Міністерство освіти та науки України

Львівський національний університет імені Івана Франка

Факультет електроніки та

Комп’ютерних технологій

**Звіт**

Про виконання лабораторної роботи №2

“Бінарні відношення та способи їх задання”

Виконав:

Студент групи ФеІ-44

Сапанюк М.І.

Перевірив:

Мостова М.Р.

Львів 2022

**Мета:**

Засвоїти способи задання бінарного відношення (матричний, через графи). Знаходити верхнеі та нижні перетини бінарного відношення. Здійснювати операції перетину, об'єднання, композиції, різниці, симетричної різниці над однорідними бінарними відношеннями.

**Хід роботи:**

Завдання1: для наперед заданих викладачем множин-носіїв бінарного відношення, написати програму, яка реалізує такі функції: ввід матриці-носія, ввід бінарного відношення, матричне виведення результату.

Текст завдання та теоретичні відомості дивитись у вкладеному файті. Множини-носії ст.8 (пункт 2, 3) вкладеного файлу.

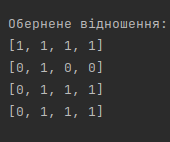
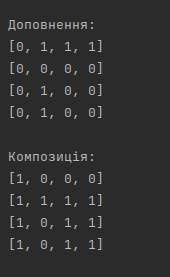
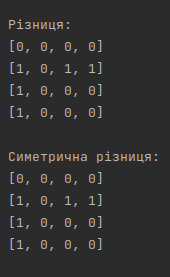
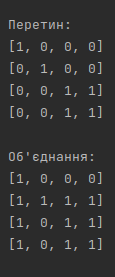
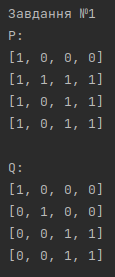
Завдання 2.На множині А = {1,2,3,…,20} задане бінарне відношення менше рівне. Знайти R+(N), R-(N). N – Ваш номер в списку студентів підгрупи згідно алфавіту. Список у Старост. Написати програму, яка знаходить верхні та нижні перетини.

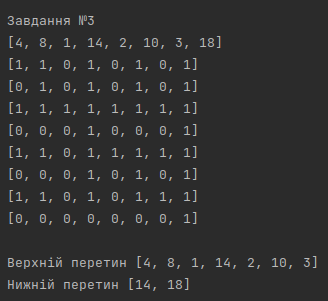
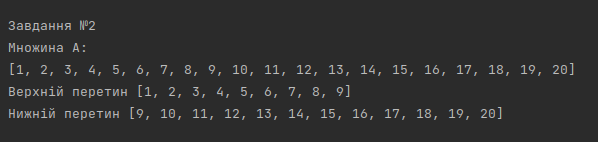
Завдання 3. На множині А = {довільні вісім чисел, букв, слів, словлсполучень} задане деяке бінарне відношення (задати його самостійно відповідно до елементів). Знайти верхній та нижній перетини бінарного відношення для елемента, під номером 4 . Написати програму, яка рандомним чином задає матрицю-носія з вісьмома елементами. та обчислює венрхні та нижні перетини.

**Виконання завдання:**

import random  
  
  
def printMatrix(mat):  
 for row in mat:  
 print(row)  
 print()  
  
  
# Завдання №1  
print("Завдання №1")  
  
  
def intersection(mat1, mat2):  
 print("Перетин:")  
 result = [[0 for x in range(len(mat1))] for y in range(len(mat1[0]))]  
 for indexRow in range(len(mat1)):  
 for indexCol in range(len(mat1[indexRow])):  
 result[indexRow][indexCol] = mat1[indexRow][indexCol] & mat2[indexRow][indexCol]  
 return result  
  
  
def union(mat1, mat2):  
 print("Об'єднання:")  
 result = [[0 for x in range(len(mat1))] for y in range(len(mat1[0]))]  
 for indexRow in range(len(mat1)):  
 for indexCol in range(len(mat1[indexRow])):  
 result[indexRow][indexCol] = mat1[indexRow][indexCol] | mat2[indexRow][indexCol]  
 return result  
  
  
def difference(mat1, mat2):  
 print("Різниця:")  
 result = [[0 for x in range(len(mat1))] for y in range(len(mat1[0]))]  
 for indexRow in range(len(mat1)):  
 for indexCol in range(len(mat1[indexRow])):  
 result[indexRow][indexCol] = mat1[indexRow][indexCol] - mat2[indexRow][indexCol]  
 if result[indexRow][indexCol] < 0:  
 result[indexRow][indexCol] = 0  
 return result  
  
  
def symmetricDifference(mat1, mat2):  
 print("Симетрична різниця:")  
 result = [[0 for x in range(len(mat1))] for y in range(len(mat1[0]))]  
 for indexRow in range(len(mat1)):  
 for indexCol in range(len(mat1[indexRow])):  
 result[indexRow][indexCol] = mat1[indexRow][indexCol] ^ mat2[indexRow][indexCol]  
 return result  
  
  
def addition(mat):  
 print("Доповнення:")  
 result = [[0 for x in range(len(mat))] for y in range(len(mat[0]))]  
 for indexRow in range(len(mat)):  
 for indexCol in range(len(mat)):  
 result[indexRow][indexCol] = 1 if mat[indexRow][indexCol] == 0 else 0  
 return result  
  
  
def composition(mat1, mat2):  
 print("Композиція:")  
 result = [[0 for x in range(len(mat1))] for y in range(len(mat1[0]))]  
 for indexRow in range(len(mat1)):  
 for indexCol in range(len(mat1[indexRow])):  
 for i in range(len(mat1[indexRow])):  
 result[indexRow][indexCol] += (mat1[indexRow][i] \* mat2[i][indexCol])  
 if result[indexRow][indexCol] > 0: result[indexRow][indexCol] = 1  
 return result  
  
  
def reverse(mat):  
 print("Обернене відношення:")  
 result = [[0 for x in range(len(mat))] for y in range(len(mat[0]))]  
 for indexRow in range(len(mat)):  
 for indexCol in range(len(mat[indexRow])):  
 result[indexRow][indexCol] = mat[indexCol][indexRow]  
 return result  
  
  
P = [  
 [1, 0, 0, 0],  
 [1, 1, 1, 1],  
 [1, 0, 1, 1],  
 [1, 0, 1, 1],  
]  
  
Q = [  
 [1, 0, 0, 0],  
 [0, 1, 0, 0],  
 [0, 0, 1, 1],  
 [0, 0, 1, 1],  
]  
print("P: ")  
printMatrix(P)  
print("Q: ")  
printMatrix(Q)  
printMatrix(intersection(P, Q))  
printMatrix(union(P, Q))  
printMatrix(difference(P, Q))  
printMatrix(symmetricDifference(P, Q))  
printMatrix(addition(P))  
printMatrix(composition(P, Q))  
printMatrix(reverse(P))  
print()  
  
  
# Завдання №2  
print("Завдання №2")  
  
  
def createMatrix(arr, condition):  
 result = [[0 for x in range(len(arr))] for y in range(len(arr))]  
 if condition == '<=':  
 for indexRow in range(len(arr)):  
 for indexCol in range(len(arr)):  
 if arr[indexRow] <= arr[indexCol]:  
 result[indexRow][indexCol] = 1  
  
 return result  
  
  
def upperCrossing(A, mat, n):  
 n -= 1  
 res = []  
 index = 0  
 for row in mat:  
 if row[n] == 1:  
 res.append(A[index])  
 index += 1  
 print(f"Верхній перетин {res}")  
  
  
def lowerCrossing(A, mat, n):  
 n -= 1  
 res = []  
 index = 0  
 for col in mat[n]:  
 if col == 1:  
 res.append(A[index])  
 index += 1  
 print(f"Нижній перетин {res}")  
  
  
A = list(range(1, 21))  
n = 9  
print("Множина А:")  
print(A)  
mat = createMatrix(A, '<=')  
upperCrossing(A, mat, n)  
lowerCrossing(A, mat, n)  
print()  
print()  
  
# Завдання №3  
print("Завдання №3")  
A = []  
n = 4  
for i in range(8):  
 A.append(random.randrange(1, 20))  
print(A)  
mat = createMatrix(A, '<=')  
printMatrix(mat)  
upperCrossing(A, mat, n)  
lowerCrossing(A, mat, n)

**Результат виконання:**





**Висновок:**

На цій лабораторній роботі я засвоїв способи задання бінарного відношення, а також навчився знаходити верхнеі та нижні перетини бінарного відношення, здійснювати операції перетину, об'єднання, композиції, різниці, симетричної різниці над однорідними бінарними відношеннями.