

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНОМУ УНІВЕРСИТЕТІ  
“ЛЬВІВСЬКА ПОЛІТЕХНІКА”**

Кафедра систем штучного інтелекту

**Звіт**

**Лабораторна робота №3**

з дисципліни «Дискретна математика»

Виконав: студент групи КН-113

Власюк Олександр

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.

## Постановка завдання : 5-Варіант

1. Чи є вірною рівність  $(A \times B) \cap (C \times D) = (A \times D) \cap (C \times B)$  ?
2. Знайти матрицю відношення  $R \subset M \times M : R = \{(x, y) \mid x \in M \text{ \& } y \in M \text{ \& } y < x + 2\}$ , де  $M = \{x \mid x \in \mathbb{Z} \text{ \& } x \leq 1\}$ ,  $\mathbb{Z}$  - множина цілих чисел.
3. Зобразити відношення графічно:  $\{(x, y) \mid x, y \in \mathbb{R} \text{ \& } x + y = 2\}$ , де  $\mathbb{R}$  - множина дійсних чисел.
4. Навести приклад бінарного відношення  $R \subset A \times A$ , де  $A = \{a, b, c, d, e\}$ , яке є рефлексивне, несиметричне, транзитивне, та побудувати його матрицю.
5. Визначити множину (якщо це можливо), на якій дане відношення є: а) функціональним; б) бієктивним:  $\{(x, y) \mid x, y \in \mathbb{R} \text{ \& } xy = 2\}$ .

### Розв'язки:

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

$$(a, b) \cap (c, d) = (a, d) \cap (c, b)$$

Ліва частина  $(Z) = \{(z_1, z_2) : (z_1 \in A \wedge z_1 \in C) \wedge (z_2 \in B \wedge z_2 \in D)\}$

Права частина  $(Y) = \{(y_1, y_2) : (y_1 \in A \wedge y_1 \in C) \wedge (y_2 \in D \wedge y_2 \in B)\}$

Для висловлювання використовуємо закон комутативності

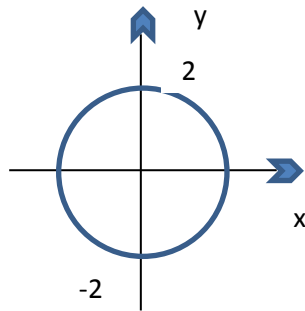
$$(Y) = \{(y_1, y_2) : (y_1 \in A \wedge y_1 \in C) \wedge (y_2 \in B \wedge y_2 \in D)\}$$

Правила для використання  $Z, Y$  однакові. Отже ці множини рівні.

2. Знайти матрицю відношення  $R \subset M \times M : R = \{(x, y) \mid x \in M \text{ \& } y \in M \text{ \& } y < x + 2\}$ , де  $M = \{x \mid x \in \mathbb{Z} \text{ \& } |x| \leq 1\}$ ,  $\mathbb{Z}$  - множина цілих чисел.

	$\{\emptyset\}$	$\{-1\}$	$\{0\}$	$\{1\}$	$\{-1; 0\}$	$\{-1; 1\}$	$\{0; 1\}$	$\{-1; 0; 1\}$
-1	1	0	0	0	0	0	0	0
0	1	1	1	1	0	0	0	0
1	1	1	1	1	1	1	1	0

3. Зобразити відношення графічно:  $\alpha = \{(x, y) \mid (x, y) \in \mathbb{R}^2 \text{ \& } (x + y)^2 = 4\}$ , де  $\mathbb{R}$  - множина дійсних чисел.



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

Рефлексивна –  $aRa$

Несиметрична –  $aRb \rightarrow \neg bRa$

Транзитивна –  $aRb \wedge bRc \rightarrow aRc$

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

- 5.** Визначити множину (якщо це можливо), на якій дане відношення є: а) функціональним; б) бієктивним:  $\alpha = \{(x, y) \mid (x, y) \in \mathbb{R}^2 \ \& \ xy = 2\}$ .

Відношення є функціональним, бо кожне  $x$  відповідає максимум одному  $y$ .

Відношення є бієктивним, бо певному значенню  $y$  відповідає певне значення  $x$  і навпаки.

## 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.

- 6.**  $\rho = \{(a, b) \mid a \in A \ \& \ b \in B \ \& \ (a + 2) > 3b\}$ ;

Код програмної реалізації:

```

1  #include <iostream>
2  #include <string>
3  #include <math.h>
4  using namespace std;
5  bool IsMatrixReflexive(int **matrix);
6  bool IsMatrixAntiReflexive(int **matrix);
7  bool IsMatrixSymetric(int **matrix);
8  bool IsMatrixAntiSymetric(int **matrix);
9  bool IsMatrixTransactive(int **matrix);
10 bool IsMatrixAntiTransactive(int **matrix);
11 void BuildMatrix(int*, int*, int**);
12 int main()
13 {
14     int array1[5] = {};
15     int array2[5] = {};
16     int ** matrix = new int *[5];
17     for (int i = 0; i < 5; i++)
18     {
19         matrix[i] = new int[5];
20     }
21
22     cout << "Enter 5 elements of array 1:" << "\n";
23
24     for (int i = 0; i < 5; i++)
25     {
26
27         cin >> array1[i];
28     }
29
30
31     cout << "Enter 5 elements of array 2:" << "\n";
32
33
34     for (int j = 0; j < 5; j++)
35     {
36
37         cin >> array2[j];
38     }
39
40     BuildMatrix(array1, array2, matrix);
41 }
42
43 void BuildMatrix(int *array1, int *array2, int **matrix)
44 {
45     for (int i = 0; i < 5; i++)
46     {
47         for (int j = 0; j < 5; j++)
48         {
49             if ((array2[i] + array1[j]) > 2)
50             {
51                 matrix[i][j] = 1;
52                 cout << "1\t";
53             }
54             else
55             {
56                 matrix[i][j] = 0;
57                 cout << "0\t";
58             }
59         }
60         cout << "\n";
61     }
62 }

```

---

```

63 bool isMatrixReflexive = IsMatrixReflexive(matrix);
64 bool isMatrixAntiReflexive = IsMatrixAntiReflexive(matrix);
65 string reflectionResult = "";
66 if (isMatrixReflexive)
67 {
68     reflectionResult = "reflexive";
69 }
70 else if (IsMatrixAntiReflexive)
71 {
72     reflectionResult = "antireflexive";
73 }
74
75
76 bool isMatrixSymetric = IsMatrixSymetric(matrix);
77 bool isMatrixAntiSymetric = IsMatrixAntiSymetric(matrix);
78 string symmetricResult = "";
79 if (isMatrixSymetric)
80 {
81     symmetricResult = "symetric";
82 }
83 else if (IsMatrixAntiSymetric)
84 {
85     symmetricResult = "antisymetric";
86 }
87
88
89 bool isMatrixTransactive = IsMatrixTransactive(matrix);
90 bool isMatrixAntiTransactive = IsMatrixAntiTransactive(matrix);
91 string transactiveResult = "";
92 if (isMatrixTransactive)
93 {
94     transactiveResult = "transitive";
95 }

```

```

94     transactiveResult = "transitive";
95 }
96 else if (isMatrixAntiTransactive)
97 {
98     transactiveResult = "antitransitive";
99 }
100
101
102 cout << "This matrix is:" << "\n";
103 cout << reflectionResult << "\n";
104 cout << symmetricResult << "\n";
105 cout << transactiveResult << endl;
106 }
107
108 bool IsMatrixReflexive(int **matrix)
109 {
110     for (int i = 0; i < 5; i++)
111     {
112         if (matrix[i][i] != 1)
113         {
114             return false;
115         }
116     }
117     return true;
118 }
119
120 bool IsMatrixAntiReflexive(int **matrix)
121 {
122     for (int i = 0; i < 5; i++)
123     {
124         if (matrix[i][i] != 0)
125         {

```

```

124         if (matrix[i][i] != 0)
125         {
126             return false;
127         }
128     }
129     return true;
130 }
131
132 bool IsMatrixSymetric(int **matrix)
133 {
134     for (int i = 0; i < 5; i++)
135     {
136         for (int j = 0; j < 5; j++)
137         {
138             if (matrix[i][j] != matrix[j][i])
139             {
140                 return false;
141             }
142         }
143     }
144     return true;
145 }
146
147 bool IsMatrixAntiSymetric(int **matrix)
148 {
149     for (int i = 0; i < 5; i++)
150     {
151         for (int j = 0; j < 5; j++)
152         {
153             if (matrix[i][j] == matrix[j][i])
154             {
155                 return false;

```

```

155         return false;
156     }
157 }
158 }
159 return true;
160 }
161
162
163 bool IsMatrixTransactive(int **matrix)
164 {
165     for (int i = 0; i < 5; i++)
166     {
167         for (int k = 0; k < 5; k++)
168         {
169             if (matrix[i][k] == 1 && i != k)
170             {
171                 for (int j = 0; j < 5; j++)
172                 {
173                     if (matrix[k][j] == 1 && matrix[i][j] != 1)
174                     {
175                         return false;
176                     }
177                 }
178             }
179         }
180     }
181     return true;
182 }
183
184 bool IsMatrixAntiTransactive(int **matrix)
185 {
186

```

```

186 {
187     for (int i = 0; i < 5; i++)
188     {
189         for (int k = 0; k < 5; k++)
190         {
191             if (matrix[i][k] == 1 && i != k)
192             {
193                 for (int j = 0; j < 5; j++)
194                 {
195                     if (matrix[k][j] == 1 && matrix[i][j] != 0)
196                     {
197                         return false;
198                     }
199                 }
200             }
201         }
202     }
203     return true;
204 }
205 }
206 }
207 }
208

```

## Результатаи виконанаої програми

```
Enter 5 elements of array 1:
1
2
3
4
5
Enter 5 elements of array 2:
3
4
5
6
6
1      1      1      1
1      1      1      1      1
1      1      1      1      1
1      1      1      1      1
1      1      1      1      1
This matrix is:
reflexive
symetric
transitive
Process returned 0 (0x0)   execution time : 11.689 s
Press any key to continue.
```