**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**НАЦІОНАЛЬНОМУ УНІВЕРСИТЕТІ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”**

**Кафедра систем штучного інтелекту**

**Лабораторна робота № 3**

з дисципліни

«Дискретна математика»

**Виконав:**

**Дребот Владислав Олегович**

студент групи КН-112

**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 4**

1. Чи є вірною рівність: (A∩B) ×C = (A×C) ∩ (B×C) ?

Нехай (х,у) є (А∩В)×С→ (х,у) є (А×С) ∩(В×С)

Закон дистрибутивності:

Нехай (х,у) є (А×С) ∩(В×С) →(х,у) є (А∩В)×С зворотній закон дистрибутивності

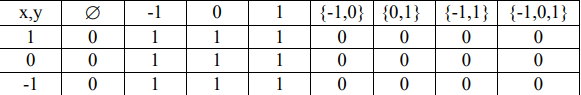
Отже (A∩B) ×C = (A×C) ∩ (B×C).

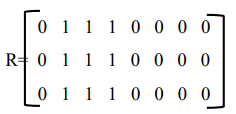
1. Знайти матрицю відношення R ⊂ M ×2 M :

R = {(x, y) x ∈ M & y ⊂ M & y = x}, де M = {x x ∈ Z & x ≤1},

Z - множина цілих чисел.

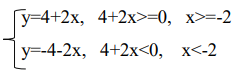
x є М={-1,0,1}

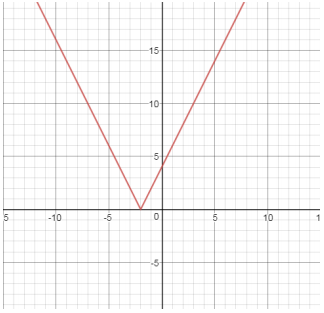




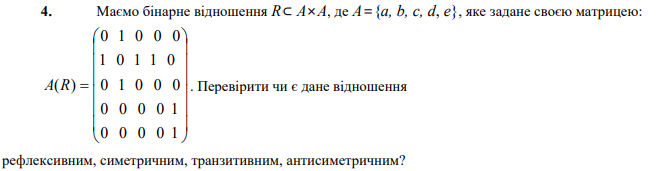
1. Зобразити відношення графічно:

α{(x, y)| (x, y) є R 2 & |4 + 2x| = y }, де R – множина дійсних чисел.





D(α)=(-∞;+∞) E(α)=[0;+∞)

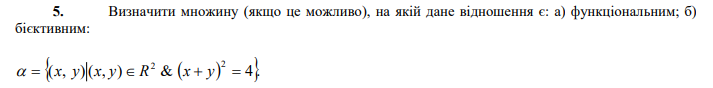


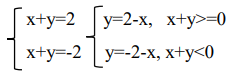
1) Не рефлексивне, не антирефлексивне, головна діагональ не складається не з одиниць, не з нулів.

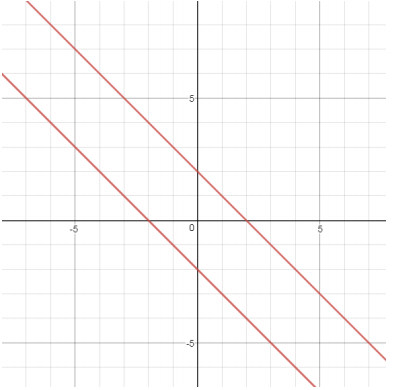
2) Не симетричне, так наприклад: а24!=a42 або а45!=а54.

3) Транзитивне а45=а55=1, то а45=1.

4) Не антисиметричне так як а12=а21.







Функціональне на проміжку (∅), а бієктивне (∅).

**Завдання №2. Написати програму, яка знаходить матрицю бінарного відношення ρ⊂ A× 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:**

1. ρ = {(a, b) a ∈ A&b∈ B &(2a +1) < b};

#include <iostream>

#include <clocale>

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 < Pot1; i++) { //

cin >> arr2[i];

} //

int\*\* arr3 = new int\*[Pot1]; //динамічний масив вказівників(рядки)

//

for (int i = 0; i < Pot1; i++) { //

arr3[i] = new int [Pot2]; //створюємо стовпці

} //

for (int i = 0; i < Pot1; i++) {

for (int j = 0; j < Pot2; j++) {

if (2\*arr1[i]+1<arr2[j]) {

arr3[i][j] = 1; //бінарний масив

}

else {

arr3[i][j] = 0;

}

}

}

delete [] arr1;

delete [] arr2;

for (int i = 0; i < Pot1; i++) {

for (int j = 0; j < Pot2; j++) { //вивожу бінарниий масив

cout << arr3[i][j]<<"\t";

}

cout << endl;

}

int check = 0;

for (int i = 0; i < Pot1; i++) {

if (arr3[i][i] == 1) { //рефлексивність

check++; //якщо на гол діагоналі 1 то check++

}

}

if (check == Pot1) { //

cout << "Reflexivna "; //якщо вся гол діагональ 1 то матриця транзитивна

} //

else if (check == 0) { //

cout << "Antireflexivna "; //якщо вся гол діагональ 0 то матриця антитранзитивна

}

else {

cout << "Nereflexivna "; //якщо є 1 і 0 то матриця нетранзитивна

}

int check0=0;

int check1=0;

for (int i = 0; i < Pot1; i++) { //перевірка на симетричність матриці

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) { //перевірка на транзитивність

for (int z = 0; z < Pot1; z++) { //якщо arr3[i][j]=1

if (arr3[j][z]==1) { //починається цикл

if (arr3[i][z] == 1) { //якщо знаходиться ще 1 одиниця від j

tranz = 1; //шукаємо 3 одиницю

} //якщо 3 одиня є тоді транзитивність є

else { //в іншому випадку немає

tranz1 = 0; //

} //

} //

} //

} //

} //

} //

if (tranz\*tranz1 == 1) { //

cout << "Tranzitivna "; //Якщо є транзитивність

} //виводимо що матриця транзитивна

else { //в іншому випадку перевіряємо на анти транзитивність

int antitranz = 0;

for (int i = 0; i < Pot1; i++) { //

for (int j = 0; j < Pot2; j++) { //

if (arr3[i][j] == 1) { //перевірка на антитранзитивність

for (int z = 0; z < Pot1; z++) { //якщо arr3[i][j]=1

if (arr3[j][z] == 1) { //починається цикл

if (arr3[i][z] == 0) { //якщо знаходиться ще 1 одиниця від j

antitranz = 1;

}

else {

antitranz = 0;

}

}

}

}

}

}

if (antitranz == 1) {

cout << "Antitranzitivna ";

} //

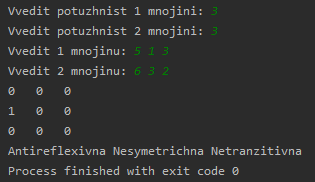
else {

cout << "Netranzitivna ";

}

}

}

Результат: