

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНОМУ УНІВЕРСИТЕТІ “ЛЬВІВСЬКА  
ПОЛІТЕХНІКА”

**Кафедра систем штучного інтелекту**  
Лабораторна робота №3  
з дисципліни  
«Дискретна математика»

Львів 2019р.

Виконав студент групи КН-115 : Вагін Микита  
Викладач: Мельникова Н.І

**Тема:** Побудова матриці бінарного відношення

**Мета роботи:** Набуття практичних вмінь та навичок при побудові матриць бінарних відношень та визначені їх типів.

**Додаток 1**

**Варіант № 3**

1. Чи є вірною рівність  $(A \cap B) \times (A \cap C) = A \times (B \cap C)$ ?

**Розв'язання**

Нехай  $(x,y) \in (A \cap B) \times (A \cap C) \Leftrightarrow x \in (A \cap B) \times y \in (A \cap C) \Leftrightarrow (x \in A \ \& \ x \in B) \ \& \ (y \in A \ \& \ y \in C) \Leftrightarrow (x \in A \ \& \ y \in A) \ \& \ (x \in B \ \& \ y \in C) \Leftrightarrow (x,y) \in (A \times A) \ \& \ (x,y) \in (B \times C) \Leftrightarrow (x,y) \in (A \times A) \cap (B \times C)$ .

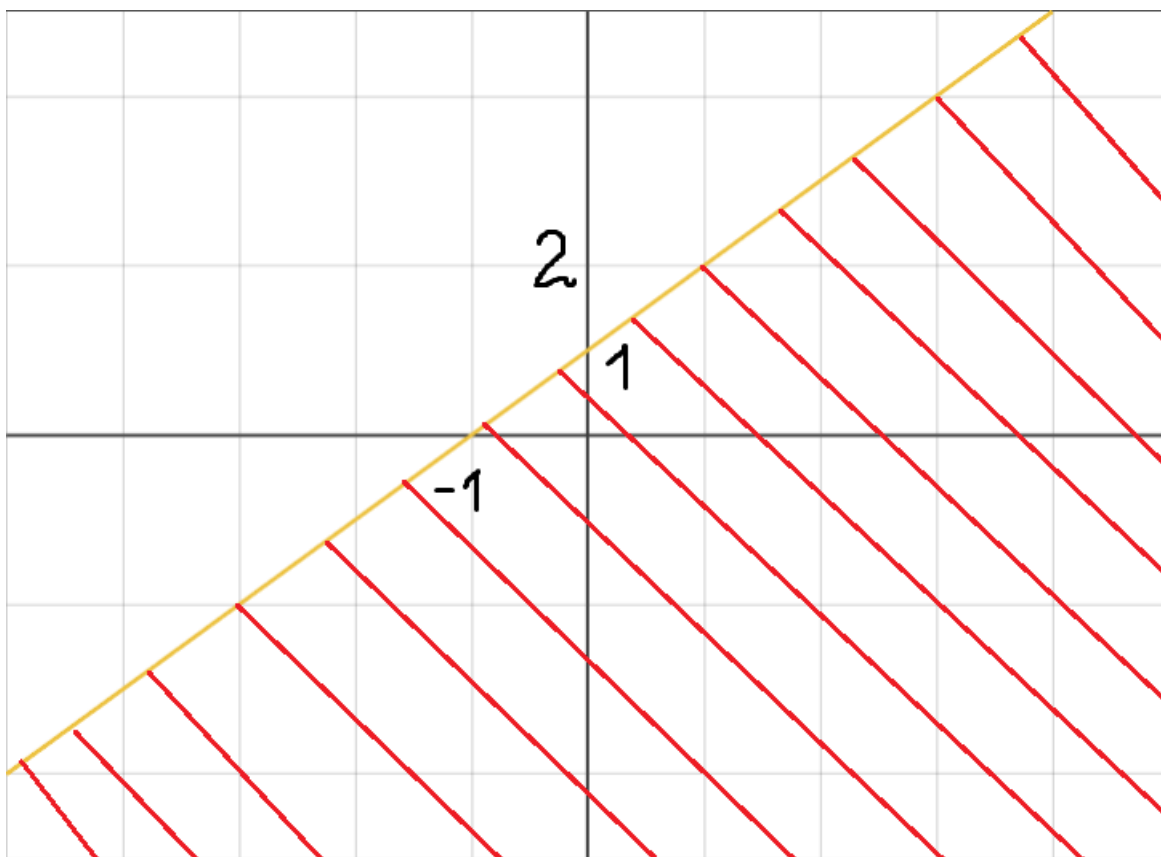
Отже,  $(A \cap B) \times (A \cap C) \neq A \times (B \cap C)$  – рівність невірна.

3. Зобразити відношення графічно:

$\alpha = \{(x,y) | (x,y) \in \mathbb{R}^2 \ \& \ x+1 \geq y\}$ , де  $\mathbb{R}$  - множина дійсних чисел.

**Розв'язання**

Область визначення  $\delta\alpha = (-\infty; +\infty)$ , область значень  $\rho\alpha = (-\infty; +\infty)$ .



4. Навести приклад бінарного відношення  $R \subset A \times A$ , де  $A = \{a,b,c,d,e\}$ , яке є антирефлексивне, антисиметричне, транзитивне, та побудувати його матрицю.

Матриця відношення  $R = \{(a,b), (a,c), (a,d), (a,e), (b,c), (b,d), (b,e), (c,d), (c,e), (d,e)\}$  буде мати вигляд:

0	1	1	1	1
0	0	1	1	1
0	0	0	1	1
0	0	0	0	1
0	0	0	0	0

Антирефлексивна :  $\sigma_{11}=0, \sigma_{22}=0, \sigma_{33}=0, \sigma_{44}=0, \sigma_{55}=0$ .

Антисиметрична :  $\sigma_{12}=1, \sigma_{21}=0, \sigma_{13}=1, \sigma_{31}=0$ .

Транзитивна :  $\sigma_{12}=1, \sigma_{23}=0, \sigma_{13}=1$ .

## Додаток 2

Написати програму, яка знаходить матрицю бінарного відношення  $\rho \subset A \times B$ , заданого на двох числових множинах. Реалізувати введення цих множин, та виведення на екран матриці відношення. Перевірити програмно якого типу є задане відношення. Навести різні варіанти тестових прикладів.

Відношення обрати згідно варіанту:

$$\rho = \{(a, b) | a \in A \ \& \ b \in B \ \& \ (a+b) > 2\}.$$

### Код програми

```
#include <iostream>
using namespace std;
void Cartes(int** AxB, int* A, int* B, int ElementsA, int ElementsB)
{
    int c = 0;
    for (int i = 0; i < ElementsA; i++)
    {
        for (int j = 0; j < ElementsB; j++)
        {
            AxB[c][0] = A[i];
            AxB[c][1] = B[j];
            c++;
        }
    }
}

void OutputMatrix(int** AxB, int** matrix, int ElementsA, int ElementsB)
{
    cout << "Matrix:" << endl;
    int c = 0;
    for (int i = 0; i < ElementsA; i++)
    {
        for (int j = 0; j < ElementsB; j++, c++)
        {
            if (AxB[c][0] + AxB[c][1] > 2) {
                matrix[i][j] = 1;
                cout << matrix[i][j] << "\t";
            }
            else {
                matrix[i][j] = 0;
                cout << matrix[i][j] << "\t";
            }
        }
        cout << endl;
    }
}

int Reflexity(int** matrix, int n, int* reflex)
{
    for (int i = 0; i < n; i++)
    {
        if (matrix[i][i] == 0)
            break;
        if (matrix[i][i] == 1 && i == n - 1) {
            *reflex = 1;
            return 0;
        }
    }
    for (int i = 0; i < n; i++)
    {
        if (matrix[i][i] == 1)
            break;
    }
}
```

```

        if (matrix[i][i] == 0 && i == n - 1) {
            *reflex = 0;
            return 0;
        }
    }
    *reflex = 2;
}

int Transitivity(int** matrix, int ElementsA, int* transitive) {
    for (int i = 0; i < ElementsA; i++)
    {
        for (int j = 0; j < ElementsA; j++)
        {
            for (int c = 0; c < ElementsA; c++)
            {
                if (matrix[i][j] == 1 && matrix[j][c] == 1 && matrix[i][c] ==
0) {
                    *transitive = 0;
                    return 0;
                }
            }
        }
    }
    for (int i = 0; i < ElementsA; i++)
    {
        for (int j = 0; j < ElementsA; j++)
        {
            for (int c = 0; c < ElementsA; c++)
            {
                if (matrix[i][j] == 1 && matrix[j][c] == 1 && matrix[i][c] ==
1) {
                    *transitive = 1;
                    return 0;
                }
            }
        }
    }
    *transitive = 0;
    return 0;
}

int main() {
    int ElementsA, ElementsB;

    cout << "Enter a number of the elements of the first array: ";
    cin >> ElementsA;
    if (ElementsA < 0 || !cin) {
        cout << "Error";
        return 0;
    }

    cout << "Enter a number of the elements of the second array: ";
    cin >> ElementsB;
    if (ElementsB < 0 || !cin) {
        cout << "Error";
        return 0;
    }
    int* A = new int[ElementsA];
    int* B = new int[ElementsB];

    cout << "Enter the elements of the first array: ";
    for (int i = 0; i < ElementsA; i++) {
        cin >> A[i];
        if (!cin) {

```

```

        cout << "Error";
        return 0;
    }
}

cout << "Enter the elements of the second array: ";
for (int i = 0; i < ElementsB; i++) {
    cin >> B[i];
    if (!cin) {
        cout << "Error";
        return 0;
    }
}

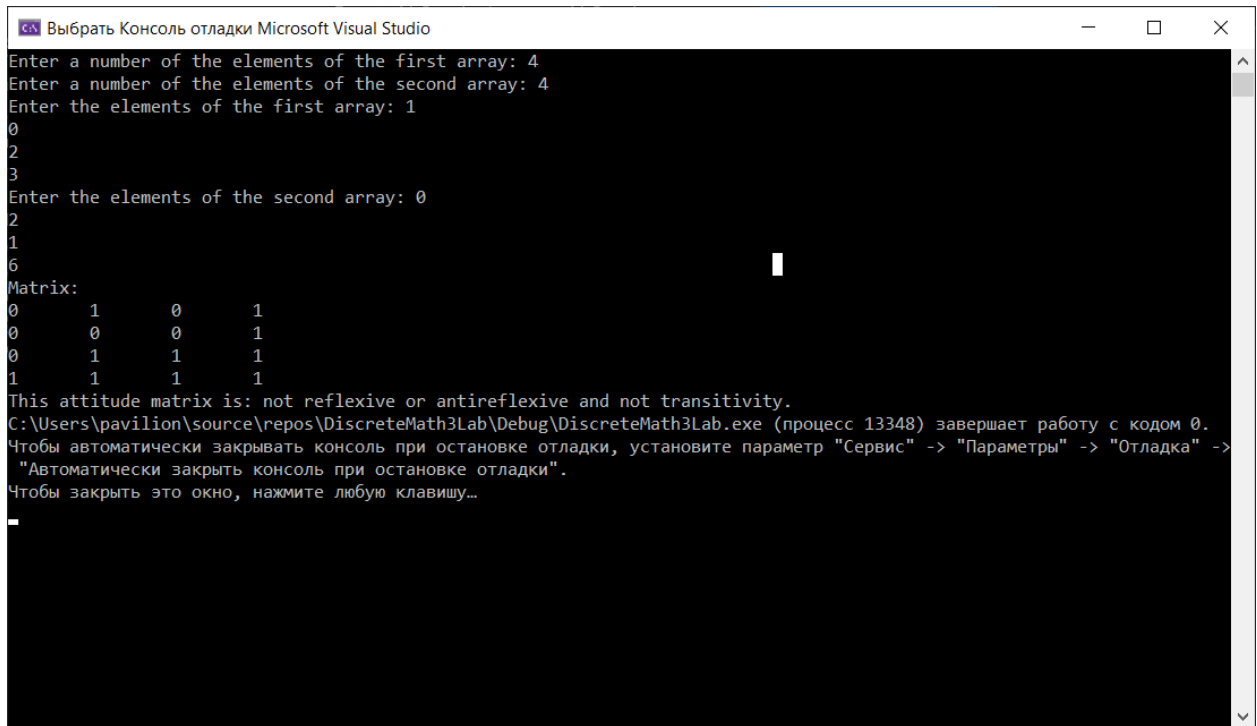
int** AxB = new int* [ElementsA * ElementsB];
for (int i = 0; i < ElementsA * ElementsB; i++)
{
    AxB[i] = new int[2];
}

int** AttitudeMatrix = new int* [ElementsA];
for (int i = 0; i < ElementsA; i++)
{
    AttitudeMatrix[i] = new int[ElementsB];
}
Cartes(AxB, A, B, ElementsA, ElementsB);
OutputMatrix(AxB, AttitudeMatrix, ElementsA, ElementsB);

if (ElementsA != ElementsB) {
    cout << "It's not a square matrix";
    return 0;
}
int reflex, transitive;
int* preflex = &reflex;
int* ptransitive = &transitive;
Reflexity(AttitudeMatrix, ElementsA, preflex);
Transitivity(AttitudeMatrix, ElementsA, ptransitive);
cout << "This attitude matrix is: ";
if (*preflex == 1)
    cout << "reflexive and ";
else if (*preflex == 0)
    cout << "antireflexive and ";
else
    cout << "not reflexive or antireflexive and ";
if (*ptransitive == 1)
    cout << "transitivity.";
else if (*ptransitive == 0)
    cout << "not transitivity.";
delete[] A;
delete[] B;
for (int i = 0; i < ElementsA * ElementsB; i++)
{
    delete AxB[i];
}
delete[] AxB;
for (int i = 0; i < ElementsA; i++)
{
    delete AttitudeMatrix[i];
}
delete[] AttitudeMatrix;
return 0;
}

```

## Результат програми



Выбрать Консоль отладки Microsoft Visual Studio

```
Enter a number of the elements of the first array: 4
Enter a number of the elements of the second array: 4
Enter the elements of the first array: 1
0
2
3
Enter the elements of the second array: 0
2
1
6
Matrix:
0      1      0      1
0      0      0      1
0      1      1      1
1      1      1      1
This attitude matrix is: not reflexive or antireflexive and not transitivity.
C:\Users\pavilion\source\repos\DiscreteMath3Lab\Debug\DiscreteMath3Lab.exe (процесс 13348) завершает работу с кодом 0.
Чтобы автоматически закрывать консоль при остановке отладки, установите параметр "Сервис" -> "Параметры" -> "Отладка" ->
"Автоматически закрыть консоль при остановке отладки".
Чтобы закрыть это окно, нажмите любую клавишу...
```