

НУЛП, ІКНІ, САПР		Тема	оцінка	підпис
КН-414	5	Ізоморфізм графів		
Коцюба В.С.				
№ залікової: 16081031				
Дискретні моделі в САПР			Викладач: к.т.н., асистент Кривий Р.З.	

Мета:

Вивчення і дослідження основних підходів до встановлення ізоморфізму графів.

Завдання:

Реалізувати метод повного перебору для встановлення ізоморфізму графів

Теоретичні відомості:

Два графа $G=(X,U,P)$ і $G'=(X',U',P')$ називаються ізоморфними, якщо між їх вершинами, а також між їхніми ребрами можна встановити взаємно однозначне співвідношення $X \leftrightarrow X'$, $U \leftrightarrow U'$, що зберігає інцидентність, тобто таке, що для всякої пари $(x,y) \in X$ ребра $u \in U$, що з'єднує їх, обов'язково існує пара $(x',y') \in X'$ і ребро $u' \in U'$, що з'єднує їх, і навпаки. Тут P - предикат, інцидентор графа G . Зауважимо, що відношення ізоморфізму графів рефлексивне, симетричне і транзитивне, тобто представляє собою еквівалентність.

Одним з найпростіших з точки зору програмної реалізації, є алгоритм перевірки ізоморфізму графів повним перебором(можливої перенумерації вершин), але складність цього алгоритму є факторіальною.

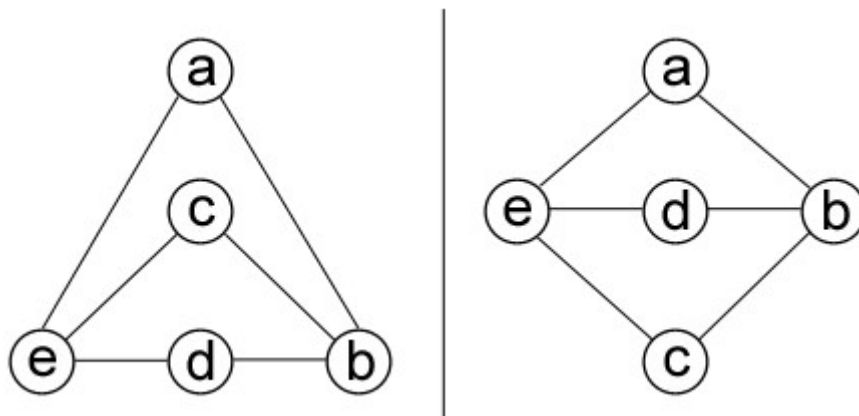


Рис.1 Графи для перевірки ізоморфізму (5 вершин)

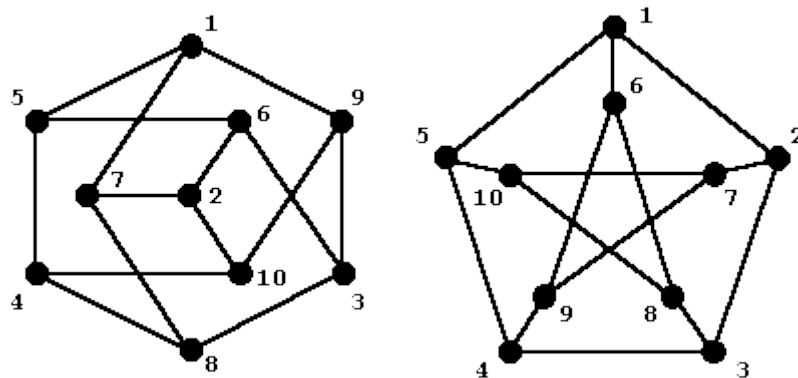


Рис.2 Графи для перевірки ізоморфізму (10 вершин)

Робота з програмою:

Після запуску програми у лівому краї вікна відображаються 2 текстбоксы з матрицями суміжності графів.

Вводимо кількість вершин у відповідному місці.

Нажимаємо кнопку 'Запуск алгоритму', у центральному нижньому текстбоксі з'явиться інформація про те, чи є графи ізоморфними.

У правому краї екрану на двох канвасах відображаються самі графи.

Якщо виникає потреба змінити матриці суміжності, то міняємо потрібні цифри в текстбоксах та знову натискаємо кнопку 'Запуск алгоритму'.

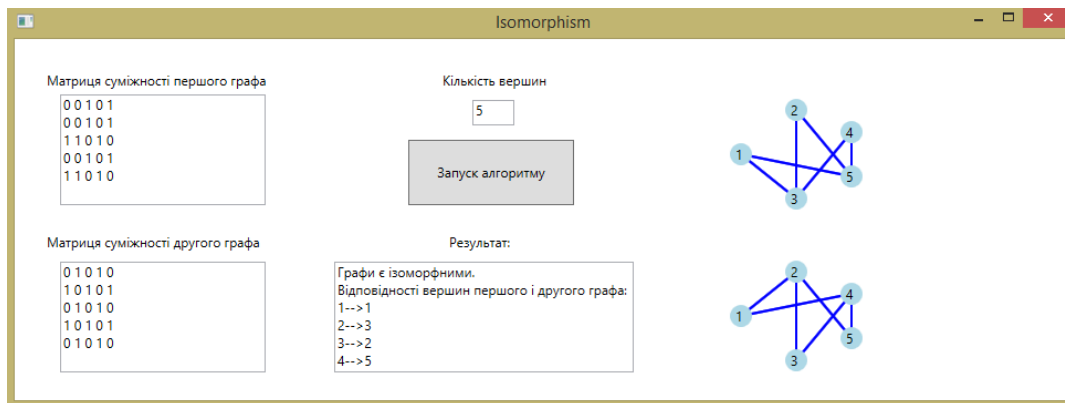


Рис.3 Графи ізоморфні (5 точок)

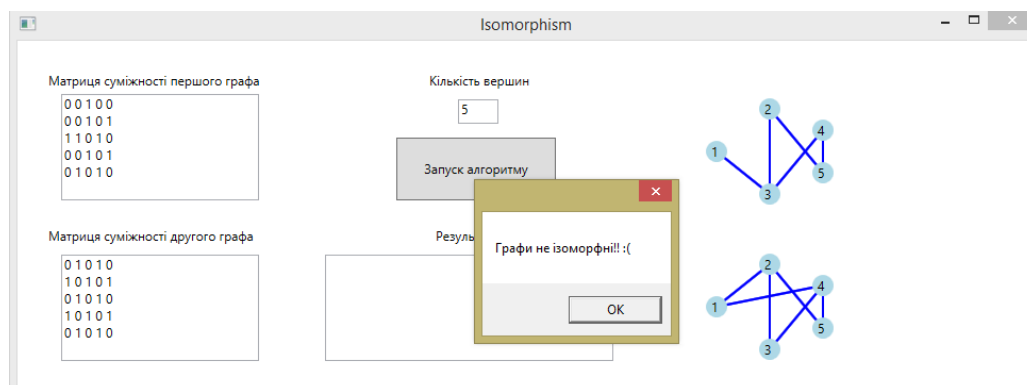


Рис.4 Графи не ізоморфні (після зміни елемента матриці)

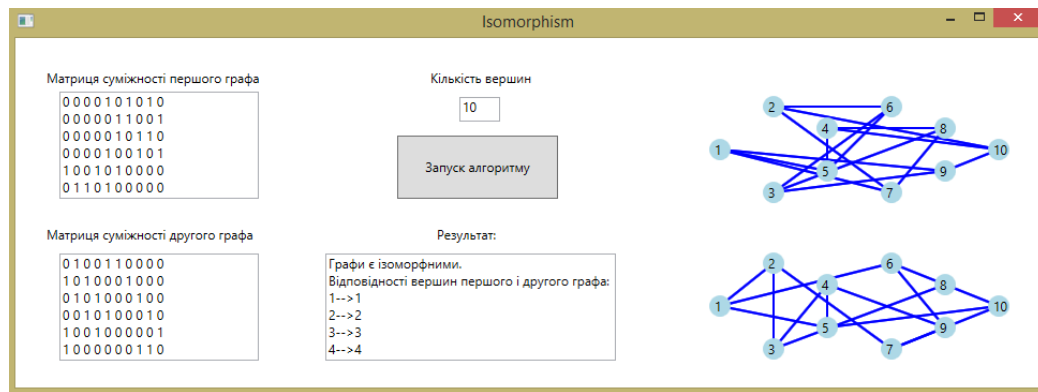


Рис.5 Результат роботи програми, графи ізоморфні (для 10 точок)

Фрагмент програми:

```
public void BrutalForce()
{
    firstSumArray = new int[numVert];
    secondSumArray = new int[numVert];
    conformity = new int[numVert];

    for (int i = 0; i < numVert; i++)//перебираємо всі вершини
        conformity[i] = -1;//ставимо їм у відповідність вершину -1

    Draw(canv1);
    Draw(canv2);
    tb4.Text = Res();
}

public String Res
{
    for (int i = 0; i < numVert; i++)//перебираємо усі рядки
    {
        for (int j = 0; j < numVert; j++)//перебираємо усі вершини
        {
            firstSumArray[i] += data1[i, j];//додаємо вагу ребра
            secondSumArray[i] += data2[i, j];//додаємо вагу ребра
        }
    }

    for (int i = 0; i < numVert; i++)//перебираємо усі вершини
    {
        for (int j = 0; j < numVert; j++)//перебираємо усі вершини
        {
            bool isDone = false;//чи перевірки завершені

            for (int k = 0; k < numVert; k++)//перебираємо усі вершини
                if (conformity[k] == j)//чи вершина має відповідну
                    isDone = true;//перевірки завершені

            if (!isDone && firstSumArray[i] == secondSumArray[j])//якщо перевірки не
                завершені і вершина має відповідну
            {
                conformity[i] = j;//встановлюємо поточне j як відповідну вершину
                break;//виходимо з циклу
            }
        }
    }

    result = "Графи є ізоморфними.";//формуємо стрічку-відповідь
    bool isrouted = true; //чи було знайдено розв'язок

    for (int i = 0; i < numVert; i++)//перебираємо усі вершини
```

```

        if (conformity[i] == -1)//якщо для вершини нема відповідної
        {
            MessageBox.Show("Графи не ізоморфні!! :(");
            isrouted = false;//рішення не знайдено
            result = "Графи не є ізоморфними.";//формуємо стрічку-відповідь
            break;//виходимо з циклу
        }

        if (isrouted)//якщо рішення знайдено
        {
            MessageBox.Show("Графи ізоморфні!! :)");
            result += "\nВідповідності вершин першого і другого графа: \n";//доповнюємо
            стрічку-відповідь
            for (int i = 0; i < numVert; i++)//перебираємо всі вершини
            {
                result += (i + 1) + "-->" + (conformity[i] + 1) + " ";//додаємо до
                стрічки відповіді відповідності вершин
                result += "\n";
            }
        }

        return result;//повертаємо отриману стрічку-відповідь
    }

```

Висновок: На цій лабораторній роботі було здійснено ознайомлення з алгоритмом повного перебору визначення ізоморфності графів.