

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНОМУ УНІВЕРСИТЕТУ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”

Кафедра систем штучного інтелекту

Лабораторна робота № 3

з дисципліни

«Дискретна математика»

Виконав:

студент групи КН-112

Матвійів Остап-Василь

Викладач:

Мельникова Н.І.

Львів – 2019 р.

Тема: Побудова матриці бінарного відношення

Мета роботи: набуття практичних вмінь та навичок при побудові матриць бінарних відношень та визначені їх типів

Варіант-11

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

Ні.

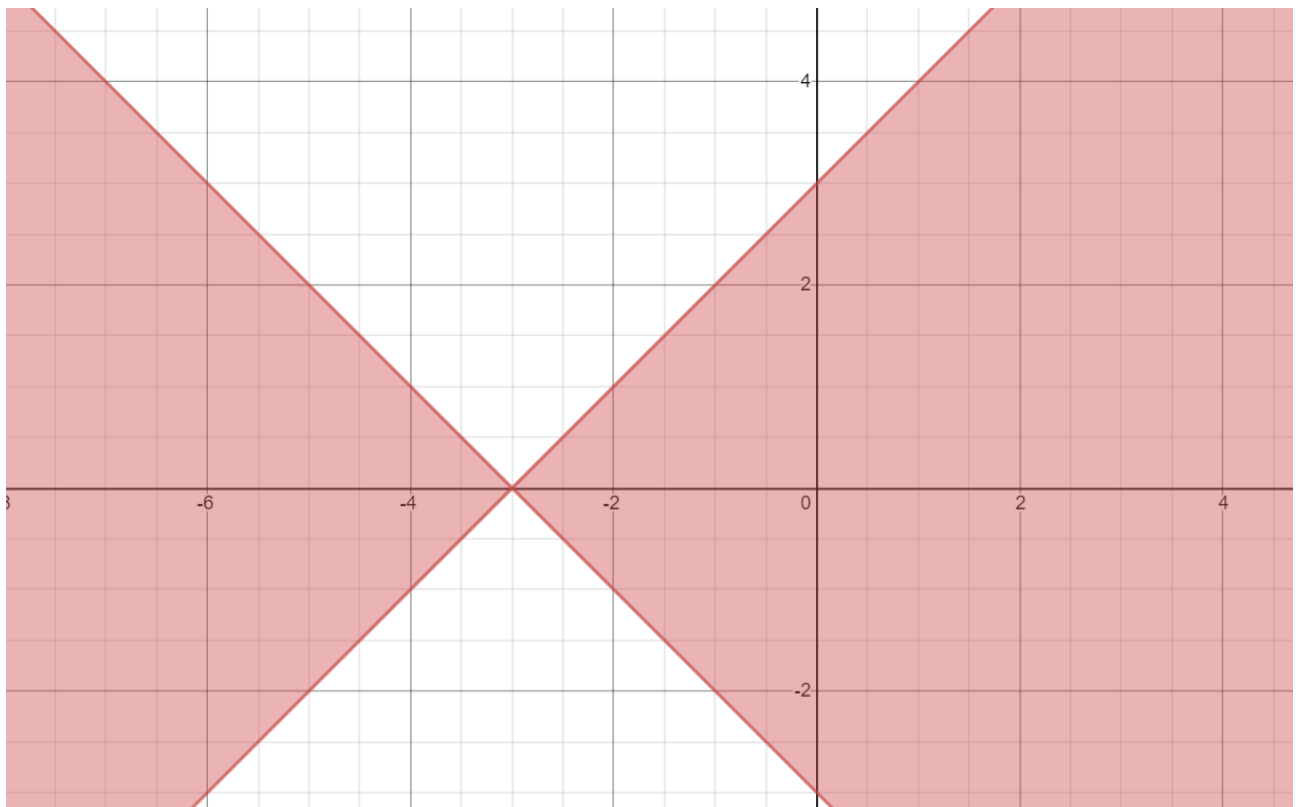
2. Знайти матрицю відношення $R \subset M \times 2^M$, де $M = \{1, 2, 3\}$:

$$R = \{(x, y) \mid x \in M \text{ \& } y \subset M \text{ \& } |y| > x\}.$$

	\emptyset	$\{1\}$	$\{2\}$	$\{3\}$	$\{1,2\}$	$\{1,3\}$	$\{2,3\}$	$\{1,2,3\}$
1	0	0	0	0	1	1	1	1
2	0	0	0	0	0	0	0	1
3	0	0	0	0	0	0	0	0

3. Зобразити відношення графічно:

$$\alpha = \{(x, y) \mid (x, y) \in \mathbb{R}^2 \text{ \& } |x+3| \geq |y|\}, \text{ де } \mathbb{R} - \text{множина дійсних чисел.}$$



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

```

0 1 0 0 1
0 0 1 0 0
A(R)= ( 1 0 0 1 0 )
1 0 0 0 1
0 1 1 0 0

```

5. Визначити множину (якщо це можливо), на якій дане відношення є: а) функціональним; б) бієктивним:

$$\alpha = \{(x, y) | (x, y) \in R^2 \text{ \& } x + \sqrt{y^2} = 1\}.$$

Це не можливо, оскільки дана функція не є ін'єктивною.

Завдання №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.

```

#include <iostream>
#include <locale>
using namespace std;

int main()
{
    int Pot1, Pot2;

    cout << "Vvedit potuzhnist 1 mnojini: ";
    cin >> Pot1;
    cout << "Vvedit potuzhnist 2 mnojini: ";
    cin >> Pot2;

    float* arr1 = new float[Pot1];
    float* arr2 = new float[Pot1];

    cout << "Vvedit 1 mnojinu: ";
    for (int i = 0; i < Pot1; i++) {
        cin >> arr1[i];
    }

    cout << "Vvedit 2 mnojinu: ";
    for (int i = 0; i < Pot2; i++) {
        cin >> arr2[i];
    }
}

```

```

for (int i = 0; i < Pot1; i++) {
    arr3[i] = new int[Pot2];
}
for (int i = 0; i < Pot1; i++) {
    for (int j = 0; j < Pot2; j++) {
        if (2 * arr1[j] > 3 * arr2[i]) {
            arr3[i][j] = 1;
        }
        else {
            arr3[i][j] = 0;
        }
    }
}

delete[] arr1;
delete[] arr2;

for (int i = 0; i < Pot1; i++) {
    for (int j = 0; j < Pot2; j++) {
        cout << arr3[i][j] << "\t";
    }
    cout << endl;
}

int check = 0;
for (int i = 0; i < Pot1; i++) {
    if (arr3[i][i] == 1) {
        check++;
    }
}

if (check == Pot1) {
    cout << "Reflexivna ";
}

```

```

        cout << "Reflexivna ";
    }

    else if (check == 0) {
        cout << "Antireflexivna ";
    }

    else {
        cout << "Nereflexivna ";
    }

    int check0 = 0;
    int check1 = 0;
    for (int i = 0; i < Pot1; i++) {
        for (int j = 0; j < Pot2; j++) {
            if (arr3[i][j] == arr3[j][i] && i != j) {
                check0++;
            }
            else {
                check1++;
            }
        }
    }

    if (check0 == Pot1 * Pot2 - Pot1) {
        cout << "Symetrichna ";
    }
    else if (check1 == Pot1 * Pot2 - Pot1) {
        cout << "Antisymetrichna ";
    }
    else {
        cout << "Nesymetrichna ";
    }

    int tranz = 0, tranz1 = 1;
    for (int i = 0; i < Pot1; i++) {
        for (int j = 0; j < Pot2; j++) {
            if (arr3[i][j] == 1) {
                for (int z = 0; z < Pot1; z++) {
                    if (arr3[j][z] == 1) {
                        if (arr3[i][z] == 1) {
                            tranz = 1;
                        }
                        else {
                            tranz1 = 0;
                        }
                    }
                }
            }
        }

        if (tranz * tranz1 == 1) {
            cout << "Tranzitivna ";
        }
        else {
            int antitrantz = 0;
            for (int i = 0; i < Pot1; i++) {
                for (int j = 0; j < Pot2; j++) {
                    if (arr3[i][j] == 1) {
                        for (int z = 0; z < Pot1; z++) {
                            if (arr3[j][z] == 1) {
                                if (arr3[i][z] == 0) {
                                    antitrantz = 1;
                                }
                            }
                        }
                    }
                }

                if (antitrantz == 0) {
                    antitrantz = 1;
                }
                else {
                    antitrantz = 0;
                }
            }

            if (antitrantz == 1) {
                cout << "Antitrantzitivna ";
            }
            else {
                cout << "Netrantzitivna ";
            }
        }
    }
}

```

```

Vvedit potuzhnist 1 mnojini: 5
Vvedit potuzhnist 2 mnojini: 5
Vvedit 1 mnojinu: 4 6 7 9 5
Vvedit 2 mnojinu: 1 3 4 2 5
1      1      1      1      1
0      1      1      1      1
0      0      1      1      0
1      1      1      1      1
0      0      0      1      0
Nereflexivna Nesymetrichna Antitrantzitivna

```

Висновок: на цій лабораторній роботі я дістав практичні вміння та навички при побудові матриць відношень, та визначення їх типу.