

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ**  
**“ЛЬВІВСЬКА ПОЛІТЕХНІКА”**

**Кафедра систем штучного інтелекту**

**Лабораторна робота №3**  
з дисципліни «Дискретна математика»

**Виконав:**

Студент групи КН-112

Шклярів Віталій

**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.

### Завдання з Додатку 1

#### Варіант № 15

**1.** Чи є вірною рівність:

$$(A \times (B \cap C)) \cap ((A \cap B) \times C) = (A \times C) \cap (B \times B) ?$$

Розв'язок:

$$\begin{aligned} (x, y) \in (A \times (B \cap C)) \cap ((A \cap B) \times C) &\sim (x, y) \in ((A \times B) \cap (A \times C) \cap (A \times C) \cap (B \times C)) \sim \\ &\sim (x, y) \in ((A \times B) \cap (A \times C) \cap (B \times C)) \sim (x, y) \in ((A \times C) \cap (B \times C) \cap (A \times B)) \sim \\ &\sim (x, y) \in ((A \times C) \cap (B \times B)) = (A \times C) \cap (B \times B) \end{aligned}$$

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

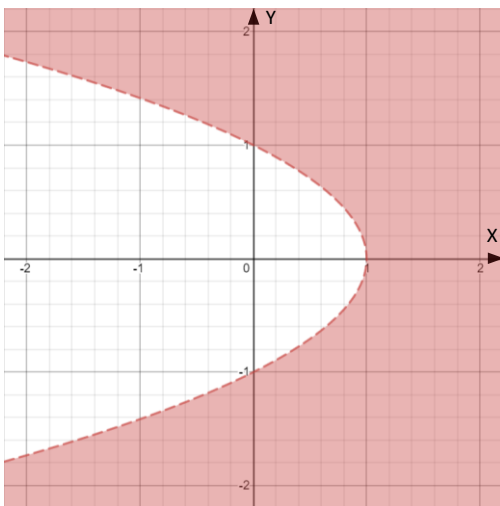
$$R = \{(x, y) | x \in M \text{ \& } y \subset M \text{ \& } |y| \leq x\}$$

Розв'язок:

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

**3.** Зобразити відношення графічно:

$$\alpha = \{(x, y) | (x, y) \in R^2 \text{ \& } x + y^2 - 1 > 0\}, \text{ де } R - \text{множина дійсних чисел.}$$



$$\delta_\alpha = (-\infty; +\infty)$$

$$\rho_\alpha = (-\infty; +\infty)$$

$$y^2 > 1 - x$$

$$\begin{cases} y > \sqrt{1 - x} \\ y < -\sqrt{1 - x} \end{cases}$$

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

$$R = \{(a,c);(a,e);(b,c);(b,d);(c,d);(e,c)\}$$

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

$$a_{bc} = a_{cd} = a_{bd} = 1$$

$$a_{ae} = a_{ec} = a_{ac} = 1$$

$$a_{aa} = a_{bb} = a_{cc} = a_{dd} = a_{ee} = 0 \Rightarrow \text{антирефлексивне}$$

$$a_{ac} \neq a_{ca}; a_{ae} \neq a_{ea}; a_{bc} \neq a_{cb}$$

$$a_{bd} \neq a_{db}; a_{cd} \neq a_{dc}; a_{ec} \neq a_{ce}$$

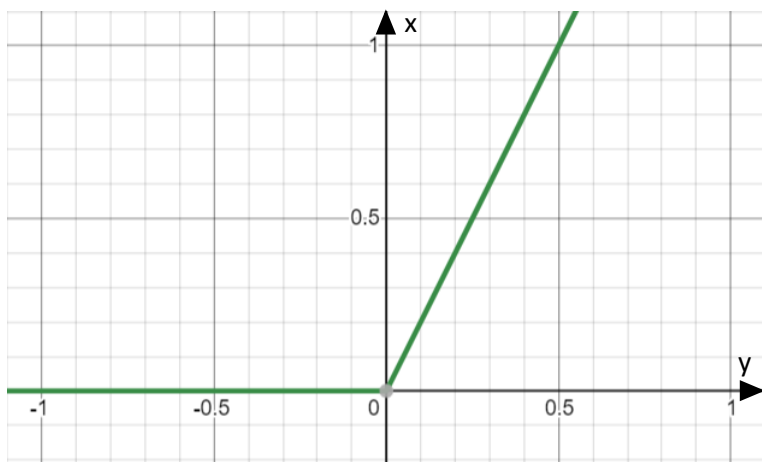
транзитивне

несиметричне

5. Визначити множину (якщо це можливо), на якій дане відношення є:

а) функціональним; б) бієктивним:

$$\alpha = \{(x, y) | (x, y) \in R^2 \ \& \ y = x + |x|\} \quad (x, y) \in R$$



$$\begin{cases} y = 2x; & x \in (0; +\infty) \\ y = 0; & x \in (-\infty; 0] \end{cases}$$

Відношення є **функціональним**, бо кожному  $x$  відповідає єдиний  $y$ .

Відношення є **бієктивним**, бо воно є ін'єктивним (різним аргументам відповідають різні значення) та сюр'єктивним (множина значень співпадає з областю значень) одночасно.

## Завдання з Додатку 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.

$$\rho = \{(a, b) | a \in A \ \& \ b \in B \ \& \ (a + b + 1) > 3\};$$

```

#include <iostream>
#include "func.h"
#include "relations.h"

using namespace std;

int main() {
    int A_B_pot;
    cout << "Enter the number of arrays elements A and B" << endl;
    cin >> A_B_pot;
    if (cin.fail()) {
        cout << "ERROR";
    }
    int *A = new int[A_B_pot];
    int *B = new int[A_B_pot];
    //ввід масивів
    cout << "Enter elements of the array A" << endl;
    enter(A, A_B_pot);
    cout << "Enter elements of the array B" << endl;
    enter(B, A_B_pot);
    //сортування масивів
    //sort(A, A_B_pot);
    //sort(B, A_B_pot);
    //.../
    int **AB = new int *[A_B_pot];
    for (int i = 0; i < A_B_pot; i++)
        AB[i] = new int[A_B_pot];
    //створення та виведення матриці відношення
    creat(A, B, AB, A_B_pot);
    output(AB, A_B_pot);
    cout << endl;
    // відношення
    reflex(AB, A_B_pot);
    anti_reflex(AB, A_B_pot);
    symmetry(AB, A_B_pot);
    anti_symmetry(AB, A_B_pot);

    int **C = new int *[A_B_pot];
    for (int i = 0; i < A_B_pot; i++)
        C[i] = new int[A_B_pot];
    for (int j = 0; j < A_B_pot; ++j) {
        for (int i = 0; i < A_B_pot; ++i) {
            C[i][j]=AB[i][j];
        }
    }
    transitivity(AB, C, A_B_pot);

    for (int j = 0; j < A_B_pot; ++j) {
        for (int i = 0; i < A_B_pot; ++i) {
            C[i][j]=AB[i][j];
        }
    }
    anti_transitivity(AB, C, A_B_pot);

    for (int i = 0; i < A_B_pot; i++)
        delete[] AB[i];
    delete[] A;
    delete[] B;
    return 0;
}

```

```

#ifndef DM_L3_RELATIONS_H
#define DM_L3_RELATIONS_H

#include <iostream>

using namespace std;

// для перевірки на рефлексивність
int reflex(int **arr, int P) {
    for (int i = 0; i < P; ++i) {
        if (arr[i][i] == 1)
            continue;
        else {
            cout << "Ne Refleksyivne" << endl;
            return 0;
        }
    }
    cout << "Refleksyivne" << endl;
    return 0;
}

// для перевірки на антирефлексивність
int anti_reflex(int **arr, int P) {
    for (int i = 0; i < P; ++i) {
        if (arr[i][i] == 0)
            continue;
        else {
            cout << "Ne AntiRefleksyivne" << endl;
            return 0;
        }
    }
    cout << "AntiRefleksyivne" << endl;
    return 0;
}

// для перевірки на симетричність
int symmetry(int **arr, int P) {
    for (int i = 0; i < P; ++i) {
        for (int j = 0; j < P; ++j) {
            if (arr[i][j] == 1) {
                if (arr[j][i] == arr[i][j])
                    continue;
                else {
                    cout << "Ne Symetrychne" << endl;
                    return 0;
                }
            }
        }
    }
    cout << "Symetrychne" << endl;
    return 0;
}

// для перевірки на антисиметричність
int anti_symmetry(int **arr, int P) {
    for (int i = 0; i < P; ++i) {
        for (int j = 0; j < P; ++j) {
            if (i == j){
                continue;
            }
            if (arr[i][j] == 1) {
                if (arr[j][i] == 0)
                    continue;
                else {
                    cout << "Ne AntiSymetrychne" << endl;
                    return 0;
                }
            }
        }
    }
    cout << "AntiSymetrychne" << endl;
    return 0;
}

```

```

#ifndef DM_L3_FUNC_H
#define DM_L3_FUNC_H

#include <iostream>
using namespace std;
//функція для вводу масиву
void enter(int *arr, int P){
    for (int i = 0; i < P; ++i) {
        cin >> arr[i];
    }
}

//функція для сортування масиву
void sort(int *arr, int P){
    int tmp;
    for(int j = 0; j < P - 1; j++){
        for(int i = 0; i < P - 1; i++){
            if (arr[i] > arr[i + 1]){
                tmp = arr[i];
                arr[i] = arr[i + 1];
                arr[i + 1] = tmp;
            }
        }
    }
}

//функція для побудови матриці
void creat(int *A, int *B, int **AB, int P){
    for (int i = 0; i < P; ++i) {
        for (int j = 0; j < P; ++j) {
            if (A[i]+B[j]>2)
                AB[i][j] = {1};
            else
                AB[i][j] = {0};
        }
    }
}

//функція для виводу матриці
void output(int **AB, int P){
    for (int i = 0; i < P; ++i) {
        for (int j = 0; j < P; ++j) {
            cout << AB[i][j] << " ";
        }
        cout << endl;
    }
}

#endif

```

```

Enter the number of arrays elements A and B
3
Enter elements of the array A
3 -1 -4
Enter elements of the array B
-4 1 4
0 1 1
0 0 1
0 0 0

Ne Refleksyvne
AntiRefleksyvne
Ne Symetrychne
AntiSymetrychne
Tranzytyvne
Ne AntiTranzytyvne

```

```

Enter the number of arrays elements A and B
3
Enter elements of the array A
3 5 6
Enter elements of the array B
3 5 6
1 1 1
1 1 1
1 1 1

Refleksyvne
Ne AntiRefleksyvne
Symetrychne
Ne AntiSymetrychne
Tranzytyvne
Ne AntiTranzytyvne

```

```

Enter the number of arrays elements A and B
4
Enter elements of the array A
5 0 -4 -9
Enter elements of the array B
-3 5 1 6
0 1 1 1
0 1 0 1
0 0 0 0
0 0 0 0

Ne Refleksyvne
Ne AntiRefleksyvne
Ne Symetrychne
AntiSymetrychne
Ne Tranzytyvne
Ne AntiTranzytyvne

```

Висновок:

Набув практичних вмінь та навичок при побудові матриць бінарних відношень та визначені їх типів.