон'юнкцію через функції алгебри Жегалкіна:

$$x \vee y = \overline{\overline{x \vee y}} = \overline{\overline{x}\,\overline{y}} = \overline{(x+1)(y+1)} = (x+1)(y+1)+1 =$$
$$= xy + y + x + 1 + 1 = x + y + xy.$$

Отже, справедлива формула, яку бажано запам'ятати

$$x \vee y = x + y + xy. \tag{5.7}$$

Вираз у правій частині рівняння (5.6) в математиці звичайно називають многочленом.

Означення. Многочленом Жегалкіна називається сума (за модулем 2) кон'юнкцій.

Із формул (5.6) та (5.7) стає зрозумілою наступна теорема.

Теорема. Довільну булеву функцію можна представити у вигляді многочлена Жегалкіна і це представлення єдине.

Приклад. Представити у вигляді многочлена Жегалкіна функцію

$$f = \overline{x}\,\overline{y} \vee xy \vee z.$$

Розв'язання.

$$f = \overline{x}\,\overline{y} \vee xy \vee z = (\overline{x}\,\overline{y} + xy + \overline{x}\,\overline{y}xy) \vee z = (\overline{x}\,\overline{y} + xy) \vee z =$$
$$= ((x+1)(y+1) + xy) \vee z = (xy + y + x + 1 + xy) \vee z =$$
$$= (x + y + 1) \vee z = x + y + 1 + z + (x + y + 1)z =$$
$$= x + y + 1 + z + xz + yz + z = 1 + x + y + xz + yz.$$

Ми отримали многочлен Жегалкіна. Функція нелінійна.

5.8. Логічні схеми.

У даному пункті розглянемо одне із застосувань булевих функцій – складання логічних схем, які реалізуються в комп'ютерах та інших електронних пристроях з скінченим числом входів та виходів. Причому, на кожному вході та виході можуть з'являтися лише два значення: нуль чи одиниця. Такі пристрої описуються булевими функціями і збираються із логічних елементів. Стандартні позначення основних логічних елементів зображено на рис. 5.5.

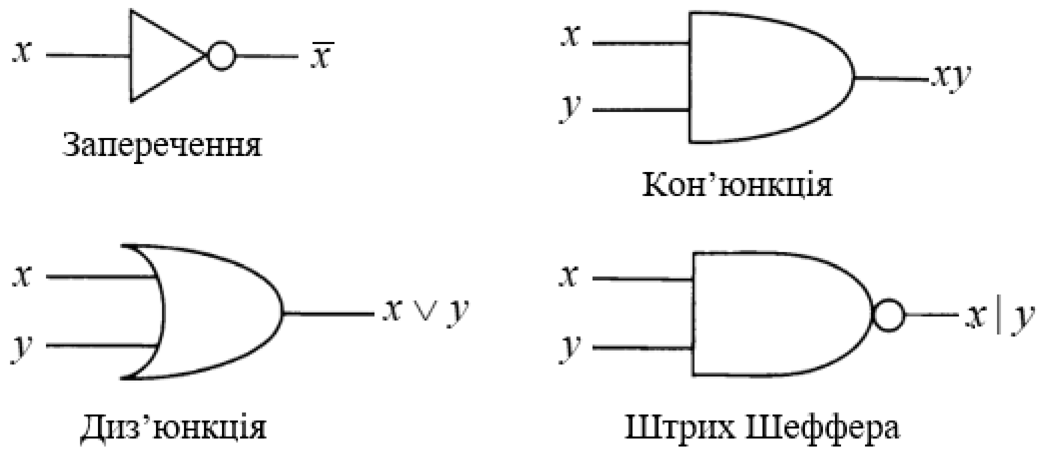


Рис. 5.5

За допомогою цих логічних елементів можна скласти логічну схему, яка може реалізувати будь яку булеву функцію.

Приклад. Скласти булеву функцію, що реалізується логічною схемою, яка представлена на рис. 5.6.

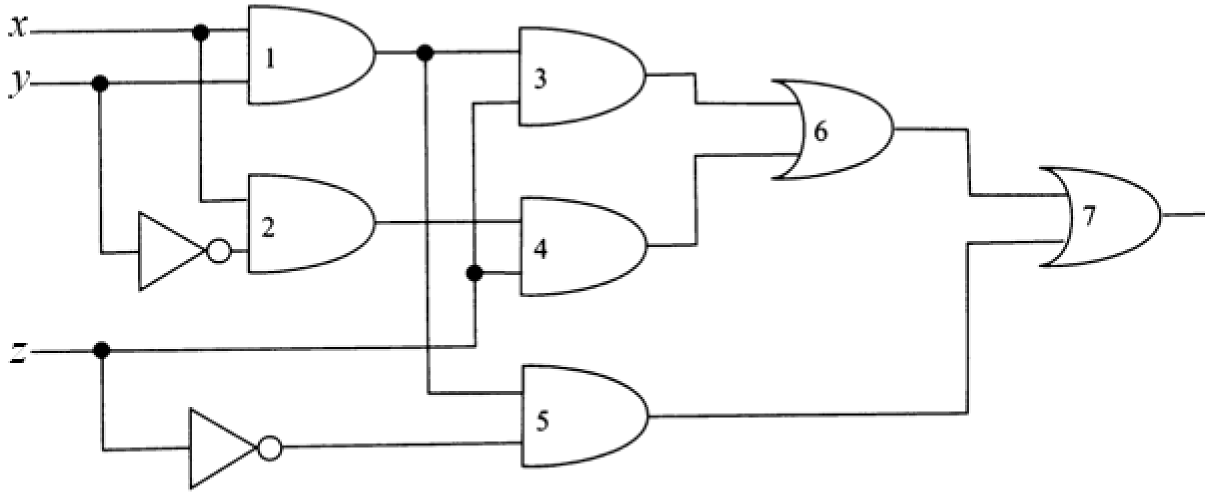


Рис. 5.6

Розв'язання. В таблицю 5.2 запишемо що входить в кожний логічний елемент і що з нього виходить.

Таблиця 5.2

Логічний елемент	Вхід	Вихід
1	x, y	xy
2	x, $\overline{y}$	$x\overline{y}$
3	xy, z	xyz
4	$x\overline{y}$, z	$x\overline{y}z$
5	xy, $\overline{z}$	$xy\overline{z}$
6	xyz, $x\overline{y}z$	$xyz \vee x\overline{y}z$
7	$xyz \vee x\overline{y}z$, $xy\overline{z}$	$xyz \vee x\overline{y}z \vee xy\overline{z}$

Отже, логічну схему реалізує наступна булева функція

$$f(x, y, z) = xyz \vee x\overline{y}z \vee xy\overline{z}\,.$$

Цю булеву функцію можна спростити, тобто мінімізувати, користуючись наступною картою Карно:

	xy	$\overline{x}y$	$\overline{x}\,\overline{y}$	$x\overline{y}$
z	1			1
$\overline{z}$	1			

Отримаємо

$$f(x, y, z) = xy \vee xz\,,$$

або, користуючись дистрибутивним законом,

$$f(x, y, z) = x(y \vee z).$$

Цю булеву функцію реалізує логічна схема, що зображена на рис. 5.7.

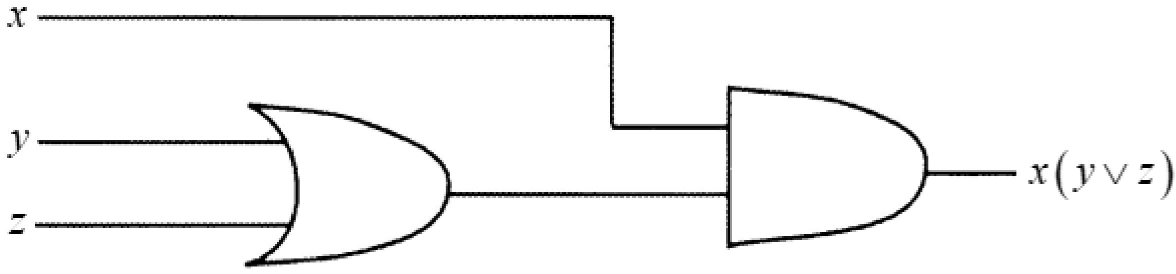


Рис. 5.7

Ми знаємо, що повна система функцій також може складатися і з однієї функції, наприклад, штриха Шеффера. Запишемо отриману функцію, використовуючи тільки штрих Шеффера і побудуємо відповідну логічну схему. Згадаємо формули, які кон'юнкцію і диз'юнкцію виражають через штрих Шеффера:

$$xy = \; x \mid y \;\mid\; x \mid y \;, \quad x \vee y = \; x \mid x \;\mid\; y \mid y \;.$$

Будемо мати

$$f(x,y,z) = x(y \vee z) = (x \mid (y \vee z)) \mid (x \mid (y \vee z)),$$

остаточно,

$$f(x,y,z) = (x \mid ((y \mid y) \mid (z \mid z))) \mid (x \mid ((y \mid y) \mid (z \mid z))).$$

Логічна схема зображена на рис. 5.8.

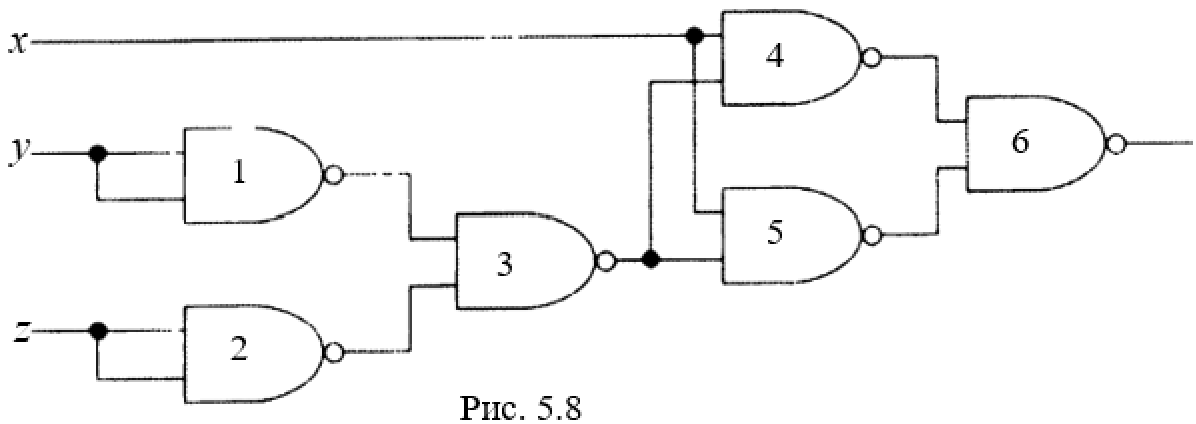


Рис. 5.8

Робота цієї логічної схеми наведена в таблиці 5.3.

Таблиця 5.3

Логічний елемент	Вхід	Вихід
1	y	$y \mid y$
2	z	$z \mid z$
3	$y \mid y,\ z \mid z$	$(y \mid y) \mid (z \mid z)$
4	$x,\ (y \mid y) \mid (z \mid z)$	$x \mid ((y \mid y) \mid (z \mid z))$
5	$x,\ (y \mid y) \mid (z \mid z)$	$x \mid ((y \mid y) \mid (z \mid z))$
6	$x \mid ((y \mid y) \mid (z \mid z)),$ $x \mid ((y \mid y) \mid (z \mid z))$	$(x \mid ((y \mid y) \mid (z \mid z))) \mid$ $(x \mid ((y \mid y) \mid (z \mid z)))$

Набір вправ до розділу 5.

5.1. Доведіть закони де Моргана.

5.2. Опираючись на закони булевої алгебри, доведіть справедливість наступних рівностей:

a) $\overline{x \wedge \overline{y}} \vee z = \overline{x} \vee y \vee z$;

b) $\overline{\overline{x} \wedge y} \wedge \left(x \vee y\right) = x$;

c) $\overline{x \wedge \overline{y} \wedge \left(z \vee x \wedge \overline{y}\right)} = \overline{x} \vee y$.

5.3. a) Функція $f\left(x, y, z\right)$ дорівнює 1 на наступних наборах: $\left(0,0,0\right)$, $\left(0,1,0\right)$, $\left(1,0,1\right)$ і $\left(1,1,1\right)$, на всіх інших наборах значення функції дорівнюють 0. Знайдіть досконалу диз'юнктивну нормальну форму (ДДНФ) і досконалу кон'юнктиву нормальну форму (ДКНФ).

b) Функція $f\left(x, y, z, t\right)$ дорівнює 1 на наступних наборах: $\left(0,0,0,1\right)$, $\left(0,0,1,0\right)$, $\left(1,0,1,1\right)$ і $\left(1,1,0,1\right)$, на всіх інших наборах значення функції дорівнюють 0. Знайдіть ДДНФ і ДКНФ.

5.4. Заповніть таблиці істинностей для наступних функцій. Знайдіть ДДНФ і ДКНФ за таблицями і без таблиць. Побудуйте многочлени Жегалкіна. Дослідіть властивості функцій: зберігає константу 0, зберігає константу 1, самодвоїстість, лінійність та монотонність:

a) $f\left(x, y, z\right) = x\left(\overline{y} \vee z\right) \vee \overline{x}\left(y \vee \overline{z}\right)$;

b) $f\left(x, y, z\right) = x\left(\overline{y} \vee z\right) \vee yz \to xy$.

5.5. Запишіть вираз $x\overline{y}z$, використовуючи: а) тільки штрих Шеффера, б) тільки стрілку Пірса.

5.6. Спростіть наступні функції, тобто знайдіть їх мінімальні ДНФ:

a) $f\left(x, y, z\right) = \overline{x}\,\overline{y}\,\overline{z} \vee \overline{x}\,y\overline{z} \vee x\overline{y}z \vee xyz$;

b) $f\left(x, y, z\right) = \overline{x}\,y\overline{z} \vee \overline{x}\,yz \vee xy\overline{z} \vee xyz$;

c)

$$f\left(x, y, z, t\right) = \overline{x}\,\overline{y}zt \vee \overline{x}\,y\overline{z}\,\overline{t} \vee \overline{x}\,y\overline{z}t \vee \overline{x}\,yzt \vee x\overline{y}\,\overline{z}t \vee x\overline{y}zt \vee xy\overline{z}\,\overline{t} \vee xy\overline{z}t$$

d) $f\left(x, y, z, t\right) = \overline{x}\,\overline{y}zt \vee x\overline{y}\,\overline{z}\,\overline{t} \vee \overline{x}\,y\overline{z}t \vee \overline{x}\,\overline{y}\,\overline{z}t \vee x\overline{y}\,\overline{z}t \vee \overline{x}\,yzt$;

e)

$$f\left(x, y, z, t\right) = \overline{x}\,\overline{y}zt \vee xyz\overline{t} \vee xyzt \vee \overline{x}\,yzt \vee x\overline{y}\,\overline{z}t \vee \overline{x}\,\overline{y}\,\overline{z}t \vee x\overline{y}\,\overline{z}\,\overline{t} \vee \overline{x}\,\overline{y}\,\overline{z}\,\overline{t}.$$

5.7. Користуючись законами булевої алгебри, доведіть справедливість рівності

$$\overline{x} \mid \left(\overline{y} \mid z\right) = x \vee \overline{y}z.$$

Побудуйте логічну схему, що відповідає лівій частині рівності і побудуйте логічну схему, що відповідає правій частині рівності.

5.8. Побудуйте булеву функцію, що реалізується логічною схемою, зображеною на рис. 5.9. Запишіть отриману функцію у вигляді ДДНФ,

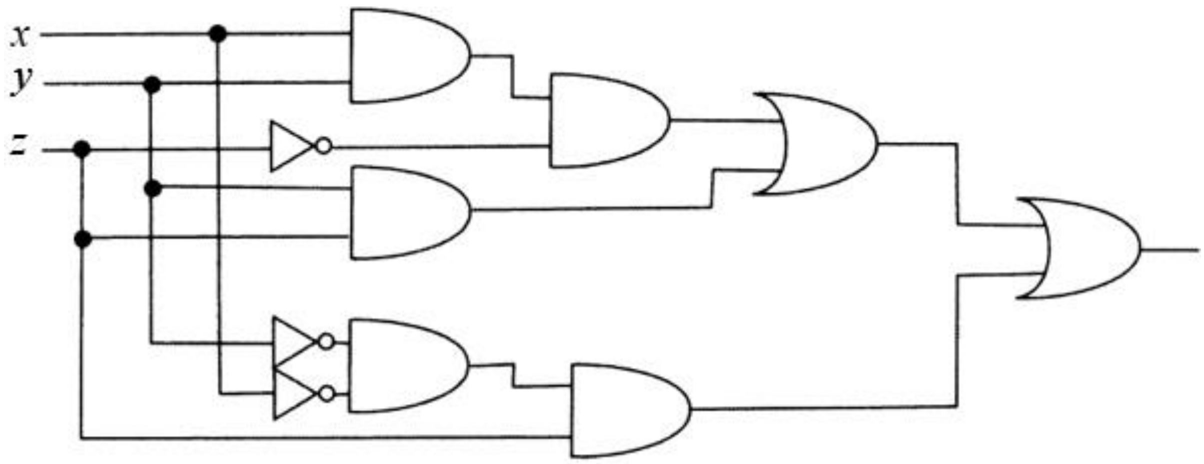


Рис. 5.9

знайдіть мінімальну ДНФ та зобразіть логічну схему мінімальної ДНФ.

5.9. Побудуйте логічну схему для стрілки Пірса $x \downarrow y$, використовуючи тільки логічний елемент штрих Шеффера $x \mid y$.

РОЗДІЛ 6. ГРАФИ

6.1. Графи. Основні поняття

Теорія графів – одна з небагатьох математичних дисциплін, дата народження якої встановлена абсолютно точно.

Перша робота з теорії графів належить Леонарду Ейлеру. Вона з'явилась у 1736 році і виникла з так званої "задачі про кенігсберзькі мости".

Сім мостів Кенігсберга – видатна історична задача з математики.

Місто Кенігсберг (нині Калінінград) розташоване на берегах річки Прегель і двох островах. Різні частини міста сполучені сімома мостами (рис. 6.1). Жителі міста протягом довгого часу намагалися розв'язати наступну задачу: чи можна здійснити прогулянку по місту, пройшовши по кожному мосту рівно один раз, і повернутися до вихідної точки.

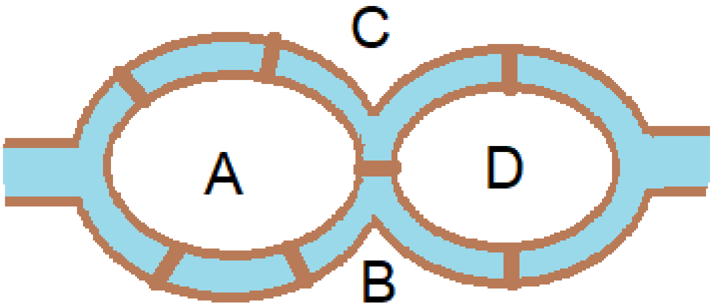


Рис. 6.1

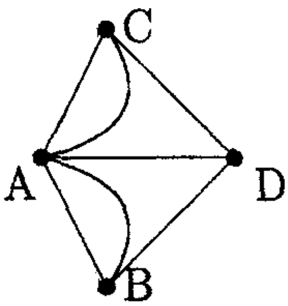


Рис. 6.2

Ця задача не піддавалась розв'язанню поки за діло не взявся Ейлер. Він склав математичну модель задачі, позначивши частини міста точками, а мости, що їх сполучають – відрізками (рис. 6.2). Отримана фігура називається графом. До задачі ми ще повернемось, але спочатку дамо необхідні означення.

Графом називається сукупність точок і відрізків, які з'єднують деякі точки. Точки називаються ***вершинами***, а відрізки – ***ребрами*** або ***дугами*** графа.

Множину вершин позначимо буквою V, а множину ребер – буквою E. Таким чином граф G задається двома множинами $G=(V,E)$.

Можна сказати, що множина E задає відношення на множині V: дві вершини з V знаходяться у відношенні, якщо вони з'єднані деяким ребром.

Якщо вершина v є кінцем ребра e, то кажуть, що v і e ***інцидентні*** (рис. 6.3).

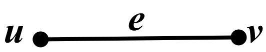


Рис. 6.3

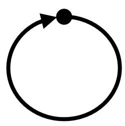


Рис. 6.4

Вершини, які з'єднані ребром, називаються ***суміжними.*** На рис. 6.3 вершини u і v суміжні.

Якщо кінці ребра належать однієї вершині, то таке ребро називається ***петлею*** (рис. 6.4).

Якщо дві вершини зв'язані декількома ребрами, то ці ребра називаються ***кратними.*** На рис.6.2 вершини A і B зв'язані двома ребрами. Ці ребра є кратними.

Степенем вершини називається число ребер, які інцидентні цій вершині.

Для графа на рис. 6.2 степені вершин B, C і D дорівнюють 3, а степінь вершини A дорівнює 5. Позначення:

$$\deg(A) = 5, \quad \deg(B) = \deg(C) = \deg(D) = 3.$$

Якщо степінь вершини дорівнює 1, то вершина називається ***кінцевою***. На рис. 6.3 вершини u і v кінцеві.

Якщо степінь вершини дорівнює 0, то вершина називається ***ізольованою***.

Граф будемо називати ***скінченним***, якщо в нього скінченна кількість вершин та ребер.

Скінчений граф називається ***простим***, якщо в нього немає кратних ребер і петель.

Граф, в якому будь-які дві вершини з'єднані ребром, називається ***повним***.

Теорема. Повний граф з n вершинами має $n(n-1)/2$ ребер.

Доведення. Оскільки кожні дві вершини зв'язані ребром, то ребер буде стільки, скількома способами можна вибрати 2 вершини з n, тобто

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

Теорему доведено.

6.2. Способи задання графа

1. Графічний спосіб. При цьому способі завдання графа вершини зображуються точками на площині чи в просторі. Ребра зображуються відрізками або дугами, які з'єднують деякі вершини. На рис. 6.5 зображений граф G з вершинами із множини V та ребрами із множини E:

$$V = \{a, b, c, d, e\}, \quad E = \{ab, ac, ad, bc, be, cd, de\}. \qquad (6.1)$$

2. Завдання графа матрицею суміжності. Матрицею суміжності графа називається матриця відношення, яке задано множиною E на множині V. Елементи цієї матриці обчислюються за формулою: $m_{ij} = 1$, якщо i-й та j-й елементи множини V з'єднані ребром і $m_{ij} = 0$ в супротивному випадку.

Матриця суміжності графа G на рис. 6.5 має вигляд:

$$M = \begin{pmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \end{pmatrix}.$$

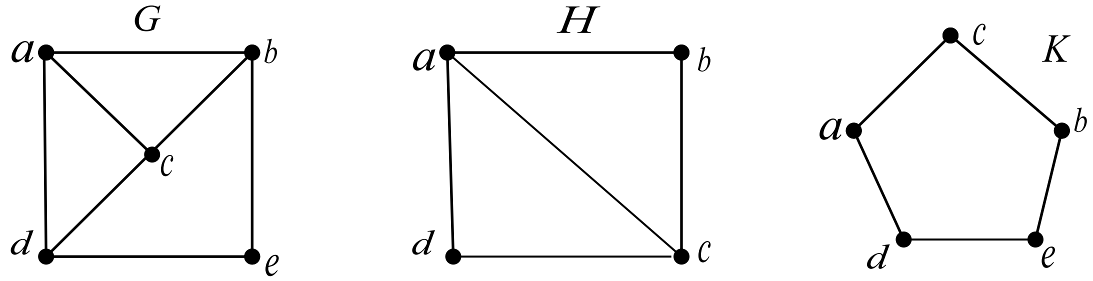


Рис. 6.5

3. Завдання графа матрицею інцидентності. Упорядкуємо елементи множин V та E так, як вони записані в (6.1). Рядки матриці інцидентності позначимо елементами множини V, а стовпці – елементами множини E. Елементи матриці інцидентності обчислюються так: $n_{ij} = 1$, якщо i-та вершина інцидентна j-му ребру і $n_{ij} = 0$ у супротивному випадку.

Матриця інцидентності графа G на рис.6.5 має вигляд:

$$N = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 \end{pmatrix}.$$

4. Задання графа у вигляді списку. При цьому способі для кожної вершини графа вказують суміжні вершини.

Для графа G на рис. 6.5 список має наступний вигляд:

$$a : b, c, d; \quad b : a, c, e; \quad c : a, b, d; \quad d : a, c, e; \quad e : b, d.$$

Список займає небагато місця в пам'яті комп'ютера, тому цей спосіб завдання графа найчастіше застосовується в програмуванні.

6.3. Зв'язність графа

Визначення. Підграфом графа $G = (V, E)$ називається граф $G' = (V', E')$, в якому $V' \subseteq V$ та $E' \subseteq E$.

Наприклад, графи H і K є підграфами графа G на рис. 6.5.

Шляхом довжини k в графі G називається послідовність вершин $v_0 v_1 ... \mathrm{v}_k$ така, що для будь-якого $i = 0,1,...,k-1$ пара $v_i v_{i+1}$ є ребром графа.

Наприклад, для графа G на рис. 6.5 шляхом буде $abcde$. Довжина цього шляху дорівнює кількості пройдених ребер, тобто 4.

Циклом називається шлях, в якому збігаються початкова й кінцева вершини ($v_0 = \mathrm{v}_k$).

Для графа G на рис. 6.5 існують два цикли довжиною 5:

$$acbeda \quad \text{та} \quad abedca.$$

В цих циклах кожне ребро і кожна вершина (окрім кінцевих вершин) пройдені тільки один раз. Ми можемо пройти ці цикли як в одному напрямі, так і в другому. Починати цикл можна з будь-якої вершини. У графа G є три різних цикли довжини 4:

$$abeda, \quad abcda, \quad bedcb$$

та два цикли довжини три:

$$abca \quad \text{та} \quad acda.$$

Граф, в якому немає циклів, називається ***ациклічним.***

Граф називається ***зв'язним***, якщо будь-які дві його вершини зв'язані шляхом.

Довільний загальний граф можна розбити на підграфи, кожен з яких виявиться зв'язним. Ці підграфи називаються компонентами зв'язності графа. Мінімальна кількість таких зв'язних компонент називається числом зв'язності графа і позначається через $c(G)$ або просто c.

Алгоритм зв'язності. Запишемо алгоритм знаходження числа зв'язності c графа $G = (V, E)$.

Крок 1. Введемо множину $V1 := V$; Числу зв’язності присвоїмо нуль $c := 0$.

Крок 2. Виберемо довільний елемент $y \in V1$ і вилучимо y з $V1$ та всі елементи, які зв'язані з y шляхом; $c := c + 1$.

Крок 3. Крок 2 продовжуємо поки множина $V1$ не стане порожньою.

Приклад. Прослідкуємо за роботою алгоритму на прикладі графа G, зображеного на рис. 6.6.

Розв'язання. Результати роботи алгоритму представлені в наступній таблиці:

$V1$	c	Вибір $y \in V1$
$\{1,2,3,4,5,6,7,8\}$	0	
$\{2,4,5,7\}$	1	$y = 1$
$\{7\}$	2	$y = 2$
$\varnothing$	3	$y = 7$

Число зв'язності $c = 3$. Це означає, що граф G розбивається на 3 компоненти зв'язності: $G1$, $G2$ та $G3$ (рис. 6.6).

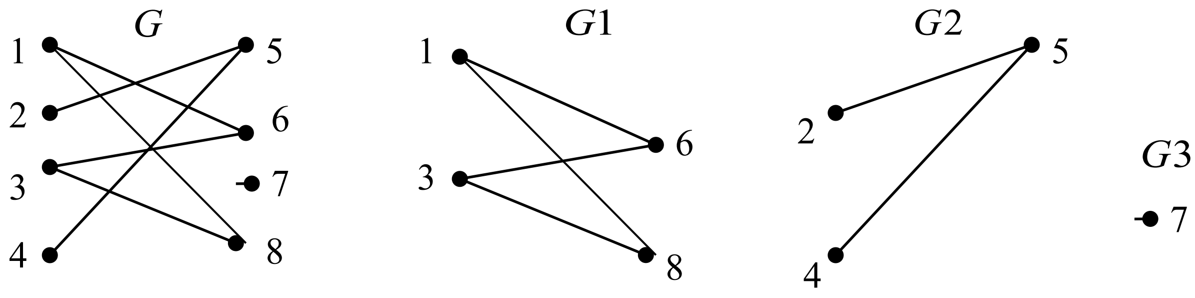


Рис. 6.6

6.4. Ейлерові графи

У даному пункті ми розв'яжемо Велику задачу про Кенігсберзькі мости. Але перед Великою грою треба добре розім'ятись, щоб у процесі гри не надірвати собі м'язи. У якості розминки розглянемо дві леми.

Лема про естафету. Сума степенів всіх вершин будь-якого графа дорівнює подвоєній кількості ребер.

Доведення. Кожне ребро графа вносить по одному степеню до інцидентних йому вершин (рис. 6.3). Тобто додає дві одиниці до суми степенів всіх вершин. Тому сума степенів всіх вершин дорівнює подвоєній кількості ребер.

Лему доведено.

Наслідок. В будь-якому графі кількість вершин непарного степеня є парною. Це твердження називається ***лемою про рукостискання*** (кількість непарних рукостискань є парною).

Доведення. З леми про естафету виходить, що сума степенів всіх вершин графа є парною. Оскільки сума степенів всіх вершин з парними степенями є парною, то і сума степенів всіх вершин з непарними степенями повинна бути парною. Останнє можливо тільки тоді, якщо кількість вершин з непарними степенями буде парною. Наслідок доведено.

Тепер перейдемо до основної теми.

Означення. Цикл в графі називається ***ейлеревим***, якщо кожне ребро графа буде пройденим тільки один раз.

Граф називається ***ейлеревим***, якщо в ньому є ейлерів цикл.

Теорема 6.1. Скінченний граф G має ейлерів цикл тоді і тільки тоді коли він зв'язний і кожна вершина графа має парний степінь.

Доведення складається з двох частин. 1) Нехай скінченний граф має ейлерів цикл. Тоді, очевидно, він зв'язний, бо інакше з однієї компоненти зв'язності неможливо потрапити в іншу. Оскільки існує ейлерів цикл, то потрапивши в будь-яку вершину по одному ребру, з іншого ребра з цієї вершини треба вийти (по кожному ребру можна пройти тільки один раз!). Тому степінь кожної вершини парний (рис. 6.7).

2) Нехай граф зв'язний і степінь кожної вершини парний. Доведемо, що існує ейлерів цикл. Візьмемо будь-яку вершину a графа і рушимо з неї в путь P, проходячи кожне ребро графа не більше одного разу і рухаймось стільки наскільки це можливо. Враховуючи скінченну кількість ребер графа і неможливість пройти по ребру більше одного разу, робимо висновок, що шлях P буде скінченним. Кінцевою точкою шляху P буде вершина a (рис. 6.8). Будь-яка інша вершина c не може бути кінцевою, бо увійшовши в неї можна і потрібно вийти оскільки степінь вершини є парний і ми зобов'язані рухатись, якщо є така можливість.

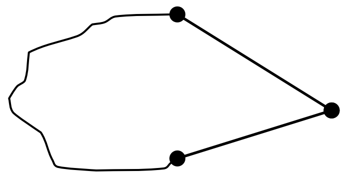


Рис. 6.7

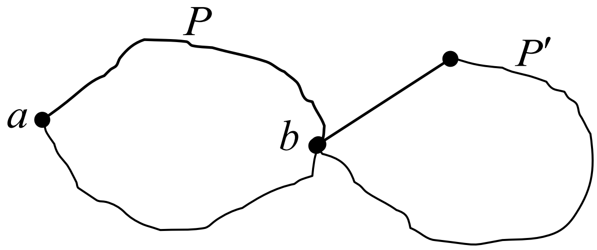


Рис. 6.8

Якщо цикл P вміщує всі ребра графа G, то задачу розв'язано. В супротивному випадку, враховуючи зв'язність графа, на шляху P знайдеться вершина b, яка інцидентна ще не пройденому ребру. Із вершини b утворимо новий цикл P' по ребрах, які не увійшли в цикл P. Розглянемо цикл

$$P1 = P(a,b) \cup P'\left(b,b\right) \cup P(b,a).$$

Цикл $P1$ складається з частини шляху P від вершини a до вершини b, з циклу P' від вершини b до вершини b і з непройденої частини P від b до a (рис. 6.8).

Якщо цикл $P1$ вміщує всі ребра графа G, то задачу розв'язано. В супротивному разі продовжимо описану процедуру доки не одержимо ейлерів цикл. Теорему доведено.

Доведення другої частини теореми є алгоритмом побудови ейлерева циклу.

Приклад. Знайти ейлерів цикл в графі, зображеному на рис. 6.9.

Розв'язання. У професійній діяльності часто зустрічаються графи хоча і скінченні, але з великою кількістю вершин та ребер. Тому розв'яжемо задачу не дивлячись на рис. 6.9, а запишемо граф у вигляді списку.

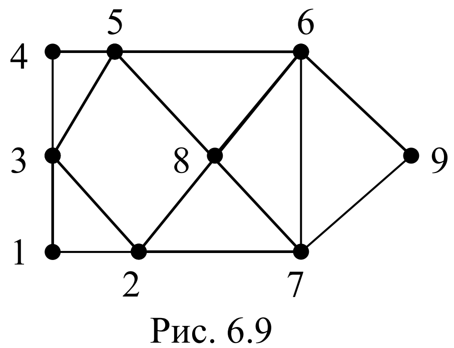


Рис. 6.9

Запис графа у вигляді списку:

1: 2,3;
2: 1,3,7,8;
3: 1,2,4,5;
4: 3,5;
5: 3,4,6,8;
6: 5,7,8,9;
7: 2,6,8,9;
8: 2,5,6,7;
9: 6,7.

Передусім зауважимо, що степінь кожної вершини є парним. Значить, існує ейлерів цикл. Утворимо 2 стека: шлях і цикл.

Стек шлях: $1,2,3,(1),4,5,(3),6,7,2,8,(5),6,9,7,(8)$

Стек цикл: $1,3,5,8,7,9,6,8,2,7,6,5,4,3,2,1$.

Стек шлях заповнюється так. Розпочнемо путь з вершини 1. Зі списку 1 видно, що з 1 можна потрапити в 2. Щоб по ребру 1,2 більше не ходити, перекреслимо 2 зі списку 1 і 1 зі списку 2. Знаходячись в 2 можна потрапити в 3, а з 3 – в 1. Зі списків не слід забувати перекреслювати вершини, відповідні пройденим ребрам. Утворився цикл $1,2,3,1$, який записаний в стеку шлях. Останню 1 зі стеку шлях переводимо у стек цикл. В стеку шлях записуємо новий цикл, який починається з 3: 3,4,5,3. Останнє 3 переводимо у стек цикл. Новий цикл у стеку шлях розпочинаємо з вершини 5. Описаний алгоритм продовжуємо до тих пір поки не будуть пройдені всі ребра (тобто не будуть перекреслені всі числа у списках). Стек цикл після цього прийме вигляд: $1,3,5,8$. Дописавши у цей стек вершини зі стеку шлях у зворотному порядку, отримаємо шуканий ейлерів цикл.

Отже, відповідь задачі записана у стеку цикл.

Означення. Ейлеровим шляхом в графі називається шлях, в якому кожне ребро графа пройдено рівно один раз.

Теорема 6.2. Скінченний граф G має ейлерів шлях тоді і тільки тоді коли він зв'язний і має не більше двох вершин непарного степеня.

Доведення. Нехай граф G має 2 вершини непарного степеня. Позначимо їх через u і v. Якщо u і v не зв'язані ребром (рис. 6.10), то зв'яжемо їх. Утворений новий граф буде мати всі вершини парного степеня, а значить, ейлерів цикл $u\ \dots\ vu$. Вилучивши введене ребро, отримаємо ейлерів шлях $u\ \dots\ v$.

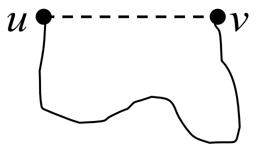


Рис. 6.10

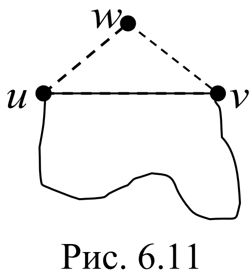


Рис. 6.11

Якщо вершини u і v зв'язані ребром, то введемо додаткову вершину w і ребра uw та vw (рис. 6.11). В новому графі всі вершини мають парний степінь. Значить, існує ейлерів цикл $wu\ldots vw$. Вилучивши введену вершину і ребра, отримаємо ейлерів шлях $u\ \dots\ v$. Теорему доведено.

Перейдемо тепер до Великої задачі про Кенігсберзькі мости (рис. 6.12).

Тут 4 вершини і всі вони мають непарний степінь. Згідно з теорією (теореми 6.1 та 6.2) задача не має розв'язків.

Розв'язок Кайзера. Вільгельм Кайзер – германський імператор 19 ст. Він славився своєю простотою мислення і солдатської прямотою. Одного разу, перебуваючи на світської вечірці, вчені мужі вирішили над ним пошуткувати. Вони попросили його спробувати вирішити цю знамениту задачку, яка була просто не вирішуваною. На загальний подив, Кайзер попросив аркуш паперу і перо, і при цьому уточнив, що вирішить це завдання всього за півтори хвилини. Приголомшені вчені не могли повірити своїм вухам, але чорнило і папір швидко знайшли для нього. Кайзер поклав аркуш на стіл, взяв перо, і написав: «Наказую побудувати восьмий міст на острові Ломза». І все, – задача вирішена ... . Так в місті Кенігсберг і з'явився новий восьмий міст через річку, який так і назвали - міст Кайзера. А задачку з вісьмома мостами (рис. 6.13) тепер легко розв'язати.

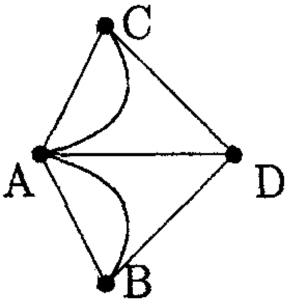


Рис. 6.12

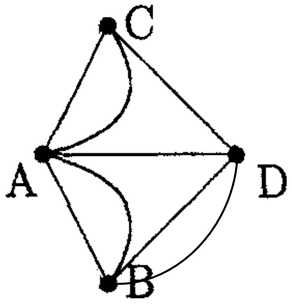


Рис. 6.13

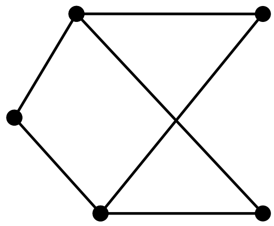


Рис. 6.14

Прикладів застосування ейлеревих графів дуже багато. Не вдаючись у велику науку, наведемо приклад з комунального господарства. Поливальна машина повинна проїхатись по всім вулицям району рівно один раз і повернутись у місце, звідки виїхала.

Приклад. Знайти ейлерів шлях для графа на рис. 6.14.

6.5. Гамільтонові графи

Означення. Цикл в графі називається ***гамільтоновим***, якщо проходить через кожну вершину графа рівно один раз.

Граф називається ***гамільтоновим***, якщо в ньому є гамільтонів цикл.

На жаль, теорія гамільтонових графів не розроблена так детально як теорія ейлерових графів. Немає зручних умов, за якими можна було б виявити, чи є даний граф гамільтоновим. Наукові дослідження в цьому напрямі ведуться і можна взяти в них участь. А поки що задачу дослідження кожного графа на гамільтоновість будемо розв'язувати окремо.

Приклад. Розглянемо повний граф з n вершинами (тобто граф, в якому будь-які дві вершини зв'язані ребром). Очевидно, що повний граф має гамільтонів цикл, і не один. Підрахуємо кількість гамітонових циклів.

Першу вершину можна вибрати n способами. З неї можна потрапити в будь-яку з $n-1$ вершин. Після цього наступну вершину можна вибрати $n-2$ способами, і так далі. Останню вершину (тобто вершину з якої вийшли) можна вибрати 1 способом. За принципом множення комбінаторики гамільтонових циклів існує $n(n-1)(n-2)...1=n!$.

Якщо немає значення з якої вершини починається цикл, то кількість гамільтонових циклів буде дорівнювати $(n-1)!$ Врахуємо ще, що кожний цикл можна пройти як в одному напрямі, так і в зворотному. В такому разі кількість гамільтонових циклів буде дорівнювати $\frac{1}{2}(n-1)!$. Так вважається в теорії графів.

Наведемо ще одну достатню умову гамільтоновості графа.

Теорема (Оре, 1960). Якщо в графі G з n вершинами ($n \geq 3$) сума степенів будь-яких двох несуміжних вершин є не меншою, ніж n, то граф G гамільтонів.

Приклад. Довести, що граф на рис. 6.15 не є гамільтоновим.

Доведення. Припустимо супротивне, що в цьому графі є гамільтонів цикл. Тоді кожна вершина у гамільтоновому циклі має 2 інцидентних ребра, які входять в цей цикл. Значить, в цикл обов'язково увійдуть ребра 01, 12, 23, 34, 57, 58, 79. Ребро 26 увійти в цей цикл не може. Це означає, що у цикл увійдуть ребра 69 і 68. Звідси витікає, що у цикл не увійдуть ребра 09 і 48. Значить, увійде ребро 04. Ребра, які ми вибрали утворюють два незалежних цикли: 012340 і 697586 (рис. 6.16), які зв'язати неможливо. Ми довели, що граф на рис. 6.15 не є гамільтоновим.

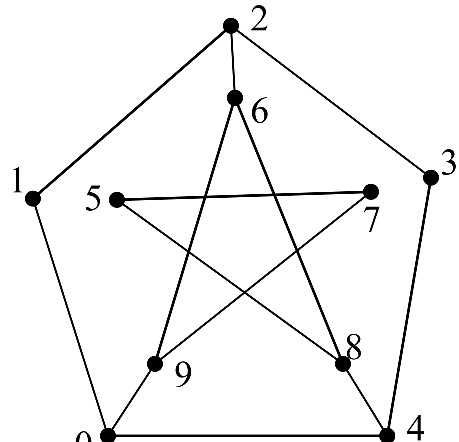


Рис. 6.15

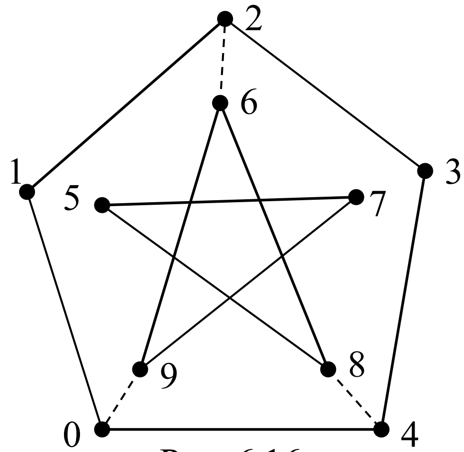


Рис. 6.16

Гамільтонові графи мають багато застосувань. Вони служать моделлю при складанні розкладів руху поїздів, комунікаційних мереж, і так далі. Розглянемо ще один важливий приклад застосування гамільтонових графів.

Задача комівояжера. Комівояжер (торговець) повинен здійснити поїздку по містах і повернутися назад, побувавши в кожному місті рівно один раз, звівши при цьому витрати на поїздку до мінімуму.

Кажуть, що товар у комівояжера такий, що приїжджати в одне місто більше одного разу йому не безпечно.

Математичною моделлю задачі є гамільтонів граф. Вершини графа зображують міста, а ребра – дороги, що їх з'єднують. Кожне ребро підписують числом, яке називається ***вагою ребра***. Вага може позначати відстань між містами, вартість проїзду, час у дорозі, тощо.

Граф, ребра якого мають вагу, називається ***зваженим графом***.

Задача полягає в знаходженні гамільтонова циклу з найменшою загальною вагою.

На жаль, ефективний метод розв'язання даної задачі поки не відомий. Можна знайти всі гамільтонові цикли, підрахувати їх загальні ваги і вибрати цикл з найменшою загальною вагою. Але для великих графів цей спосіб не прийнятний. Він потребує багато місця в пам'яті комп'ютера і багато машинного часу. Але існують зручні способи пошуку ***субоптимального розв'язку***. Субоптимальний розв'язок не обов'язково дасть цикл мінімальної загальної ваги, але знайдений цикл буде меншої ваги, ніж більшість довільних гамільтонових циклів. Розглянемо один з таких алгоритмів.

Алгоритм найближчого сусіда. Цей алгоритм видає субоптимальний розв'язок задачі комівояжера для графа з множиною вершин V. Отриманий цикл буде кінцевим значенням змінної *шлях*, а його довжина – кінцеве значення змінної w.

Крок 1. Виберемо довільну вершину $v \in V$. Змінній $шлях := v$. Вага цього шляху поки що дорівнює нулю: $w := 0$. Введемо ще одну змінну $v1 := v$. Помітимо $v1$.

Крок 2. Вибираємо непомічену вершину u, найближчу до $v1$. До змінної $шлях$ приєднаємо u: $шлях := шлях\, u$. Вага нового шляху збільшиться на вагу ребра $uv1$: $w := w + вага\ ребра\ uv1$. Змінній $v1$ присвоїмо u, $v1 := u$ та помічаємо $v1$.

Крок 3. Якщо є непомічені вершини, то повертаємось до кроку 2. В супротивному випадку $шлях := шлях\ v$; $w := w + вага\ ребра\ vv1$.

Приклад. Прослідкуємо за роботою алгоритму на прикладі графа, зображеного на рис. 6.17. Результати роботи запишемо у наступну таблицю.

шлях	w	$v1$	u
A	0	A	
AD	3	D	D
ADB	10	B	B
ADBC	15	C	C
ADBCA	23		

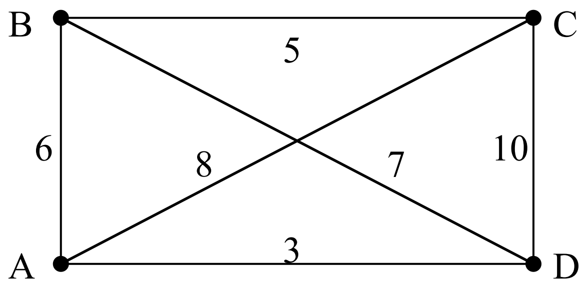


Рис. 6.17

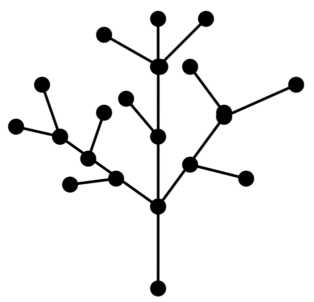
Ми знайшли гамільтонів цикл ADBCA загальною вагою 23. Окрім знайденого циклу існують ще два: ABCDA із загальною вагою 24 та ABDCA із загальною вагою 31. Отже, знайдений нами субоптимальний розв'язок задачі збігається з оптимальним.

Завдання. Прослідкуйте за роботою алгоритму, починаючи з вершини D.

Приклад застосування. Уявимо собі прилад "4 в 1", який вміє вирішувати 4 задачі A, B, C, D. Для вирішування конкретної задачі його треба привести у відповідний стан. Ці стани приладу можна позначити вершинами попереднього графу. Часи переведення приладу з одного стану в інший задані вагами ребер цього графу. Роботу треба організувати так, щоб всі 4 задачі вирішити у найкоротший час. Математичною моделлю цієї задачі є задача комівояжера, яку ми щойно розв'язали.

6.6. Дерева

Означення. Деревом називається зв'язний граф, в якому немає циклів (рис. 6.18).



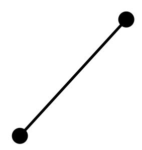


Рис.6.18 Рис.6.19 Рис.6.20

Із означення випливають наступні властивості дерева. Їх можна сформулювати як необхідні і достатні умови, при яких заданий граф є деревом.

1. Будь-яка пара вершин дерева з'єднана єдиним шляхом.

Дійсно, будь-яка пара вершин дерева з'єднана шляхом бо граф зв'язний. Цей шлях єдиний, бо якби був ще й другий шлях, то був би цикл.

2. Дерево – це зв'язний граф, але якщо вилучити хоча б одне ребро, то зв'язність буде порушено.

Дійсно, вилучене ребро єдиним способом з'єднувало певні дві вершини дерева.

3. У дереві немає циклів, але якщо будь-які дві вершини з'єднати ребром, то утвориться цикл.

4. У дереві з n вершинами кількість ребер дорівнює $n-1$.

Доведення проведемо за методом математичної індукції. Очевидно, що ця властивість справедлива при $n=1$ (рис. 6.19), при $n=2$ (рис. 6.20). Припустимо, що вона справедлива при $n \le k$ і доведемо її справедливість при $n=k+1$. У дереві з $k+1$-ю вершиною вилучимо одне ребро. Отримаємо 2 підграфи, які є деревами. Нехай перший підграф має n_1 вершин, а другий – n_2 вершин. Очевидно, що n_1 і n_2 не перевищують k і $n_1+n_2=k+1$. Тому перший підграф має n_1-1 ребер, у другому підграфу кількість ребер дорівнює n_2-1. Обидва підграфи мають

$$n_1-1+n_2-1=n_1+n_2-2=k+1-2=k-1$$

ребер. Згадаємо, що із дерева з $k+1$-ю вершиною ми вилучили ребро. Додавши це ребро, отримаємо k ребер. Властивість доведено.

Зауважимо, що ця, остання властивість стане більш зрозумілою та наочною пізніше, коли будемо вивчати дерева з коренем.

6.7. Остовні дерева

У будь-якому зв'язному графі знайдеться підграф, який є деревом.

Означення. Підграф, який є деревом і містить всі вершини даного графа, називається його ***остовним деревом.***

Алгоритм побудови остовного дерева. Виберемо довільне ребро графа. Потім послідовно додаємо інші ребра графу, не утворюючи циклів. Алгоритм закінчиться тоді, коли не можна буде приєднати ребро, не утворивши циклу. Якщо граф має n вершин, то остовне дерево буде мати $n-1$ ребро.

Скільки остовних дерев у графі? Для відповіді на це запитання наведемо без доведення 2 теореми.

Теорема Келі. Кількість остовних дерев у повному графі з n вершинами дорівнює n^{n-2}.

Теорема Кірхгофа. Нехай дано зв'язний граф. Кратні ребра допускаються. Запишемо матрицю суміжності графа, замінивши кожний її елемент на протилежний. На головній діагоналі запишемо степені відповідних вершин. Тоді всі алгебраїчні доповнення рівні між собою і дорівнюють кількості остовних дерев графа.

Приклад. Скільки остовних дерев у графі, зображеному на рис. 6.21?

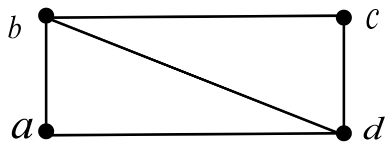


Рис. 6.21

Розв'язання. Складемо матрицю суміжності графа

$$\begin{array}{c} \\ a \\ b \\ c \\ d \end{array}\begin{array}{c} \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix} \end{array}.$$

Замінимо кожний елемент матриці на протилежний. На головній діагоналі запишемо степені відповідних вершин. Отримаємо

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

Тепер можна обчислити алгебраїчне доповнення будь-якого елемента отриманої матриці. Оберемо, наприклад, елемент, який знаходиться у першому рядку і в першому стовпці. Його алгебраїчне доповнення дорівнює

$$\begin{vmatrix} 3 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 3 \end{vmatrix} = 3\cdot 5 + 1(-4) - 1\cdot 3 = 15 - 4 - 3 = 8.$$

Отже, граф, зображений на рис. 6.21 має 8 остовних дерев. Зобразимо ці дерева на рис. 6.22.

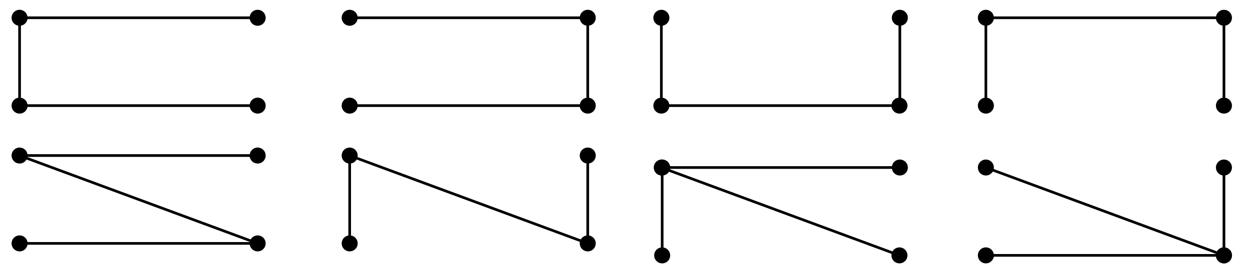


Рис. 6.22

Задача. Побудувати залізничну мережу, що зв'язує деяку кількість міст. Відома вартість будівництва відрізків шляхів між будь-якою парою міст. Потрібно знайти мережу мінімальної вартості.

Розв'язання. Математичною моделлю задачі є зважений орграф, в якому потрібно знайти остовне дерево з найменшою загальною вагою.

Означення. Мінімальним остовним деревом (коротко – ***МОД***) графа називається остовне дерево з мінімальною загальною вагою.

Мінімальне остовне дерево графа можна знайти за допомогою алгоритму Прима, який ми вивчили раніше, або за допомогою алгоритму Краскала.

Алгоритм Краскала. В цьому алгоритмі послідовно вибирають ребра найменшої можливої ваги до утворення остовного дерева. Запишемо алгоритм докладніше.

Крок 1. Взяти ребро з найменшою вагою.

Крок 2. Із ребер, що залишилися, взяти ребро з найменшою вагою, чиє додавання до попередніх взятих ребер не приведе до утворення циклів.

Крок 3. Крок 2 слід повторювати до тих пір, поки неможливо буде додавання ребер без утворення циклів.

Приклад. Граф на рис. 6.23 зображує мережу доріг, що зв'язують 7 сіл. Відстань між селами задано в милях. Знайти мережу доріг мінімальної загальної довжини, що охоплює всі села.

Розв’язання. Спочатку можна взяти ребро AD з вагою 5; другим – EC з вагою 5; третім – DF з вагою 6; четвертим – AB з вагою 7; п’ятим – BE з вагою 7; шостим – EG з вагою 9. Таким чином отримаємо граф на рис 6.24, який є мінімальним остовним деревом заданого графа.

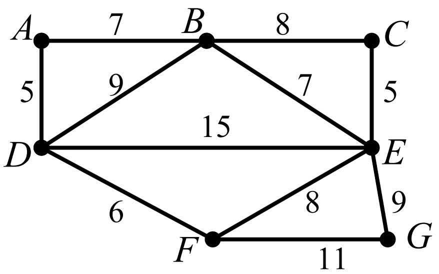


Рис. 6.23

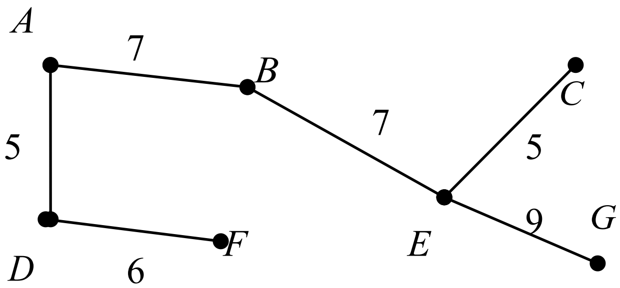


Рис. 6.24

6.8. Дерева з коренем

Коренем дерева називається будь-яка виділена вершина дерева. Дерево з виділеною вершиною називається ***деревом з коренем***.

Корінь дерева звичайно зображують наверху – це вершина нульового рівня (рис. 6.25). Суміжні з коренем вершини зображують нижче – вершини 1 рівня. Вони називаються ***синами*** кореня, а корінь – ***батько*** синів. Далі йдуть вершини 2 рівня. Вершини, розташовані в самому низу (вони не мають синів), називаються ***листями***. Вершини, відмінні від кореня і листя, називаються ***внутрішніми*** вершинами дерева.

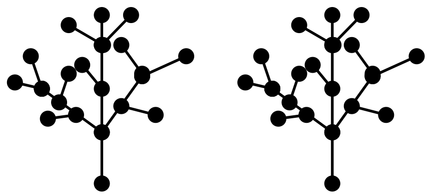


Рис. 6.25 Рис. 6.26

Нульове дерево, - це дерево, яке не має жодної вершини.

Кожна вершина дерева є коренем іншого дерева, яке росте з неї, це дерево називається ***піддеревом*** заданого дерева.

Глибиною вершини дерева називається довжина єдиного шляху, який зв'язує цю вершину з коренем. ***Глибиною дерева*** називається максимальна глибина його вершин.

Сукупність дерев називається ***лісом*** (рис. 6.26).

Якщо $T_1, T_2, ..., T_n$ не зв'язані один з одним дерева з коренями $v_1, v_2, ..., v_n$, то граф, який отримаємо приєднанням кожної з вершин $v_1, v_2, ..., v_n$ до нової вершини v окремим ребром, буде деревом (рис. 6.27).

Дерево називається ***двійковим (бінарним)***, якщо кожна його вершина має не більше двох синів (рис. 6.28). Два піддерева вершини називаються ***лівим і правим піддеревами.***

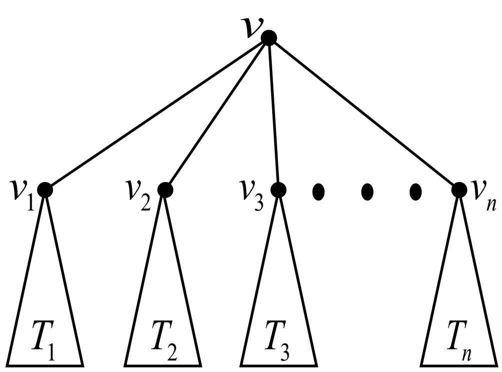


Рис. 6.27

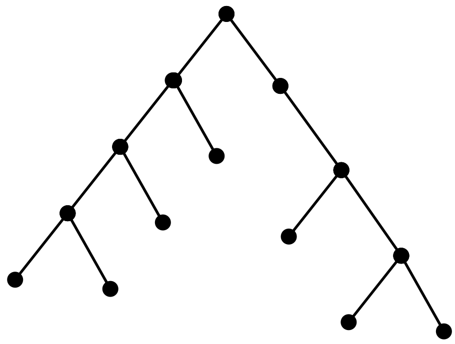


Рис. 6.28

Двійкове дерево називається ***повним***, якщо кожна вершина (окрім листя) має рівно два сина.

Двійкові дерева найчастіше зустрічаються в інформатиці, а дерева взагалі – в різних галузях науки і техніки. Наведемо приклад генеалогічного дерева деяких членів родини Бернуллі (рис. 6.29).

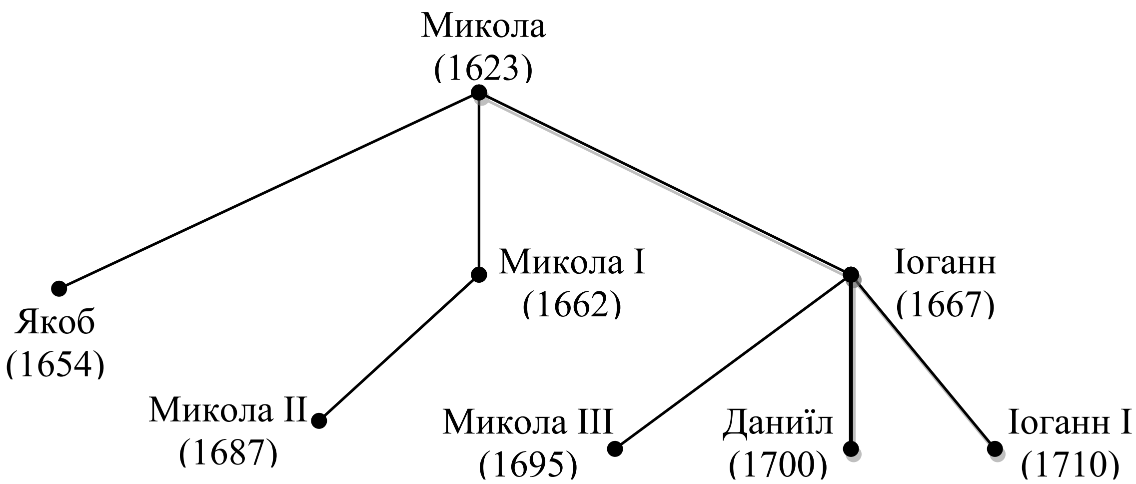


Рис. 6.29

Сімейство Бернуллі дало світу 9 математиків, 3 з яких – великі (Якоб, Іоганн та Даниїл).

Батько Бернуллі займав в місті помітне положення. Був підприємцем. Володів мережею аптек, був членом великої міської ради. У нього було 11 дітей, але математиками стало троє. У старшого брата Якова був син художник. У Йоганна було 5 синів, але науковою діяльністю займалися тільки троє. Сімейство Бернуллі - чудове сімейство.

6.9. Орієнтовані графи

Орієнтований граф (або орграф) також задається двома множинами $G=(V,E)$, де V – скінченна множина вершин, E – множина орієнтованих ребер, які зв'язують деякі з цих вершин. Орієнтовані ребра графа називаються також ***дугами.***

Можна сказати, що множина E задає відношення на множині V: вершина u знаходяться у відношенні з вершиною v, якщо вони зв'язані дугою від u до v (рис. 6.30). Цю дугу можна позначити uv, при цьому вершина u називається ***антецедентом*** вершини v.

Орграф називається ***простим***, якщо в ньому нема кратних дуг і петель. Граф на рис. 6.31 простий. Дуги uv та vu не кратні. Ці дві дуги можна замінити одним неорієнтованим ребром (рис. 6.32).

Задаються орграфи аналогічно неорієнтованим графам.

Наприклад, орграф $G=(V,E)$ з множиною вершин $V=\{a,b,c,d\}$ і множиною дуг $E=\{ab,bd,cb,db,dc\}$ можна зобразити графічно рис. 6.33.

Матриця суміжності цього графа має вигляд

$$\begin{array}{c c} & \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{array}{c} a \\ b \\ c \\ d \end{array} & \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{pmatrix} \end{array}$$

Шляхом довжини k в орграфі G називається послідовність різних вершин $v_0v_1...v_k$ така, що для будь-якого $i=0,1,...,k-1$ пара v_iv_{i+1} є дугою графа.

Наприклад, для графа G на рис. 6.33 шляхом буде $abdc$. Довжина цього шляху дорівнює кількості пройдених дуг, тобто 3.

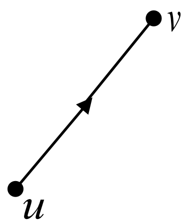


Рис. 6.30

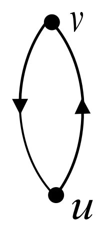


Рис. 6.31

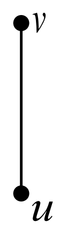


Рис. 6.32

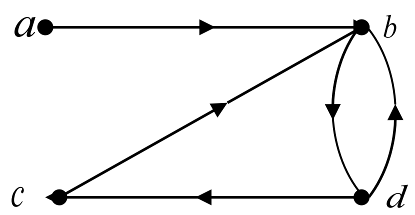


Рис. 6.33

Контуром називається шлях, в якому збігаються початкова й кінцева вершини ($v_0=v_k$).

Для графа G на рис. 6.33 є контур довжиною 3: $bdcb$.

Граф називається ***безконтурним***, якщо в ньому немає контурів.

Безконтурні графи застосовуються при плануванні завдань. Тут контур означав би, що деякі завдання повторювались би необмежену кількість разів. Система планування завдань вперше була розроблена в США в 1958 р. при конструюванні підводного човна. Ця система має кодову назву ПЕРТ. ПЕРТ– це скорочення від "Program Evaluation and Review Technique".

Приклад 6.1. Для засвоєння дискретної математики необхідно вивчити 8 розділів. Залежність розділів один від одного представлена в таблиці 6.1. Треба розробити систему ПЕРТ для визначення послідовності вивчення розділів.

Розв'язання. Математичною моделлю задачі є орграф, зображений на рис. 6.34. Вершини графа позначають розділи, а дуги – послідовність їх вивчення. Наприклад, дуга BA означає, що перед вивченням розділу A слід вивчити розділ B. З цього рисунку видно, що курс можна вивчити, наприклад, в такій послідовності: BADCEFHG.

Тепер уявимо собі, що завдань у нас не 8, а набагато більше. Тоді відповідний граф буде важко уявленим. В цьому разі задачу розв'язують за допомогою так званого алгоритму ***топологічного сортування***.

Таблиця 6.1

	Розділи	Попередні розділи
A	Математична логіка	B
B	Множини	Ніяких вимог
C	Відношення	A
D	Комбінаторика	B
E	Неорієнтовані графи	A, C, D
F	Орієнтовані графи	E
G	Булеві функції	D, H
H	Алгебраїчні структури	C

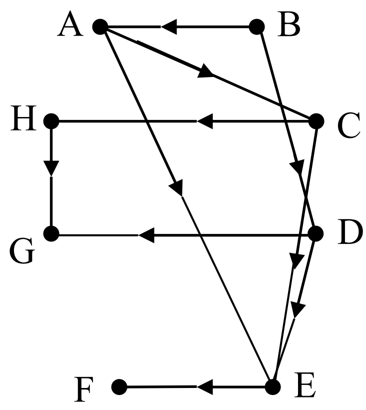


Рис. 6.34

6.10. Алгоритм топологічного сортування

Топологічне сортування – це присвоєння вершинам графа міток $1, 2, ..., n$ (n – число вершин графа) таким чином, щоб виконувалась умова: якщо uv дуга, то мітка вершини u повинна бути менша за мітку вершини v. Запишемо такий алгоритм, позначивши через $A(v)$ – множину антецедентів вершини v, а через l – мітку, яку буде присвоєно вершині.

Крок 1. Обчислимо множину антецедентів $A(v)$ для кожної вершини $v \in V$. Значення мітки l присвоїмо нуль, $l := 0$.

Крок 2. Значення мітки збільшимо на одиницю, $l := l + 1$. Виберемо довільну вершину u з порожнім антецедентом, $A(u) = \varnothing$ та присвоїмо їй мітку l. Із множини антецедентів кожної непоміченої вершини $v \in V$ вилучимо вершину u: $A(v) := A(v) \setminus \{u\}$.

Крок 3. Продовжуємо крок 2 поки є непомічені вершини v з порожнім антецедентом ($A(v) = \varnothing$).

Приклад. Прослідкуйте за роботою алгоритму топологічного сортування на прикладі 6.1.

Розв'язання. Запишемо множини антецедентів всіх вершин графа:

$A(\mathrm{A})=\{\mathrm{B}\}$, $A(\mathrm{B})=\varnothing$, $A(\mathrm{C})=\{\mathrm{A}\}$, $A(\mathrm{D})=\{\mathrm{B}\}$, $A(\mathrm{E})=\{\mathrm{A,C,D}\}$, $A(\mathrm{F})=\{E\}$, $A(\mathrm{G})=\{\mathrm{D,H}\}$, $A(\mathrm{H})=\{\mathrm{C}\}$; мітка $l=0$.

1 крок циклу: є непомічені вершини з порожнім антецедентом. Тому $l=1$, $u:=\mathrm{B}$, присвоїмо вершині u мітку l: B_1; з множин антецедентів всіх непомічених вершин вилучимо вершину u. Отримаємо:

$A(\mathrm{A})=\varnothing$, $A(\mathrm{C})=\{\mathrm{A}\}$, $A(\mathrm{D})=\varnothing$, $A(\mathrm{E})=\{\mathrm{A,C,D}\}$, $A(\mathrm{F})=\{E\}$, $A(\mathrm{G})=\{\mathrm{D,H}\}$, $A(\mathrm{H})=\{\mathrm{C}\}$.

2 крок циклу: є непомічені вершини з порожнім антецедентом. Тому $l=2$, $u:=\mathrm{A}$, присвоїмо вершині u мітку l: A_2; з множин антецедентів всіх непомічених вершин вилучимо вершину u. Отримаємо: $A(\mathrm{C})=\varnothing$, $A(\mathrm{D})=\varnothing$, $A(\mathrm{E})=\{\mathrm{C,D}\}$, $A(\mathrm{F})=\{E\}$, $A(\mathrm{G})=\{\mathrm{D,H}\}$, $A(\mathrm{H})=\{\mathrm{C}\}$.

3 крок циклу: є непомічені вершини з порожнім антецедентом. Тому $l=3$, $u:=\mathrm{C}$, присвоїмо вершині u мітку l: C_3; з множин антецедентів всіх непомічених вершин вилучимо вершину u. Отримаємо:

$A(\mathrm{D})=\varnothing$, $A(\mathrm{E})=\{\mathrm{D}\}$, $A(\mathrm{F})=\{E\}$, $A(\mathrm{G})=\{\mathrm{D,H}\}$, $A(\mathrm{H})=\varnothing$.

4 крок циклу: є непомічені вершини з порожнім антецедентом. Тому $l=4$, $u:=\mathrm{D}$, присвоїмо вершині u мітку l: D_4; з множин антецедентів всіх непомічених вершин вилучимо вершину u. Отримаємо:

$A(\mathrm{E})=\varnothing$, $A(\mathrm{F})=\{E\}$, $A(\mathrm{G})=\{\mathrm{H}\}$, $A(\mathrm{H})=\varnothing$.

5 крок циклу: є непомічені вершини з порожнім антецедентом. Тому $l=5$, $u:=\mathrm{E}$, присвоїмо вершині u мітку l: E_5; з множин антецедентів всіх непомічених вершин вилучимо вершину u. Отримаємо:

$A(\mathrm{F})=\varnothing$, $A(\mathrm{G})=\{\mathrm{H}\}$, $A(\mathrm{H})=\varnothing$.

6 крок циклу: є непомічені вершини з порожнім антецедентом. Тому $l=6$, $u:=\mathrm{F}$, присвоїмо вершині u мітку l: F_6; з множин антецедентів всіх непомічених вершин вилучимо вершину u. Отримаємо:

$A(\mathrm{G})=\{\mathrm{H}\}$, $A(\mathrm{H})=\varnothing$.

7 крок циклу: є непомічені вершини з порожнім антецедентом. Тому $l = 7$, $u := \mathrm{H}$, присвоїмо вершині u мітку l: H_7; з множин антецедентів всіх непомічених вершин вилучимо вершину u. Отримаємо: $A(\mathrm{G})=\varnothing$.

8 крок циклу: є непомічені вершини з порожнім антецедентом. Тому $l = 8$, $u := \mathrm{G}$, присвоїмо вершині u мітку l: G_8; з множин антецедентів всіх непомічених вершин треба вилучити вершину u.

Непомічених вершин немає, тому виходимо з циклу і отримаємо

$$\mathrm{B}_1\mathrm{A}_2\mathrm{C}_3\mathrm{D}_4\mathrm{E}_5\mathrm{F}_6\mathrm{H}_7\mathrm{G}_8.$$

Відповідь. Алгоритм топологічного сортування пропонує вивчати дискретну математику в наступній послідовності:

1) Множини.
2) Математична логіка.
3) Відношення.
4) Комбінаторика.
5) Неорієнтовані графі.
6) Орієнтовані графи.
7) Алгебраїчні структури.
8) Булеві функції.

6.11. Досяжність в орграфах

Орграфи застосовуються для схематичного зображення авроліній, що з'єднують міста усього світу, комунікаційних мереж між комп'ютерами. У таких мережах важливо знати наслідки виключень будь-якого з'єднання (дуги або вершини) по всій мережі.

Наприклад, якщо літак не може приземлитися для дозаправки в деякому місті внаслідок несприятливих погодних умов, то помилка в його переадресації загрожує катастрофою: йому може не вистачити пального для досягнення неправильно призначеного аеропорту. Аналогічно, якщо одна чи кілька ланцюгів в комп'ютерній мережі не працюють, то для деяких користувачів окремі сервери можуть виявитися недоступними.

Таким чином, ми приходимо до задачі пошуку шляхів між довільною парою вершин в орграфі.

Нехай $G=(V,E)$ орграф з n вершинами, M – його матриця суміжності. Дуга являє собою шлях, довжини 1. Розглянемо булевий добуток матриць $M^2 = M \cdot M$. Якщо в i-му рядку та j-му стовпці цієї матриці стоїть 1, то це означає, що від i-ї до j-ї вершини існує шлях довжини 2. За матрицею $M^3 = M^2 \cdot M$ можна знайти всі шляхи, довжиною 3. За матрицею M^k можна знайти всі шляхи, довжиною k. Нарешті, за

матрицею досяжності, $M^* = M \vee M^2 \vee ... \vee M^n$ можна знайти всі шляхи між вершинами. Про довжини цих шляхів нічого сказати не можна.

Матриця досяжності орграфа є матрицею замикання за транзитивністю відношення E на вершинах орграфа.

Приклад. Знайти матрицю досяжності орграфа на рис. 6.35.

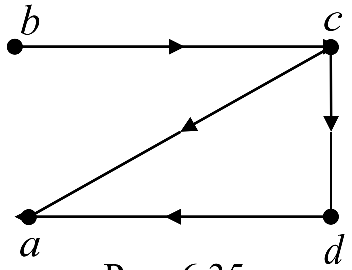


Рис. 6.35

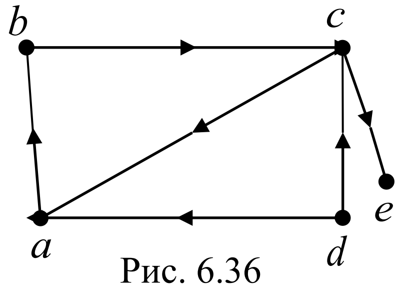


Рис. 6.36

Розв'язання. Спочатку знайдемо матрицю суміжності графа:

$$M = \begin{array}{c} \\ a \\ b \\ c \\ d \end{array} \begin{array}{c} \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix} \end{array}.$$

Ця матриця показує всі шляхи одиничної довжини. Для визначення шляхів, довжини 2, знайдемо матрицю M^2:

$$M^2 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

В цій матриці 3 одинички. Це означає, що існують троє таких шляхів: від b до a, від b до d та від c до a, що узгоджується з рис. 6.35.

Знайдемо тепер шляхи, довжиною 3:

$$M^3 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

В матриці M^3 всього одна одиничка: існує всього один шлях (від b до a), довжини 3. Знайдемо, нарешті, шляхи, довжини 4;

$$M^4 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

Ми отримали нульову матрицю. В нашому графі немає жодного шляху такої довжини.

Тепер знайдемо всі шляхи в нашому орграфі. Матриця досяжності має вигляд

$$M^* = M \vee M^2 \vee M^3 \vee M^4 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix}.$$

Зауваження 1. Замість булевого добутку матриць можна взяти звичайний. В цьому випадку позначимо матрицю суміжності буквою C. Тоді елемент матриці C^k, що знаходиться в i-му рядку та j-му стовпці буде дорівнювати числу шляхів довжини k із i-ї вершини в j-ту. Матриця досяжності в цьому випадку буде мати вигляд

$$C^* = C + C^2 + \ldots + C^n.$$

Елемент цієї матриці, що знаходиться в i-му рядку та j-му стовпці буде дорівнювати числу шляхів із i-ї вершини в j-ту.

В нашому прикладі $M^k = C^k$, $k = 1,2,3,4$. Але

$$C^* = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 2 & 0 & 1 & 1 \\ 2 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix}.$$

Зауваження 2. Замість одиничок матриці суміжності можна записати відповідні ребра. Перед цим ребра потрібно пойменувати. Замість суми потрібно записати об’єднання, а замість множення – приписування символів. В результаті будемо отримувати матриці зі словами. Ці слова будуть вказувати шляхи.

В попередньому прикладі, якщо одинички матриці суміжності замінити відповідними ребрами, то вона прийме вигляд:

$$M = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & bc & 0 \\ ca & 0 & 0 & cd \\ da & 0 & 0 & 0 \end{pmatrix}.$$

А матриці M^2 та M^3 будуть такими:

$$M^2=\begin{pmatrix} 0 & 0 & 0 & 0 \\ bc\,ca & 0 & 0 & bc\,cd \\ cd\,da & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix},\ M^3=\begin{pmatrix} 0 & 0 & 0 & 0 \\ bc\,cd\,da & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

Матриця M^4 буде нульовою. Запишемо матрицю суміжності:

$$M^*=\begin{pmatrix} 0 & 0 & 0 & 0 \\ bc\,ca \cup bc\,cd\,da & 0 & bc & bc\,cd \\ ca \cup cd\,da & 0 & 0 & cd \\ da & 0 & 0 & 0 \end{pmatrix}$$

Розглянутий нами спосіб знаходження матриці досяжності є трудомістким для матриць великої розмірності. Найбільш ефективним є наступний алгоритм Уоршелла.

Алгоритм Уоршелла. Нехай $G=(V,E)$ орграф з n вершинами, M – його матриця суміжності. Алгоритм Уоршелла послідовно обчислює матриці $W_0=M$, W_1, W_2, ... , $W_n=M^*$. Звернемо увагу, що на останньому кроці алгоритму обчислюється матриця досяжності. Якщо в i - му рядку та j - му стовпці матриці W_k стоїть 1, то це означає, що існує шлях невідомої довжини від вершини i до вершини j. Внутрішніми вершинами цього шляху можуть бути лише вершини 1, ... , *k*. Запишемо сам алгоритм.

begin
  $W=M$;
  for $k:=1$ ***to*** n ***do***
    for $i:=1$ ***to*** n ***do***
      for $j:=1$ ***to*** n ***do***

$$w(i,j)=w(i,j)\vee\big(w(i,k)\wedge w(k,j)\big); \tag{6.1}$$

end

З формули (6.1) витікає наступне:

$$w(i,k)=0\ \Rightarrow w(i,j)=w(i,j)\vee\big(0\wedge w(k,j)\big)=w(i,j)\vee 0=w(i,j);$$

$$w(i,k)=1\ \Rightarrow w(i,j)=w(i,j)\vee\big(1\wedge w(k,j)\big)=w(i,j)\vee w(k,j).$$

Звідси ***правило***. Для побудови матриці W_k переглядаємо k-й стовпець попередньої матриці:

якщо в i - му рядку стоїть 0, то цей рядок переписуємо без змін;

якщо в i - му рядку стоїть 1, то цей рядок спаровуються з k-м рядком диз'юнкцією.

Приклад. Знайти матрицю досяжності орграфа на рис. 6.36.

Розв'язання. За алгоритмом Уоршелла будуємо послідовно матриці

$$W_0 = M = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \quad W_1 = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

При побудові матриці W_1 дивимось на 1 стовпець матриці W_0. Там в першому, другому і в п'ятому рядках стоять нулі. Тому ці рядки переписуємо в матрицю W_1. В третьому і четвертому рядках стоять одиниці. Ці рядки спарюємо з першим рядком диз'юнкцією. Аналогічно знаходимо

$$W_2 = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \; M^* = W_5 = W_4 = W_3 = \begin{pmatrix} 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

Тут $W_4 = W_3$ тому, що в четвертому стовпці матриці W_3 стоять нулі, значить, всі рядки цієї матриці переписуються. Нулями заповнений і 5-й рядок матриці W_3, тому його спарювання з будь-яким рядком нічого не міняє, отже $W_5 = W_4$.

Проаналізуємо знайдені матриці. Якщо матриця W_1 вказує на шляхи, в яких є внутрішні вершини, то внутрішньою вершиною може бути лише вершина a. Якщо матриця W_2 вказує на шляхи, в яких є внутрішні вершини, то внутрішніми вершинами можуть бути лише вершини a і b. Нарешті, якщо матриця W_3 вказує на шляхи, в яких є внутрішні вершини, то внутрішніми вершинами можуть бути лише вершини a, b і c.

6.12. Найкоротші шляхи в орграфах

В попередньому пункті ми займались знаходженням шляхів в орграфі. На практиці не менш важливою є задача знаходження найкоротших шляхів. Розглянемо приклад. Дана мережа автомобільних доріг, що з'єднують міста Одеської області. Деякі дороги односторонні. Знайти найкоротші шляхи від Одеси до кожного міста області. Математичною моделлю цієї задачі буде, очевидно, наступна задача.

Дано зважений граф без петель і ребер від'ємної ваги. Треба знайти найкоротші шляхи від заданої вершини А до всіх інших вершин графа.

Найбільш популярним алгоритмом розв'язання цієї задачі є ***алгоритм Дейкстри***. Ідея алгоритму полягає в наступному.

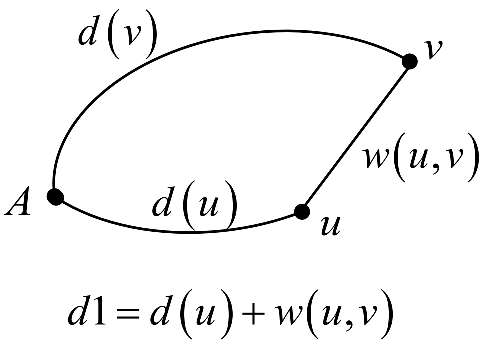
Спочатку складемо вагову матрицю, елементи якої обчислюються за формулою

$$w_{ij}=\begin{cases}0, & \text{якщо} \quad i=j,\\ \infty, & \text{якщо вершини } i \text{ та } j \text{ не з'єднані ребром,}\\ d, & \text{якщо } ij - \text{ребро ваги } d.\end{cases}$$

Протягом роботи алгоритму кожній вершині v графа присвоюється число $d(v)$, що дорівнює відстані від вершини А до v. Спочатку ці значення можна взяти із першого рядка вагової матриці W, тобто $d(v)$ збігається з вагою ребра Av, якщо таке існує, або дорівнює ∞ в протилежному випадку. Вершину A потрібно помітити.

На кожному наступному кроці серед непомічених вершин вибирається вершина u, найближча до A і помічається. Відстань $d(u)$ від цієї вершини до A далі змінюватися не буде. Для непомічених вершин v число $d(v)$ може змінитися, якщо існує ребро uv і якщо довжина шляху Auv менша за $d(v)$ (рис. 6.37). Алгоритм закінчить роботу в той момент, коли всі вершини будуть помічені і отримають свої остаточні значення $d(v)$.

Приклад. За допомогою алгоритму Дейкстри знайдіть найкоротші шляхи від вершини A до вершин графа на рис. 6.38.



$$d1=d(u)+w(u,v)$$

Якщо d1<d(v), то d(v):=d1

Рис. 6.37

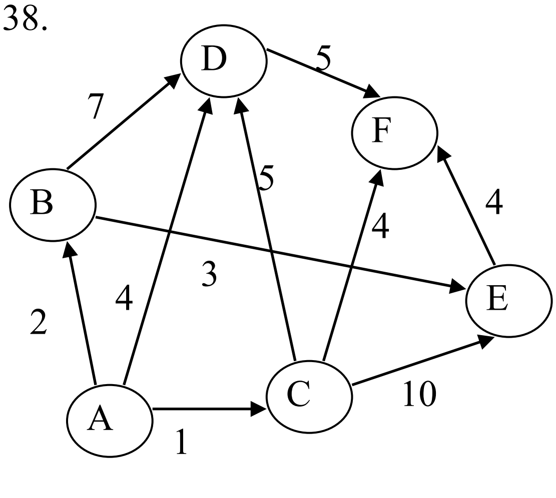


Рис.6.38

Розв'язання. Складемо вагову матрицю

$$
W = \begin{array}{c|c}
 & \begin{array}{cccccc} A & B & C & D & E & F \end{array} \\
\begin{array}{c} A \\ B \\ C \\ D \\ E \\ F \end{array} &
\begin{pmatrix}
0 & 2 & 1 & 4 & \infty & \infty \\
\infty & 0 & \infty & 7 & 3 & \infty \\
\infty & \infty & 0 & 5 & 10 & 4 \\
\infty & \infty & \infty & 0 & \infty & 5 \\
\infty & \infty & \infty & \infty & 0 & 4 \\
\infty & \infty & \infty & \infty & \infty & 0
\end{pmatrix}
\end{array}.
$$

Результати роботи алгоритму запишемо у таблицю 6.2.

Крок 0. Помічаємо вершину A і в першому рядку запишемо відстані від A до всіх інших вершин (перший рядок вагової матриці).

Крок 1. Найменше число в першому рядку дорівнює 1: $d(C)=1$. Отже C – найближча вершина до A. Помічаємо вершину C. Попередньою вершиною до C є вершина A (відстань від C до A стала остаточною, коли була помічена вершина A).

Обчислимо відстані від A до непомічених вершин через вершину C. Якщо нові значення відстаней стануть менше за старих, то змінимо нові значення замість старих. Так, від C можна потрапити до D за 5 од., до E – за 10 од, і до F – за 4 од. (це видно з вагової матриці W). Враховуючи, що відстань від A до C дорівнює 1, то відстані ACD, ACE і ACF дорівнюватимуть 6, 11 і 5 од. відповідно. Отже, відстані від A до E і до F стали менше ніж були раніше (∞). Заповнюючи рядок таблиці, який відповідає кроку 1, ми замінили $d(E)$ на 11 і $d(F)$ на 5.

Таблиця 6.2

Крок	Помічені вершини	Попередні вершини	Відстань до вершини					
			A	B	C	D	E	F
0	A		0	2	**1**	4	∞	∞
1	C	A	0	**2**	1	4	11	5
2	B	A	0	2	1	**4**	5	5
3	D	A	0	2	1	4	**5**	5
4	E	B	0	2	1	4	5	**5**
5	F	C	0	2	1	4	5	5

Крок 2. В рядку кроку 1 найменше число 2, яке відповідає відстані від А до непоміченої вершини В. Помічаємо вершину В. Попередньою для неї буде вершина А.

Обчислимо відстані від А до непомічених вершин через вершину В. Від В можна потрапити до D за 7 од. і до Е – за 3 од. Враховуючи, що відстань від А до В дорівнює 2, то відстані ABD і АВЕ дорівнюватимуть відповідно 9 і 5 од. відповідно. Отже, відстань від А до Е стала менше ніж були раніше (11). Заповнюючи рядок таблиці, який відповідає кроку 2, ми замінили $d(E)$ на 5.

Крок 3. Серед непомічених вершин (D, E, F) найближчою до А є вершина D: $d(D)=4$. Помічаємо вершину D. Попередньою до вершини D буде вершина А. Від D можна потрапити до F за 5 од. Відстань ADF становитиме 4+5=9 од. Вона не зменшує попередню (5 од.). Отже, відстань AF залишається рівною 5 од.

Крок 4. Залишилися дві непомічені вершин Е та F, обидві знаходяться на однакової відстані від А. Помітити можна будь-яку з них. Помічаємо вершину Е. Від неї можна потрапити до F за 4 од. Відстань AEF становитиме 5+4=9 од. Вона не зменшує попередню(5 од.). Отже, відстань AF залишається рівною 5 од. Попередньою до вершини Е буде вершина В.

Крок 5. Помічаємо вершину F: $d(F)=5$. Попередньою для неї буде вершина С.

Алгоритм Дейкстри закінчив свою роботу. В останньому рядку таблиці представлені найкоротші відстані від вершини A до всіх інших вершин орграфа. У третьому стовпці таблиці для кожної вершини вказана попередня вершина найкоротшого шляху. Це дає можливість вказати найкоротший шлях для будь-якої вершини. Наприклад, довжина найкоротшого шляху від A до E дорівнює 5. Цей шлях проходить через вершини ABE.

Набір вправ до розділу 6. Графи

6.1. Повний граф має n вершин. При яких значення n граф буде ейлеревим? Який вигляд має матриця суміжності повного графа?

6.2. Позначте вершини графів на рис. 6,39 та запишіть для них відповідні матриці суміжностей:

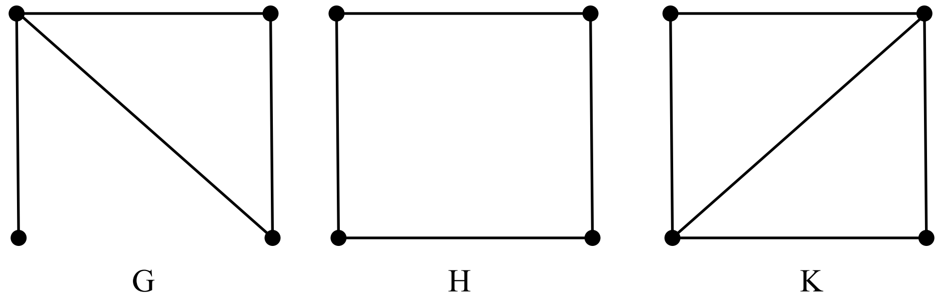


Рис. 6.39

6.3. Зобразіть граф G, матриця суміжності якого має вигляд:

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

Які із графів на рис. 6.40 можуть бути підграфами графа G?

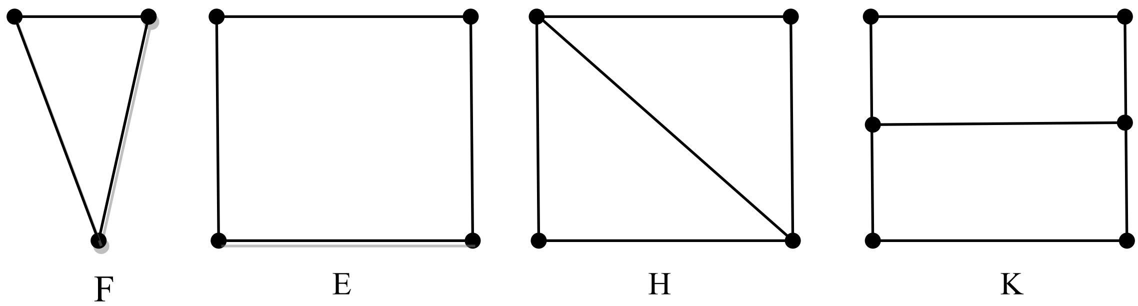


Рис. 6.40

6.4. За принципом Діріхле доведіть, що якщо простий граф має більше однієї вершини, то у нього знайдуться хоча б дві вершини однакового степеня.

6.5. Знайдіть гамільтоновий цикл у графі, що зображений на рис. 6.41.

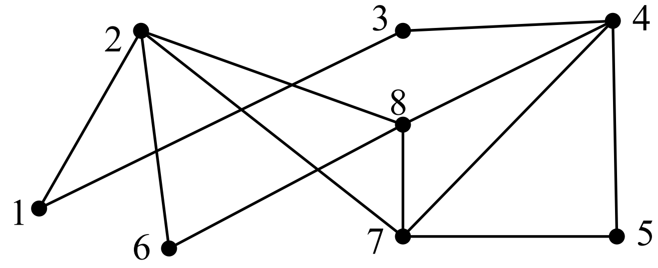


Рис. 6.41

6.6. На рис. 6.42 зображено граф Петерсена. Знайдіть в ньому цикл довжини 9. Доведіть, що граф не є гамільтоновим.

6.7. За алгоритмом найближчого сусіда знайдіть гамільтонів цикл у зваженому графі, що зображений на рис. 6.43, взявши за початкову вершину: а) вершину A; б) вершину D.

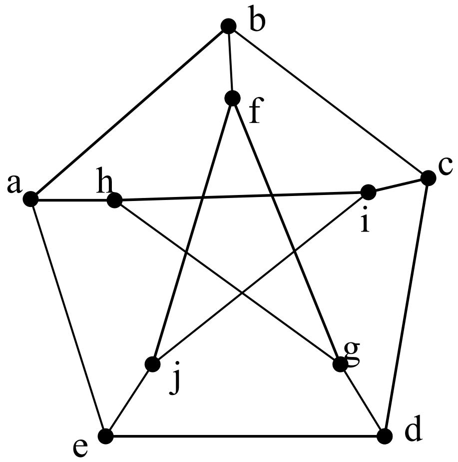


Рис. 6.42

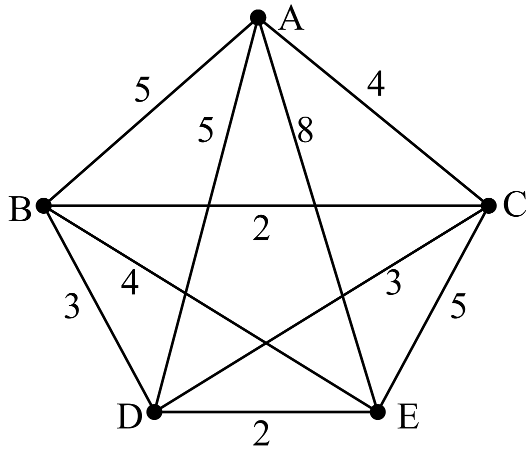


Рис. 6.43

6.8. Доведіть, що граф $G=(V,E)$ є ейлеровим і знайдіть ейлерів цикл, якщо

$$V=\{1,2,3,4,5,6\},\ E=\{(1,2),(1,4),(2,3),(2,4),(2,5),(3,4),(4,6),(5,6)\}$$

6.9. Знайдіть ейлерові цикли або шляхи в графах, що зображені на рис. 6.44 – 6.47.

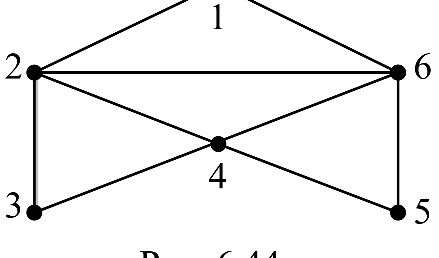


Рис. 6.44

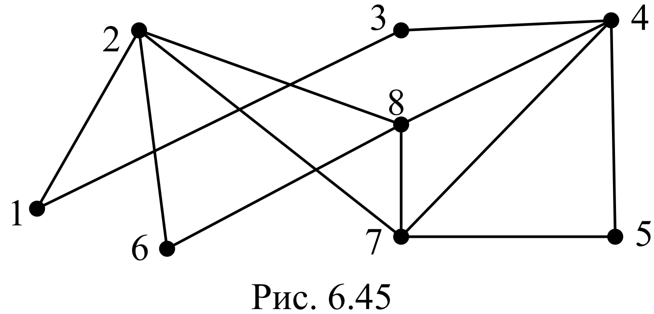


Рис. 6.45

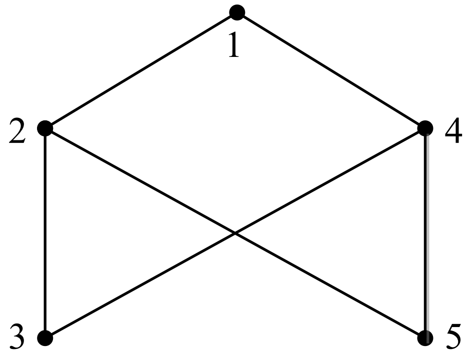


Рис. 6.46

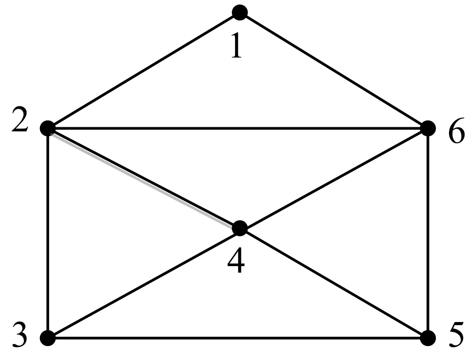


Рис. 6.47

6.10. З’ясуйте, які із наступних графів, що задані матрицями суміжностей, є деревами:

$$M = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \end{pmatrix} \quad N = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}$$

6.11. Дерево T має 3 вершини степеня 3 і 4 вершини степеня 2. Решта вершин дерева мають степінь 1. Скільки вершин степеня 1 має дерево?

6.12. Лісом називається граф, кожна компонента зв’язності якого – дерево. Нехай G - ліс з n вершинами і k компонентами зв’язності.

А) Доведіть, що G має $n-k$ ребер.

Б) Доведіть, що якщо в кожній компоненті зв’язності лісу G є більше однієї вершини, то G містить принаймні $2k$ вершин степеня 1.

В) Зобразіть ліс з дев’ятьма вершинами та шістьма ребрами, в якому не більше п’яти вершин степеня 1.

6.13. Знайдіть мінімальне остовне дерево графа, зображеного на рис. 6.48.

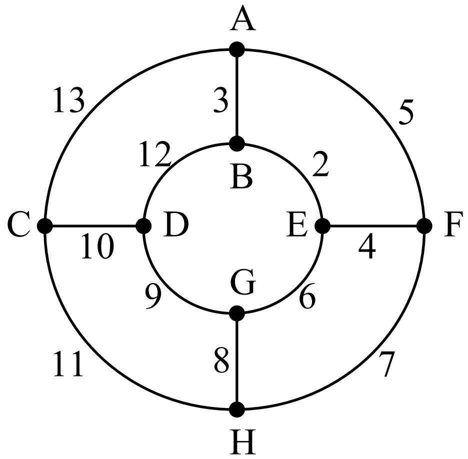


Рис. 6.48

6.14. В таблиці 6.3 наведено відстані між шістьма містами. Знайдіть мережу доріг мінімальної загальної довжини, що зв'язує всі шість міст.

Таблиця 6.3

	А	Б	В	Г	Д	Е
А		78	56	73	71	114
Б	78		132	121	135	96
В	56	132		64	85	154
Г	73	121	64		144	116
Д	71	135	85	144		185
Е	114	96	154	116	185	

6.15. Глибина вершини v дерева T з коренем визначається як довжина єдиного шляху від неї до кореня дерева. Глибина графа T - це максимальна глибина його вершин. Зобразіть наступні дерева:

А) дерево з коренем глибини 1 з шістьма вершинами;

Б) повне двійкове дерево з коренем глибини 2;

В) дерево з коренем глибини 3, кожна вершина глибини $i\left(i \geq 0\right)$ якого має $\left(i+1\right)$ сина.

Доведіть за індукцією, що повне двійкове дерево з коренем глибини n має 2^n листя.

6.16. Зобразіть орграф з вершинами $\left\{1,2,3,4,5,6\right\}$ та матрицею суміжності M. Уявіть, що вага кожної дуги дорівнює 1 та знайдіть (якщо він існує)

А) найкоротший шлях від вершини 1 до вершини 2;

Б) найкоротший шлях від вершини 3 до вершини 6;

В) контур довжини 5.

$$M = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 \end{pmatrix}.$$

6.17. Зв'язним називається такий орграф, з якого виходить зв'язний граф, якщо забути про орієнтацію дуг. Якщо для будь якої упорядкованої пари вершин існує шлях від однієї вершини до другої, то такий орграф називається сильно зв'язним.

А) Визначте, який із графів на рис. 6.49 є сильно зв'язним.

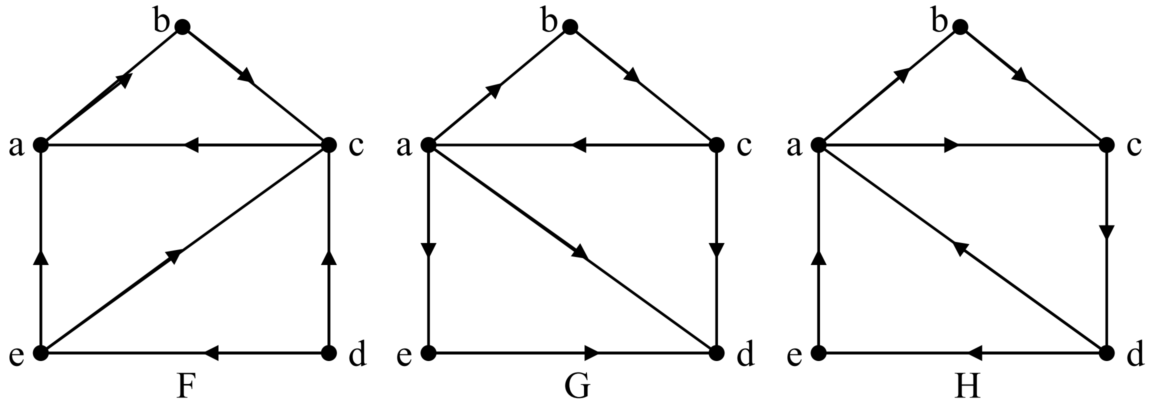


Рис. 6.49

Б) Поясніть, як потрібно орієнтувати ребра гамільтонового графу, щоб з нього вийшов сильно зв'язний граф.

В) Чому орграф, що являє собою систему односторонніх доріг у місті, повинен бути сильно зв'язним?

6.18. Застосуйте алгоритм топологічного сортування до орграфу з вершинами (a,b,c,d,e,f) та матрицею суміжності

$$M=\begin{pmatrix}0&0&0&1&0&0\\1&0&0&0&0&1\\0&1&0&0&1&0\\0&0&0&0&0&1\\0&0&0&1&0&1\\0&0&0&0&0&0\end{pmatrix}$$

Запишіть нову матрицю суміжності, рядки та стовпці якої упорядковані у відповідності з новими позначеннями вершин. Що можна сказати про нову матрицю?

Що можна сказати про алгоритм топологічного сортування для орграфа із задачі 6.16?

6.19. Для засвоєння елементарної математики необхідно вивчити 8 розділів. Залежність розділів один від одного представлена в таблиці 6.4. Треба розробити систему ПЕРТ для визначення послідовності вивчення розділів.

6.20. Матриця суміжності орграфа G має вигляд

$$M=\begin{pmatrix}0&1&0&0\\0&0&1&0\\0&0&0&1\\1&0&0&0\end{pmatrix}.$$

Обчисліть M^2, M^3 та M^4. Знайдіть матрицю досяжності M^*.

Таблиця 6.4

Завдання	Зміст завдання	Послідовність попередніх дій
А	Арифметика і алгебра	Ніяких
Б	Рівняння і нерівності	А,В
В	Тригонометрія	А
Г	Початки аналізу	А,В,Д
Д	Геометрія	А, В
Е	Вектори	А,Д
Ж	Елементи теорії імовірностей	А
З	Функції і графіки	А,В

6.21. За алгоритмом Уоршелла обчисліть W_1, W_2, W_3 та W_4. Для орграфа G із вправи 6.20. Чому дорівнює матриця досяжності M^*?

6.22. Прослідкуйте за роботою алгоритму Дейкстри на прикладі орграфа, що зображений на рисунку та знайдіть найкоротші шляхи до кожної вершини

А) від вершини A;

Б) від вершини C.

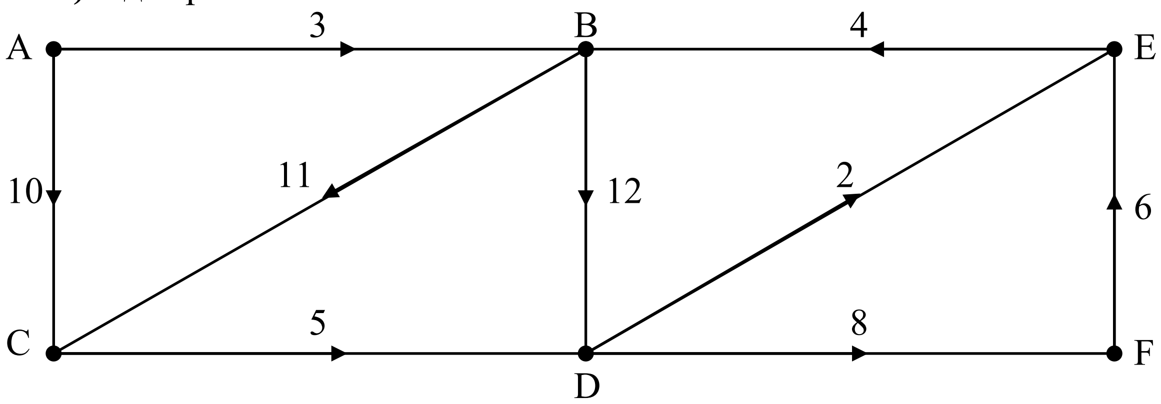


Рис. 6.50

ВІДПОВІДІ

РОЗДІЛ 5. БУЛЕВІ ФУНКЦІЇ

5.3. a) ДДНФ: $\overline{x}\overline{y}\overline{z} \vee \overline{x}y\overline{z} \vee x\overline{y}z \vee xyz$.
ДКНФ: $\left(x \vee y \vee \overline{z}\right)\left(x \vee \overline{y} \vee \overline{z}\right)\left(\overline{x} \vee y \vee z\right)\left(\overline{x} \vee \overline{y} \vee z\right)$.
b) ДДНФ: $\overline{x}\overline{y}\overline{z}t \vee \overline{x}\overline{y}z\overline{t} \vee x\overline{y}zt \vee xy\overline{z}t$. ДКНФ:
$\left(x \vee y \vee z \vee t\right)\left(x \vee y \vee \overline{z} \vee \overline{t}\right)\left(x \vee \overline{y} \vee z \vee t\right)\left(x \vee \overline{y} \vee z \vee \overline{t}\right)$
$\left(x \vee \overline{y} \vee \overline{z} \vee t\right)\left(x \vee \overline{y} \vee \overline{z} \vee \overline{t}\right)\left(\overline{x} \vee y \vee z \vee t\right)\left(\overline{x} \vee y \vee z \vee \overline{t}\right)$
$\left(\overline{x} \vee y \vee \overline{z} \vee t\right)\left(\overline{x} \vee \overline{y} \vee z \vee t\right)\left(\overline{x} \vee \overline{y} \vee \overline{z} \vee t\right)\left(\overline{x} \vee \overline{y} \vee \overline{z} \vee t\right)$.

5.4. a)

x	y	z	$\overline{y}$	$\overline{y} \vee z$	$x\left(\overline{y} \vee z\right)$	$\overline{x}$	$\overline{z}$	$y \vee \overline{z}$	$\overline{x}\left(y \vee \overline{z}\right)$	$f\left(x,y,z\right)$
0	0	0	1	1	0	1	1	1	1	1
0	0	1	1	1	0	1	0	0	0	0
0	1	0	0	0	0	1	1	1	1	1
0	1	1	0	1	0	1	0	1	1	1
1	0	0	1	1	1	0	1	1	0	1
1	0	1	1	1	1	0	0	0	0	1
1	1	0	0	0	0	0	1	1	0	0
1	1	1	0	1	1	0	0	1	0	1

ДДНФ: $f\left(x,y,z\right) = \overline{x}\overline{y}\overline{z} \vee \overline{x}y\overline{z} \vee \overline{x}yz \vee x\overline{y}\overline{z} \vee x\overline{y}z \vee xyz$,

ДКНФ: $f\left(x,y,z\right) = \left(x \vee y \vee \overline{z}\right)\left(\overline{x} \vee \overline{y} \vee z\right)$,

Многочлен Жегалкіна: $f\left(x,y,z\right) = xy + xz + yz + z + 1$, не зберігає константу 0, зберігає константу 1, не самодвоїсна, не лінійна, не монотонна.

b)

x	y	z	$\overline{y}$	$\overline{y} \vee z$	$x\left(\overline{y} \vee z\right)$	yz	$x\left(\overline{y} \vee z\right) \vee yz$	xy	$f\left(x,y,z\right)$
0	0	0	1	1	0	0	0	0	1
0	0	1	1	1	0	0	0	0	1
0	1	0	0	0	0	0	0	0	1
0	1	1	0	1	0	1	1	0	0
1	0	0	1	1	1	0	1	0	0
1	0	1	1	1	1	0	0	0	1
1	1	0	0	0	0	1	1	1	1
1	1	1	0	1	1	1	1	1	1

ДДНФ: $f\left(x,y,z\right) = \overline{x}\overline{y}\overline{z} \vee \overline{x}\overline{y}z \vee \overline{x}y\overline{z} \vee x\overline{y}z \vee xy\overline{z} \vee xyz$, ДКНФ: $f\left(x,y,z\right)=\left(x\vee\overline{y}\vee\overline{z}\right)\left(\overline{x}\vee y\vee z\right)$,

Многочлен Жегалкіна: $f\left(x,y,z\right)=xy+xz+yz+x+1$, не зберігає константу 0, зберігає константу 1, не самодвоїсна, не лінійна, не монотонна.

5.5. a) $\left(\left(\left(x\left(y\mid y\right)\right)\mid\left(x\left(y\mid y\right)\right)\right)\mid z\right)\mid\left(\left(\left(x\left(y\mid y\right)\right)\mid\left(x\left(y\mid y\right)\right)\right)\mid z\right)$; **b)** $\overline{x}=x\downarrow x$; $xy=\left(x\downarrow x\right)\downarrow\left(y\downarrow y\right)$; $x\vee y=\left(x\downarrow y\right)\downarrow\left(x\downarrow y\right)$; $x\overline{y}z=\left(\left(\left(x\downarrow x\right)\downarrow\left(\left(y\downarrow y\right)\downarrow\left(y\downarrow y\right)\right)\right)\right)\downarrow\left(z\downarrow z\right)$. **5.6. a)** $xz\vee\overline{x}\overline{z}$; **b)** y; **c)** $y\overline{z}\vee\overline{x}zt\vee x\overline{y}t$; **d)** $\overline{x}t\vee x\overline{y}\overline{z}$; **e)** $\overline{y}\overline{z}\vee xyz\vee\overline{x}zt$. **5.7.** Наведемо доведення справедливості рівності: $\overline{x}\mid\left(\overline{y}\mid z\right)=\overline{\overline{x}\left(\overline{\overline{y}z}\right)}=\overline{\overline{x}\left(y\vee\overline{z}\right)}=x\vee\overline{y}z$. Ліва частина рівності $\overline{x}\mid\left(\overline{y}\mid z\right)$ реалізується логічною схемою, що представлена на рис. В.5.1, а права частина $x\vee\overline{y}z$ - логічною схемою, що зображена на рис. В.5.2.

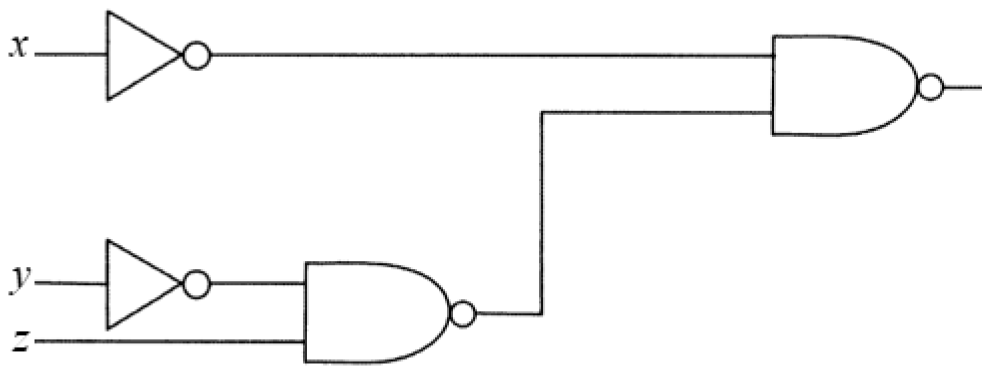


Рис. В.5.1

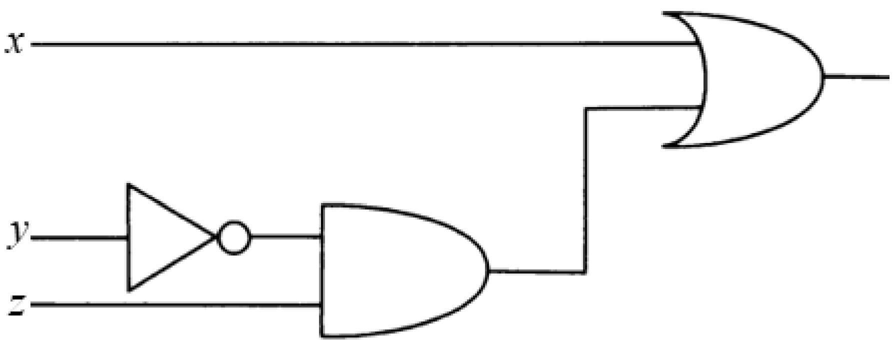


Рис. В.5.2

5.8. Пронумеруємо логічні елементи логічної схеми, як показано на рис. В.5.3 і запишемо в таблицю В.5.1 що входить в кожний логічний елемент і що з нього виходить. Отже, булева функція, що реалізується логічною схемою, зображеною на рис. В.5.3 має вигляд

$$f\left(x,y,z\right)=yz\vee xy\overline{z}\vee\overline{x}\,\overline{y}z.$$

Запишемо її у вигляді ДДНФ:

$$f\left(x,y,z\right)=\underline{xyz}\vee\bar{x}yz\vee\underline{xy\bar{z}}\vee\bar{x}\,\bar{y}z\,.$$

Групуючи першу кон'юнкцію з третьою, а другу − з четвертою, знайдемо мінімальну ДНФ.

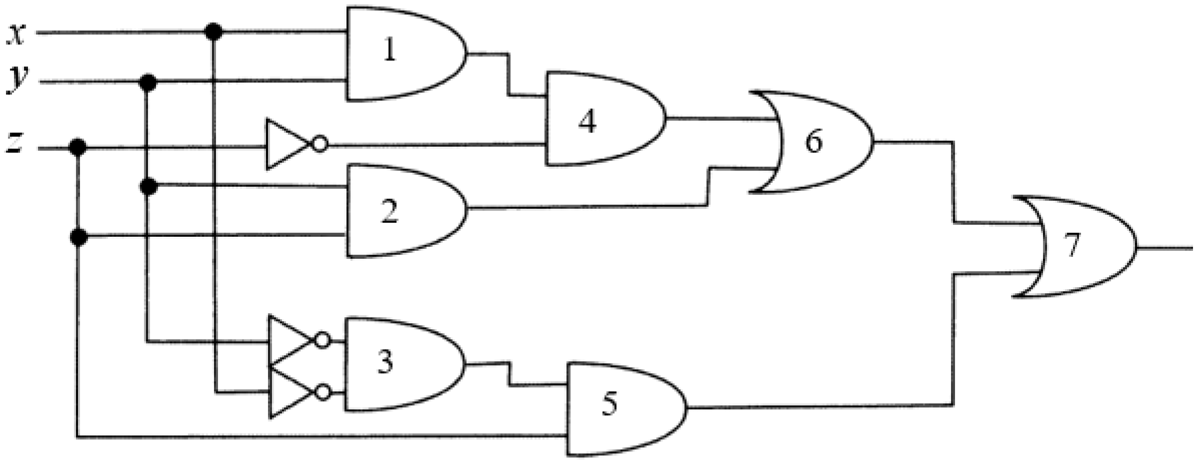


Рис. В.5.3

Таблиця В.5.1

Логічний елемент	Вхід	Вихід
1	x, y	xy
2	y, z	yz
3	$\bar{x}$, $\bar{y}$	$\bar{x}\,\bar{y}$
4	xy, $\bar{z}$	$xy\bar{z}$
5	$\bar{x}\,\bar{y}$, z	$\bar{x}\,\bar{y}z$
6	yz, $xy\bar{z}$	$yz\vee xy\bar{z}$
7	$yz\vee xy\bar{z}$, $\bar{x}\,\bar{y}z$	$yz\vee xy\bar{z}\vee\bar{x}\,\bar{y}z$

Запишемо мінімальну ДНФ

$$f\left(x,y,z\right)=xy\vee\bar{x}z\,.$$

Мінімальна ДНФ реалізується наступною логічною схемою (рис. В.5.4).

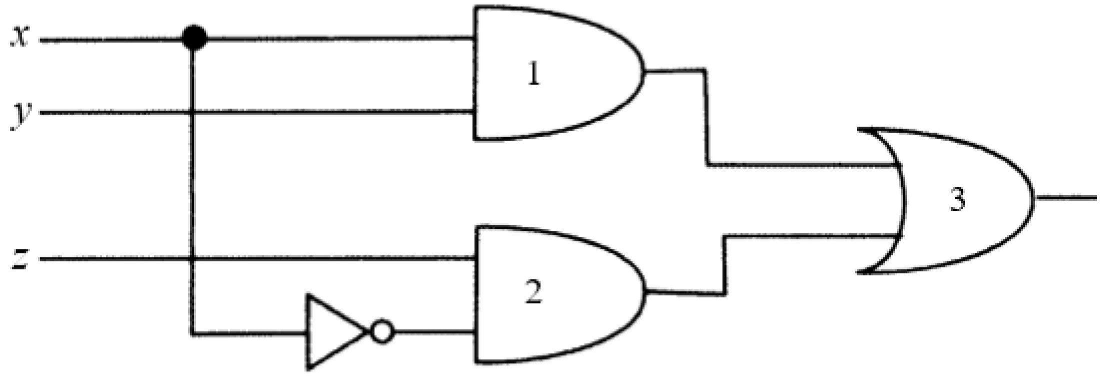


Рис. В.5.4

В цьому можна переконатись із таблиці В.5.2.

Таблиця В.5.2

Логічний елемент	Вхід	Вихід
1	x, y	xy
2	$\overline{x}$, z	$\overline{x}z$
3	xy, $\overline{x}z$	$xy \vee \overline{x}z$

5.9. Спочатку виразимо стрілку Пірса через штрих Шеффера:

$$x \downarrow y = \overline{x \vee y} = \overline{x}\,\overline{y} = \overline{\overline{(x \mid x)(y \mid y)}} = \overline{(x \mid x) \mid (y \mid y)},$$

$$x \downarrow y = \big((x \mid x) \mid (y \mid y)\big) \mid \big((x \mid x) \mid (y \mid y)\big).$$

Логічна схема зображена на рис. В.5.5.

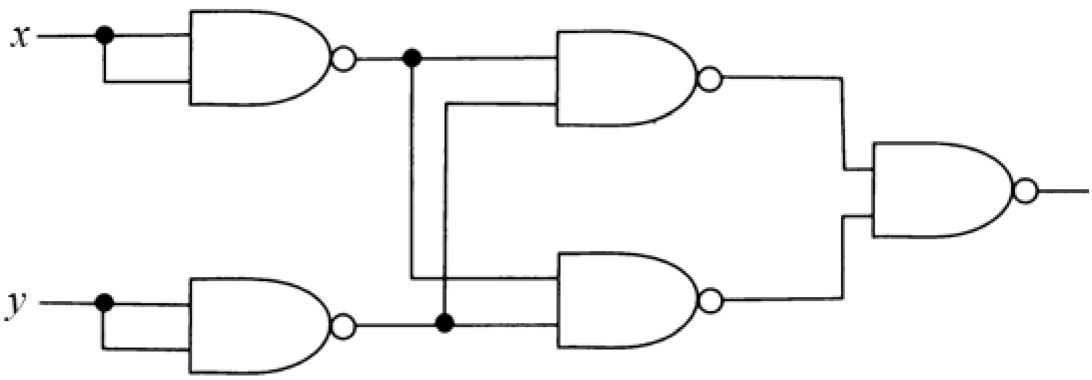


Рис. В.5.5

РОЗДІЛ 6. ГРАФИ

6.1. Граф буде ейлеровим, якщо n - непарне число більше одиниці. На головній діагоналі повного графа стоять одиниці, всі інші елементи дорівнюють одиниці. **6.2.** Якщо нижню ліву вершину позначимо через a, а решту проти годинникової стрілки - через b, c і d, то

$$\text{G: } \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix}, \quad \text{H: } \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix}, \quad \text{K: } \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix}.$$

6.3. Граф G зображено на рис. 6.51. Підграфами цього графа можуть бути графи E і K. **6.5.** 1,2,6,8,7,5,4,3,1. **6.6.** a,b,i,j,f,g,d,c,b,a. **6.7.** **а)** ACBDEA, довжина циклу 19. **b)** DEBCAD, довжина циклу 17. **6.8.** 1,2,3,4,2,5,6,4,1. **6.9.** Рис. 6.44: 1,2,3,4,2,6,4,5,6,1; Рис. 6.45: 1,3,4,8,7,5,4,7,2,8,6,2,1; Рис. 6.46: 2,1,4,5,2,3,4; Рис. 6.47: 3,2,1,6,2,4,3,5,4,6,5. **6.10.** Деревом є граф, що задається матрицею N . **6.11.** 5. **6.12. B)** Рис. 6.52. **6.13.** Рис. 6.53. **6.14.** Рис. 6.54.

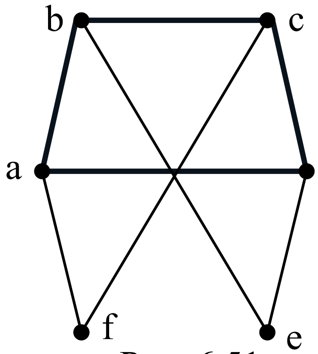


Рис. 6.51

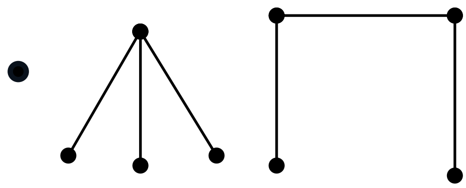


Рис. 6.52

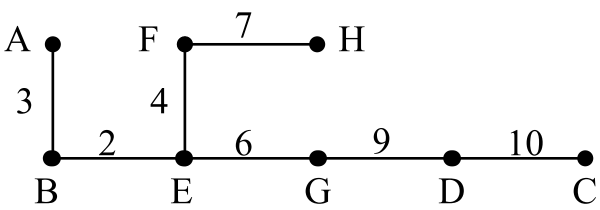


Рис. 6.53

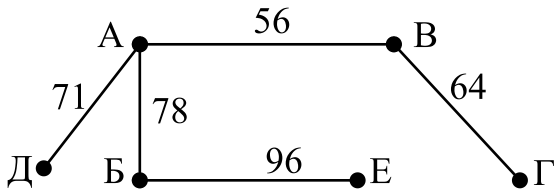


Рис. 6.54

6.15. Рис. 6.55.

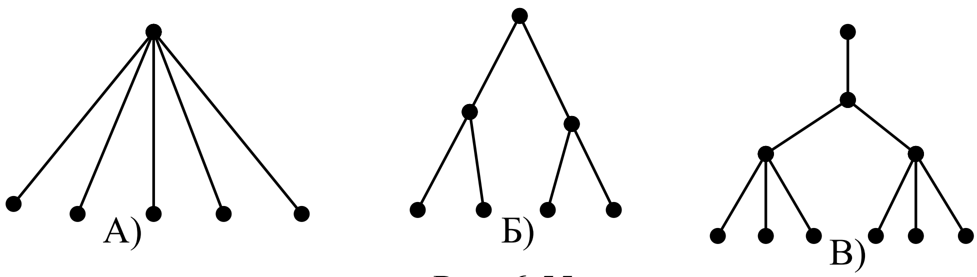


Рис. 6.55

6.16. Рис. 6.56. А) Найкоротший шлях від вершини 1 до вершини 2 є 1,4,6,2. Б) Є 2 найкоротших шляхів від вершини 3 до вершини 6: 3,5,6 та 3,2,6. Один із контурів довжини 5 – це контур 1,4,6,3,2,1.

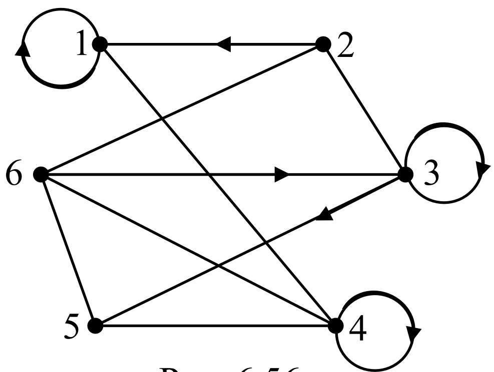


Рис. 6.56

6.17. А) Серед графів, що зображені на рис. 6.49 сильно зв'язним є лише граф H . **Б)** Оскільки граф гамільтонів, то запишемо гамільтонів цикл. Ребра цього циклу зорієнтуємо від попередньої вершини до суміжної наступної. **В)** Для того, щоб від довільного району міста можна було б потрапити до будь-якого іншого. **6.18.** Алгоритм топологічного сортування може видати наступну послідовність вершин: c_1, b_2, a_3, e_4, d_5, f_6. Нова матриця суміжності має наступний вигляд:

$$
\begin{array}{c}
\\ c \\ b \\ a \\ e \\ d \\ f
\end{array}
\begin{array}{c}
\begin{array}{cccccc} c & b & a & e & d & f \end{array} \\
\begin{pmatrix}
0 & 1 & 0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}
\end{array}
$$

Частина матриці, що розташована над головною діагоналлю дає повну інформацію про орграф. Для орграфа із задачі 6.16 не можна застосувати алгоритм топологічного сортування: антецеденти всіх вершин непорожні, орграф має контури. **6.19.** Алгоритм топологічного сортування може видати наступну послідовність вершин: $А_1$, $В_2$, $Б_3$, $Д_4$, $Е_5$, $Ж_6$, $З_7$, $Г_8$. **6.20.**

$$
M^2 = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix},\
M^3 = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix},\
M^4 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix},
$$

$$
M^* = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}.
$$

6.21. $M = W_0 = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix}$, $W_1 = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \end{pmatrix}$, $W_2 = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}$,

$W_3 = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}$, $W_4 = M^* = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}$.

6.22.

Таблиця 6.22. А)

Крок	Помічені вершини	Попередні вершини	Відстань до вершини					
			A	B	C	D	E	F
0	A		0	**3**	10	∞	∞	∞
1	B	A	0	3	**10**	15	∞	∞
2	C	A	0	3	10	**15**	∞	∞
3	D	B	0	3	10	15	**17**	23
4	E	D	0	3	10	15	17	**23**
5	F	D	0	3	10	15	17	23

Таблиця 6.22. Б)

Крок	Помічені вершини	Попередні вершини	Відстань до вершини					
			A	B	C	D	E	F
0	C		∞	∞	0	**5**	∞	∞
1	D	C	∞	∞	0	5	**7**	13
2	E	D	∞	**11**	0	5	7	13
3	B	E	∞	11	0	5	**7**	**13**
4	F	D	∞	11	0	5	7	13
5	A	C	∞	11	0	5	7	13

ЛІТЕРАТУРА

Базова

1. Бондаренко М.Ф., Білоус Н.В., Руткас А.Г. Комп’ютерна дискретна математика. Підручник. – Харків, Компанія СМІТ, 2004. – 480 с. https://bit.ly/3VRF1BC

2. Бардачов Ю.М., Соколова Н.А., Ходаков В.Є. Дискретна математика. Підручник. – К., 2022. – 287 с. https://bit.ly/45WqAAK

3. Капітонова Ю.В., Кривий С.Л., Летичевський О.А., Луцький Г.М., Печурін М.К. Основи дискретної математики. Підручник. – К., 2002. – 580 с. https://bit.ly/3Lcnejp

Допоміжна

1. Трохимчук Р. М., Нікітченко М. С. Дискретна математика у прикладах і задачах : навч. посібник /– Київ : Київський університет, 2017. **https://bit.ly/3zrX8X0**

2. Johnsonbaugh, R. (2001) Discrete Mathematics, 5th edn, New Jersey: Prentice Hall. https://bit.ly/4eTmKML

3. Gersting, J. L. (1999) Mathematical Structures for Computer Science, 4th edn, San Francisco: W. H. Freeman. https://bit.ly/4cSgXp8

4. Rosen, K. H. (1998) Discrete Mathematics and Its Applications, New York: M Graw-Hill. https://bit.ly/3WmRoXT

ПРЕДМЕТНИЙ ПОКАЗЧИК

ОНМУ

ДИСКРЕТНА МАТЕМАТИКА
Частина 2

Навчальний посібник

Одеса — 2026