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 №2

на тему «Методи оптимізації за бінарними відношеннями»

З дисципліни «Теорія прийняття рішень»

7 варіант

Виконала:

студентка гр. ІС-91

Ковальська А.А.

Перевірила:

Жураковська О.С.

# Завдання 1

|  |  |  |  |
| --- | --- | --- | --- |
| Відношення | Клас, до якого належить БВ | Опт. альтернативи за принципом домінування | Опт. альтернативи за принципом блокування |
| R1 | Не відноситься до БВ | X\*R = {∅}, X\*\*R = {∅} | X0R = {∅}, X00R = {∅} |
| R2 | Не відноситься до БВ | X\*R = {∅}, X\*\*R = {∅} | X0R = {∅}, X00R = {∅} |
| R3 | Еквівалентність | X\*R = {∅}, X\*\*R = {∅} | X0R = {1, 2, 3, 4, 5, 6, 7, 8},  X00R = {∅} |
| R4 | Слабке упорядкування | X\*P = {∅} | X0P = {2,3,7} |
| R5 | Слабке упорядкування | ∀𝑦 ∈ Ω , 𝑦 ≠ 7 | 7𝑃y  X\*P = {7} | ∀𝑦 ∈ Ω | y!𝑃7  X\*P = {7} |
| R6 | Слабке упорядкування | ∀𝑦 ∈ Ω , 𝑦 ≠ 1 | 1𝑃y  X\*P = {1} | ∀𝑦 ∈ Ω | y!𝑃1  X0P = {1} |
| R7 | Квазіпорядок | ∀𝑦 ∈ Ω 3𝑅y  X\*R = {3}, X\*\*R = {3} | ∀𝑦 ∈ Ω 3𝑅y  X0R = {3}, X00R = {3} |
| R8 | Слабкий порядок | ∀𝑦 ∈ Ω 5𝑅y  X\*R = {5}, X\*\*R = {5} | ∀𝑦 ∈ Ω 5𝑅y  X0R = {5}, X00R = {5} |

Таблиця 1. Результати виконання завдання 1

|  |  |  |  |
| --- | --- | --- | --- |
| Відношення | Ациклічне/неациклічне | Розв'язок Неймана-Моргенштерна | Опт. альтернативи за принципом К-оптимізації |
| R1 | + | ХНМ = {1,4,5} |  |
| R2 | + | ХНМ = {1,3,5,10,11} |  |
| R3 | - |  | k1 max = {1,3,4,6,11}  k1 opt = {1,3,4,6,11}  k2 max = {1,6,11}  k2 opt = {1,6,11}  k3 max = {}  k3 opt = {}  k4 max = {1,6,11}  k4 opt = {} |
| R4 | + | ХНМ = {1,6,8,10} |  |
| R5 | + | ХНМ = {3,5,7,10} |  |
| R6 | + | ХНМ = {1,2,4,7,9,13} |  |
| R7 | + | ХНМ = {2,4} |  |
| R8 | + | ХНМ = {1,4,5,6,13} |  |
| R9 | - |  | k1 max = {14}  k1 opt = {14}  k2 max = {14}  k2 opt = {}  k3 max = {14}  k3 opt = {14}  k4 max = {14}  k4 opt = {} |
| R10 | - |  | k1 max = {}  k1 opt = {}  k2 max = {}  k2 opt = {}  k3 max = {}  k3 opt = {}  k4 max = {}  k4 opt = {} |

Таблиця 2. Результати виконання завдання 2

# Завдання 1

## №1 -------------

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 | 1 | 1 | 1 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 |
| 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| N | P | N |  | P | P | P | N |
|  | I | N | P |  | P | P | P |
| N | N | I |  | N |  | N |  |
| P |  | P | I | I | I | I | P |
|  | P | N | I | I |  | P |  |
|  |  | P | I | P | N | P | P |
|  |  | N | I |  |  | N |  |
| N |  | P |  | P |  | P | I |

*R1 не відноситься до бінарного класу.*

## №2 -------------

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| N | I | N |  | I | N |  | I |
| I | N |  |  |  | P | N |  |
| N | P | I |  |  | I | N |  |
| P | P | P | N | I | N | P | I |
| I | P | P | I | I | P | P |  |
| N |  | I | N |  | I | I | I |
| P | N | N |  |  | I | N | N |
| I | P | P | I | P | I | N | N |

*R2 не відноситься до бінарного класу.*

## №3 -------------

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 |
| 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 |
| 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| I | N | I | N | N | N | N | N |
| N | I | N | N | I | N | I | N |
| I | N | I | N | N | N | N | N |
| N | N | N | I | N | I | N | I |
| N | I | N | N | I | N | I | N |
| N | N | N | I | N | I | N | I |
| N | I | N | N | I | N | I | N |
| N | N | N | I | N | I | N | I |

*R3 відноситься до еквівалентності.*

Оптимізація за домінуванням:

I≠∅, тому підмножина найкращих альтернатив – найбільший та строго найбільший елемент по R на множині Ω. На множині не існує такого елементу ХR\* та ХR\*\*, для якого виконується R-(x\*) = Ω та R-(x\*) = Ω i R+(x\*) = {x\*}, тому переходимо до оптимізації за блокуванням.

Відповідь: X\*R = {∅}, X\*\*R = {∅}.

Оптимізація за блокуванням:

I≠∅, тому множина ненайгірших альтернатив – максимальний та строго максимальний елемент по R на множині Ω.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| I | N | I | N | N | N | N | N |
| N | I | N | N | I | N | I | N |
| I | N | I | N | N | N | N | N |
| N | N | N | I | N | I | N | I |
| N | I | N | N | I | N | I | N |
| N | N | N | I | N | I | N | I |
| N | I | N | N | I | N | I | N |
| N | N | N | I | N | I | N | I |

Відповідь: X0R = {1,2,3,4,5,6,7,8}, X00R = {∅}.

## №4 -------------

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| N |  |  |  | N |  | N |  |
| P | N | N | P | P | P | P | N |
| P | N | N | P | P | P | P | N |
| P |  |  | N | P | N | P |  |
| N |  |  |  | N |  | N |  |
| P |  |  | N | P | N | P |  |
| N |  |  |  | N |  | N |  |
| P | N | N | P | P | P | P | N |

*R4 відноситься до слабкого упорядкування.*

Оптимізація за домінуванням:

I=∅, тому підмножина найкращих альтернатив – найбільший за домінуванням елемент по Р на множині Ω. Таких елементів не існує, тому переходимо до оптимізації за блокуванням.

Відповідь: X\*P = {∅}.

Оптимізація за блокуванням:

I=∅, тому множина ненайгірших альтернатив – максимальний елемент по Р на множині Ω.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 |

Відповідь: X0P = {2,3,7}.

## №5 -------------

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 |
| 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| N |  | P |  | P | P |  |  |
| P | N | P |  | P | P |  |  |
|  |  | N |  |  |  |  |  |
| P | P | P | N | P | P |  |  |
|  |  | P |  | N | P |  |  |
|  |  | P |  |  | N |  |  |
| P | P | P | P | P | P | N | P |
| P | P | P | P | P | P |  | N |

*R5 відноситься до слабкого упорядкування.*

Оптимізація за домінуванням:

I=∅, тому підмножина найкращих альтернатив – найбільший за домінуванням елемент по Р на множині Ω.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 |
| 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 |

∀𝑦 ∈ Ω , 𝑦 ≠ 7 | 7𝑃y

Відповідь: X\*P = {7}.

Оптимізація за блокуванням:

I=∅, тому множина ненайгірших альтернатив – максимальний елемент по Р на множині Ω.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 |
| 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 |

∀𝑦 ∈ Ω | y!𝑃7

Відповідь: X\*P = {7}.

## №6 -------------

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| N | P | P | P | P | P | P | P |
|  | N | P | P |  | P |  | P |
|  |  | N | P |  |  |  | P |
|  |  |  | N |  |  |  |  |
|  | P | P | P | N | P |  | P |
|  |  | P | P |  | N |  | P |
|  | P | P | P | P | P | N | P |
|  |  |  | P |  |  |  | N |

*R6 відноситься до слабкого упорядкування.*

Оптимізація за домінуванням:

I=∅, тому підмножина найкращих альтернатив – найбільший за домінуванням елемент по Р на множині Ω.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |

∀𝑦 ∈ Ω , 𝑦 ≠ 1 | 1𝑃y

Відповідь: X\*P = {1}.

Оптимізація за блокуванням:

I=∅, тому множина ненайгірших альтернатив – максимальний та строго максимальний елемент по Р на множині Ω.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |

∀𝑦 ∈ Ω | y!𝑃1

Відповідь: X0P = {1}.

## №7 -------------

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| I |  |  |  | I | I | I |  |
| P | I |  | I | P | P | P | I |
| P | P | I | P | P | P | P | P |
| P | I |  | I | P | P | P | I |
| I |  |  |  | I | I | I |  |
| I |  |  |  | I | I | I |  |
| I |  |  |  | I | I | I |  |
| P | I |  | I | P | P | P | I |

*R7 відноситься до квазіпорядку.*

Оптимізація за домінуванням:

I≠∅, тому підмножина найкращих альтернатив – найбільший та строго найбільший елемент по R на множині Ω.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 |

∀𝑦 ∈ Ω 3𝑅y

Відповідь: X\*R = {3}, X\*\*R = {3}.

Оптимізація за блокуванням:

I≠∅, тому множина ненайгірших альтернатив – максимальний та строго максимальний елемент по R на множині Ω.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 |

∀𝑦 ∈ Ω 3𝑅y

Відповідь: X0R = {3}, X00R = {3}.

## №8 -------------

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| I |  |  |  |  |  |  |  |
| P | I |  |  |  |  | P |  |
| P | P | I | P |  |  | P |  |
| P | P |  | I |  |  | P |  |
| P | P | P | P | I | P | P | P |
| P | P | P | P |  | I | P | P |
| P |  |  |  |  |  | I |  |
| P | P | P | P |  |  | P | I |

*R8 відноситься до нестрогого слабкого порядку.*

Оптимізація за домінуванням:

I≠∅, тому підмножина найкращих альтернатив – найбільший та строго найбільший елемент по R на множині Ω.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 |

∀𝑦 ∈ Ω 5𝑅y

Відповідь: X\*R = {5}, X\*\*R = {5}.

Оптимізація за блокуванням:

I≠∅, тому множина ненайгірших альтернатив – максимальний та строго максимальний елемент по R на множині Ω.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 |

∀𝑦 ∈ Ω 5𝑅y

Відповідь: X0R = {5}, X00R = {5}.

# Завдання 2

На початку відношення перевіряється на ациклічність.

Обраний алгоритм – пошук у глибину (DFS).

1. Створити граф із заданими ребрами та вершинами

2. Створити рекурсивну функцію, що ініціалізує поточну вершину та список відвідуваних вершин і рекурсивний стек.

3. Позначити поточну вершину як відвідану і позначити індекс в рекурсивному стеку.

4. Знайти ті вершини, що ще не відвідані та знайти суміжні вершини до поточного. Рекурсивно викликати функцію до цих вершин. Якщо рекурсивна функція повертає true, то повернути true.

5. Якщо суміжні вершини вже позначені в рекурсивному стеку, то повернути true.

Часова складність: O(V+E).

Код, де перевіряється ациклічність:

def check\_acyclic():  
 used = []  
 for node in range(size):  
 if used.count(node) == 0:  
 if dfs(used, [], node):  
 return True  
 return False  
  
# dfs  
def dfs(used, checked, row):  
 checked.append(row)  
 for col in range(size):  
 if matrix[row][col] == 0:  
 continue  
 if matrix[row][col] == 1 and checked.count(col) == 0:  
 if dfs(used, checked, col):  
 return True  
 else:  
 return True  
 checked.remove(row)  
 used.append(row)  
 return False

В залежності від того, яке відношення, далі обираємо спосіб оптимізації.

Оптимізація Неймана-Моргенштерна.

Спочатку будується множина С0 для ациклічного відношення.

1. Формуємо послідовність множин S.

def get\_S\_NM():  
 S = []  
 up\_sets = []  
 for i in range(size):  
 s = get\_upper\_contour\_set(matrix, i)  
 up\_sets.append(s)  
 S0 = []  
 for i in range(size):  
 if len(up\_sets[i]) == 0:  
 S0.append(i)  
 print('S0:', S0)  
 S.append(S0)  
 count\_s = 1  
 while S[-1] != list(range(size)):  
 Si = []  
 for i in range(size):  
 if up\_sets[i].issubset(S[-1]):  
 Si.append(i)  
 print('S{}: {}'.format(count\_s, Si))  
 S.append(Si)  
 count\_s += 1  
 return S

1. Формуємо послідовність множин Q.

def get\_Q\_NM(S):  
 Q = [S[0]]  
 print('Q0: {}'.format(Q[0]))  
 up\_sets = []  
 for i in range(size):  
 s = get\_upper\_contour\_set(matrix, i)  
 up\_sets.append(s)  
 for i in range(1, len(S)):  
 Q.append(Q[-1].copy())  
  
 dif = list(set(S[i]) - set(S[i - 1]))  
 for j in dif:  
 if len(set(up\_sets[j]).intersection(Q[i - 1])) == 0:  
 Q[i].append(j)  
 print('Q{}: {}'.format(i, Q[i]))  
  
 return Q

С0 = QL, якщо множина є внутрішньо і зовнішньо стійкою.

Тому в оптимізації Неймана-Моргенштерна множина перевіряється на внутрішню та зовнішню стійкість.

Множина X називається внутрішньо стійкою, якщо два будь-які її елемента не можуть бути порівняними за відношенням переваги, тобто:

∀ x∈ X : R+(x) ∩ (X \{x}) = ∅∧ R-(x) ∩ (X \{x}) =∅

Код з перевіркою внутрішньої стійкості:

def check\_internal\_stability(arr):  
 for row in arr:  
 for col in arr:  
 if matrix[row][col] == 1:  
 return False  
 return True

Множина Х називається зовнішньо стійкою, якщо для будь-якого у неналежаного Х знайдеться х більш переважний, тобто:

∀y ∈Ω \X:R+ (y)∩ X ≠ ∅

Код з перевіркою зовнішньої стійкості:

def check\_external\_stability(arr):  
 for col in range(size):  
 if (col in arr):  
 continue  
 res = False  
 for row in arr:  
 if matrix[row][col] == 1:  
 res = True  
 break  
 if not res:  
 return False  
 return True

Множина, яка має властивості внутрішньої та зовнішньої стійкості – це розв’язок Неймана-Моргенштерна.

К-оптимізація

Спочатку формуємо таблицю з P,I,N частинами відношення:

def get\_pin():  
 res\_mat = []  
 for row in range(len(matrix)):  
 res\_mat.append([])  
 for col in range(len(matrix[row])):  
 if matrix[row][col] == 1 and matrix[col][row] == 1:  
 res\_mat[row].append('I')  
 elif matrix[row][col] == 0 and matrix[col][row] == 0:  
 res\_mat[row].append('N')  
 elif matrix[row][col] == 1 and matrix[col][row] == 0:  
 res\_mat[row].append('P')  
 else:  
 res\_mat[row].append('-')  
 return res\_mat

Далі в залежності від того, який тип оптимізації (к = 1, 2, 3,4) формуємо множину SkR(x).

S1R(x) = H-P (x) ∪ H-I (x) ∪ H-N (x),

S2R(x) = H-P (x) ∪ H-N (x),

S3R(x) = H-P (x) ∪ H-I (x),

S4R(x) = H-P (x).

Фрагмент коду побудови SkR(x) в залежності від параметру pin\_str:

S1 = np.zeros((size, size))  
for row in range(len(pin)):  
 for col in range(len(pin[row])):  
 if pin[row][col] in pin\_str:  
 S1[row][col] = 1

Далі знаходимо максимальні та оптимальні елементи.

Х буде к-максимальним елементом по відношенню R, якщо SkR(x) є максимальним по включенню елементом сімейства { SkR(x)| x ∈Ω}.

Х називається к-оптимальним елементом по відношенню R на Ω, якщо SkR(x) = Ω.

max = []  
opt = []  
  
for row in range(size):  
 is\_max\_element = True  
 is\_opt\_element = True  
 for col in range(size):  
 if S1[row][col] != 1:  
 is\_opt\_element = False  
 if not check\_null(S1, col):  
 is\_max\_element = False  
 if is\_max\_element:  
 max.append(row)  
 if is\_opt\_element:  
 opt.append(row)

Елемент можу бути максимальний, проте не оптимальним, для цього робимо перевірку. Тобто, якщо у рядку зустрічається 0 і весь стовпчик також дорівнює 0, то цей елемент можна віднести до максимального.

def check\_null(matrix, col):  
 for row in range(len(matrix)):  
 if matrix[row][col] != 0:  
 return False  
 return True

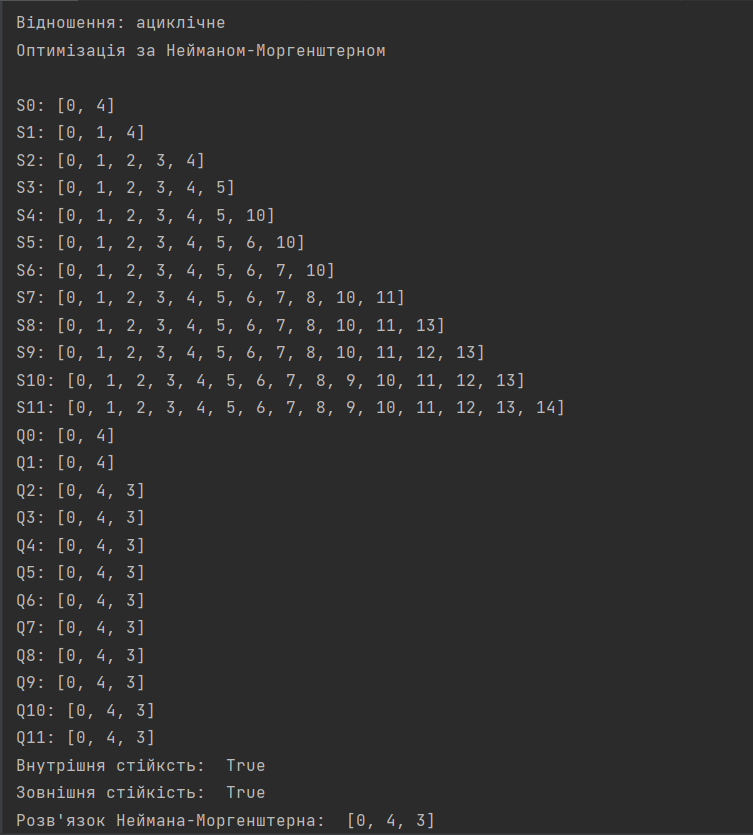
Отже, для К-оптимізації для кожного к=1,2,3,4 будується матриця та знаходиться максимальний елемент, а також перевіряється чи є максимальний елемент оптимальним.

|  |  |  |  |
| --- | --- | --- | --- |
| Функція/метод | Параметри | Опис | Значення, що повертає |
| dfs | used, checked, row | Обхід відношення у глибину, для перевірки ациклічності | True – відношення має цикл  False – відношення не має цикла |
| check\_acycle |  | Перевірка ациклічності | True – відношення циклічне  False – відношення ациклічне |
| get\_upper\_countour\_set | node | Знаходження множини верхніх перерізів для вузла | up\_set - множина верхніх перерізів для вузла |
| get\_S\_NM |  | Формування послідовних множин S | S – послідовність множин |
| get\_Q\_NM | S | Формування послідовних множин Q | Q – послідовність множин |
| check\_internal\_stability | arr | Перевірка на внутрішню стійкість | True – множина має властивість внутрішньої стабільності  False – множина не має цієї властивості |
| check\_external\_stability | arr | Перевірка на зовнішню стійкість | True – множина має властивість зовнішньої стабільності  False – множина не має цієї властивості |
| get\_pin |  | Для відношення R формуємо нижні перерізи  P, I, N | res\_mat – результуюча матриця з нижніми перерізами множин P, I, N |
| check\_null | matrix, col | Перевірка чи у заданому стовпці всі рядки дорівнюють 0 | True – всі рядки у заданому стовпці дорівнюють 0  False – знайдено хоча б 1 рядок з 1 |
| get\_S | pin, pin\_str | В залежності від способу оптимізації, який передаємо з параметром pin\_str знаходимо множину домінованих альтернатив і формуємо множину к-максимальних та к-оптимальних елементів | max – к-максимальний елемент  opt – к-оптимальний елемент |

Таблиця 3. Опис функцій та методів

Далі наведені скріншоти роботи програми з проміжними даними.

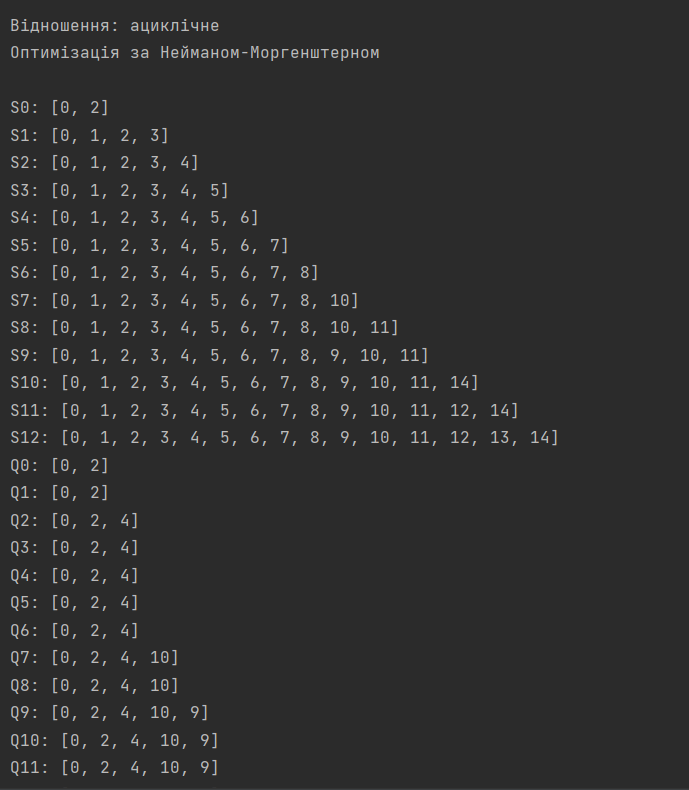
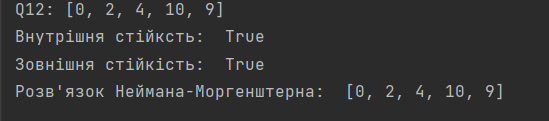
## №1 -------------



Відношення ациклічне, тому проводимо оптимізацію Неймана-Моргенштерна.

Так як множина внутрішньо та зовнішньо стійка, то розв’язок Неймана-Моргенштерна C0 = Q11= [1,5,4].

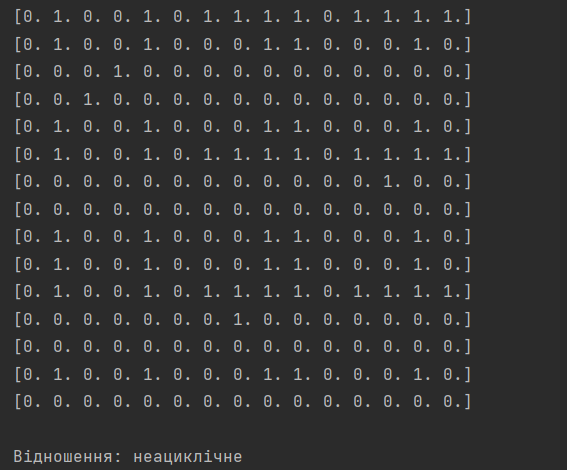
## №2 -------------

Відношення ациклічне, тому проводимо оптимізацію Неймана-Моргенштерна.

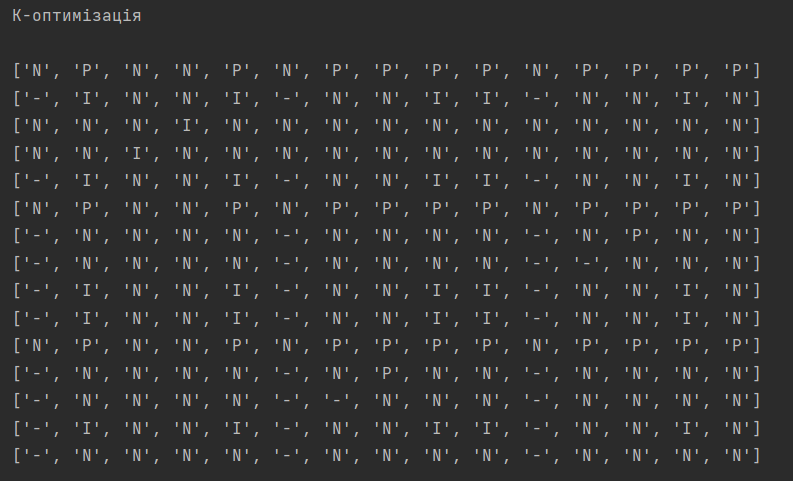
Так як множина внутрішньо та зовнішньо стійка, то розв’язок Неймана-Моргенштерна C0 = Q12= [1,3,5,11,10].

## №3 -------------

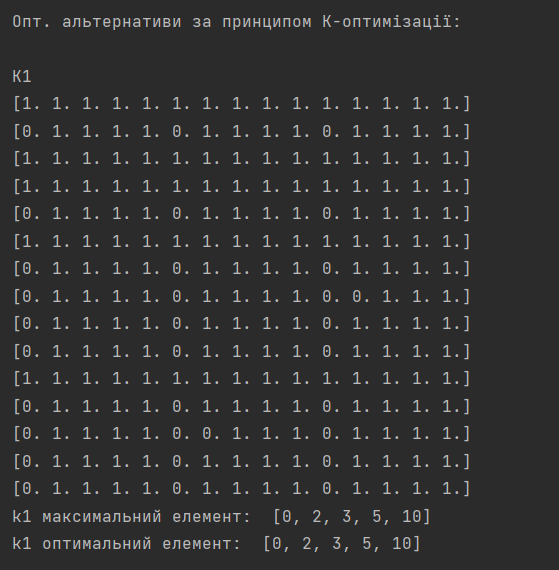


Відношення неациклічне, тому проводимо к-оптимізацію.

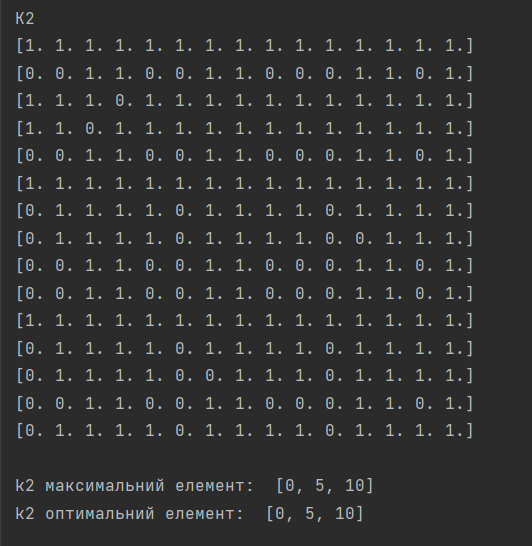
Будуємо відношення з P,I,N частинами:



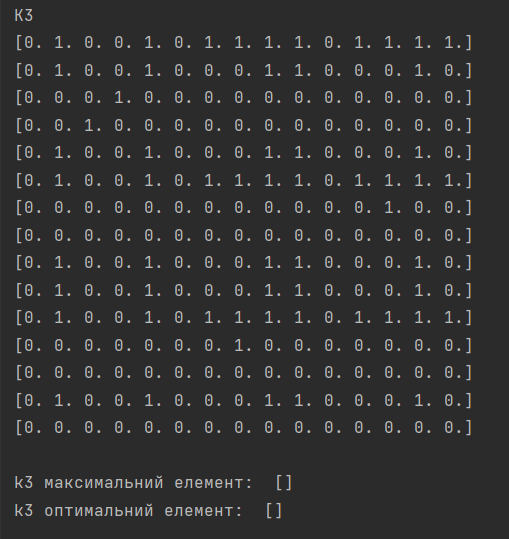
Будуємо SkR(x) та знаходимо максимальні та оптимальні елементи за наведеним вище алгоритмом.

S1R(x) = H-P (x) ∪ H-I (x) ∪ H-N (x):

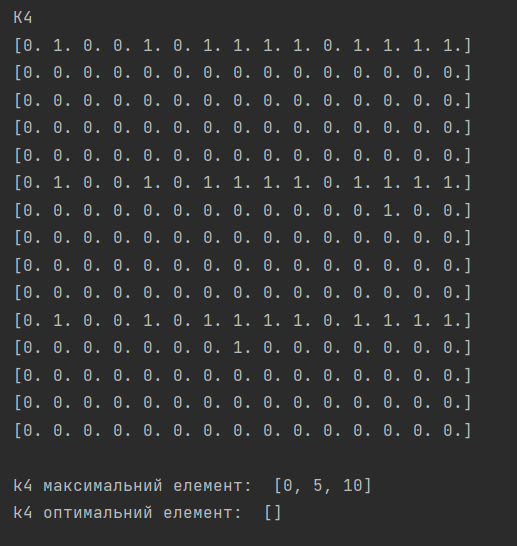
S2R(x) = H-P (x) ∪ H-N (x):



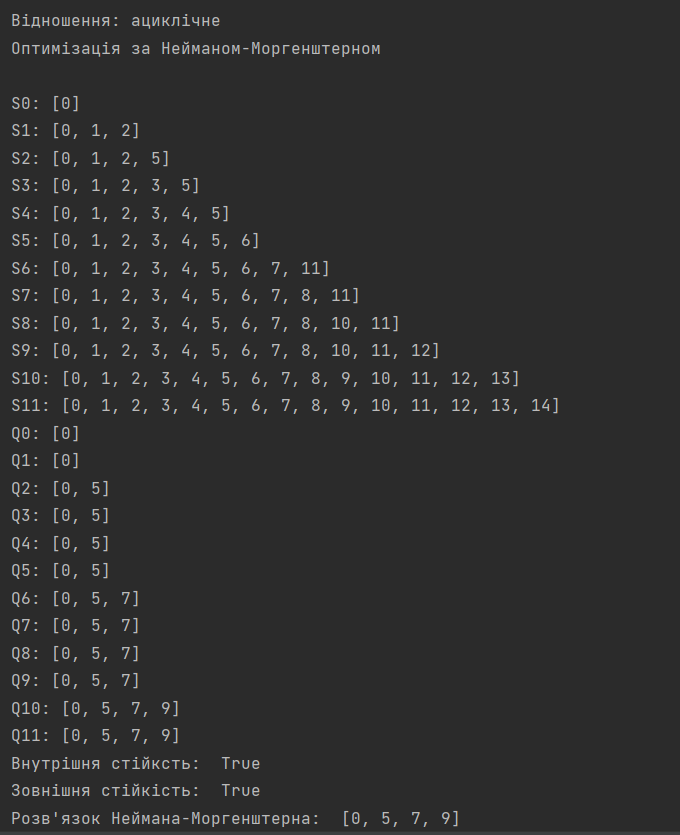
S3R(x) = H-P (x) ∪ H-I (x):



S4R(x) = H-P (x):



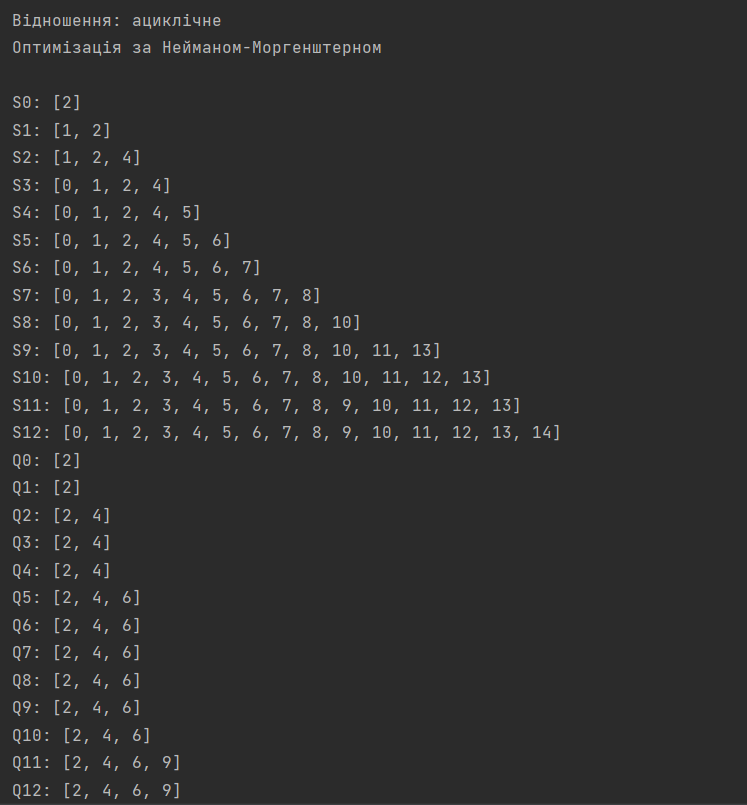
## №4 -------------

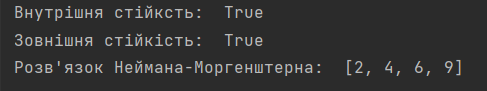


Відношення ациклічне, тому проводимо оптимізацію Неймана-Моргенштерна.

Так як множина внутрішньо та зовнішньо стійка, то розв’язок Неймана-Моргенштерна C0 = Q11= [1,6,8,10].

## №5 -------------

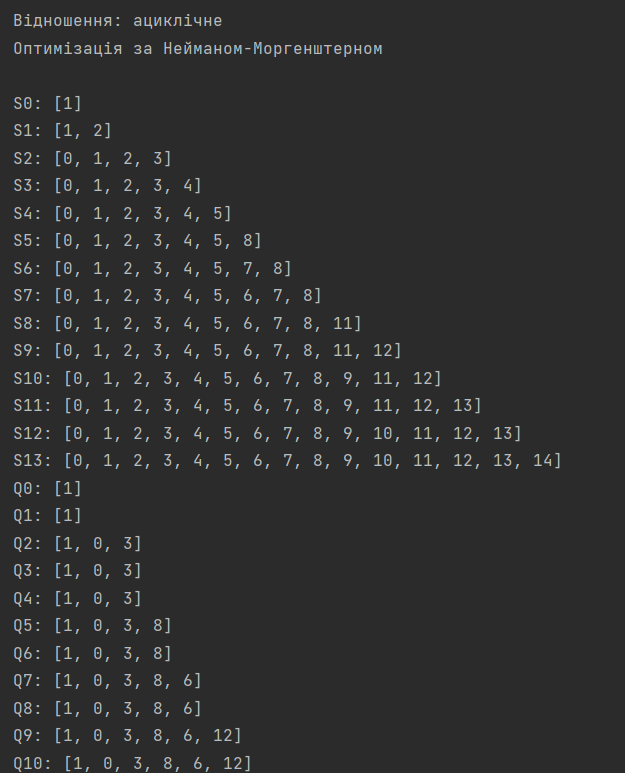


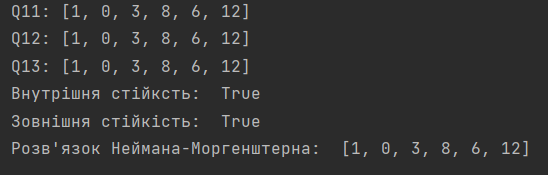


Відношення ациклічне, тому проводимо оптимізацію Неймана-Моргенштерна.

Так як множина внутрішньо та зовнішньо стійка, то розв’язок Неймана-Моргенштерна C0 = Q12= [3,5,7,10].

## №6 -------------

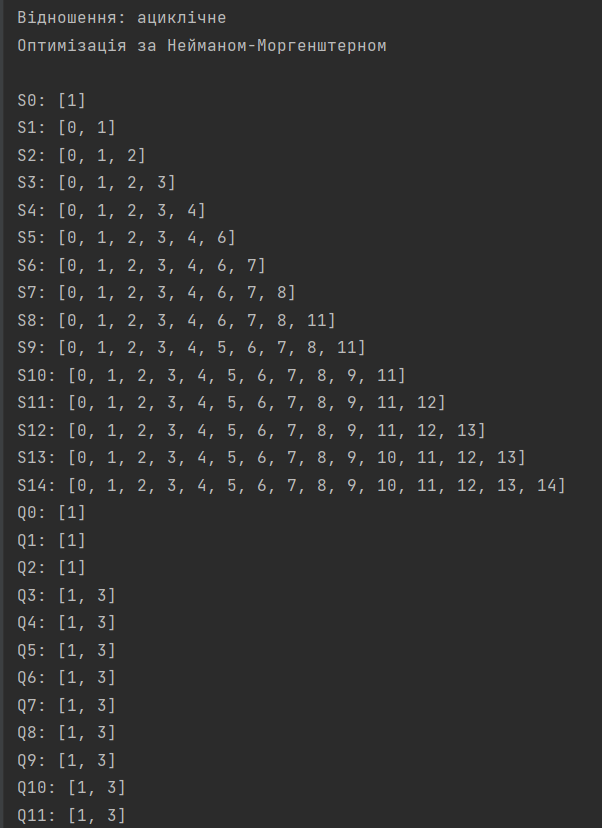


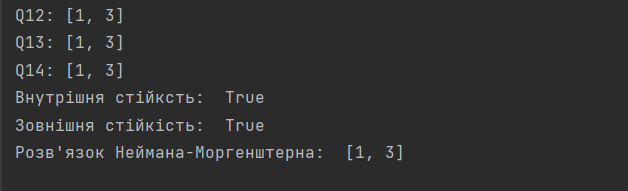


Відношення ациклічне, тому проводимо оптимізацію Неймана-Моргенштерна.

Так як множина внутрішньо та зовнішньо стійка, то розв’язок Неймана-Моргенштерна C0 = Q13= [2,1,4,9,7,13].

## №7 -------------

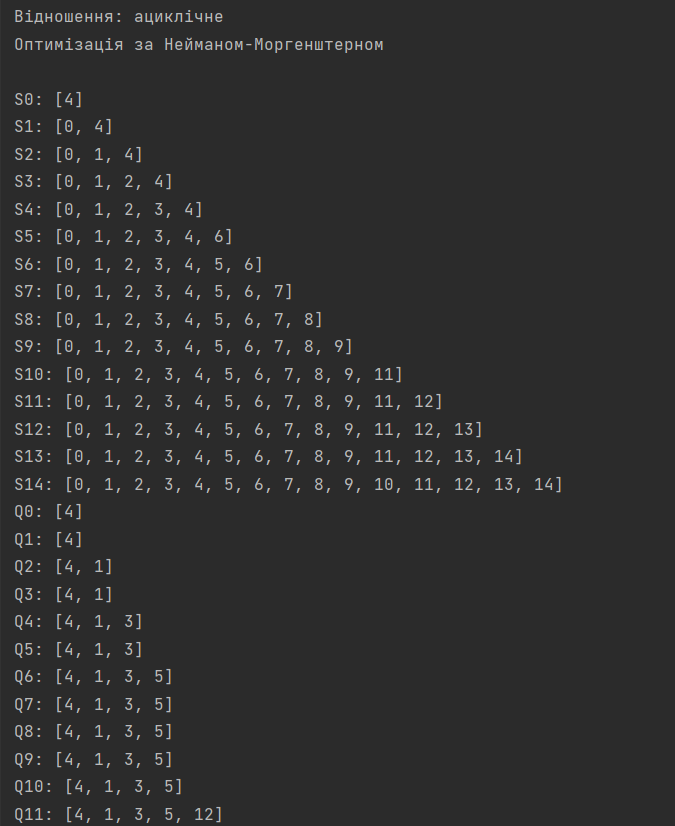


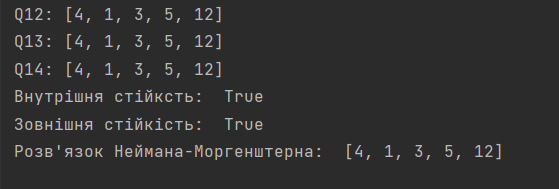


Відношення ациклічне, тому проводимо оптимізацію Неймана-Моргенштерна.

Так як множина внутрішньо та зовнішньо стійка, то розв’язок Неймана-Моргенштерна C0 = Q14= [2,4].

## №8 -------------

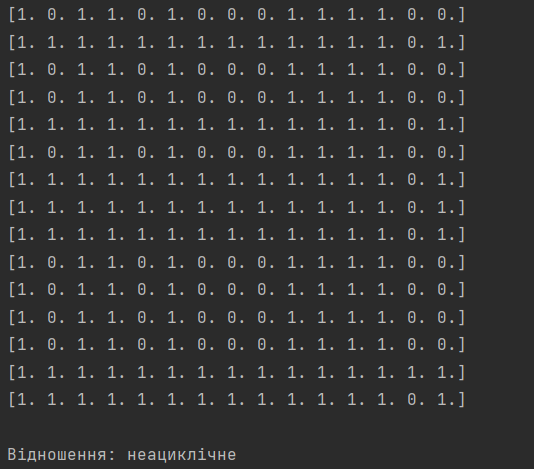




Відношення ациклічне, тому проводимо оптимізацію Неймана-Моргенштерна.

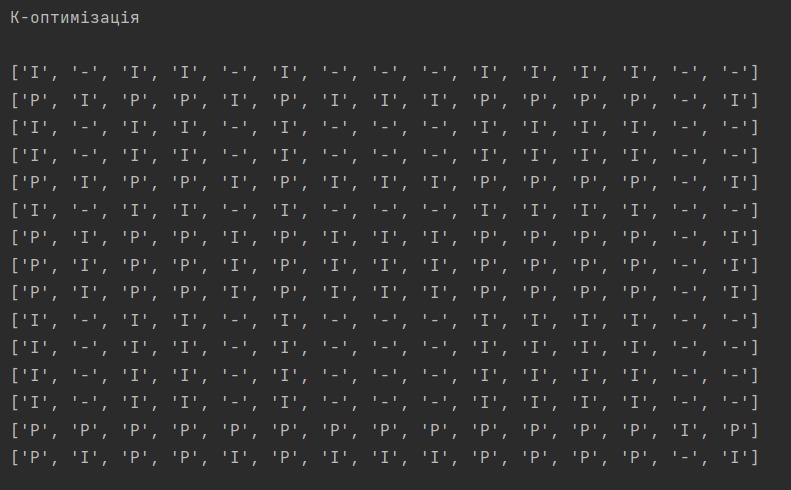
Так як множина внутрішньо та зовнішньо стійка, то розв’язок Неймана-Моргенштерна C0 = Q14= [5,2,4,6,13].

## №9 -------------



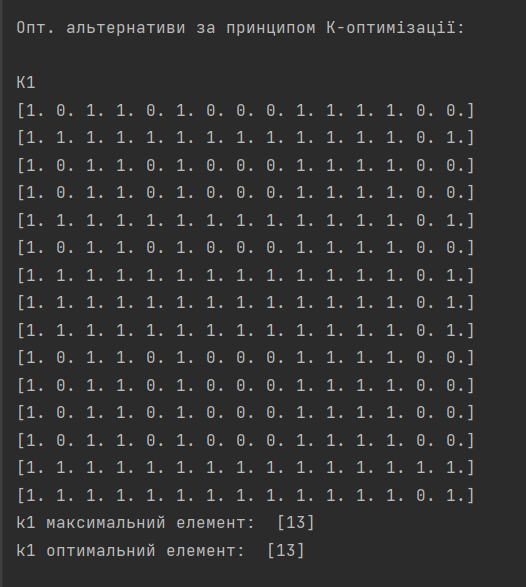
Відношення неациклічне, тому проводимо к-оптимізацію.

Будуємо відношення з P,I,N частинами:

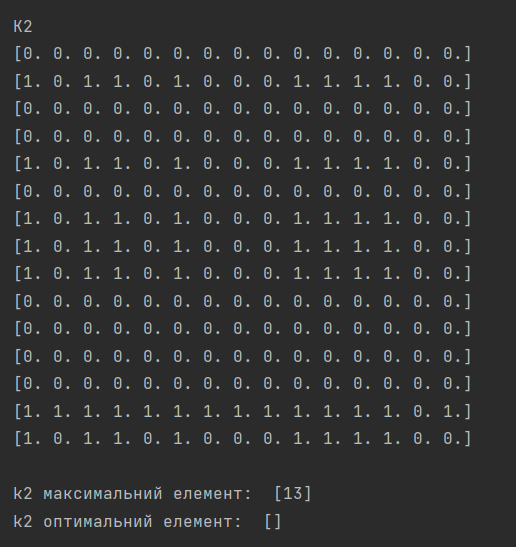


Будуємо SkR(x) та знаходимо максимальні та оптимальні елементи за наведеним вище алгоритмом.

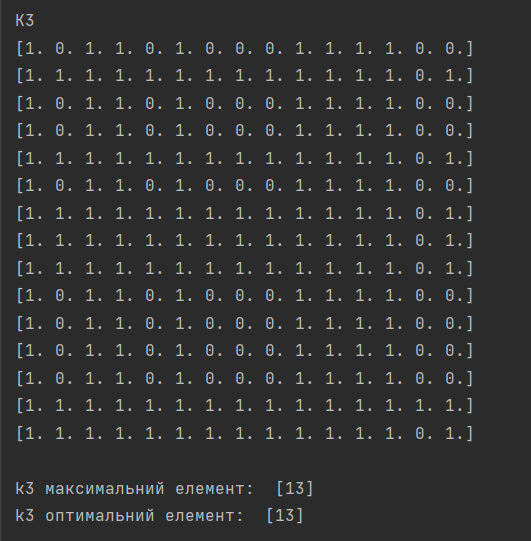
S1R(x) = H-P (x) ∪ H-I (x) ∪ H-N (x):



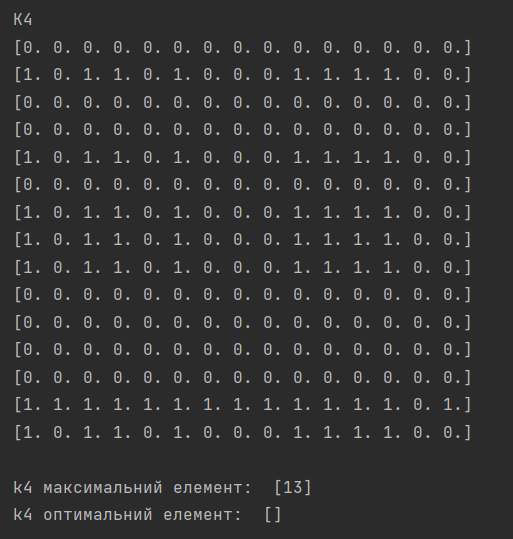
S2R(x) = H-P (x) ∪ H-N (x):



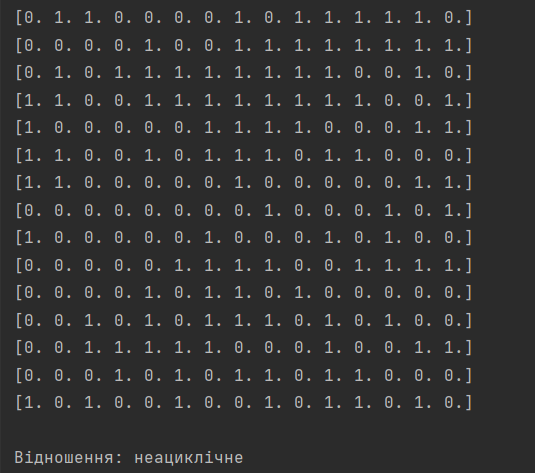
S3R(x) = H-P (x) ∪ H-I (x):



S4R(x) = H-P (x):

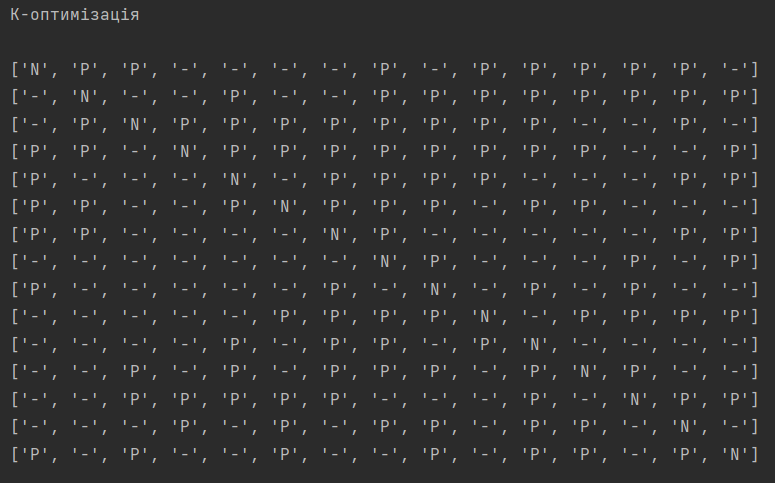


## №10 -------------



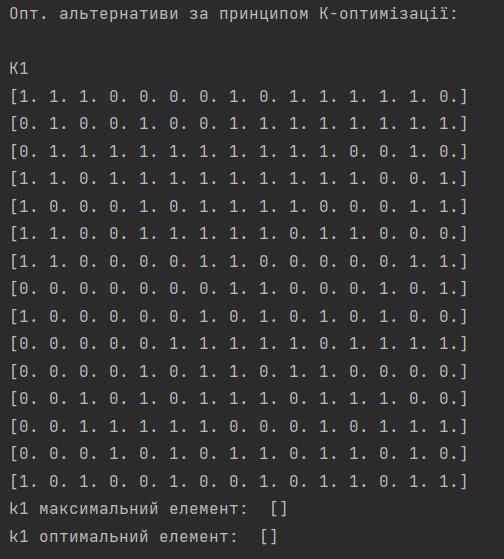
Відношення неациклічне, тому проводимо к-оптимізацію.

Будуємо відношення з P,I,N частинами:

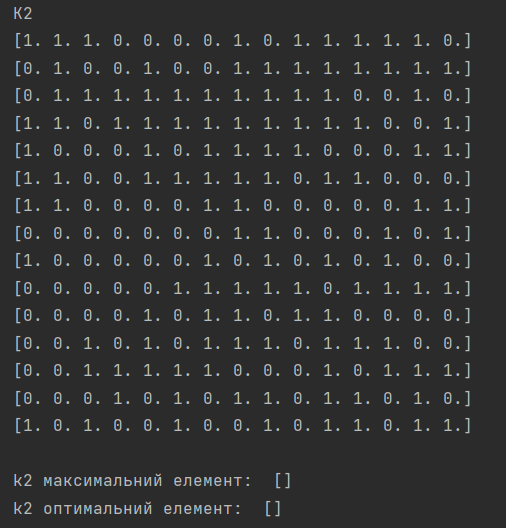


Будуємо SkR(x) та знаходимо максимальні та оптимальні елементи за наведеним вище алгоритмом.

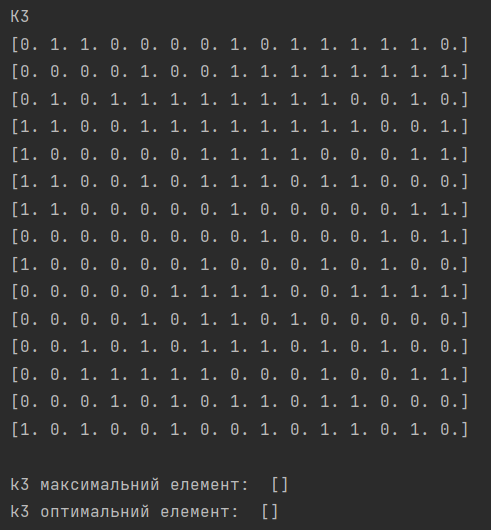
S1R(x) = H-P (x) ∪ H-I (x) ∪ H-N (x):



S2R(x) = H-P (x) ∪ H-N (x):



S3R(x) = H-P (x) ∪ H-I (x):



S4R(x) = H-P (x):

