

Міністерство освіти і науки України  
Національний технічний університет України «Київський політехнічний  
інститут імені Ігоря Сікорського»  
Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

Звіт

з лабораторної роботи № 8 з дисципліни  
«Алгоритми та структури даних-1.  
Основи алгоритмізації»

«Дослідження алгоритмів пошуку та сортування»

Варіант 5

Виконав студент

ІП-15, Буяло Дмитро Олександрович  
(шифр, прізвище, ім'я, по батькові)

Перевірив

Вечерковська Анастасія Сергіївна  
(прізвище, ім'я, по батькові)

Київ 2021

## Лабораторна робота 8

### Дослідження алгоритмів пошуку та сортування

**Мета** – дослідити алгоритми пошуку та сортування, набути практичних навичок використання цих алгоритмів під час складання програмних специфікацій..

#### Індивідуальне завдання

#### Варіант 5

#### Завдання

Розробити алгоритм та написати програму, яка складається з наступних дій:

1. Опису змінної індексованого типу (двовимірний масив) згідно з варіантом
2. Ініціювання змінної, що описана в п.1 даного завдання
3. Створення нової змінної індексованого типу (одновимірний масив) та її ініціювання значеннями, що обчислюються згідно з варіантом

№	Розмірність	Тип даних	Обчислення значень елементів одновимірного масиву
5	$5 \times 7$	Цілий	Із суми значень елементів стовпців двовимірного масиву. Відсортувати методом бульбашки за зростанням.

#### 1. Постановка задачі

1) Термінологія в формуванні задачі повністю зрозуміла та не потребує пояснень. 2) Маємо розмірність матриці, тобто двовимірного масиву, та цілий тип даних значень елементів. 3) Необхідно з суми значень елементів стовпців двовимірного масиву створити одновимірний масив, використовуючи вхідні дані, користуючись методом бульбашки, маємо відсортувати масив. 4) Як загальну властивість можна виділити те, що в нас задана розмірність. 5) Існує багато розв'язків даної задачі, будемо

використовувати, на мою думку, найпростіший, найефективніший та найшвидший.

б) Даних цілком достатньо, всі потрібні та припущень робити не потрібно.

## 2. Побудова математичної моделі

Складемо таблицю імен змінних

<i><b>Змінна</b></i>	<i><b>Тип</b></i>	<i><b>Ім'я</b></i>	<i><b>Призначення</b></i>
Кількість рядків	Цілочисельний	m	Вхідні дані
Кількість стовпчиків	Цілочисельний	n	Вхідні дані
Двовимірний масив	Цілочисельний	matrix	Проміжні дані
Одновимірний масив	Цілочисельний	arr	Проміжні дані, результат
Лічильник 1	Цілочисельний	i	Проміжні дані
Лічильник 2	Цілочисельний	j	Проміжні дані
Генерація масиву	Процедура	createArr	Проміжні дані
Вивід матриці	Процедура	outputM	Проміжні дані
Сума елементів стовпців	Процедура	arrOfSum	Проміжні дані, результат
Вивід масиву	Процедура	outputA	Проміжні дані
Сортування масиву	Процедура	sortA	Проміжні дані, результат

Щоб обчислити значення елементів одновимірного масиву із суми значень елементів стовпців двовимірного масиву та відсортувати отриманий масив методом бульбашки за зростанням, нам потрібно застосувати підпрограми, в яких ми будемо генерувати двовимірний масив з заданою розмірністю, виводити в консоль отриману матрицю, знаходити суму елементів стовпців та додавати їх у масив, виводити отриманий масив та підпрограму для сортування цього масиву по методу бульбашки за зростанням.

Вхідними параметрами функції `createArr` є пустий двовимірний масив та його розмірність, підпрограма буде створювати масив-матрицю.

Вхідними параметрами функції `outputM` є двовимірний масив та його розмірність, підпрограма буде виводити у консоль матрицю, з якою ми працюємо.

Вхідними параметрами функції `arrOfSum` є пустий одновимірний масив, двовимірний масив та його розмірність, підпрограма рахує суму елементів у кожному стовпчику матриці та додає їх у одновимірний масив.

Вхідними параметрами функції `outputA` є масив та довжина масиву, підпрограма виводить одновимірний масив у консоль.

Вхідними параметрами функції `sortA` є масив та довжина масиву, підпрограма за методом бульбашки сортує масив у порядку зростання.

Всі виклики функцій відбуваються в основній програмі.

Також зазначимо, що оператор `+=` - це оператор складання з привласненням, додає значення правого операнда до змінної і привласнює змінній результат.

Функція `Rand()` повертає рівномірно розподілене випадкове дійсне число, яке більше або дорівнює 0 та менше за 1. Під час кожного обчислення аркуша повертається нове випадкове дійсне число.

### Розв’язання

Програмні специфікації запишемо у псевдокоді та графічній формі у вигляді блок-схеми.

*Крок 1.* Визначимо основні дії

*Крок 2.* Деталізуємо дію знаходження відсортованого масиву з елементів, що є сумами стовпців двовимірного масиву.

### 3. Псевдокод алгоритму

#### Основна програма

<i>Крок 1</i>	<i>Крок 2</i>
<b>Початок</b>	<b>Початок</b>
Задання розмірності	m=5;n=7
Створення та вивід двовимірного масиву	matrix[m][n]
	createArr(matrix, n,m)
	outputM(matrix,n,m)
<u>Створення та вивід масиву з елементами, як сума стовпців</u>	arr[n]
	arrOfSum(arr,matrix,n,m)
	outputA(arr,n)
<u>Сортування масиву</u>	sortA(arr,n)
Вивід масиву	outputA(arr,n)
<b>Кінець</b>	<b>Кінець</b>

**Підпрограми:**

<b>createArr(matrix, n, m)</b>
<b>Початок</b>
Повторити для <b>i</b> від 0 до <b>m</b>
Повторити для <b>j</b> від 0 до <b>n</b>
matrix[i][j] = Rand()
Все повторити
Все повторити
<b>Кінець</b>

<b>outputM(matrix, n, m)</b>
<b>Початок</b>
Повторити для <b>i</b> від 0 до <b>m</b>
Повторити для <b>j</b> від 0 до <b>n</b>
Виведення matrix[i][j]
Все повторити
Все повторити
<b>Кінець</b>

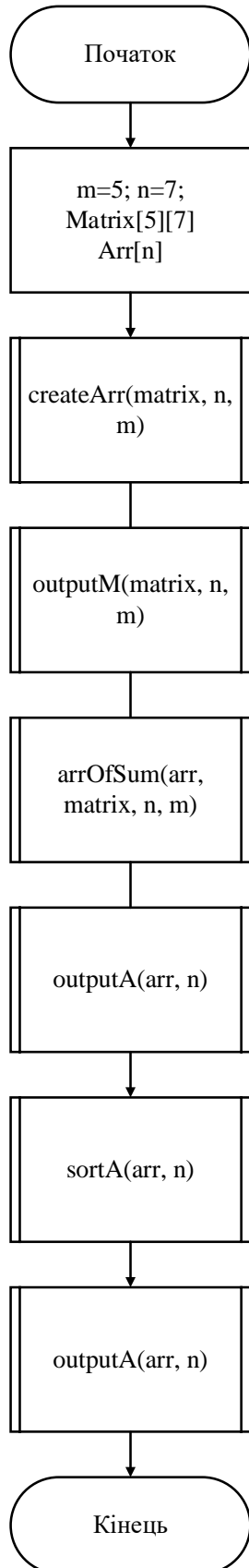
<b>outputA(arr, n)</b>
<b>Початок</b>
Повторити для <b>i</b> від 0 до <b>n</b>
Виведення arr[i]
Все повторити
<b>Кінець</b>

<b>arrOfSum(arr, matrix, n, m)</b>
<b>Початок</b>
Повторити для j від 0 до n
Повторити для i від 0 до m
arr[j] += matrix[i][j]
Все повторити
Все повторити
<b>Кінець</b>

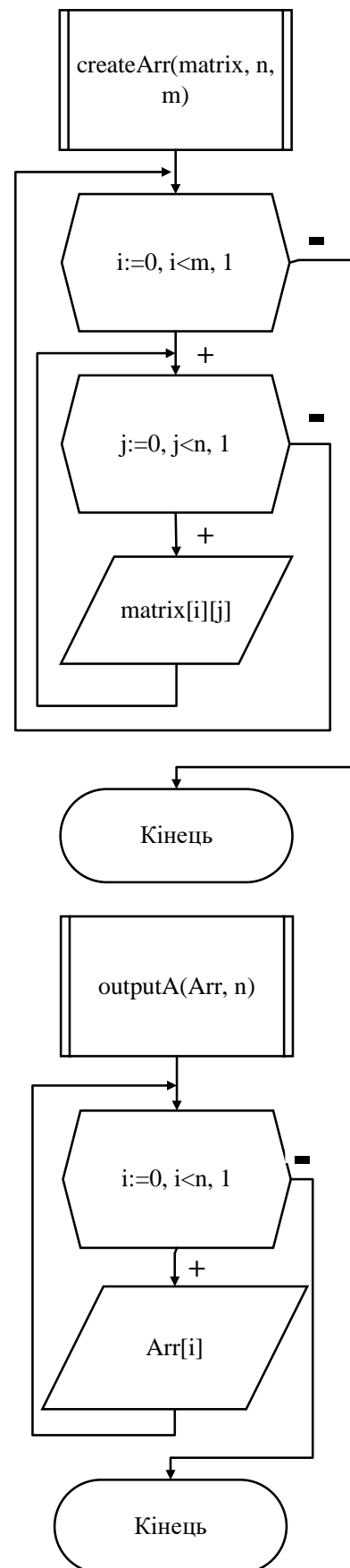
<b>sortA(arr, n)</b>
<b>Початок</b>
isSorted = false; buf=0
<b>Повторити</b>
<b>Поки</b> !isSorted
isSorted = true
<b>Повторити для i від 0 до n-1</b>
<b>Якщо</b> arr[i]>arr[i+1]
<b>то</b>
isSorted=false
buf = arr[i]
arr[i] = arr[i+1]
arr[i+1] = buf
<b>Все Якщо</b>
<b>Все повторити</b>
<b>Все повторити</b>
<b>Кінець</b>

#### 4. Блок-схема алгоритму

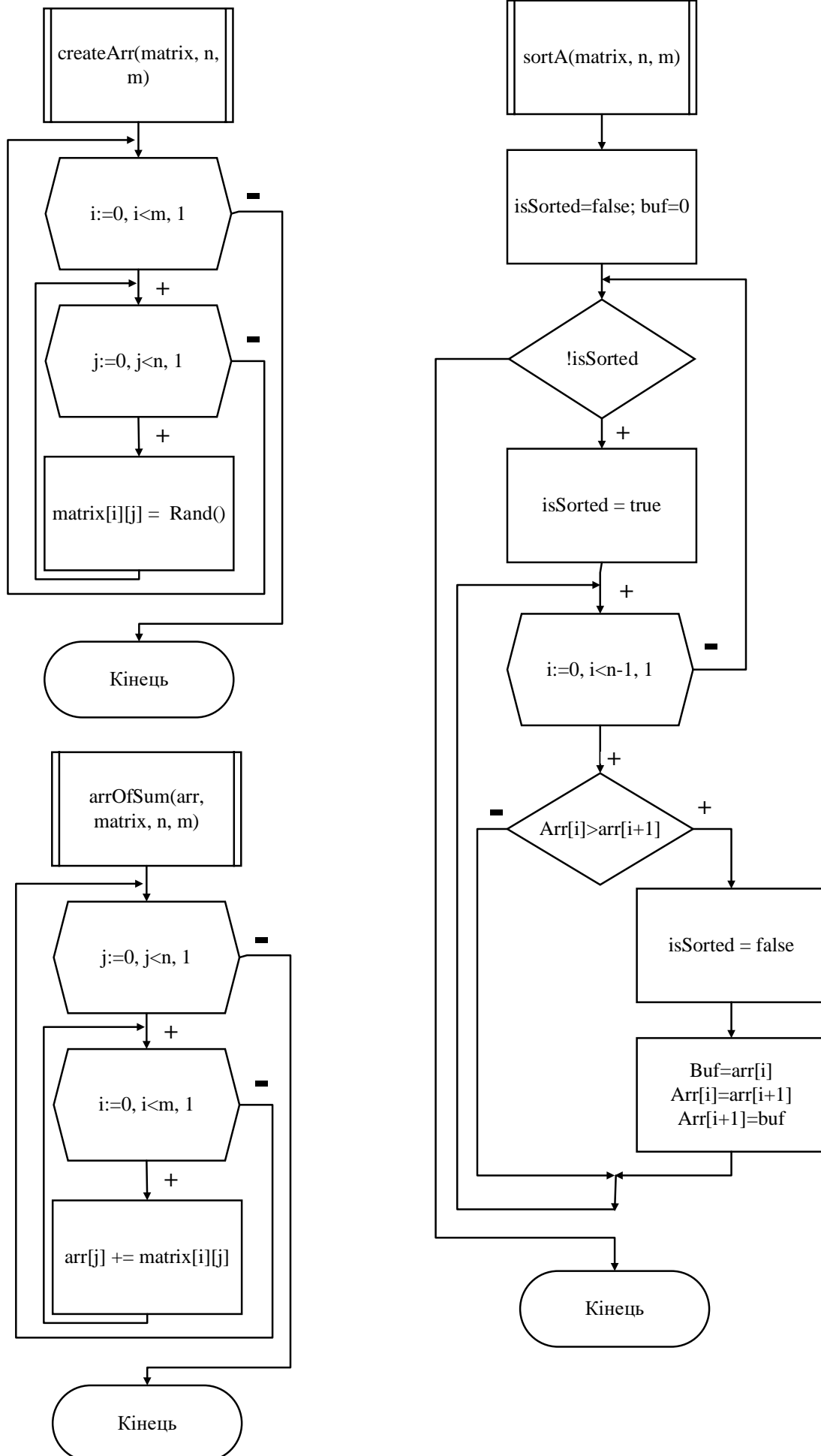
##### Основна програма



##### Підпрограми







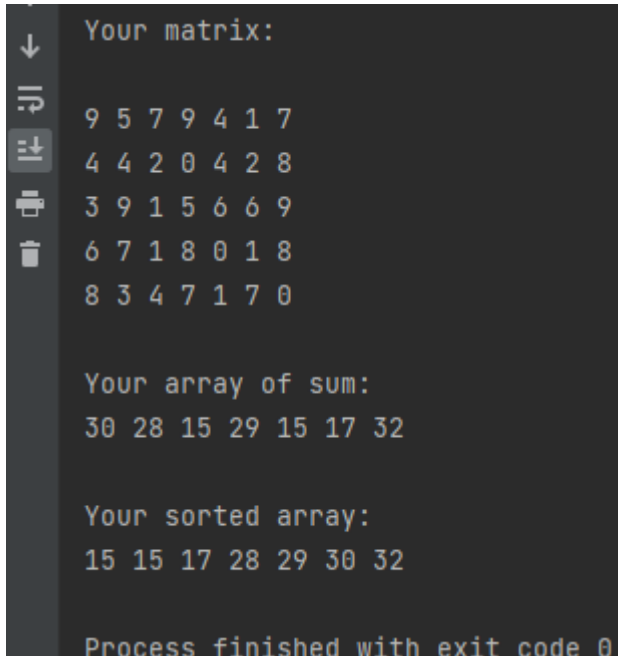
## 5. Код програми

Код написаний на мові програмування Java

```
public class Main {
    public static void main(String[] args) {
        int m=5;
        int n=7;
        int[][] matrix = new int[m][n];
        int[] arr=new int[n];
        createArr(matrix,n,m);
        System.out.println("Your matrix:");
        outputM(matrix,n,m);
        arrOfSum(arr,matrix,n,m);
        System.out.println("\nYour array of sum:");
        outputA(arr,n);
        sortA(arr,n);
        System.out.println("\nYour sorted array:");
        outputA(arr,n);
    }
    public static void createArr(int[][] matrix, int n, int m) {
        for(int i=0; i<m;i++) {
            for(int j=0;j<n;j++) {
                matrix[i][j] = (int) (Math.random()*10);
            }
        }
    }
    public static void outputM(int[][] matrix, int n, int m) {
        for(int i=0; i<m;i++) {
            System.out.println();
            for(int j=0;j<n;j++) {
                System.out.print(matrix[i][j]+" ");
            }
            System.out.println();
        }
    }
    public static void arrOfSum(int[] arr,int[][] matrix, int n, int m) {
        for(int j = 0;j<n;j++) {
            for(int i=0;i<m;i++) {
                arr[j]+=matrix[i][j];
            }
        }
    }
    public static void outputA(int[] arr,int n) {
        for(int i=0;i<n;i++) {
            System.out.print(arr[i]+" ");
        }
        System.out.println();
    }
    public static void sortA(int[] arr,int n) {
        boolean isSorted = false;
        int buf;
        while(!isSorted) {
            isSorted = true;
            for (int i = 0; i < n-1; i++) {
                if(arr[i] > arr[i+1]){
                    isSorted = false;

                    buf = arr[i];
                    arr[i] = arr[i+1];
                    arr[i+1] = buf;
                }
            }
        }
    }
}
```

Результат виконання:



```
Your matrix:
9 5 7 9 4 1 7
4 4 2 0 4 2 8
3 9 1 5 6 6 9
6 7 1 8 0 1 8
8 3 4 7 1 7 0

Your array of sum:
30 28 15 29 15 17 32

Your sorted array:
15 15 17 28 29 30 32

Process finished with exit code 0
```

## 6. Висновки

Ми дослідили алгоритми пошуку та сортування та набули практичних навичок використання цих алгоритмів під час складання програмних специфікацій. В результаті виконання лабораторної роботи ми отримали алгоритм для створення масиву, що складається з суми значень елементів стовпців двовимірного масиву., відсортували його методом бульбашки за зростанням. Дискретували задачу на 2 кроки: визначили основні дії, деталізували знаходження масиву та його сортування. Алгоритм є ефективним та результативним, бо забезпечує розв'язок за мінімальний час із мінімальними витратами ресурсів та отримує чіткий кінцевий результат.