

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”
ІНСТИТУТ КОМП’ЮТЕРНИХ НАУК ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра систем штучного інтелекту

Лабораторна робота №3
із дисципліни
«Дискретна математика»

Виконав:
студент групи КН-113
Калапунь.Н.Т.

Викладач:
Мельникова Н.І.

Львів – 2019 р.

Тема роботи: Побудова матриці бінарного відношення.

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

Варіант – 10

Завдання 1

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

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

Завдання 2

Знайти матрицю відношення $R \subset 2^A \times 2^B$:

$R = \{(x, y) \mid x \subset A \text{ \& } y \subset B \text{ \& } y \subset x\}$, де $A = \{2, 4\}$, $B = \{1, 2, 4\}$.

| | $\{\emptyset\}$ | $\{1\}$ | $\{2\}$ | $\{4\}$ | $\{1, 2\}$ | $\{1, 4\}$ | $\{2, 4\}$ | $\{1, 2, 4\}$ |
|-----------------|-----------------|---------|---------|---------|------------|------------|------------|---------------|
| $\{\emptyset\}$ | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $\{2\}$ | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| $\{4\}$ | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| $\{2, 4\}$ | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 |

Завдання 3

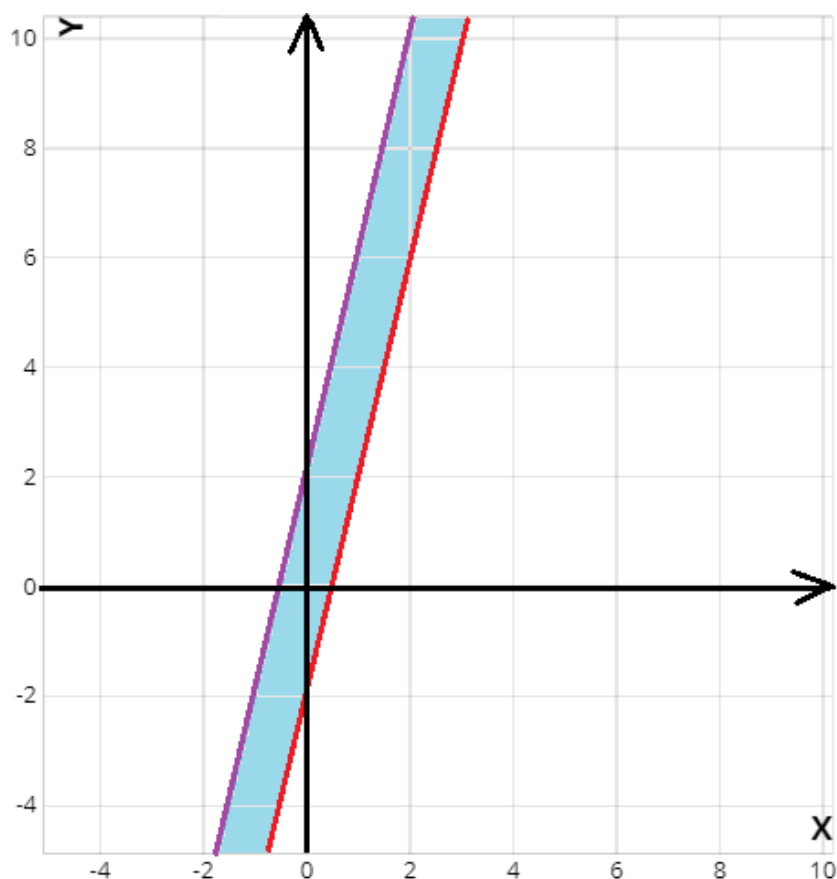
Зобразити відношення графічно: $\alpha = \{(x, y) \mid (x, y) \in \mathbb{R}^2 \text{ \& } |y - 4x| < 2\}$, де

\mathbb{R} - множина дійсних чисел.

Використовуючи означення модуля створимо систему рівнянь:

$$\begin{cases} y - 4x < 2 ; \\ y - 4x > -2 ; \end{cases} \quad \begin{cases} y < 2 + 4x ; \\ y > 4x - 2 ; \end{cases}$$

Побудуємо графік цих прямих і визначимо множину розв'язків, які задовольняють цю систему:



Завдання 4

Маємо бінарне відношення $R \subset A \times A$, де $A = \{a, b, c, d, e\}$, яке задане своєю матрицею:

$$A(R) = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \end{pmatrix}$$

Перевірити чи є дане відношення рефлексивним, симетричним, транзитивним, антисиметричним?

- **Рефлексивна**, бо всі елементи головної діагоналі дорівнюють 1.
- **Не симетрична**, бо $b_{23}=1$, а $b_{32}=0$ і т.д.
- **Антисиметрична**, бо не має жодної пари одиниць, які знаходяться на симетричних місцях по відношенню до головної діагоналі.
- **Транзитивна**, бо $b_{22}=b_{23}=b_{33}=1$; $b_{23}=b_{33}=b_{33}=1$; $b_{24}=b_{44}=b_{44}=1$; $b_{45}=b_{53}=b_{43}=1$; $b_{53}=b_{33}=b_{53}=1$; $b_{55}=b_{53}=b_{53}=1$; $b_{43}=b_{33}=b_{43}=1$;

Завдання 5

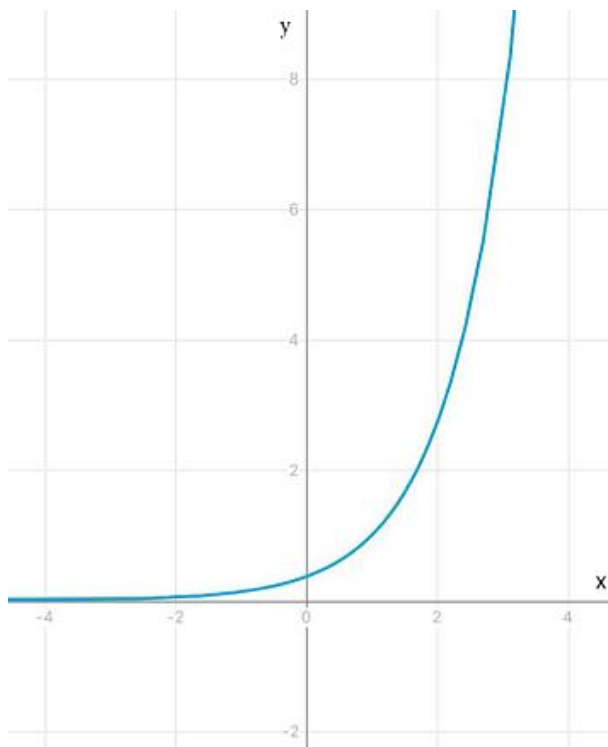
Визначити множину (якщо це можливо), на якій дане відношення є: а) функціональним; б) бієктивним:

$$\alpha = \{(x, y) \mid (x, y) \in \mathbb{R}^2 \ \& \ y = e^{x-1}\}$$

Функціональне, коли одному x відповідає одне y . Оскільки це степенева функція, то таке відношення є завжди функціональним і бієктивним (кожному значенню x відповідає тільки одне значення y і кожному значенню y відповідає тільки одне значення x) на всій області визначення, отже :

$$\alpha = \{(x, y) \mid (x, y) \in \mathbb{R}^2 \ \& \ y = e^{x-1} \ \& \ y > 0\}$$

Зобразимо графік:



Додаток 2

Написати програму, яка знаходить матрицю бінарного відношення $\rho \subseteq A \times B$, 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.

$$\rho = \{(a, b) \mid a \in A \ \& \ b \in B \ \& \ (2b + 1) > a\};$$

Програмна реалізація:

```
#include<iostream>

using namespace std;
int SIZE1=1; //створення змінних та масивів.
int SIZE2=1;

void FillArray(int* const arr, const int size) //функція заповнення масиву
{
    for (int i = 0; i < size; i++)
    {
        cout << "\t";
        cin>>arr[i];
    }
}

void PrintArray( const int* const arr, const int size) //функція виведення масиву
{
    for (int i = 0; i < size; i++)
    {
        cout << arr[i] << " ";
    }
    cout << endl;
}

void Check(int* arr, int size) //функція перевірки елементів масиву на унікальність
{
    bool AlreadyThere;

    for (int i = 0; i < size; i++)
    {
        AlreadyThere = false;
        for (int j = 0; j < size; j++)
        {
            if (i == j)
                continue;
            if (arr[i] == arr[j])
            {
                cout << "Такий елемент \"" << arr[i] << "\" у множині вже існує,
введіть новий: ";
                cin >> arr[i];
                break;
            }
        }
    }
}
```

```

    }
}

void PrintMatrix(int** arr, int ROWS, int COLLS) // функція виведення матриці на екран
{
    for (int i = 0; i < ROWS; i++)
    {
        for (int j = 0; j < COLLS; j++)
        {
            cout << "\t" << arr[i][j];
        }
        cout << endl << endl << endl;
    }
}

int main()
{
    setlocale(LC_ALL, "Ukrainian");

    cout << "Введіть кількість елементів множини 1: ";
    cin >> SIZE1;
    int* mas1 = new int[SIZE1]; //Створення і виведення матриць, перевірка
    елементів на унікальність.
    cout << "Введіть елементи множини 1\n";
    FillArray(mas1, SIZE1);
    Check(mas1, SIZE1);

    cout << "\nВведіть кількість елементів множини 2: ";
    cin >> SIZE2;
    int* mas2 = new int[SIZE2];
    cout << "Введіть елементи множини 2\n";
    FillArray(mas2, SIZE2);
    Check(mas2, SIZE2);
    cout << "\n\nМножина 1: ";
    PrintArray(mas1, SIZE1);
    cout << "\n\nМножина 2: ";
    PrintArray(mas2, SIZE2);

    int** matrix = new int* [SIZE2]; //створюємо динамічний двовірний масив
    for (int i = 0; i < SIZE2; i++)
    {
        matrix[i] = new int[SIZE1];
    }

    for (int i = 0; i < SIZE2; i++) //створення матриці бінарних відношень
    {
        for (int j = 0; j < SIZE1; j++)
        {
            if (mas2[i] * 2 + 1 > mas1[j])
                matrix[i][j] = 1;
            else
                matrix[i][j] = 0;
        }
    }
    cout << "\n-----" << endl <<
endl << endl;
    PrintMatrix(matrix, SIZE2, SIZE1);
    cout << "-----" << endl << endl
<< endl;

    bool reflexivity;
    for (int i = 0; i < SIZE1; i++) //перевірка на рефлексивність
    {
        reflexivity = matrix[0][0];
        if (matrix[i][i] != reflexivity)
        {
            cout << "Матриця не рефлексивна.\n\n";
        }
    }
}

```

```

        goto link_ref;    //якщо виконується ця умова, переходимо за лінком.
        break;
    }
}
if (reflexivity)
    cout << "Матриця рефлексивна.\n\n";
else
    cout << "Матриця антирефлексивна.\n\n";
link_ref:

for (int i = 0; i < SIZE1; i++) //перевірка матриці на симетричність
{
    for (int j = 0; j < SIZE1; j++)
    {
        if (matrix[i][j] != matrix[j][i] && i != j)
        {
            goto link_sym;
        }
        else if (i == j && j == SIZE1-1)
            cout << "Матриця симетрична.\n\n";
    }
}
link_sym:
for (int i = 0; i < SIZE1; i++) //перевірка матриці на асиметричність і
антисиметричність
{
    for (int j = 0; j < SIZE1; j++)
    {
        if (matrix[i][j] == matrix[j][i] && i != j)
        {
            cout << "Матриця асиметрична.\n\n";
            goto link_sym_end;
        }
        else if (i == j && j == SIZE1 - 1)
            cout << "Матриця антисиметрична.\n\n";
    }
}
link_sym_end:

bool transitivity = true;
for (int i = 0; i < SIZE1; i++) //перевірка на транзитивність
{
    for (int j = 0; j < SIZE1; j++)
    {
        for (int k = 0; k < SIZE1; k++)
        {
            if (i == j && i == k && j == k)
                continue;
            else if (matrix[i][j] == 1 && matrix[j][k] == 1 && matrix[i][k] != 1)
            {
                transitivity = false;
                goto link;
            }
        }
    }
}
link:
if (transitivity)
    cout << "Матриця транзитивна.\n\n";
else
    cout << "Матриця не транзитивна.\n\n";

delete[] mas1; // Видалення динамічних масивів.
delete[] mas2;
for (int i = 0; i < SIZE2; i++)
{
    delete[] matrix[i];
}

```

}

Результати виконання програми:

```
Введіть кількість елементів множини 1: 4
Введіть елементи множини 1
  3
 67
  8
  0

Введіть кількість елементів множини 2: 4
Введіть елементи множини 2
 29
  2
  5
  2

Такий елемент "2" у множині вже існує, введіть новий: 7

Множина 1: 3 67 8 0

Множина 2: 29 7 5 2

-----

      1      0      1      1

      1      0      1      1

      1      0      1      1

      1      0      0      1

-----

Матриця не рефлексивна.
Матриця асиметрична.
Матриця не транзитивна.
```

```
Введіть кількість елементів множини 1: 3
Введіть елементи множини 1
  3
  6
  6

Такий елемент "6" у множині вже існує, введіть новий: 8

Введіть кількість елементів множини 2: 3
Введіть елементи множини 2
  3
  2
  9

Множина 1: 3 8 6

Множина 2: 3 2 9

-----

      1      0      1

      1      0      0

      1      1      1

-----

Матриця не рефлексивна.
Матриця асиметрична.
Матриця не транзитивна.
```

```
Введіть кількість елементів множини 1: 3
Введіть елементи множини 1
  9
  7
  6

Введіть кількість елементів множини 2: 3
Введіть елементи множини 2
  1
  1
  2

Такий елемент "1" у множині вже існує, введіть новий: 5

Множина 1: 9 7 6

Множина 2: 5 1 2

-----

      1      1      1

      0      0      0

      0      0      0

-----

Матриця не рефлексивна.
Матриця асиметрична.
Матриця транзитивна.
```

```
Введіть кількість елементів множини 1: 4
Введіть елементи множини 1
  2
  4
  5
 56

Введіть кількість елементів множини 2: 3
Введіть елементи множини 2
  0
 76
  3

Множина 1: 2 4 5 56

Множина 2: 0 76 3

-----

      0      0      0      0

      1      1      1      1

      1      1      1      0

-----
```


Висновок: На цій лабораторній роботі я набув практичних вмінь та навичок при побудові матриць бінарних відношень та визначені їх типів.