

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ТАРАСА ШЕВЧЕНКА

Факультет радіофізики, електроніки та комп'ютерних систем

Кафедра комп'ютерної інженерії

**СТВОРЕННЯ ЗАСТОСУНКУ ДЛЯ РЕАЛІЗАЦІЇ ТЕОРЕТИКО-
МНОЖИННИХ ТА АЛГЕБРАЇЧНИХ ОПЕРАЦІЙ НАД ГРАФАМИ**

Кваліфікаційна робота бакалавра
студента 4 курсу

спеціальність: 123 «Комп'ютерна
інженерія»

Максима ХОТЯЧУКА

Науковий керівник:

доктор техн. наук, професор

Сергій ПОГОРІЛИЙ

Рецензент:

кандидат фіз.-мат. наук, асистент

Іван КОЛОМІЄЦЬ

До захисту допускаю

Протокол засідання кафедри від

«__»_____2022 р. №__

Завідувач кафедри

к.ф-м.н., доцент

Юрій БОЙКО

РЕФЕРАТ

Проведено аналіз наукових публікацій, теоретичних матеріалів з теми «Теорія графів. Способи задання графів. Операції над графами». Обрано 9 основних операцій над графами для подальшого вивчення та їх реалізацій (об'єднання, перетину, доповнення за відношенням, різниці, з'єднання, диз'юнктивної суми графів, добутку, суми та композиції графів.).

Для кожної операції розроблено застосунок мовою програмування C#, який генерує випадкову пару матриць та виконує вищенаведені операції над ними.

Підготовлено матеріали для створення методичних матеріалів «Посібник студента» та «Посібник викладача», які включають 60 варіантів завдань для студентів. Посібник викладача містить розв'язки та відповіді до всіх задач.

Кваліфікаційна робота бакалавра за обсягом складає 46 сторінок, містить 1 рисунок та використано 6 літературних джерел.

Ключові слова: граф, алгебраїчні властивості графів, теоретико-множинні властивості графів, операції над графами.

ЗМІСТ

Вступ.....	5
РОЗДІЛ 1. ЗАГАЛЬНІ ВІДОМОСТІ.....	6
1.1. Орієнтовані графи. Способи задання графів.....	6
1.1.1. Аналітичний метод задання.....	6
1.1.2. Геометричний метод задання.....	7
1.1.3. Матричний метод задання.....	7
1.2. Теоретико-множинні властивості графів.....	8
1.2.1. Перетин графів.....	8
1.2.2. Об'єднання графів.....	9
1.2.3. Доповнення до відображення.....	10
1.2.4. Різниця графів.....	11
1.2.5. З'єднання графів.....	12
1.2.6. Диз'юнктивна сума графів.....	13
1.3. Алгебраїчні властивості графів.....	14
1.3.1. Добуток графів.....	14
1.3.2. Сума графів.....	15
1.3.3. Композиція графів	17
РОЗДІЛ 2. ЕКСПЕРИМЕНТАЛЬНА ЧАСТИНА.....	19
2.1. Задача експериментальної частини.....	19
2.2. Робота застосунку.....	19
2.2.1. Робота застосунку для операцій перетину, об'єднання, доповнення за відношенням та різниці графів.....	19

2.2.2. Робота застосунку для операцій з'єднання та диз'юнктивної суми графів.....	23
2.2.3. Робота застосунку для операцій добутку, суми та композиції графів.....	25
Висновки.....	29
Літературні джерела.....	30
Додатки.....	31

Вступ

Науково-технічний прогрес у другій половині XX сторіччя визначається тим фактом, що відбувається домінування інформаційних технологій. Це призвело до того, що в галузі дискретної математики найбільш динамічного розвиваються розділи, на яких ґрунтуються ці технології. Зростають вимоги до інженерно-математичного рівня підготовки студентів спеціальності 123 – Комп'ютерна інженерія. Все це вимагає підготовки нових науково-методичних матеріалів з метою розширення освітніх профілів у галузі дискретної математики.

Важливим розділом цієї математичної дисципліни є теорія графів. Низка сучасних задач комп'ютерної інженерії (проектування топології доменів, мікропроцесорів, створення новітніх відеоадаптерів, композиція та декомпозиція скінчених автоматів і багато інших) передбачають вільне оперування студентами основоположними поняттями теорії графів та операцій із ними.

В умовах сучасних реалій, коли дистанційне навчання набуває все більшого попиту, об'єктивна перевірка знань студентів є майже нереальною в зв'язку з тим, що необхідно кожному студенту дати окремий варіант завдань та перевірити їх. Тож існує необхідність створення методичних посібників, а саме «Посібник студента» для студентів з варіантами завдань та «Посібник викладача» з розв'язками задач і відповідями до них.

Задачею роботи є поглиблене ознайомлення з теорією графів, властивостями та операціями над графами, розробка застосування для перевірки коректності вирішення завдань на тему «операції над графами», розробка двох посібників «Посібник студента» з варіантами завдань і «Посібник викладача» з відповідями до завдань.

РОЗДІЛ 1. ЗАГАЛЬНІ ВІДОМОСТІ

1.1. Орієнтовані графи. Способи задання графів

Вивчення теорії графів є важливим, оскільки вона допомагає вирішити ряд сучасних проблем в різних галузях життя: інформатиці та програмуванні (граф-схеми алгоритму, файлових системах комп'ютерів), комунікаційних і транспортних системах, логістиці, автоматизації технологічних процесів та виробництв тощо.

Поняття орієнтованого графа спирається на понятті множин і відображень. Для того, щоб задати деяку множину необхідно вказати закон, по якому відносно довільного елементу можна сказати, чи належить він даній множині. Для задання відображення на множині необхідно кожному елементу множини зіставити деяку підмножину даної множини.

Серед множини графів виділяють три типи графів: орієнтовані (графи Бержа), неорієнтовані та мішані граfi. Оскільки в даній роботі будуть розглянуті лише орієнтовані граfi, для зручності під словом «граф» буде матись на увазі поняття «орієнтований граф».

Існують три еквівалентних методи задання графів: аналітичний, геометричний і матричний.

1.1.1. Аналітичний метод задання

Якщо задаємо деяку множину елементів X та виділимо деяку підмножину $F = \{(x_i, x_j)\} \in X * X$, що задає бінарне відношення на X , то множина X і бінарне відношення F на цій множині визначатимуть деякий граф $G = (X, F)$. При заданні відношення F на X необхідно кожному елементу $x_i \in X$ зіставити певну підмножину X . Для акцентування того, що елементу x_i зіставляється саме підмножина, відповідна відношенню F , вона позначається через $Fx_i \in X$. Наприклад, маємо множину $X = \{x_1, x_2, x_3, x_4\}$, $Fx_1 = \{x_2\}$, $Fx_2 = \{x_2, x_3, x_4\}$, $Fx_3 = \{x_2, x_4\}$, $Fx_4 = \{x_2, x_1\}$. Тоді множини X та F задають граф G .

1.1.2. Геометричний метод задання

Множину елементів X графа G зображують кружками та називають множиною вершин. Кожну вершину $x_i \in X$ з'єднують лініями з тими вершинами $x_j \in X$, для яких виконується умова $x_j \in Fx_i$. Множина ліній, яка відповідає множині впорядкованих пар вершин (x_i, x_j) , де $x_i \in X$, а $x_j \in Fx_i$, називається множиною ребер графа. Якщо $x_i \neq x_j$, то ребро (x_i, x_j) зображується лінією зі стрілкою – дугою, що спрямована від x_i до x_j . Якщо $x_i = x_j$, то ребро (x_i, x_j) зображується лінією без стрілки, яка поєднує вершину x_i із собою, і називається петлею. На рис. 1 показано геометричне задання графа $G = (X, F)$ з попереднього прикладу.

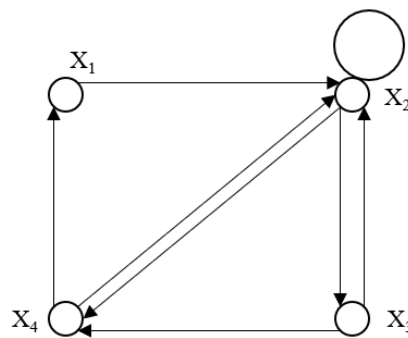


Рис 1. Геометричне задання графа

1.1.3. Матричний метод задання

Квадратна матриця $R = \|r_{ij}\| = \begin{pmatrix} r_{11} & r_{12} & \dots & r_{1n} \\ r_{21} & r_{22} & \dots & r_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ r_{n1} & r_{n2} & \dots & r_{nn} \end{pmatrix}$, елементи якої – 0 та 1

називається матрицею суміжності графа $G = (X, F)$ тоді й тільки тоді, коли її елемент утворюється за правилом: елемент r_{ij} , що стоїть на перерізі x_i -го рядка та x_j -го стовпця, дорівнює 1, якщо існує дуга, що йде з вершини x_i до вершини x_j , а $r_{ij} = 0$ – в протилежному випадку. Скорочено це записується в такий спосіб:

$$r_{ij} = \begin{cases} 1, \text{ якщо } x_j \in Fx_i \\ 0, \text{ якщо } x_j \notin Fx_i \end{cases}$$

Для прикладу з попередніх пунктів матриця суміжності матиме вигляд:

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

Найбільш зручними способами задання графів є аналітичний та матричний, оскільки в такому випадку з графами зручно працювати та виконувати операції як для графа в цілому, так і для кожної вершини графа окремо. Геометричний метод задання графа є більш наочним і слугує для ілюстрації та загального представлення отриманих результатів.

1.2. Теоретико-множинні властивості графів

В роботі буде розглянуто та реалізовано наступні теоретико-множинні операції над графами:

- перетин графів;
- об'єднання графів;
- доповнення до відображення;
- різниця графів;
- з'єднання графів;
- диз'юнктивна сума графів.

1.2.1. Перетин графів

Граф $K = (V, E)$ називається перетином графів G і H і позначається $K = G \cap H$, якщо

$$V = V_G \cap V_H,$$

$$E_K = E_G \cap E_H, \text{ де } u \in V.$$

Якщо $G \cap H = \emptyset$ (в тому випадку, коли $X \cap Y = \emptyset$), то кажуть, що G та H не перетинаються.

Приклад 1. Нехай $G = (X, F)$, $X = \{x_1, x_2, x_3\}$, $Fx_1 = \{x_2\}$, $Fx_2 = Fx_3 = \{x_1, x_2\}$ і $H = (Y, P)$, $Y = \{x_1, x_2\}$, $Px_1 = \{x_1\}$, $Px_2 = \{x_1, x_2\}$. Визначити граф, який дорівнює перетину графів G і H .

Знайдемо B і T графа K :

$$B = X \cap Y = \{x_1, x_2\},$$

$$Tx_1 = Fx_1 \cap Py_1 = \emptyset,$$

$$Tx_2 = Fx_2 \cap Py_2 = \{x_1, x_2\}.$$

Матриця суміжності для графа K має вигляд:

$$R_K = R_G \cap R_H = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \end{pmatrix} \cap \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ 1 & 1 \end{pmatrix}.$$

1.2.2. Об'єднання графів

Граф $Q = (A, S)$ називається об'єднанням графів G та H і позначається $Q = G \cup H$, якщо

$$A = X \cup Y,$$

$$Sx = Fx \cup Px, \text{ де } x \in A.$$

Причому, якщо $x \notin X$, то $Fx = \emptyset$, або якщо $x \notin Y$, то $Px = \emptyset$.

Приклад 2. Нехай $G = (X, F)$ і $H = (Y, P)$ – два графа з прикладу 1. Тоді результатом об'єднання графів G та H буде граф $Q = (A, S)$, який дорівнює:

$$A = X \cup Y = \{x_1, x_2, x_3\},$$

$$Sx_1 = Fx_1 \cup Px_1 = \{x_1, x_2\},$$

$$Sx_2 = Fx_2 \cup Px_2 = \{x_1, x_2\},$$

$$Sx_3 = Fx_3 = \{x_1, x_2\}.$$

Матриця суміжності графа Q буде рівна:

$$R_Q = R_G \cup R_H = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \end{pmatrix} \cup \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \end{pmatrix}.$$

1.2.3. Доповнення до відображення

Нехай $G = (X, F)$ – довільний підграф графа L . Доповненням по відображенню графа G до насиченого графа G_x називають граф $\bar{G} = (X, \bar{F})$, в якого $\bar{F}x = X \setminus Fx$, де $x \in X$. Якщо G_y – інший насичений граф такий, що $X \subseteq Y$, то доповненням по відображенню графа H до графа G_y називають граф $\bar{H} = (Y, \bar{F})$, де $\bar{F}y = Y \setminus Fy$ і $y \in Y$, причому для $y \notin X$ $Fy = \emptyset$. Аналогічно визначається доповнення по відображенню графа H до графа L .

Приклад 3. Нехай $G = (X, F)$ і $H = (Y, P)$ – два графа з прикладу 1. В якості універсального графа оберемо насичений граф третього порядку G_x , де $X = \{x_1, x_2, x_3\}$. Знайдемо доповнення до відображення графів G і H до графа G_x .

Знаходимо графи $\bar{G} = (X, \bar{F})$ і $\bar{H} = (Y, \bar{P})$:

$$X = Y = \{x_1, x_2, x_3\};$$

$$\bar{F}x_1 = \{x_1, x_3\}, \quad \bar{P}x_1 = \{x_2, x_3\},$$

$$\bar{F}x_2 = \{x_3\}, \quad \bar{P}x_2 = \{x_3\},$$

$$\bar{F}x_3 = \{x_3\}, \quad \bar{P}x_3 = \{x_1, x_2, x_3\}.$$

Матриці суміжності графів $\bar{G}$ та $\bar{H}$ наведені нижче:

$$R_{\bar{G}} = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix}; \quad R_{\bar{H}} = \begin{pmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix}.$$

1.2.4. Різниця графів

Граф, який позначається як $Q = G \setminus H$, називається різницею графів G і H , якщо він визначається формулами:

$$Q = G \cap \bar{H},$$

де $\bar{H}$ – доповнення по відображенню графа G до графа L . Важливо відзначити, що операція різниці графів не комутативна.

Приклад 4. Нехай $G = (X, F)$ і $H = (Y, P)$ – два графа з прикладу 1. В якості універсального графа в даному випадку обираємо насичений граф третього порядку O_x , де $X = \{x_1, x_2, x_3\}$. Знайти доповнення по відображенню графів G і H до графа G_x і різниці графів $G \setminus H$.

Відповідно до визначень, знаходимо доповнення по відображенню графа H :

$$\bar{H} = (Y, \bar{P}), \text{ де } \bar{P}x_1 = \{x_2, x_3\}, \bar{P}x_2 = \{x_3\}, \bar{P}x_3 = \{x_1, x_2, x_3\}.$$

Знаходимо граф $Q = (A, S)$, де:

$$A = \{x_1, x_2, x_3\},$$

$$Sx_1 = Fx_1 \cap \bar{P}x_1 = \{x_2\},$$

$$Sx_2 = Fx_2 \cap \bar{P}x_2 = \emptyset,$$

$$Sx_3 = Fx_3 \cap \bar{P}x_3 = \{x_1, x_2\}.$$

Розглянемо розв'язок операції різниці графів за допомогою матриць суміжності. Запишемо їх для графів G та $\bar{H}$:

$$R_G = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \end{pmatrix}; \quad R_{\bar{H}} = \begin{pmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix}.$$

. Матриця суміжності графа Q рівна:

$$R_Q = R_G \cap R_{\bar{H}} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \end{pmatrix} \cap \begin{pmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 0 \end{pmatrix}.$$

1.2.5. З'єднання графів

Нехай $G = (X, F)$ і $H = (Y, P)$ – довільні графи. Граф $M = (C, U)$ називається з'єднанням графів G і H та позначається $M = G \boxplus H$, якщо:

$$C = (\{1\} \times X) \cup (\{2\} \times Y),$$

$$U(i, z) = ((\{1\} \times Fz) \cup (\{2\} \times Y)) \cup ((\{1\} \times X) \cup (\{2\} \times Pz)),$$

де $(i, z) \in C$, $i \in I = \{1, 2\}$, причому, якщо $z \notin X$, то $Fz = \emptyset$ і $((\{1\} \times Fz) \cup (\{2\} \times Y)) = \emptyset$. Аналогічно, якщо $z \notin Y$, то $Pz = \emptyset$ і $((\{1\} \times X) \cup (\{2\} \times Pz)) = \emptyset$.

Приклад 5. Нехай $G = (X, F)$ і $H = (Y, P)$ – графи з прикладу 1. Знайти граф, рівний з'єднанню графів G і H .

З визначення знаходимо множину C і відображення U графа M :

$$C = \{(1, x_1), (1, x_2), (1, x_3), (2, x_1), (2, x_2)\},$$

$$U(1, x_1) = (\{1\} \times Fx_1) \cup (\{2\} \times Y) = \{(1, x_2), (2, x_1), (2, x_2)\},$$

$$U(1, x_2) = (\{1\} \times Fx_2) \cup (\{2\} \times Y) = \{(1, x_1), (1, x_2), (2, x_1), (2, x_2)\},$$

$$U(1, x_3) = (\{1\} \times Fx_3) \cup (\{2\} \times Y) = \{(1, x_1), (1, x_2), (2, x_1), (2, x_2)\},$$

$$U(2, x_1) = (\{1\} \times X) \cup (\{2\} \times Px_1) = \{(1, x_1), (1, x_2), (1, x_3), (2, x_1)\},$$

$$U(2, x_2) = (\{1\} \times X) \cup (\{2\} \times Px_2) = \{(1, x_1), (1, x_2), (1, x_3), (2, x_1), (2, x_2)\}.$$

Матриця суміжності графа C рівна:

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

1.2.6. Диз'юнктивна сума графів

Диз'юнктивною сумою графів $G = (X, F)$ і $H = (Y, P)$ називається граф, який позначається $Q = G \oplus H$ та утворюється по будь-якій з наступних еквівалентних формул:

$$G \oplus H = (G \cup H) \setminus (G \cap H) = (G \cup H) \cap (\overline{G \cap H}) = (G \cap \bar{H}) \cup (\bar{G} \cap H).$$

Приклад 6. Нехай $G = (X, F)$ і $H = (Y, P)$ – графи з прикладу 1. Знайти граф $Q = (Z, S)$ – диз'юнктивну суму графів G і H .

З визначення знаходимо множину Z і відображення S графа Q :

$$Z = \{x_1, x_2, x_3\},$$

$$Sx_1 = \{x_1, x_2\},$$

$$Sx_2 = \emptyset,$$

$$Sx_3 = \{x_1, x_2\}.$$

Розв'яжемо даний приклад за допомогою матриць суміжності:

$$R_G = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \end{pmatrix}; R_{\bar{G}} = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix}; R_H = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}; R_{\bar{H}} = \begin{pmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix}.$$

Матриця суміжності для графа Q має наступний вигляд:

$$\begin{aligned} R_Q &= (R_G \cap R_{\bar{H}}) \cup (R_{\bar{G}} \cap R_H) = \\ &= \left(\begin{pmatrix} 0 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \end{pmatrix} \cap \begin{pmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix} \right) \cup \left(\begin{pmatrix} 1 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix} \cap \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} \right) = \\ &= \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 0 \end{pmatrix} \cup \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 0 \end{pmatrix}. \end{aligned}$$

1.3. Алгебраїчні властивості графів

В даній роботі буде розглянуто та реалізовано наступні алгебраїчні операції над графами:

- добуток графів;
- сума графів;
- композиція графів;

1.3.1. Добуток графів

Нехай $G = (X, F)$ і $H = (Y, P)$ – два довільних орієнтованих графа. Граф $Q = (Z, U)$ називається добутком графів G і H і позначається $Q = G \times H$, якщо множина Z і відображення U множини Z в себе визначаються наступним чином:

$$Z = X \times Y,$$

$$Uz = Fx \times Py,$$

де $z \in Z$, $x \in X$, $y \in Y$ і $z = (x, y)$. Результат добутку графів G і H утворюється як декартовий добуток множин X і Y , а відображення довільного елемента z цього добутку є декартовим добутком підмножин X і Y , які відповідають відображенню F елемента $pr_1(z) = x$ і відображенню P елемента $pr_2(z) = y$.

Приклад 7. Нехай дані графи $G = (X, F)$ і $H = (Y, P)$, де $X = \{x_1, x_2\}$, $Fx_1 = \{x_2\}$, $Fx_2 = \{x_1, x_2\}$, $Y = \{y_1, y_2, y_3\}$, $Py_1 = \{y_1, y_3\}$, $Py_2 = \{y_2\}$, $Py_3 = \{y_1, y_2\}$. Знайдемо граф, який буде рівний добутку графів G і H .

Спочатку знаходимо множину елементів Z : $Z = \{(x_1, y_1), (x_1, y_2), (x_1, y_3), (x_2, y_1), (x_2, y_2), (x_2, y_3)\}$, де $z_1 = (x_1, y_1)$, $z_2 = (x_1, y_2)$, $z_3 = (x_1, y_3)$, $z_4 = (x_2, y_1)$, $z_5 = (x_2, y_2)$, $z_6 = (x_2, y_3)$.

Таким чином:

$$Z = \{z_1, z_2, z_3, z_4, z_5, z_6\},$$

$$U_{z_1} = Fx_1 \times Py_1 = \{z_4, z_6\},$$

$$U_{z_2} = Fx_1 \times Py_2 = \{z_5\},$$

$$U_{z_3} = Fx_1 \times Py_3 = \{z_4, z_5\},$$

$$U_{z_4} = Fx_2 \times Py_1 = \{z_1, z_3, z_4, z_6\},$$

$$U_{z_5} = Fx_2 \times Py_2 = \{z_2, z_5\},$$

$$U_{z_6} = Fx_2 \times Py_3 = \{z_1, z_2, z_4, z_5\}.$$

Розглянемо розв'язок прикладу за допомогою матриць суміжності. Для тих самих графів запишемо матриці суміжності:

$$R_G = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}; \quad R_H = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \end{pmatrix}.$$

Знайдемо матрицю суміжності графа Z_1 , який рівний добутку графів G і H :

$$R_{Z_1} = R_G \times R_H = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix} \times \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 \end{pmatrix}.$$

1.3.2. Сума графів

Граф $N = (Z, R)$ називається сумою графів G і H і позначається як $N = G + H$, якщо Z і R визначаються виразами:

$$Z = X \times Y,$$

$$Rz = (Fx \times \{y\}) \cup (\{x\} \times Py),$$

де $z \in Z$, $x \in X$, $y \in Y$ і $z = (x, y)$.

Приклад 8. Нехай дано графи G і H з прикладу 6. Знайдемо граф, який дорівнює сумі графів G і H .

Визначимо множину Z :

$$Z = \{z_1, z_2, z_3, z_4, z_5, z_6\}.$$

Визначимо відображення R множини Z в себе:

$$Rz_1 = (Fx_1 \times \{y_1\}) \cup (\{x_1\} \times Py_1) = \{z_1, z_3, z_4\},$$

$$Rz_2 = (Fx_1 \times \{y_2\}) \cup (\{x_1\} \times Py_2) = \{z_2, z_5\},$$

$$Rz_3 = (Fx_1 \times \{y_3\}) \cup (\{x_1\} \times Py_3) = \{z_1, z_2, z_6\},$$

$$Rz_4 = (Fx_2 \times \{y_1\}) \cup (\{x_2\} \times Py_1) = \{z_1, z_4, z_6\},$$

$$Rz_5 = (Fx_2 \times \{y_2\}) \cup (\{x_2\} \times Py_2) = \{z_2, z_5\},$$

$$Rz_6 = (Fx_2 \times \{y_3\}) \cup (\{x_2\} \times Py_3) = \{z_3, z_4, z_5, z_6\}.$$

Розглянемо розв'язок операції суми графів за допомогою матриць суміжності.

Для тих самих графів запишемо матриці суміжності:

$$R_G = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}; \quad R_H = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \end{pmatrix}.$$

Знайдемо матрицю суміжності графа Z_2 , який рівний сумі графів G і H

(матриці E_G та E_H – одиничні матриці порядку R_G і R_H відповідно):

$$\begin{aligned} R_{Z_2} &= (R_G \times E_H) \cup (E_G \times R_H) = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix} \times \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \cup \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \times \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \end{pmatrix} = \\ &= \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix} \cup \begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \end{pmatrix} = \end{aligned}$$

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

1.3.3. Композиція графів

Граф $K = (Z, T)$ називається композицією графів G і H і позначається як $K = G \otimes H$, якщо Z і T визначаються як:

$$Z = X \times Y,$$

$$Tz = (Fx \times Y) \cup (X \times Py),$$

де $z \in Z, x \in X, y \in Y$ і $z = (x, y)$.

Приклад 9. Знайти композицію графів G і H з попередніх прикладів.

Визначимо множину Z :

$$Z = \{z_1, z_2, z_3, z_4, z_5, z_6\}.$$

Визначимо відображення T множини Z в себе:

$$Tz_1 = \{z_1, z_3, z_4, z_5, z_6\},$$

$$Tz_2 = \{z_2, z_4, z_5, z_6\},$$

$$Tz_3 = \{z_1, z_2, z_4, z_5, z_6\},$$

$$Tz_4 = \{z_1, z_3, z_4, z_5, z_6\},$$

$$Tz_5 = Tz_6 = \{z_1, z_2, z_3, z_4, z_5, z_6\}.$$

Розглянемо розв'язок операції композиції графів за допомогою матриць суміжності. Для тих самих графів запишемо матриці суміжності:

$$R_G = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}; \quad R_H = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \end{pmatrix}.$$

Знайдемо матрицю суміжності графа Z_3 , який рівний композиції графів G і H (матриці P_G та P_H – матриці порядку R_G і R_H відповідно, які заповнені одиницями):

$$\begin{aligned}
 R_{Z_3} &= (R_G \times P_H) \cup (P_G \times R_H) = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix} \times \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \cup \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \times \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \end{pmatrix} = \\
 &= \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \cup \begin{pmatrix} 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 \end{pmatrix} = \\
 &= \begin{pmatrix} 1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}.
 \end{aligned}$$

РОЗДІЛ 2. ЕКСПЕРИМЕНТАЛЬНА ЧАСТИНА

2.1. Задача експериментальної частини

Основна задача практичної частини роботи полягає в розробці застосунку для реалізації теоретико-множинних та алгебраїчних операцій над графами мовою програмування C# з використанням .NET Framework 4.7.2 в середовищі Visual Studio 2022. Мова програмування C# має усі необхідні функції і методи для роботи з масивами, якими задаються графи, що дає змогу реалізувати теоретико-множинні та алгебраїчні операції.

Необхідно розробити функції для реалізації наступних операцій над графами: об'єднання, перетину, доповнення за відношенням та різниці графів для графів розмірності 8 і 10; з'єднання та диз'юнктивної суми графів для графів розмірності 6 і 8; добутку, суми та композиції графів для графів розмірності 3 і 4 та посібники студента і викладача. Приклад «Посібника викладача» та «Посібника студента» наведено в додатку 1 та додатку 2 відповідно, а реалізація кожної з операцій наведена в додатку 3.

2.2. Робота застосунку

2.2.1. Робота застосунку для операцій перетину, об'єднання, доповнення за відношенням та різниці графів

Для операцій об'єднання, перетину, доповнення за відношенням та різниці графів генеруються графи розмірністю 8 і 10 ($X = (Q, F)$, $Y = (E, T)$):

$$Q = \{q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8\};$$

$$E = \{e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8, e_9, e_{10}\};$$

$$R_X = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 \end{pmatrix};$$

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

Результатом перетину даних графів буде граф $G_2 = (B, N)$, який рівний:

$$B = \{b_1, b_2, b_3, b_4, b_5, b_6, b_7, b_8\};$$

$$R_{G_2} = R_X \cap R_Y = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 \end{pmatrix} \cap$$

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

Результатом об'єднання даних графів буде граф $G_1 = (A, S)$, який дорівнює:

$$A = \{a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}\};$$

$$R_{G_1} = R_X \cup R_Y = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 \end{pmatrix} \cup$$

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

Доповненням до відображення даних графів будуть графи $\bar{X} = (Q, \bar{F})$ та $\bar{Y} = (E, \bar{T})$:

$$R_{\bar{X}} = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix};$$

$$R_{\bar{Y}} = \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 \end{pmatrix}.$$

Результатом різниці даних графів будуть графи $G_3 = (U, I)$, який дорівнює $X - Y$, та $H = (O, P)$, який дорівнює $Y - X$:

$$U = \{u_1, u_2, u_3, u_4, u_5, u_6, u_7, u_8\};$$

$$R_{G_3} = R_X \cap R_{\bar{Y}} = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 \end{pmatrix} \cap$$

$$\cap \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix};$$

$$O = \{o_1, o_2, o_3, o_4, o_5, o_6, o_7, o_8, o_9, o_{10}\};$$

$$R_H = R_Y \cap R_{\bar{X}} = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \cap$$

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

2.2.2. Робота застосунку для операцій з'єднання та диз'юнктивної суми графів

Для операції з'єднання та диз'юнктивної суми графів генеруються два графи розмірністю 6 і 8 ($X = (I, N)$, $Y = (M, V)$).

$$I = \{i_1, i_2, i_3, i_4, i_5, i_6\};$$

$$M = \{m_1, m_2, m_3, m_4, m_5, m_6, m_7, m_8\};$$

$$R_X = \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 0 \end{pmatrix}; R_Y = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}.$$

Результатом з'єднання даних графів буде граф $G_7 = (P, T)$, який дорівнює:

$$P = \{p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8, p_9, p_{10}, p_{11}, p_{12}, p_{13}, p_{14}\};$$

$$R_{G_7} = \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}.$$

Результатом диз'юнктивної суми графів даних графів буде граф $G_8 = (S, L)$, який дорівнює:

$$S = \{s_1, s_2, s_3, s_4, s_5, s_6\};$$

$$R_{G_8} = (R_X \cap R_{\bar{Y}}) \cup (R_{\bar{X}} \cup R_Y) = \left(\begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 0 \end{pmatrix} \cap \begin{pmatrix} 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \end{pmatrix} \right) \cup \left(\begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \cap \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \end{pmatrix} \right) = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \end{pmatrix} \cup \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}$$

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

2.2.3. Робота застосунку для операцій добутку, суми та композиції графів

Робота застосунку: для добутку, суми та композиції застосунок генерує випадкові графи розмірністю 3 і 4 ($X = (Q, F)$, $Y = (E, T)$):

$$Q = \{q_1, q_2, q_3\};$$

$$E = \{e_1, e_2, e_3, e_4\};$$

$$R_X = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{pmatrix}; \quad R_Y = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}.$$

Результатом добутку даних графів буде граф $G_5 = (K, L)$, який дорівнює:

$$K = \{k_1, k_2, k_3, k_4, k_5, k_6, k_7, k_8, k_9, k_{10}, k_{11}, k_{12}\};$$

$$R_{G_5} = R_X \times R_Y = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{pmatrix} \times \begin{pmatrix} 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} =$$

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

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

Результатом композиції даних графів буде граф $G_6 = (C, V)$, який дорівнює:

$$C = \{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8, c_9, c_{10}, c_{11}, c_{12}\};$$

$$R_{G_6} = (R_X \times P_Y) \cup (P_X \times R_Y) = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{pmatrix} \times \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix} \cup \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \times$$

$$\times \begin{pmatrix} 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \cup$$

Висновки

В роботі проаналізовано виконання теоретико-множинних та алгебраїчних операцій над орієнтованими графами (а саме: перетин, об'єднання, доповнення до відображення, різницю, з'єднання, диз'юнктивну суму, добуток, суму та композицію) та продемонстровано способи розв'язку. Обґрунтовано доцільність використання структур даних подання графа у вигляді матриці суміжності в процесі проектування застосунку.

Спроековано алгоритми та мовою C# розроблено застосунок, який реалізує перераховані вище операції над графами і налічує 671 рядків коду. Роздруківка застосунку наведена у Додатку 1.

Підготовлено матеріали для створення методичних матеріалів «Посібник студента» та «Посібник викладача», які включають 60 варіантів завдань для студентів. Посібник викладача містить розв'язки та відповіді до всіх задач.

Літературні джерела

1. Погорілий С.Д. Програмне конструювання. Підручник серії Автоматизація наукових досліджень за редакцією академіка АПН України Третяка О.В. ВПЦ Київський університет., 2-е видання, 2007.
2. А. Е. Кононюк. Парадигма развития науки. Истины и информация. Фундаментальная теория представления истин и информации. Отображение информации языками графов. Книга 5, 2016.
3. Пронькин Ю.С., Егоров Ю.А. Элементы теории графов и их технические приложения. Учебно-методическое пособие для студентов технических специальностей, 2007.
4. Р. Беллман. Введение в теорию матриц. Изд-во «Наука». Главная редакция физ.-мат. литературы, 1969.
5. Г. Корн, Т. Корн. Справочник по математике для научных работников и инженеров. Главная редакция физ.-мат. литературы, 1968.
6. Э. Троелсен. С# и платформа .NET. «Питер», 2003.

Додатки

Додаток 1. Приклад варіанту «Посібника викладача»

Варіант 1

Теоретико множинні операції.

Перетин, об'єднання, доповнення до відображення, різниця графів для графів:

$$X = \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix};$$

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

$$\text{Перетин : } G_1 = \begin{pmatrix} 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix};$$

Об'єднання: $G_2 =$

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

Доповнення до відображення X : $\bar{X} =$

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

Доповнення до відображення Y : $\bar{Y} =$

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

Різниця $X - Y$: $G_3 =$

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

Різниця $Y - X$: $G_4 =$

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

З'єднання, диз'юнктивна сума для графів:

$$X = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}; Y = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \end{pmatrix}.$$

З'єднання: $G_5 =$

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

Диз'юнктивна сума: $G_6 =$

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

Алгебраїчні операції

Добуток, сума, композиція для графів:

$$X = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}; \quad Y = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 \end{pmatrix}.$$

$$\text{Добуток: } G_7 = \begin{pmatrix} 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

$$\text{Сума: } G_8 = \begin{pmatrix} 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

Композиція: $G_9 =$

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

Додадок 2. Приклад варіанту «Посібника студента»

Варіант 1

Теоретико множинні операції.

Знайти перетин, об'єднання, доповнення до відображення, різницю графів для графів:

$$X = \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix};$$

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

Знайти з'єднання, диз'юнктивну суму для графів:

$$X = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}; Y = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \end{pmatrix}.$$

Алгебраїчні операції

Знайти добуток, суму, композицію для графів:

$$X = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}; Y = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 \end{pmatrix}.$$

Додаток 3. Роздруківка програми

Реалізація операції перетину графів

```
static int[,] Peretin(int[] X, int[] Y, int[,] Fx, int[,] Py)
{
    int[,] Z = new int[X.Length, X.Length]; //створюємо масив результату операції
перетину
    for (int i = 0; i < Z.GetLength(0); i++)
    {
        for (int j = 0; j < Z.GetLength(1); j++)
        {
            Z[i, j] = 0; //заповнюємо масив нулями
        }
    }
    for (int i = 0; i < Z.GetLength(0); i++)
    {
        for (int j = 0; j < Z.GetLength(1); j++)
        {
            if ((Fx[i, j] == 1) && (Py[i, j] == 1)) //виконуємо перевірку відповідних
елементів графів на рівність 1
            {
                Z[i, j] = 1; //заповнюємо відповідний елемент одиницею
            }
        }
    }
    return Z;
}
```

Реалізація операції об'єднання графів

```
static int[,] Obednannya(int[] X, int[] Y, int[,] Fx, int[,] Py)
{
    int[,] Z = new int[Y.Length, Y.Length]; //створюємо масив результату операції
об'єднання
    for (int i = 0; i < Z.GetLength(0); i++)
    {
        for (int j = 0; j < Z.GetLength(1); j++)
        {
            Z[i, j] = 0; //заповнюємо масив нулями
        }
    }
    for (int i = 0; i < Y.Length; i++)
    {
        for (int j = 0; j < Y.Length; j++)
        {
            if ((i < X.Length) && (j < X.Length))
            {
                if ((Fx[i, j] == 1) || (Py[i, j] == 1)) //виконуємо перевірку
відповідних елементів графів на рівність 1
                {
                    Z[i, j] = 1; //заповнюємо відповідний елемент одиницею
                }
            }
            else
            {
                if (Py[i, j] == 1) //виконуємо перевірку відповідних елементів графа Y
на рівність 1
                {
                    Z[i, j] = 1; //заповнюємо відповідний елемент одиницею
                }
            }
        }
    }
}
```

```

    return Z;
}

```

Реалізація операції доповнення до відображення графа X

```

static int[,] DopovnennyaX(int[] X, int[] Y, int[,] Fx, int[,] Py)
{
    int[,] Z = new int[Y.Length, Y.Length]; //створюємо масив результату операції
    доповнення до відображення графа X
    for (int i = 0; i < Z.GetLength(0); i++)
    {
        for (int j = 0; j < Z.GetLength(1); j++)
        {
            Z[i, j] = 0; //заповнюємо масив нулями
        }
    }
    for (int i = 0; i < Z.GetLength(0); i++)
    {
        for (int j = 0; j < Z.GetLength(0); j++)
        {
            if ((i > Fx.GetLength(0) - 1) || (j > Fx.GetLength(1) - 1))
            {
                Z[i, j] = 1; //заповнюємо одиницями при i та j > довжини масиву X
            }
            else if (Fx[i, j] == 0)
            {
                Z[i, j] = 1; //обертаємо значення графа на протилежні
            }
            else
            {
                Z[i, j] = 0;
            }
        }
    }
    return Z;
}

```

Реалізація операції доповнення до відображення графа Y

```

static int[,] DopovnennyaY(int[] X, int[] Y, int[,] Fx, int[,] Py)
{
    int[,] Z = new int[Y.Length, Y.Length]; //створюємо масив результату операції
    доповнення до відображення графа Y
    for (int i = 0; i < Z.GetLength(0); i++)
    {
        for (int j = 0; j < Z.GetLength(1); j++)
        {
            Z[i, j] = 0; //заповнюємо масив нулями
        }
    }
    for (int i = 0; i < Z.GetLength(0); i++)
    {
        for (int j = 0; j < Z.GetLength(0); j++)
        {
            if (Py[i, j] == 0)
            {
                Z[i, j] = 1; //обертаємо значення графа на протилежні
            }
            else
            {
                Z[i, j] = 0;
            }
        }
    }
}

```

```

    }
}
return Z;
}

```

Реалізація операції різниці графів X та Y

```

static int[,] RiznitsyaXY(int[] X, int[] Y, int[,] Fx, int[,] Py)
{
    int[,] Z = new int[X.Length, X.Length]; //створюємо масив результату операції
    різниці X - Y
    for (int i = 0; i < Z.GetLength(0); i++)
    {
        for (int j = 0; j < Z.GetLength(1); j++)
        {
            Z[i, j] = 0; //заповнюємо масив нулями
        }
    }
    int[,] nPy = new int[Y.Length, Y.Length]; //створимо граф nY, який дорівнює
    доповненню до відображення графа Y
    nPy = DopovnnnyaY(X, Y, Fx, Py); //виконуємо операцію доповнення до відображення
    Z = Peretin2(X, Y, Fx, nPy); //виконуємо перетин графів X та nY
    return Z;
}

```

Реалізація операції різниці графів Y та X

```

static int[,] RiznitsyaYX(int[] X, int[] Y, int[,] Fx, int[,] Py)
{
    int[,] Z = new int[Y.Length, Y.Length]; //створюємо масив результату операції
    різниці Y - X
    for (int i = 0; i < Z.GetLength(0); i++)
    {
        for (int j = 0; j < Z.GetLength(1); j++)
        {
            Z[i, j] = 0; //заповнюємо масив нулями
        }
    }
    int[,] nFx = new int[Y.Length, Y.Length]; //створимо граф nX, який дорівнює
    доповненню до відображення графа X
    nFx = DopovnnnyaX(X, Y, Fx, Py); //виконуємо операцію доповнення до відображення
    Z = Peretin2(X, Y, nFx, Py); //виконуємо перетин графів nX та Y
    return Z;
}

```

Реалізація операції з'єднання графів

```

static int[,] Ziednnnya(int[] X, int[] Y, int[,] Fx, int[,] Py)
{
    int[,] Z = new int[X.Length + Y.Length, X.Length + Y.Length]; //створюємо масив
    результату операції з'єднання
    for (int i = 0; i < Z.GetLength(0); i++)
    {
        for (int j = 0; j < Z.GetLength(1); j++)
        {
            Z[i, j] = 0; //заповнюємо масив нулями
        }
    }
    int numberOfZ = 0;
    for (int i = 0; i < Fx.GetLength(0); i++)
    {
        for (int j = 0; j < Fx.GetLength(1); j++)

```

```

    {
        if (Fx[i, j] == 1)
        {
            Z[numberOfZ, j] = 1; //виконуємо операцію {1} * Fx
        }
    }
    numberOfZ++;
}
numberOfZ = 0;
do
{
    for (int i = 0; i < Y.Length; i++)
    {
        Z[numberOfZ, i + X.Length] = 1; //виконуємо операцію {2} * Y
    }
    numberOfZ++;
}
while (numberOfZ < X.Length);
numberOfZ = X.Length;
do
{
    for (int i = 0; i < X.Length; i++)
    {
        Z[numberOfZ, i] = 1; //виконуємо операцію {1} * X
    }
    numberOfZ++;
}
while (numberOfZ < X.Length + Y.Length);
numberOfZ = X.Length;
for (int i = 0; i < Py.GetLength(0); i++)
{
    for (int j = 0; j < Py.GetLength(1); j++)
    {
        if (Py[i, j] == 1)
        {
            Z[numberOfZ, j + X.Length] = 1; //виконуємо операцію {2} * Py
        }
    }
    numberOfZ++;
}
return Z;
}

```

Реалізація операції диз'юнктивної суми графів

```

static int[,] DizSum(int[] X, int[] Y, int[,] Fx, int[,] Py)
{
    int[,] Z = new int[X.Length, X.Length]; //створюємо масив для результату операції
    диз'юнктивної суми
    int[,] Z1 = new int[Y.Length, Y.Length]; //створюємо масив для результату операції
    об'єднання X та Y
    int[,] Z2 = new int[X.Length, X.Length]; //створюємо масив для результату операції
    перетину X та Y
    int[,] Z3 = new int[X.Length, X.Length]; //створюємо масив для результату операції
    доповнення до відображення Z2
    //виконуємо об'єднання X та Y
    for (int i = 0; i < Y.Length; i++)
    {
        for (int j = 0; j < Y.Length; j++)
        {
            if ((i < X.Length) && (j < X.Length))
            {
                if ((Fx[i, j] == 1) || (Py[i, j] == 1))
                {
                    Z1[i, j] = 1;
                }
            }
            else

```

```

        {
            if (Py[i, j] == 1)
            {
                Z1[i, j] = 1;
            }
        }
    }
}
//виконуємо перетин X та Y
for (int i = 0; i < Z2.GetLength(0); i++)
{
    for (int j = 0; j < Z2.GetLength(1); j++)
    {
        if ((Fx[i, j] == 1) && (Py[i, j] == 1))
        {
            Z2[i, j] = 1;
        }
    }
}
//виконуємо доповнення до відображення Z2
for (int i = 0; i < Z2.GetLength(0); i++)
{
    for (int j = 0; j < Z2.GetLength(0); j++)
    {
        if (Z2[i, j] == 0)
        {
            Z3[i, j] = 1;
        }
        else
        {
            Z3[i, j] = 0;
        }
    }
}
//виконуємо операцію диз'юнктивної суми
for (int i = 0; i < Z3.GetLength(0); i++)
{
    for (int j = 0; j < Z3.GetLength(1); j++)
    {
        if ((Z1[i, j] == 1) && (Z3[i, j] == 1))
        {
            Z[i, j] = 1;
        }
    }
}
return Z;
}

```

Реалізація операції добутку графів

```

static int[,] Dobutok(int[] X, int[] Y, int[,] Fx, int[,] Py)
{
    int[,] Z = new int[X.Length * Y.Length, X.Length * Y.Length]; //створюємо масив
результату операції добутку
    for (int i = 0; i < Z.GetLength(1); i++)
    {
        for (int j = 0; j < Z.GetLength(0); j++)
        {
            Z[i, j] = 0; //заповнюємо масив нулями
        }
    }
    int numberOfZ = 0;
    int numberOfZValue = 0;
    //виконуємо Fx * Py
    for (int i = 0; i < Fx.GetLength(0); i++)
    {
        for (int j = 0; j < Py.GetLength(0); j++)

```

```

    {
        for (int k = 0; k < Fx.GetLength(1); k++)
        {
            for (int u = 0; u < Py.GetLength(1); u++)
            {
                if ((Fx[i, k] == 1) && (Py[j, u] == 1))
                {
                    numberOfZValue = returnNumberX(k) + u;
                    Z[numberOfZ, numberOfZValue] = 1;
                }
            }
        }
        numberOfZ++;
    }
}
return Z;
}

```

Реалізація операції суми графів

```

static int[,] Suma(int[] X, int[] Y, int[,] Fx, int[,] Py)
{
    int[,] Z = new int[X.Length * Y.Length, X.Length * Y.Length]; //створюємо масив
результату операції суми
    for (int i = 0; i < Z.GetLength(1); i++)
    {
        for (int j = 0; j < Z.GetLength(0); j++)
        {
            Z[i, j] = 0; //заповнюємо масив нулями
        }
    }
    int numberOfZ = 0;
    int numberOfZValue = 0;
    //виконуємо Fx * {y}
    for (int i = 0; i < Fx.GetLength(0); i++)
    {
        for (int k = 0; k < Y.Length; k++)
        {
            for (int j = 0; j < Fx.GetLength(1); j++)
            {
                if (Fx[i, j] == 1)
                {
                    numberOfZValue = returnNumberX(j) + k;
                    Z[numberOfZ, numberOfZValue] = 1;
                }
            }
            numberOfZ++;
        }
    }
    numberOfZ = 0;
    //виконуємо {x} * Py
    for (int i = 0; i < X.Length; i++)
    {
        for (int j = 0; j < Py.GetLength(0); j++)
        {
            for (int k = 0; k < Py.GetLength(1); k++)
            {
                if (Py[j, k] == 1)
                {
                    numberOfZValue = returnNumberX(i) + k;
                    Z[numberOfZ, numberOfZValue] = 1;
                }
            }
            numberOfZ++;
        }
    }
    return Z;
}

```

Реалізація операції композиції графів

```

static int[,] Compositia(int[] X, int[] Y, int[,] Fx, int[,] Py)
{
    int[,] Z = new int[X.Length * Y.Length, X.Length * Y.Length]; //створюємо масив
    результату операції композиції
    for (int i = 0; i < Z.GetLength(1); i++)
    {
        for (int j = 0; j < Z.GetLength(0); j++)
        {
            Z[i, j] = 0; //заповнюємо масив нулями
        }
    }
    int counter = 0;
    int numberOfZ = 0;
    int numberOfZValue = 0;
    //виконуємо Fx * Y
    for (int i = 0; i < Fx.GetLength(0); i++)
    {
        counter = 0;
        do
        {
            for (int j = 0; j < Fx.GetLength(1); j++)
            {
                if (Fx[i, j] == 1)
                {
                    for (int k = 0; k < Y.Length; k++)
                    {
                        numberOfZValue = returnNumberX(j) + k;
                        Z[numberOfZ, numberOfZValue] = 1;
                    }
                }
            }
            numberOfZ++;
            counter++;
        }
        while (counter < Y.Length);
    }
    counter = 0;
    numberOfZ = 0;
    //виконуємо X * Py
    do
    {
        for (int j = 0; j < Py.GetLength(0); j++)
        {
            for (int k = 0; k < Py.GetLength(1); k++)
            {
                for (int i = 0; i < X.Length; i++)
                {
                    if (Py[j, k] == 1)
                    {
                        numberOfZValue = returnNumberX(i) + k;
                        Z[numberOfZ, numberOfZValue] = 1;
                    }
                }
            }
            numberOfZ++;
        }
        counter++;
    }
    while (counter < X.Length);
    return Z;
}

```

Основна функція для виклику обрахунку всіх операцій над графами

```

static void Main(string[] args)
{
    Random rnd = new Random();
    string path = @"F:\Diplom\Variant1.txt"; //вказуємо файл, де буде зберігатись
    варіанти завдань та відповіді до них
    string text = "";
    int[] X = new int[8] { 1, 2, 3, 4, 5, 6, 7, 8 }; //створюємо перший масив для
    перших чотирьох операцій
    int[,] Fx = new int[8, 8];
    int[] Y = new int[10] { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 }; //створюємо другий масив
    для перших чотирьох операцій
    int[,] Py = new int[10, 10];
    text += "X\n";
    for (int i = 0; i < Fx.GetLength(0); i++)
    {
        for (int j = 0; j < Fx.GetLength(1); j++)
        {
            Fx[i, j] = rnd.Next(0, 2); //заповнюємо перший граф випадковими числами від
0 до 1
            text += Fx[i, j]; //записуємо початковий граф
        }
        text += "\n";
    }
    text += "Y\n";
    for (int i = 0; i < Py.GetLength(0); i++)
    {
        for (int j = 0; j < Py.GetLength(1); j++)
        {
            Py[i, j] = rnd.Next(0, 2); //заповнюємо другий граф випадковими числами від
0 до 1
            text += Py[i, j]; //записуємо початковий граф
        }
        text += "\n";
    }

    text += "Перетин\n";
    int[,] G1 = new int[X.Length, Y.Length]; //створюємо масив результату операції
перетину
    G1 = Peretin(X, Y, Fx, Py); //записуємо результат перетину
    for (int i = 0; i < G1.GetLength(0); i++)
    {
        for (int j = 0; j < G1.GetLength(1); j++)
        {
            text += G1[i, j];
        }
        text += "\n";
    }

    text += "Об'єднання\n";
    int[,] G2 = new int[Y.Length, Y.Length]; //створюємо масив результату операції
об'єднання
    G2 = Obednannya(X, Y, Fx, Py); //записуємо результат об'єднання
    for (int i = 0; i < G2.GetLength(0); i++)
    {
        for (int j = 0; j < G2.GetLength(1); j++)
        {
            text += G2[i, j];
        }
        text += "\n";
    }

    text += "Доповнення до відображення X\n";
    int[,] G3 = new int[Y.Length, Y.Length]; //створюємо масив результату операції
    доповнення до відображення X
    G3 = DopovnennyaX(X, Y, Fx, Py); //записуємо результат доповнення до відображення
    графу X
    for (int i = 0; i < G3.GetLength(0); i++)

```

```

{
    for (int j = 0; j < G3.GetLength(1); j++)
    {
        text += G3[i, j];
    }
    text += "\n";
}

text += "Доповнення до відображення Y\n";
int[,] G4 = new int[Y.Length, Y.Length]; //створюємо масив результату операції
доповнення до відображення Y
G4 = DopovnennyaY(X, Y, Fx, Py); //записуємо результат доповнення до відображення
графу Y
for (int i = 0; i < G4.GetLength(0); i++)
{
    for (int j = 0; j < G4.GetLength(1); j++)
    {
        text += G4[i, j];
    }
    text += "\n";
}

text += "Різниця X - Y\n";
int[,] G5 = new int[X.Length, X.Length]; //створюємо масив результату операції
різниці X - Y
G5 = RznitsyaXY(X, Y, Fx, Py); //записуємо результат різниці X - Y
for (int i = 0; i < G5.GetLength(0); i++)
{
    for (int j = 0; j < G5.GetLength(1); j++)
    {
        text += G5[i, j];
    }
    text += "\n";
}

text += "Різниця Y - X\n";
int[,] G6 = new int[Y.Length, Y.Length]; //створюємо масив результату операції
різниці Y - X
G6 = RznitsyaYX(X, Y, Fx, Py); //записуємо результат різниці X - Y
for (int i = 0; i < G6.GetLength(0); i++)
{
    for (int j = 0; j < G6.GetLength(1); j++)
    {
        text += G6[i, j];
    }
    text += "\n";
}

int[] X1 = new int[6] { 1, 2, 3, 4, 5, 6 }; //створюємо перший граф для операцій
з'єднання та диз'юнктивної суми
int[,] Fx1 = new int[6, 6];
int[] Y1 = new int[8] { 1, 2, 3, 4, 5, 6, 7, 8 }; //створюємо другий граф для
операцій з'єднання та диз'юнктивної суми
int[,] Py1 = new int[8, 8];
text += "\nX\n";
for (int i = 0; i < Fx1.GetLength(0); i++)
{
    for (int j = 0; j < Fx1.GetLength(1); j++)
    {
        Fx1[i, j] = rnd.Next(0, 2); //заповнюємо перший граф випадковими числами
від 0 до 1
        text += Fx1[i, j]; //записуємо початковий граф
    }
    text += "\n";
}
text += "Y\n";
for (int i = 0; i < Py1.GetLength(0); i++)
{
    for (int j = 0; j < Py1.GetLength(1); j++)
    {

```

```

        Py1[i, j] = rnd.Next(0, 2); //заповнюємо другий граф випадковими числами
від 0 до 1
        text += Py1[i, j]; //записуємо початковий граф
    }
    text += "\n";
}

text += "З'єднання\n";
int[,] G7 = new int[X1.Length + Y1.Length, X1.Length + Y1.Length]; //створюємо
масив результату операції з'єднання
G7 = Ziednnnya(X1, Y1, Fx1, Py1); //записуємо результат операції з'єднання
for (int i = 0; i < G7.GetLength(0); i++)
{
    for (int j = 0; j < G7.GetLength(1); j++)
    {
        text += G7[i, j];
    }
    text += "\n";
}

text += "Диз'юнктивна сума\n";
int[,] G8 = new int[X1.Length + Y1.Length, X1.Length + Y1.Length]; //створюємо
масив результату операції диз'юнктивної суми
G8 = DizSum(X1, Y1, Fx1, Py1); //записуємо результат операції диз'юнктивної суми
for (int i = 0; i < G8.GetLength(0); i++)
{
    for (int j = 0; j < G8.GetLength(1); j++)
    {
        text += G8[i, j];
    }
    text += "\n";
}

int[] X2 = new int[3] { 1, 2, 3 }; //створюємо перший граф для алгебраїчних
операцій
int[,] Fx2 = new int[3, 3];
int[] Y2 = new int[4] { 1, 2, 3, 4 }; //створюємо другий граф для алгебраїчних
операцій
int[,] Py2 = new int[4, 4];
text += "\nX\n";
for (int i = 0; i < Fx2.GetLength(0); i++)
{
    for (int j = 0; j < Fx2.GetLength(1); j++)
    {
        Fx2[i, j] = rnd.Next(0, 2); //заповнюємо перший граф випадковими числами
від 0 до 1
        text += Fx2[i, j]; //записуємо початковий граф
    }
    text += "\n";
}
text += "\nY\n";
for (int i = 0; i < Py2.GetLength(0); i++)
{
    for (int j = 0; j < Py2.GetLength(1); j++)
    {
        Py2[i, j] = rnd.Next(0, 2); //заповнюємо другий граф випадковими числами
від 0 до 1
        text += Py2[i, j]; //записуємо початковий граф
    }
    text += "\n";
}

text += "Добуток\n";
int[,] G9 = new int[X2.Length * Y2.Length, X2.Length * Y2.Length]; //створюємо
масив результату операції добутку
G9 = Dobutok(X2, Y2, Fx2, Py2); //записуємо результат операції добутку
for (int i = 0; i < G9.GetLength(0); i++)
{
    for (int j = 0; j < G9.GetLength(1); j++)
    {

```

```

        text += G9[i, j];
    }
    text += "\n";
}

text += "Сума\n";
int[,] G10 = new int[X2.Length * Y2.Length, X2.Length * Y2.Length]; //створюємо
масив результату операції суми
G10 = Suma(X2, Y2, Fx2, Py2); //записуємо результат операції суми
for (int i = 0; i < G10.GetLength(0); i++)
{
    for (int j = 0; j < G10.GetLength(1); j++)
    {
        text += G10[i, j];
    }
    text += "\n";
}

text += "Композиція\n";
int[,] G11 = new int[X2.Length * Y2.Length, X2.Length * Y2.Length]; //створюємо
масив результату операції композиції
G11 = Compositia(X2, Y2, Fx2, Py2); //записуємо результат операції композиції
for (int i = 0; i < G11.GetLength(0); i++)
{
    for (int j = 0; j < G11.GetLength(1); j++)
    {
        text += G11[i, j];
    }
    text += "\n";
}

using (StreamWriter wr = new StreamWriter(path))
{
    wr.WriteLine(text); //записуємо всі результати в текстовий документ
    wr.Close();
}
}

```