Національний Технічний Університет України

Київський Політехнічний Інститут імені Ігоря Сікорського

Інститут Прикладного Системного Аналізу

Кафедра Системного Проектування

ЛАБОРАТОРНА РОБОТА №1

З дисципліни: «Теорія прийняття рішень»

На тему

«БАГАТОКРИТЕРІАЛЬНИЙ ВИБІР. ВИЗНАЧЕННЯ ОПТИМАЛЬНИХ АЛЬТЕРНАТИВ ЗА ПАРЕТО ТА СЛЕЙТЕРОМ»

Виконала:

Студентка групи ДА-81

Желєзнова В.С.

Варіант №10

Київ – 2021

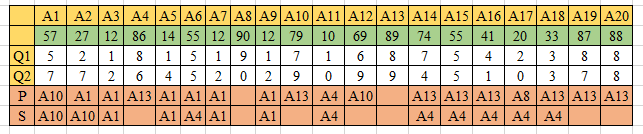
**Мета роботи:**

Ознайомитись з поняттями оптимальності за Парето та за Слейтером при багатокритеріальному виборі.

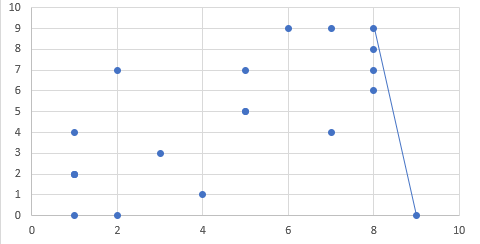
**Дані за варіантом:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| № | С | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| 10 | 1 | 57 | 27 | 12 | 86 | 14 | 55 | 12 | 90 | 12 | 79 | 10 | 69 | 89 | 74 | 55 | 41 | 20 | 33 | 87 | 88 |
| 2 | 38 | 66 | 70 | 84 | 56 | 17 | 06 | 60 | 49 | 37 | 05 | 59 | 17 | 18 | 45 | 83 | 73 | 58 | 73 | 37 |
| 3 | 89 | 83 | 07 | 78 | 57 | 14 | 71 | 29 | 51 | 59 | 18 | 38 | 25 | 88 | 74 | 33 | 57 | 81 | 93 | 58 |

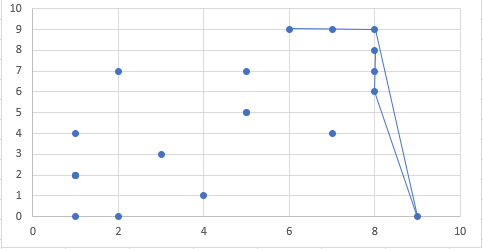
**Перший рядок:**



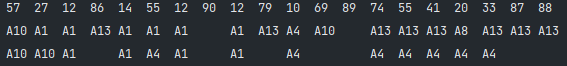
**Графічний розв’язок за Парето:**



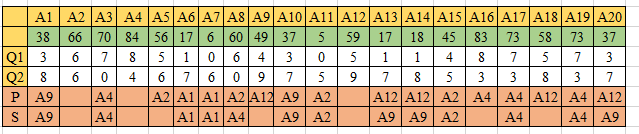
**Графічний розв’язок за Слейтером:**



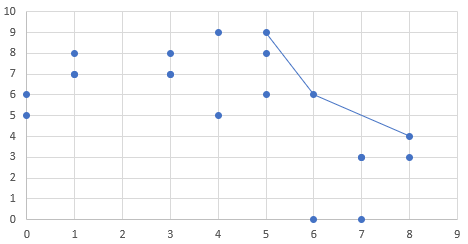
**Програмний розв’язок:**



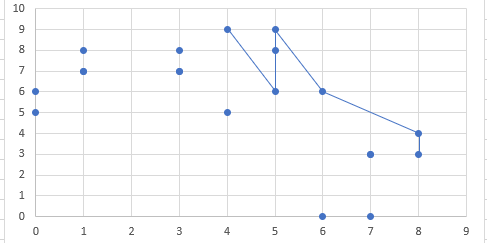
**Другий рядок:**



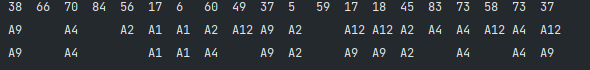
**Графічний розв’язок за Парето:**



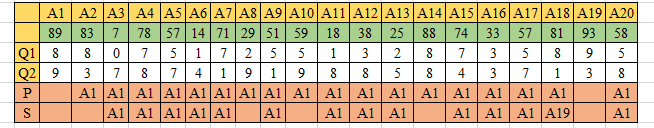
**Графічний розв’язок за Слейтером:**



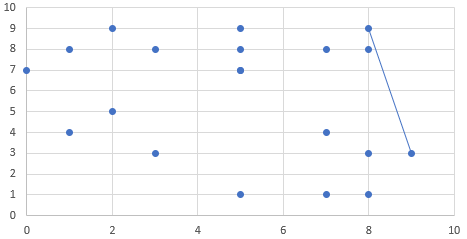
**Програмний розв’язок:**



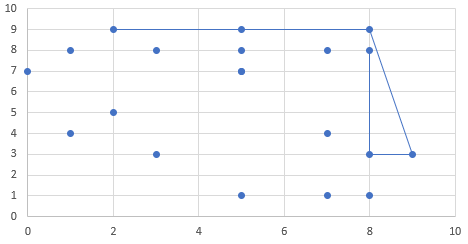
**Третій рядок:**



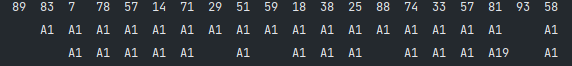
**Графічний розв’язок за Парето:**



**Графічний розв’язок за Слейтером:**

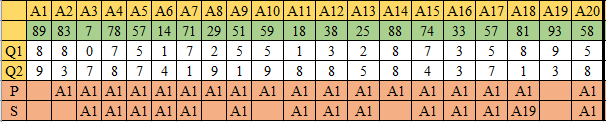


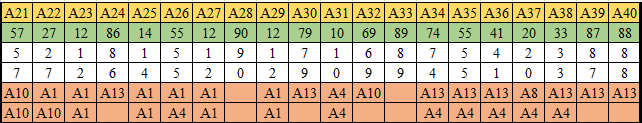
**Програмний розв’язок:**

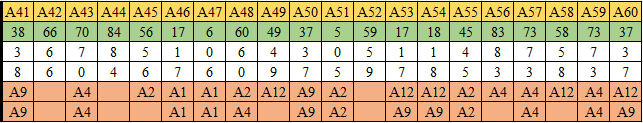


**Всі рядки:**

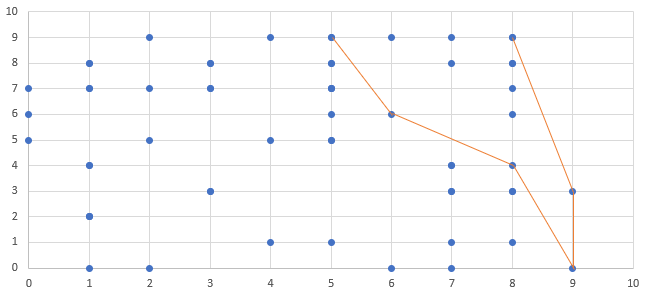
Заради полегшення обчислень я переставила третій рядок наперед, оскільки він має найбільший елемент, по якому більшість буде відсіюватися:



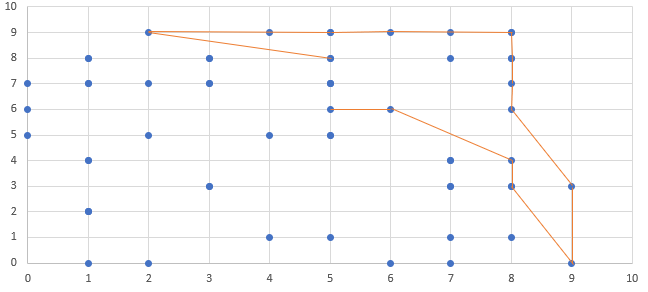




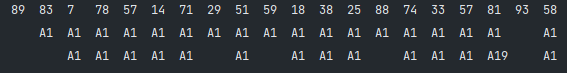
**Графічний розв’язок за Парето:**

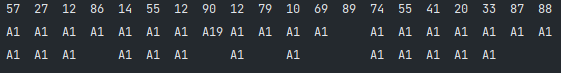


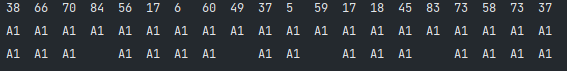
**Графічний розв’язок за Слейтером:**



**Програмний розв’язок:**







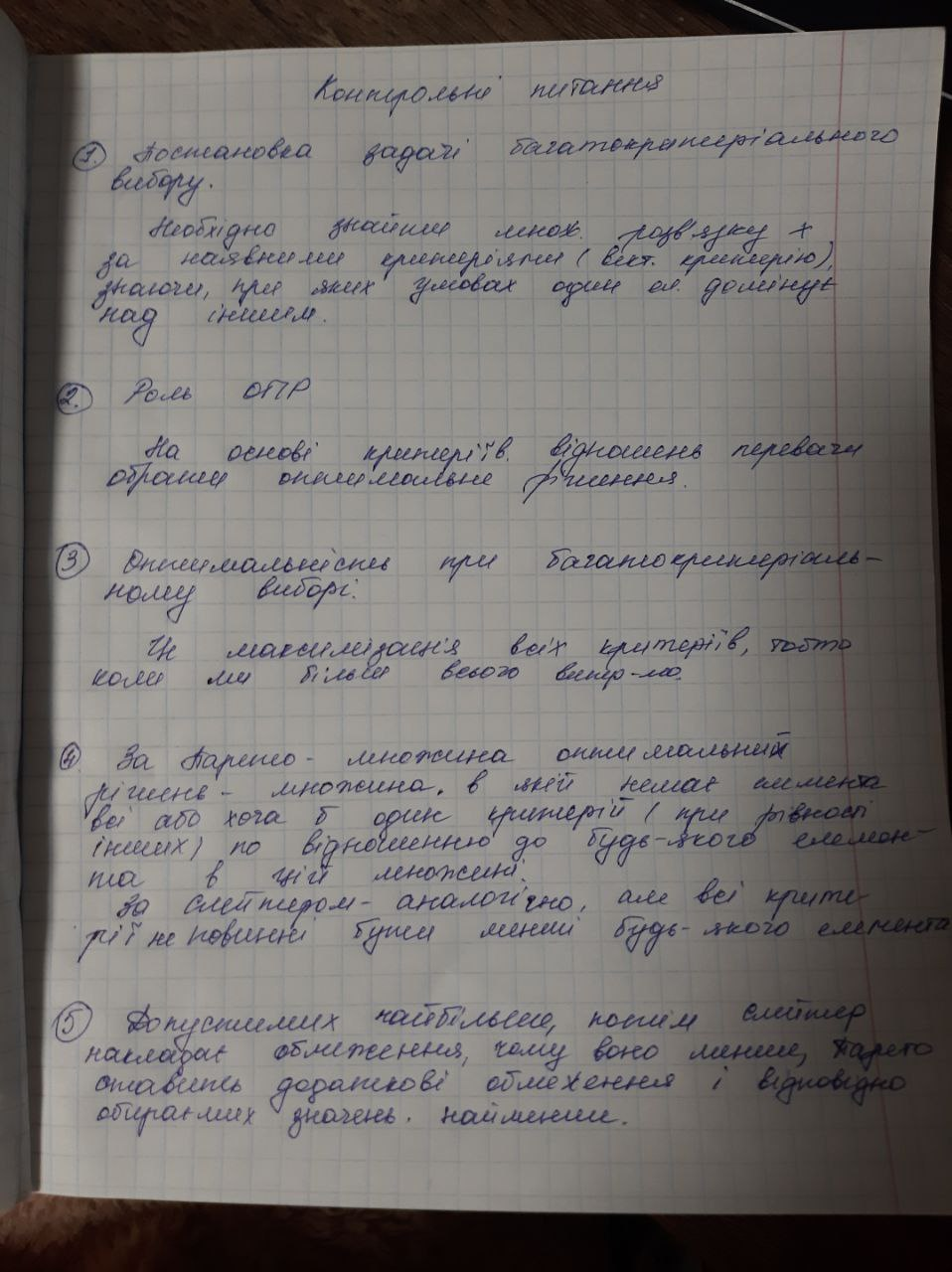
**Висновок:**

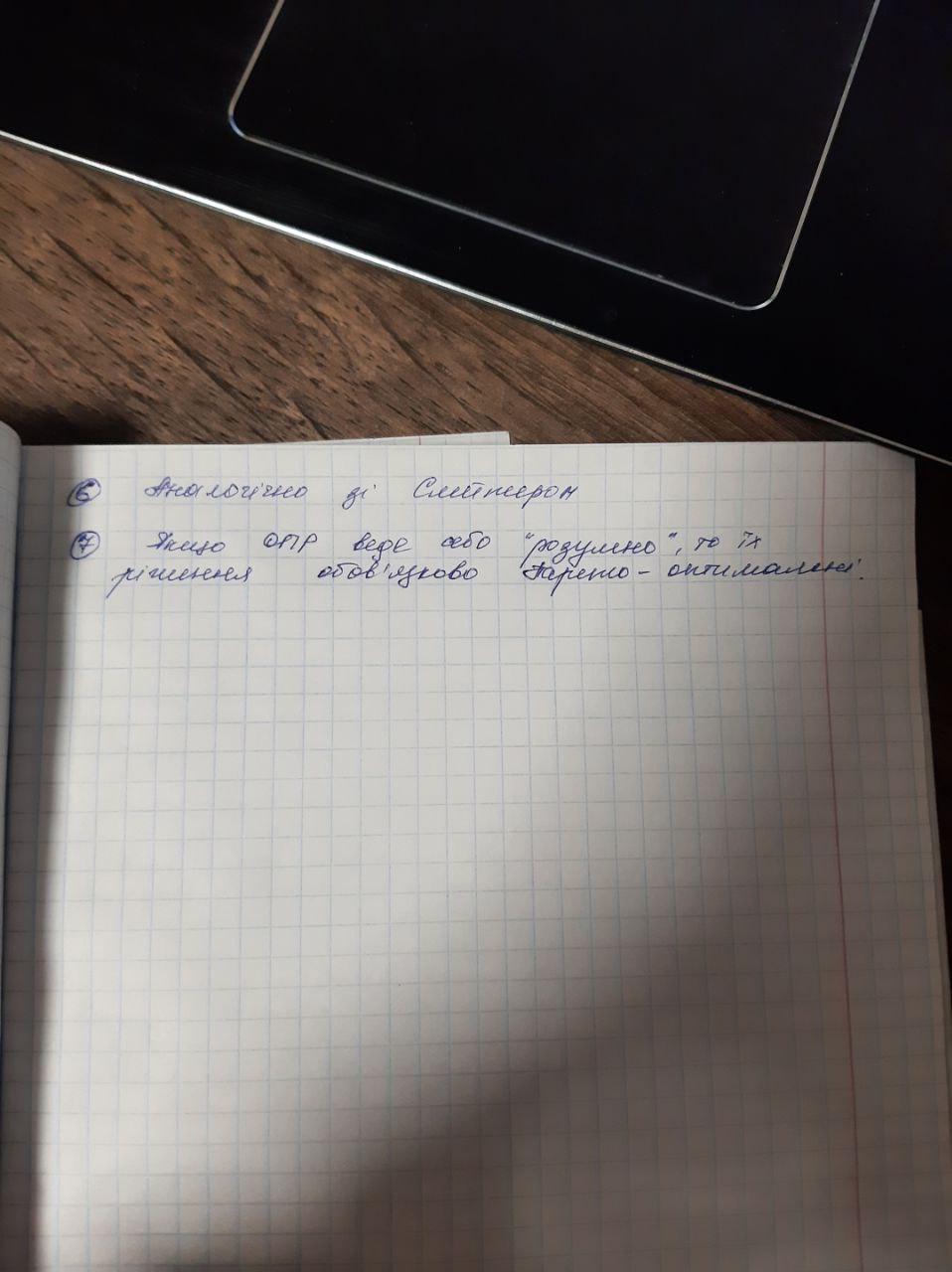
У результаті виконання даної лабораторної роботи мною було розглянуто два методи знаходження оптимального рішення: за Слейтером і за Парето. За Слейтером – у вихідній множині не повинно існувати елементів, які по всім критеріям менші за множини розв’язку. Парето має більш строгу умову – необхідно, щоб не було елемента, який по одному з критеріїв менший, а умови рівності інших критеріїв, та у випадку, коли всі критерії менші за інший елемент у множині результатів. За виконаним графічним розв’язком робимо висновок, що розв’язок є правильним.

У випадку з Парето немає можливості покращити критерій, не змінюючи інший, в який додаються елементи, в які входять критерії, які хоча б по одному не менші за інші.

Робимо висновок, що Парето накладає більше обмежень, що робить меншим кількість результатів, а Слейтер потребує менше порівнянь, тому він буде швидшим за Парето.

**Контрольні питання:**





**Лістинг:**

public class ParetoSleyter {

    static int[][] arr = new int[][]{

            {57, 27, 12, 86, 14, 55, 12, 90, 12, 79, 10, 69, 89, 74, 55, 41, 20, 33, 87, 88},

            {38, 66, 70, 84, 56, 17, 06, 60, 49, 37, 05, 59, 17, 18, 45, 83, 73, 58, 73, 37},

            {89, 83, 07, 78, 57, 14, 71, 29, 51, 59, 18, 38, 25, 88, 74, 33, 57, 81, 93, 58}};

    static int[] arr1 = new int[]{89, 83, 07, 78, 57, 14, 71, 29, 51, 59, 18, 38, 25, 88, 74, 33, 57, 81, 93, 58,

            57, 27, 12, 86, 14, 55, 12, 90, 12, 79, 10, 69, 89, 74, 55, 41, 20, 33, 87, 88,

            38, 66, 70, 84, 56, 17, 06, 60, 49, 37, 05, 59, 17, 18, 45, 83, 73, 58, 73, 37};

    //домінація по парето

    static boolean ParetoDominant(int first, int second) {

        if ((first / 10 >= second / 10 && first % 10 > second % 10) || (first / 10 > second / 10 && first % 10 >= second % 10))

            return true;

        return false;

    }

    //домінація по слейтеру

    static boolean SleyterDominant(int first, int second) {

        if (first / 10 > second / 10 && first % 10 > second % 10) return true;

        return false;

    }

    static int[] Pareto(int array[]) {

        int[] res = new int[array.length];

        for (int i = 1; i < array.length; i++) {//по одному перевіряємо вхідний массив

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

                if (res[i] == 0 && res[j] == 0) {

                    if (ParetoDominant(array[j], array[i])) {//перевіряємо чи з массиву результатів немає елемента

                        // який домінує над новим елементом

                        res[i] = j + 1;

                        //"вилучаємо елемент" з множини результатів

                        break;

                    }

                    if (ParetoDominant(array[i], array[j])) {

                        //навпаки з минулим

                        res[j] = i + 1;

                        //"вилучаємо елемент" з множини результатів

                    }

                }

            }

        }

        return res;

    }

    static int[] Sleyter(int array[]) {

        int[] res = new int[array.length];

        for (int i = 1; i < array.length; i++) {

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

                if (res[i] == 0 && res[j] == 0) {

                    if (SleyterDominant(array[j], array[i])) {//перевіряємо чи з массиву результатів немає елемента

                        // який домінує над новим елементом

                        res[i] = j + 1;

                        //"вилучаємо елемент" з множини результатів

                        break;

                    }

                    if (SleyterDominant(array[i], array[j])) {

                        //навпаки з минулим

                        res[j] = i + 1;

                        //"вилучаємо елемент" з множини результатів

                    }

                }

            }

        }

        return res;

    }

    public static void main(String[] args) {

        int[] ParetoRes = Pareto(arr1);

        int[] SlayterRes = Sleyter(arr1);

        for (int i = 0; i < 60; i++)

            System.out.print(arr1[i] + "\t");

        System.out.println();

        for (int i = 0; i < 60; i++) {

            if (ParetoRes[i] == 0)

                System.out.print(" " + "\t");

            else

                System.out.print("A" + (ParetoRes[i] + "") + "\t");

        }

        System.out.println();

        for (int i = 0; i < 60; i++) {

            if (SlayterRes[i] == 0)

                System.out.print(" " + "\t");

            else

                System.out.print("A" + (SlayterRes[i] + "") + "\t");

        }

    }

}