Міністерство освіти і науки України

Національний технічний університет України

«Київський політехнічний інститут ім. Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра інформаційних систем та технологій

Комп’ютерний практикум №4

на тему «Методи багатокритеріального ПР. ELECTRE»

З дисципліни «Теорія прийняття рішень»

7 варіант

Виконала:

студентка гр. ІС-91

Ковальська А.А.

Перевірила:

Жураковська О.С.

# Постановка задачі

Задано множину з 15 альтернатив, які оцінені за множиною критеріїв K = {ki}, i = 1,…,12. Кожен критерій має ваговий коефіцієнт wi. Дано пару порогових значень індексів узгодження та неузгодження с, d відповідно.

Таблиця оцінок альтернатив А1-А15 за критеріями к1-к12

9 6 5 1 5 10 10 10 3 4 7 9

2 5 7 9 3 1 7 3 10 8 1 10

10 8 4 3 5 10 10 9 6 8 7 8

8 4 4 6 3 9 1 2 10 2 2 1

10 7 9 9 4 4 5 5 7 7 7 2

8 5 8 7 3 1 5 10 10 5 9 1

4 9 6 4 9 1 1 1 1 10 5 2

6 3 2 1 6 3 7 9 6 5 3 7

10 5 10 9 6 8 6 10 10 7 1 10

2 5 3 5 8 7 4 9 4 10 9 4

7 10 5 8 5 9 4 3 1 5 8 3

9 10 4 6 5 4 10 2 3 8 10 4

6 6 2 1 4 5 6 7 5 4 4 5

7 5 8 1 4 1 4 2 3 6 10 4

2 5 1 10 6 5 3 1 8 5 3 8

Вагові коефіцієнти критеріїв к1-к12:

5 2 3 3 8 4 2 6 7 7 1 10

Значення порогів для індексів узгодження та неузгодження c, d

0.802 0.494

# Теоретичні відомості

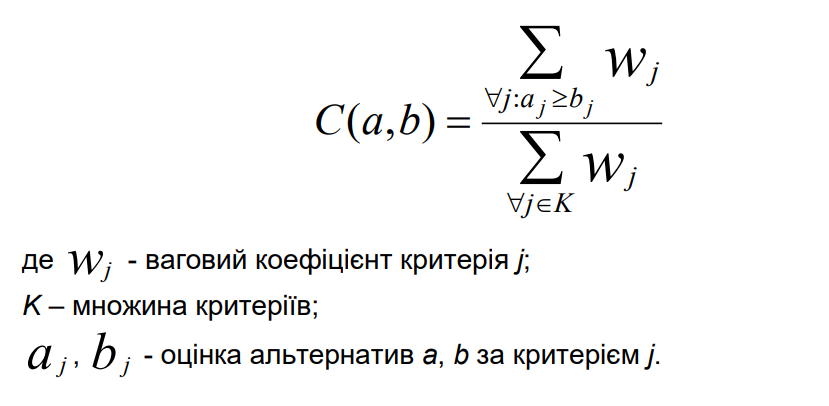
Метод ELECTRE I орієнтований на пошук розв’язку у формі підмножини альтернатив (ядра). В цьому методі для визначення відношення переваги на множині альтернатив вводяться два показники – індекс узгодження та індекс неузгодження. Індекс узгодження C(a, b) визначає ступінь переваги альтернативи a над альтернативою b (або наскільки a «не гірша» за b). Індекс неузгодження D(a, b) визначає ступінь строгої переваги альтернативи b над альтернативою a. На основі індексів узгодження та неузгодження, визначених для всіх пар альтернатив, на множині альтернатив задається відношення переваги, яке є основою для визначення ядра.

# Завдання 1

## Опис етапів вирішення задачі

Визначити підмножину найкращих альтернатив (ядро), використовуючи метод ELECTRE I (для заданих значень порогів індексів узгодження та неузгодження с, d).

Для початку, побудуємо матрицю індексів узгодження C для заданих альтернатив. Для кожної пари альтернатив (a,b) значення індекса узгодження С(a,b) обчислюється за формулою:



де – ваговий коефіцієнт критерія j;

K – множина критеріїв;

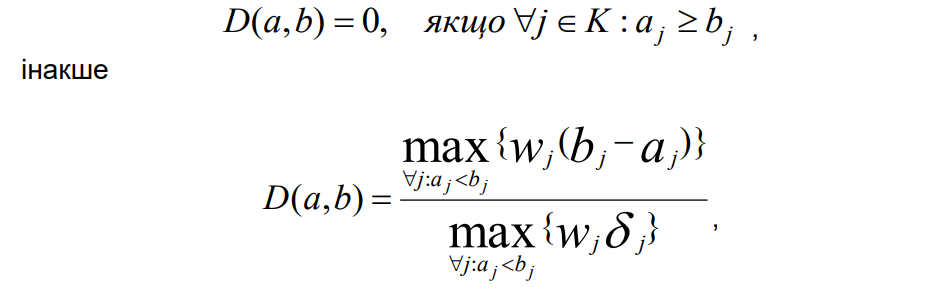
– оцінка альтернатив a, b за критерієм j.

На головній діагоналі індекси узгодженності дорівнюють 1, бо коли пара альтирнатив порівнюється сама з собою, вона рівноцінна по всім критеріям.

Фрагмент коду:

def calc\_concordance(a: int, b: int):  
 numerator = 0  
 for col in range(k\_count):  
 if matrix[a][col] >= matrix[b][col]:  
 numerator += weights[col]  
  
 return round(numerator / weights\_sum, 3)

Далі, будуємо матрицю індексів неузгодження D. Для кожної пари альтернатив (a,b) значення індекса неузгодження D(a,b) обчислюється за формулами:



де – ваговий коефіцієнт критерія j;

– оцінка альтернатив a, b за критерієм j;

– величина діапазону оцінок для критерія j;

– максимальне та мінімальне значення відповідно оцінок за критерієм j.

Фрагмент коду:

def calc\_discordance(a: int, b: int):  
 max\_numerator = 0  
 max\_denumerator = 0  
 for col in range(k\_count):  
 diff = matrix[b][col] - matrix[a][col]  
 if matrix[a][col] < matrix[b][col] and max\_numerator < diff \* weights[col]:  
 max\_numerator = diff \* weights[col]  
 if matrix[a][col] < matrix[b][col] and max\_denumerator < max\_dif[col] \* weights[col]:  
 max\_denumerator = max\_dif[col] \* weights[col]  
  
 if max\_numerator == 0:  
 return 0  
 else:  
 return round(max\_numerator / max\_denumerator, 3)

return round(max\_numerator / max\_denumerator, 3)

Тут використовується допоміжний масив, max\_diff, який містить у собі максимальну різницю за одним критерієм серед усіх альтернатив, його знаходимо за наступним кодом:

max\_dif = []  
  
for col in range(k\_count):  
 arr = []  
 for row in range(a\_count):  
 arr.append(matrix[row][col])  
 max\_dif.append(max(arr) - min(arr))

Побудуємо на множині альтернатив А відношення переваги R на основі отриманих матриць C, D на попередніх кроках, використовуючи співвідношення:



де c, d – порогові значення індексів узгодження та неузгодження відповідно.

Задані порогові значення c та d:

c = 0,802; d = 0,494.

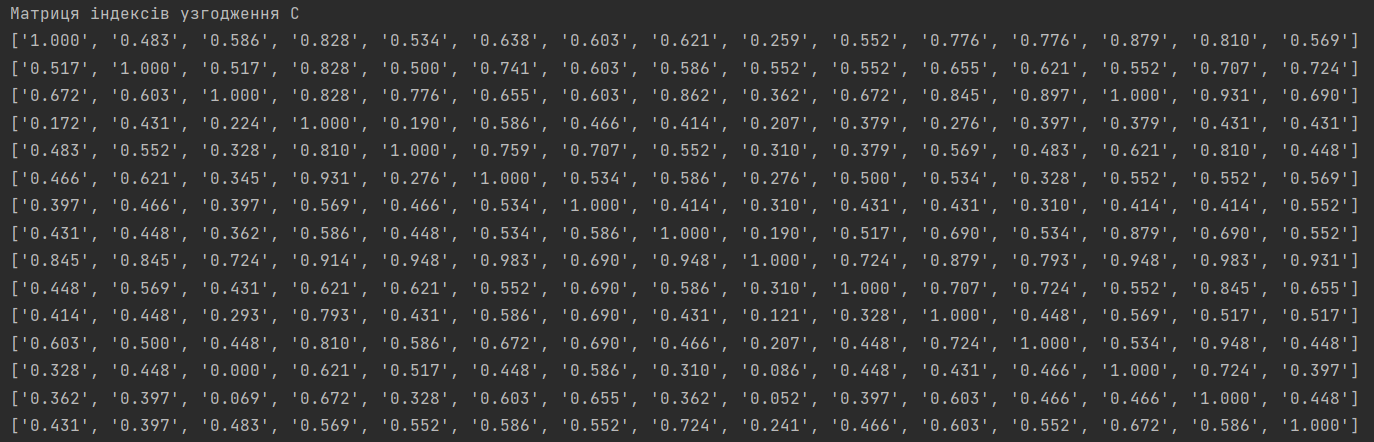
Фрагмент коду:

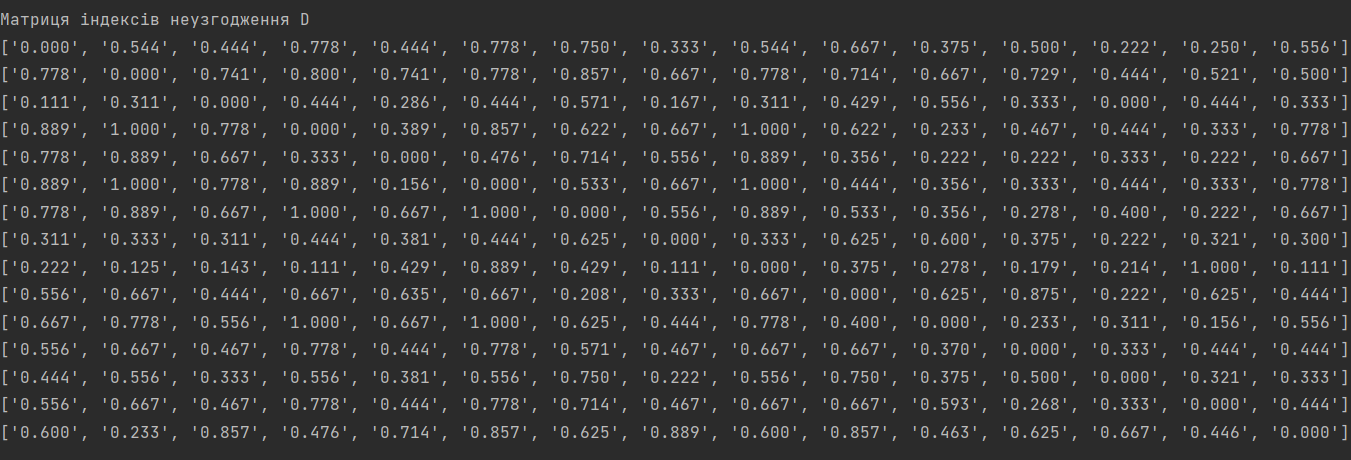
relation = []  
matrix\_C = Concordance.solve()  
matrix\_D = Discordance.solve()  
  
for row in range(a\_count):  
 relation.append([])  
 for col in range(a\_count):  
 if matrix\_C[row][col] >= C and matrix\_D[row][col] <= D:  
 relation[row].append(1)  
 else:  
 relation[row].append(0)

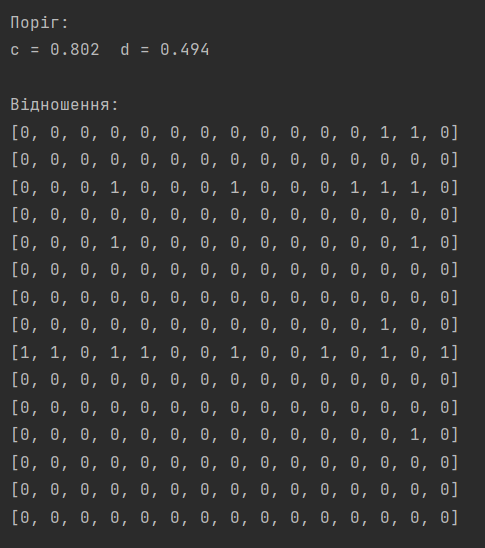
Для перевірки ациклічності та визначення ядра Х\* використаємо фрагмент коду з практикуму №2. Спочатку перевіряємо на ациклічність за допомогою алгоритму DFS, а потім в залежності від результату знаходимо оптимальне рішення.

## Розв’язок та проміжні результати завдання

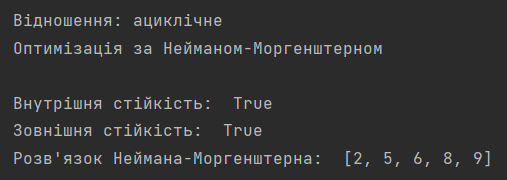
Матриця індексів узгодження С:



Матриця індексів неузгодження D:Результуюче відношення R:



Перевірка на ациклічність та визначення ядра Х\*:



Отже, отримане відношення є ациклічним, внутрішньо та зовнішньо стійким.

Відповідь: Х\* = [3, 6, 7, 9, 10]

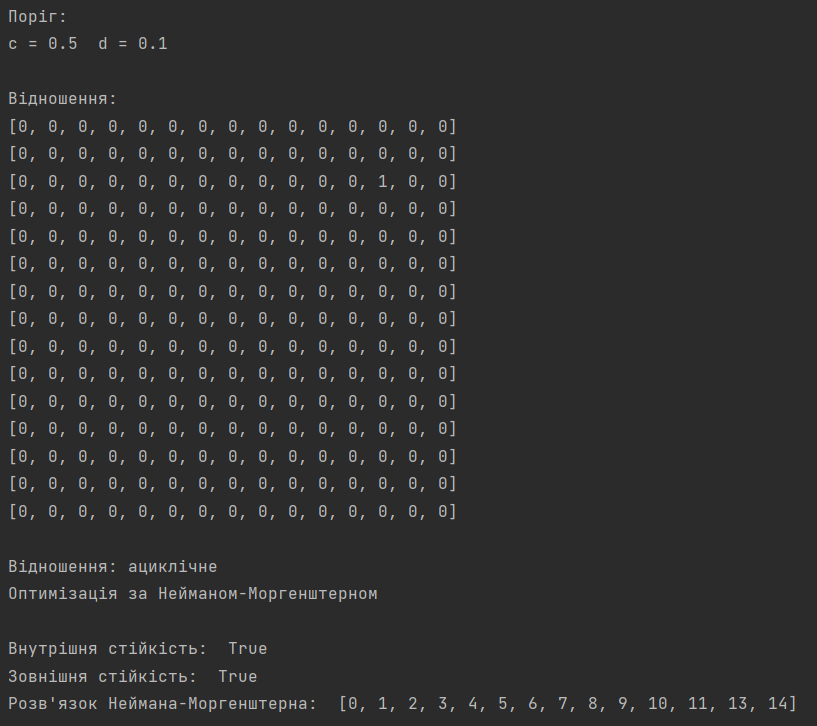
# Завдання 2

Дослідницька задача. Аналіз впливу зміни порогового значення d на склад та розмір ядра.

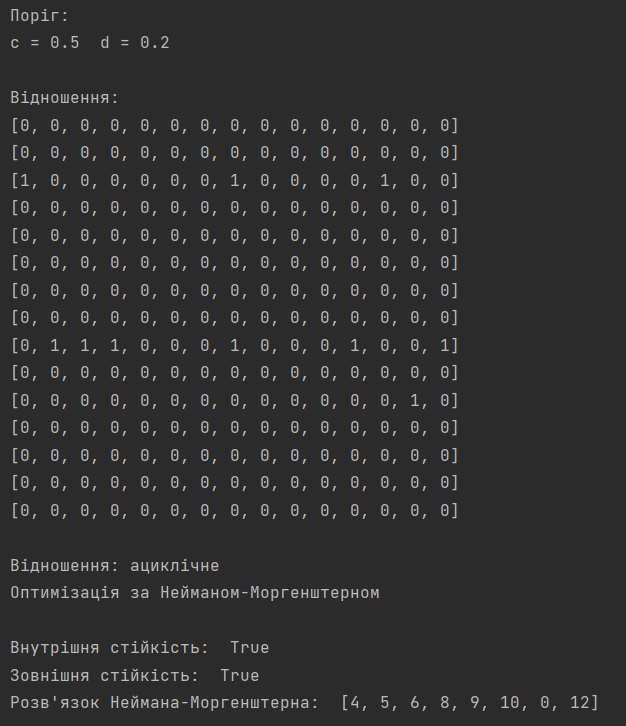
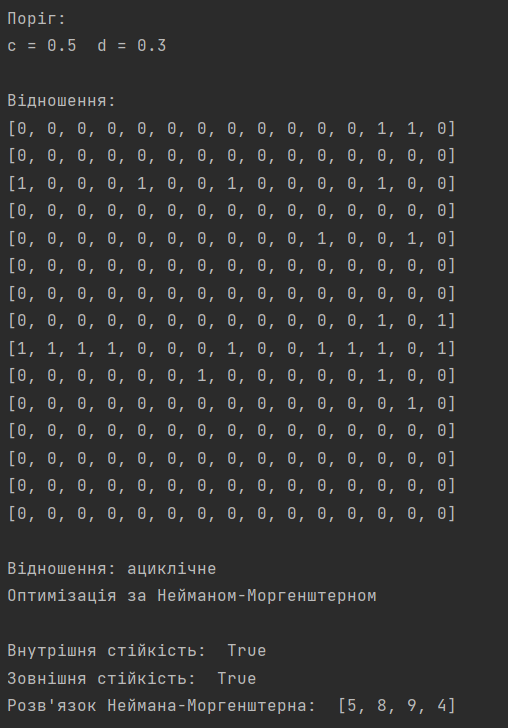
**2.1. Визначення впливу зміни порогового значення d на склад та розмір ядра**

Зафіксуємо порогове значення c = 0.5. Будемо змінювати порогове значення d в інтервалі (0; 0.5) з кроком 0.1.

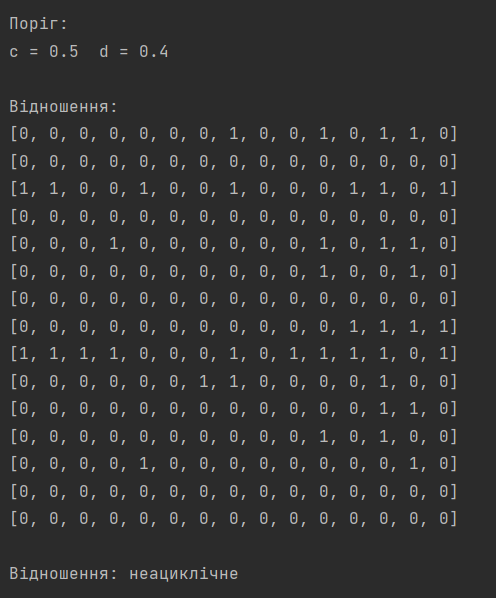
Результати експерименту при c = 0.5, d = 0.1.



Результати експерименту при c = 0.5, d = 0.2 і d = 0.3 відповідно.

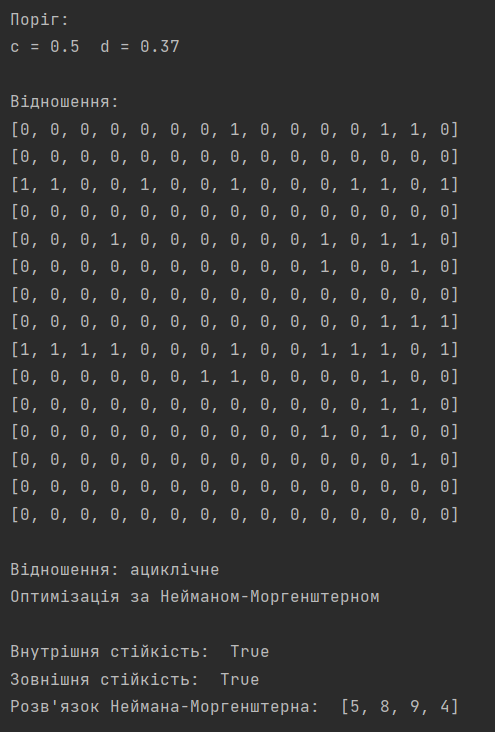
Результати експерименту при c = 0.5, d = 0.4.



Для заданих порогових значень c = 0.5, d = 0.4 отримане відношення є неациклічним. При подальшому змінені порогового значенння d відношення також є неациклічними, оскільки під задану умову підпадають ті ж самі альтернативи, тому продовжувати немає сенсу.

Зазначимо, що при кожній наступній ітерації, можна спострігати, що ядро зменшується і є підмножиною для попереднього відношення.

Для ілюстрації цього встановимо поріг d більше 0.3, та менше 0.4, наприклад 0.37.

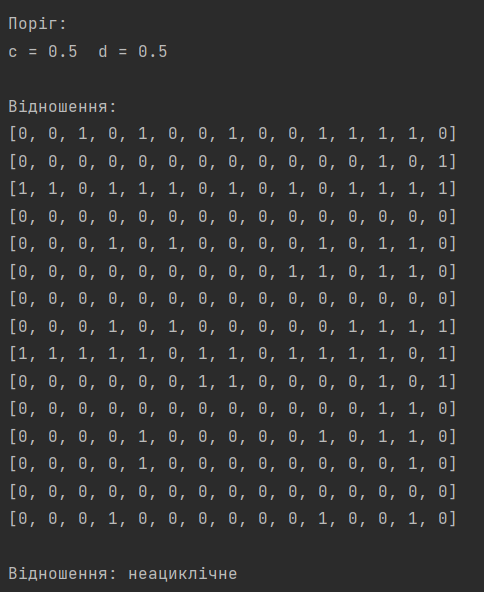


Ядро даного відношення є підмножиною ядра відношення для d = 0.3, що, в свою чергу, є підмножиною ядра відношення для d = 0.2 та, аналогічно, d = 0.1, тобто d0.37 <= d0.3 <= d0.2 <= d0.1.

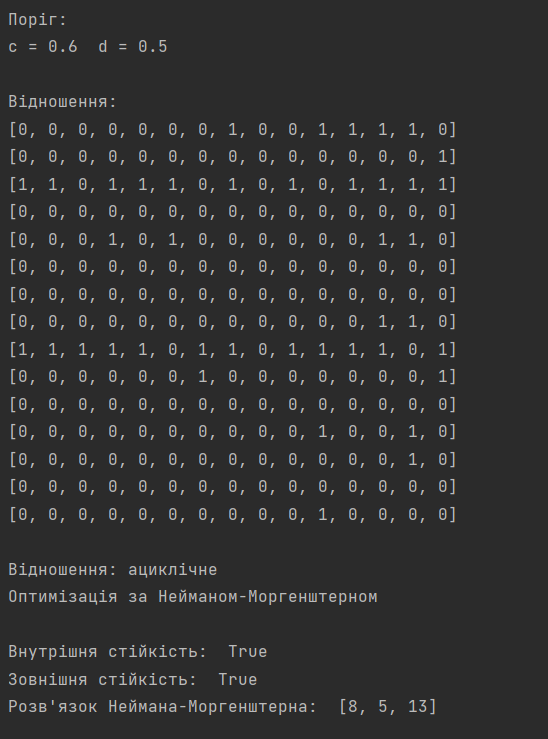
**2.2. Визначення впливу зміни порогового значення c на склад та розмір ядра**

Зафіксуємо значення порогу d = 0.5. Будемо змінювати порогове значення c в інтервалі [0.5; 1] з кроком 0.1.

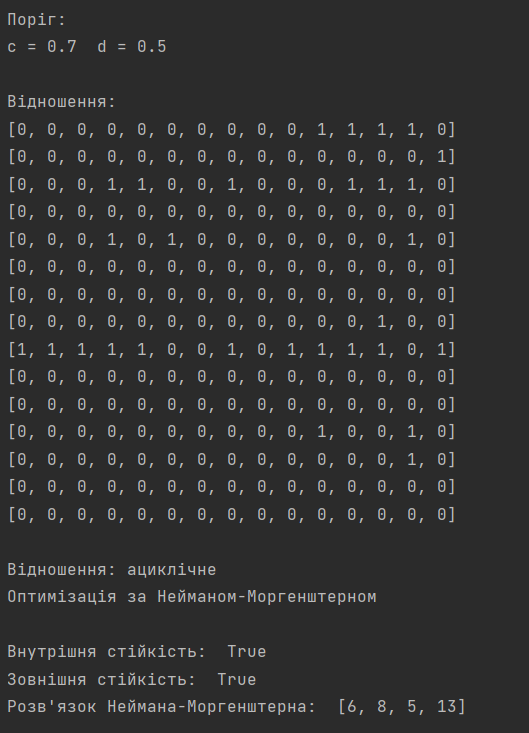
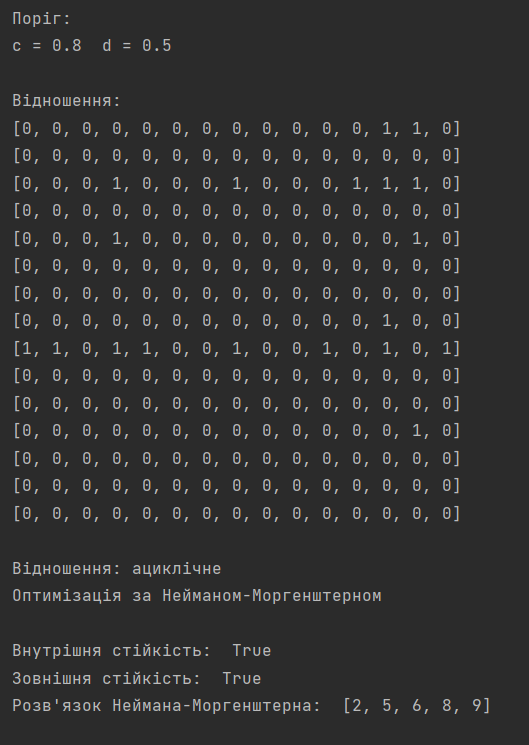
Результати експерименту при c = 0.5, d = 0.5.

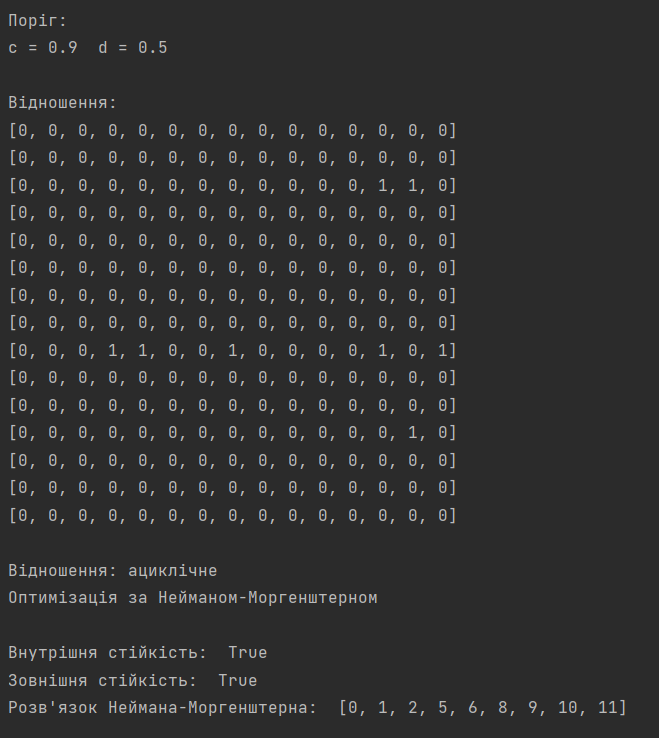


Результати експерименту при c = 0.6, d = 0.5.

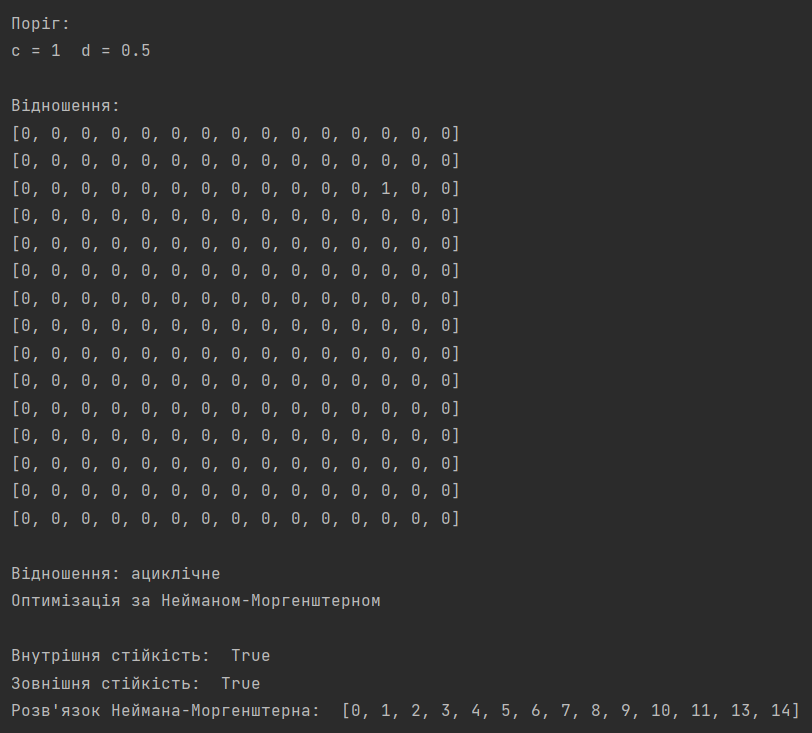


Результати експерименту при c = 0.7, 0.8 та 0.9 відповідно, d = 0.5.



Результати експерименту при c = 1, d = 0.5.



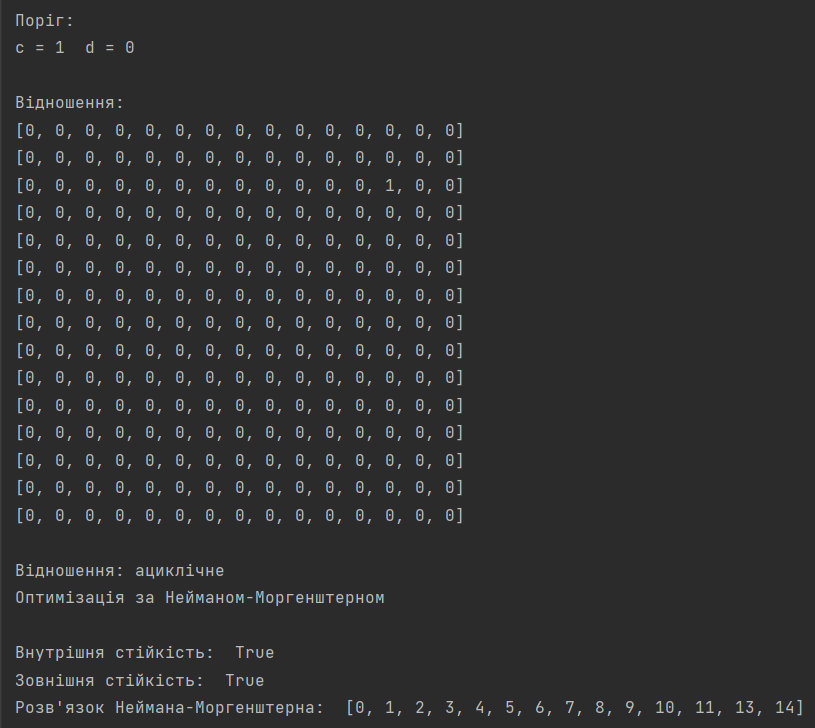
На початку експерименту отримане відношення є неациклічним, оскільки значення с є занадто слабким. При збільшенні порогу с відношення стновиться менш зв’язним, а ядро відповідно зібльшується.

Тут з кожною новою ітерацією, тобто при збільшенні порогу с, кожне попереднє відношення є підмножиною поточного. Вкінці, при с = 1 відношення складається з усіх 0, тобто немає жодних зв’язків між вершинами, отже всі елементи стають ядром.

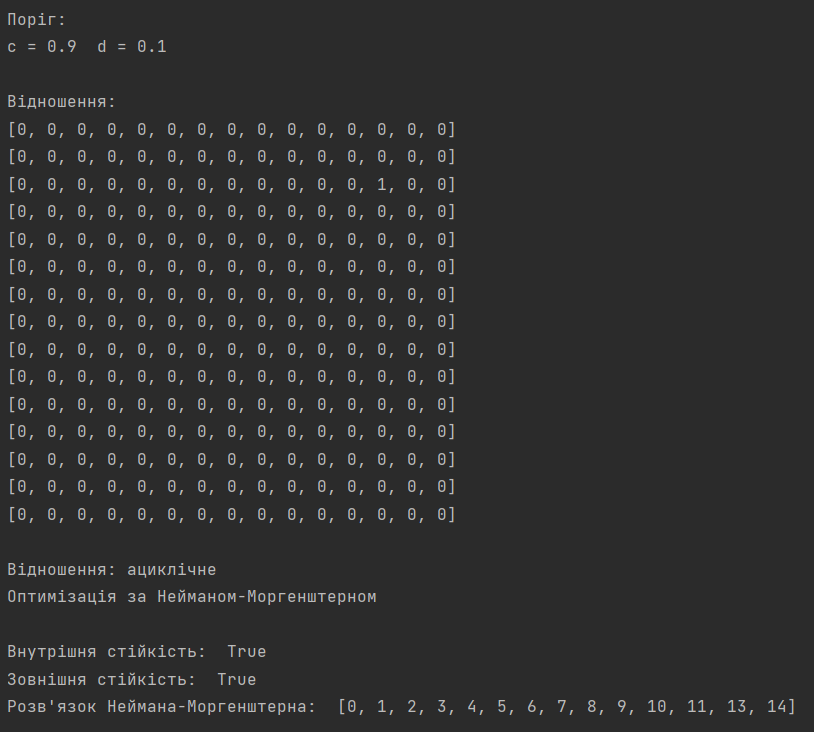
**2.3. Визначення впливу одночасної зміни порогових значень DI та CI на склад та розмір ядра**

Дослідимо вплив одночасної зміни порогів значень c, d на склад та розмір ядра, починаючи від пари значень cmax і dmin (яка відповідає максимальному складу ядра). Виконаємо одночасну зміну порогів (збільшуючи поріг d і зменшуючи поріг с, в межах заданих раніше інтервалів) з кроком 0.1.

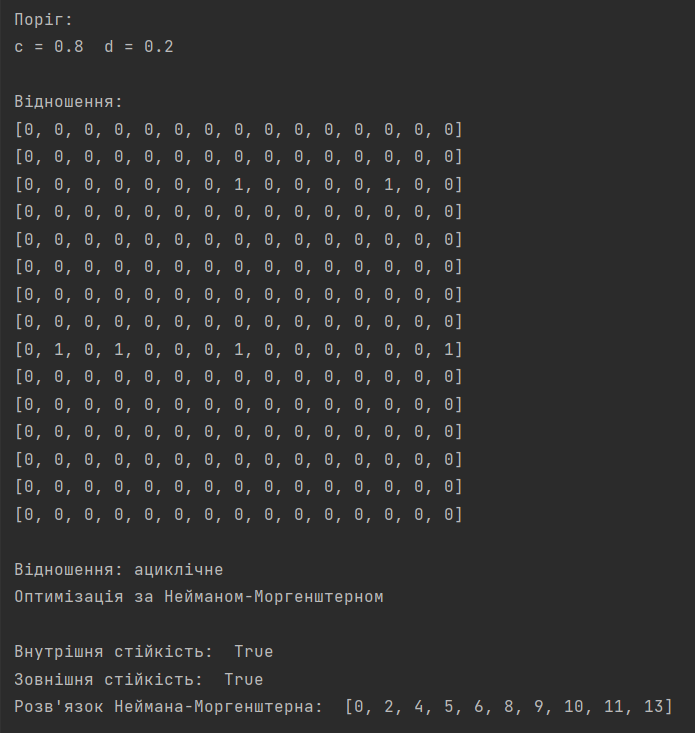
Результати експерименту при c = 1, d = 0.



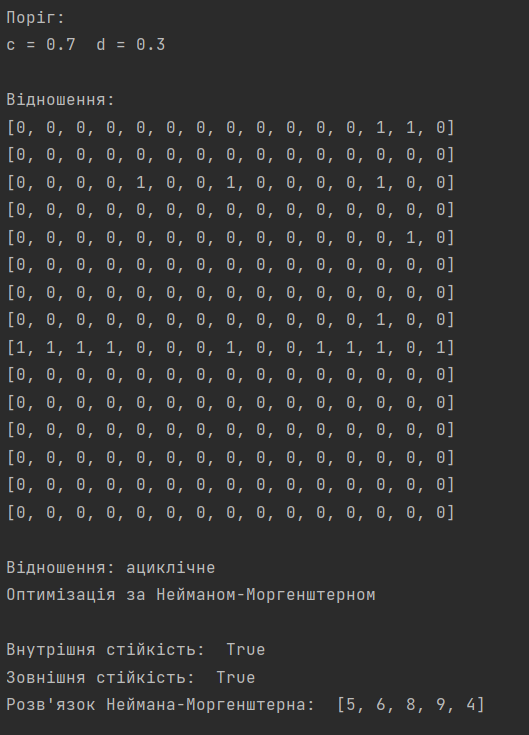
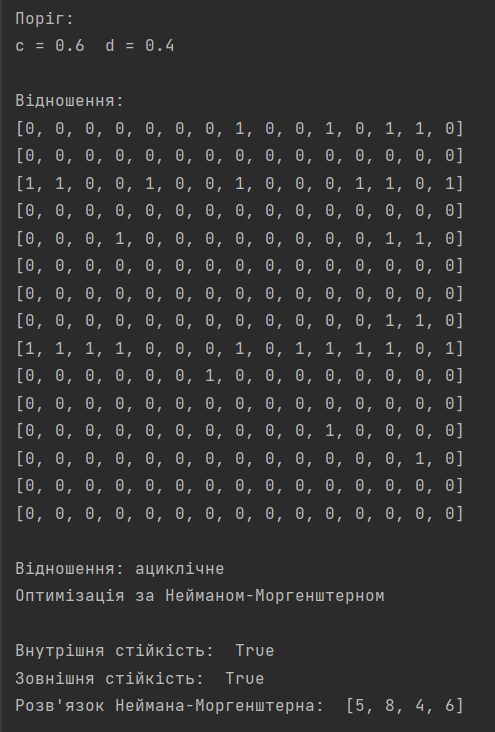
Результати експерименту при c = 0.9, d = 0.1



Результати експерименту при c = 0.8, d = 0.2



Результати експерименту при c = 0.7, d = 0.3 та c = 0.6, d = 0.4 відповідно.

При значеннях c = 0.5, d = 0.5 експеримент вже проводився.

При послабленні порогів с і d розмір ядра результуючого відношення зменшується, доки відношення не стане циклічним.

Для знаходження розв’язку задачі знаходження ядра візьмемо:

* ядро такого ациклічного відношення, що при наступній зміні порогів с і d результуюче відношення було б циклічними та не мало б ядра;
* найменше ядро з п 2.1;
* найменше ядро з п 2.2.

Перетин ядер результуючих відношень кожного експерименту є розв’язком задачі. Знайдемо цей розв’язок:

X\*2.1 = {5, 6, 9, 10}, X\*2.2 = {6, 9, 14}, X\*2.3 = {5, 6, 7, 9}

# Опис функцій та методів

Класи

|  |  |  |
| --- | --- | --- |
| Назва файлу | Назва класу | Опис |
| main.py | main | Виклик алгоритмів |
| utils.py |  | Задається умова і знаходяться допоміжні функції |
| algorithm.py | Concordance | Клас, що об’єднує методи алгоритму побудови матриці індексів узгодження С |
| Discordance | Клас, що об’єднує методи алгоритму побудови матриці індексів неузгодження D |
| Relation | Клас, що об'єднує методи побудови відношення переваги R на основі значень індексів узгодження та неузгодження |
| Optimum | Клас, що об’єднує методи алгоритму знаходження множини Х\* - ядра (або розв'язок Неймана-Моргенштерна) |

Таблиця 1. Таблиця з описами файлів та класів проекту

|  |  |  |  |
| --- | --- | --- | --- |
| Функція/метод | Параметри | Опис | Значення, що повертає |
| print\_array | array - вхідний масив | Виведенння у консоль масиву |  |
| max | array - вхідний масив | Знаходження максимального елементу в масиві | max – максимальній елемент масиву |
| min | array - вхідний масив | Знаходження мінімального елементу в масиві | min – мінімальний елемент масиву |

Таблиця 2. Таблиця опису Utils.py

|  |  |  |  |
| --- | --- | --- | --- |
| Функція/метод | Параметри | Опис | Значення, що повертає |
| solve |  | Побудова матриці узгодження С | Матрицю узгодження С |
| calc\_concordance | a, b - альтернативи | Знаходження індексу матриці узгодження С | Значення індексу |

Таблиця 3. Таблиця опису класу Concordance у файлі Algorithm.py

|  |  |  |  |
| --- | --- | --- | --- |
| Функція/метод | Параметри | Опис | Значення, що повертає |
| solve |  | Побудова матриці неузгодження D | Матрицю неузгодження D |
| calc\_discordance | a, b - альтернативи | Знаходження індексу матриці неузгодження D | Значення індексу |

Таблиця 4. Таблиця опису класу Discordance у файлі Algorithm.py

|  |  |  |  |
| --- | --- | --- | --- |
| Функція/метод | Параметри | Опис | Значення, що повертає |
| solve |  | Побудова відношення переваги R на основі значень індексів узгодження та неузгодження | Відношення переваги R |

Таблиця 5. Таблиця опису класу Relation у файлі Algorithm.py

|  |  |  |  |
| --- | --- | --- | --- |
| Функція/метод | Параметри | Опис | Значення, що повертає |
| dfs | used, checked, row, matrix | Обхід відношення у глибину, для перевірки ациклічності | True – відношення має цикл  False – відношення не має цикла |
| check\_acycle | matrix – відношення переваги R | Перевірка ациклічності | True – відношення циклічне  False – відношення ациклічне |
| get\_upper\_countour\_set | matrix,node | Знаходження множини верхніх перерізів для вузла | up\_set - множина верхніх перерізів для вузла |
| get\_Neumann\_Morgenstern\_S | matrix | Формування послідовних множин S | S – послідовність множин |
| get\_Neumann\_Morgenstern\_A | matrix, S | Формування послідовних множин Q | Q – послідовність множин |
| check\_internal\_stability | matrix, arr | Перевірка на внутрішню стійкість | True – множина має властивість внутрішньої стабільності  False – множина не має цієї властивості |
| check\_external\_stability | matrix, arr | Перевірка на зовнішню стійкість | True – множина має властивість зовнішньої стабільності  False – множина не має цієї властивості |

Таблиця 6. Таблиця опису класу Optimum у файлі Algorithm.py