

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

і

Звіт

з лабораторної роботи № 8 з дисципліни

«Алгоритми та структури даних-1.

Основи алгоритмізації»

«Дослідження алгоритмів пошуку та сортування»

Варіант 23

Виконав студент ІП-15, Мочалов Дмитро Юрійович

Перевірив Вечерковська Анастасія Сергіївна

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.

Варіант 23

Задача.

Завдання

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

1. Опису змінної індексованого типу (двовимірний масив) згідно з варіантом (табл. 1).
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 (табл. 1).

| | | | |
|----|-------|---------|--|
| 23 | 8 x 4 | Дійсний | Із добутку від'ємних значень елементів рядків двовимірного масиву. Відсортувати методом вставки за зростанням. |
|----|-------|---------|--|

Мат. модель

| Змінна | Тип | Ім'я | Призначення |
|---|---------|------------|-------------------|
| Розмірність рядків двовимірного масиву | цілий | n | К-ть елементів |
| Розмірність стовпців двовимірного масиву | цілий | m | К-ть елементів |
| Двовимірний масив | дійсний | arr | Проміжні значення |
| Зміння для розрахунку добутку від'ємних елементів рядків масиву | дійсний | dobutok | Проміжні значення |
| лічильник | цілий | k | Проміжні значення |
| Змінна для сортування | дійсний | tmp | Проміжні значення |
| Результатуючий масив | дійсний | arr_result | результат |

Таким чином, математичне модулювання зводиться до генерування масиву arr. Функція set_result_arr заповнює масив arr_result добутками від'ємних значень рядків масиву arr.

Функція sort сортує масив arr_result алгоритмом вставки за зростанням. Алгоритм вставки працює таким чином, що починаючи з другого елементу масиву вставляє його замість попереднього елементу масиву поки там є значення і вони більші за поточний елемент.

В розгалуженні “і” позначимо як &&.

Крок1: визначитись з алгоритмом

Крок2: деталізуємо алгоритм задання значень одновимірного масиву за допомогою підпрограми

Крок2: деталізуємо алгоритм сортування одновимірного масиву методом вставки за зростанням

Псевдокод

Крок1

Початок

деталізуємо алгоритм задання значень одновимірного масиву за допомогою

деталізуємо алгоритм сортування одновимірного масиву методом вставки за зростанням

Кінець

Крок2

Початок

$n = 8$

$m = 4$

`set_result_arr(arr, arr_result)`

деталізуємо алгоритм сортування одновимірного масиву методом вставки за зростанням

Кінець

Підпрограми:

set_result_arr(arr, arr_result)

початок

повторити для i від 0 до n

dobutok = 1

повторити для j від 0 до m

якщо arr[i][j] < 0 **то**

dobutok = dobutok * arr[i][j]

все якщо

все повторити

arr_result[i] = dobutok

все повторити

кінець

Крок3

Початок

n = 8

m = 4

set_result_arr(arr, arr_result)

sort(arr_result)

Кінець

Підпрограми:

set_result_arr(arr, arr_result)

початок

повторити для i від 0 до n

dobutok = 1

повторити для j від 0 до m

якщо arr[i][j] < 0 **то**

dobutok = dobutok * arr[i][j]

все якщо

все повторити

arr_result[i] = dobutok

все повторити

кінець

sort(arr_result)

початок

повторити для i від 0 до n

k = i

поки k>0 **i** a[k-1] > a[k]

tmp = a[k-1]

a[k-1] = a[k]

a[k] = tmp

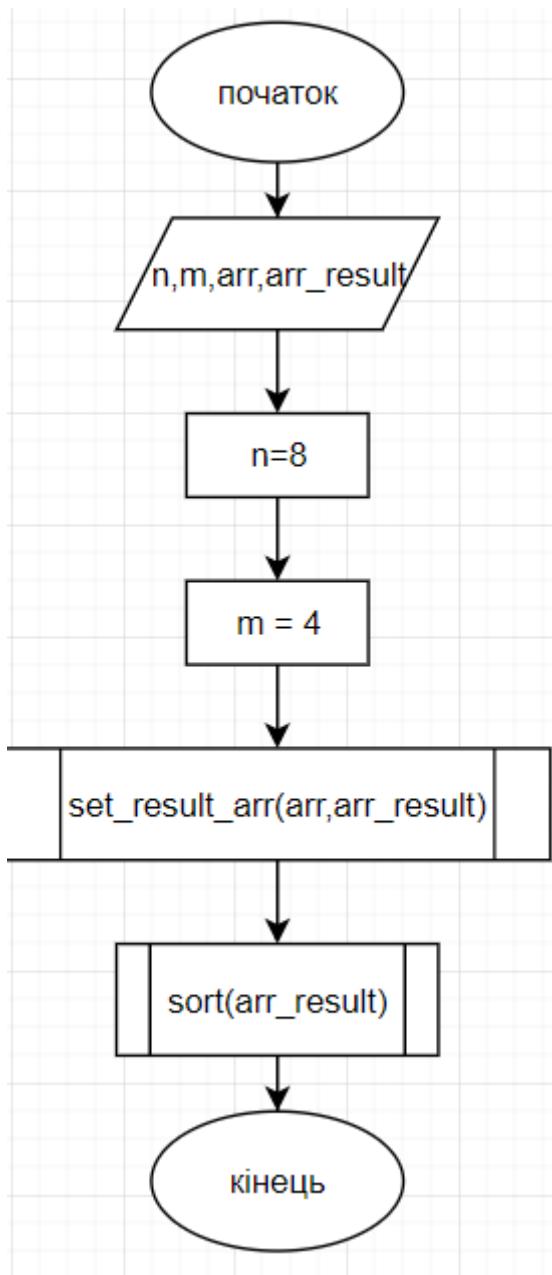
k = k-1

все поки

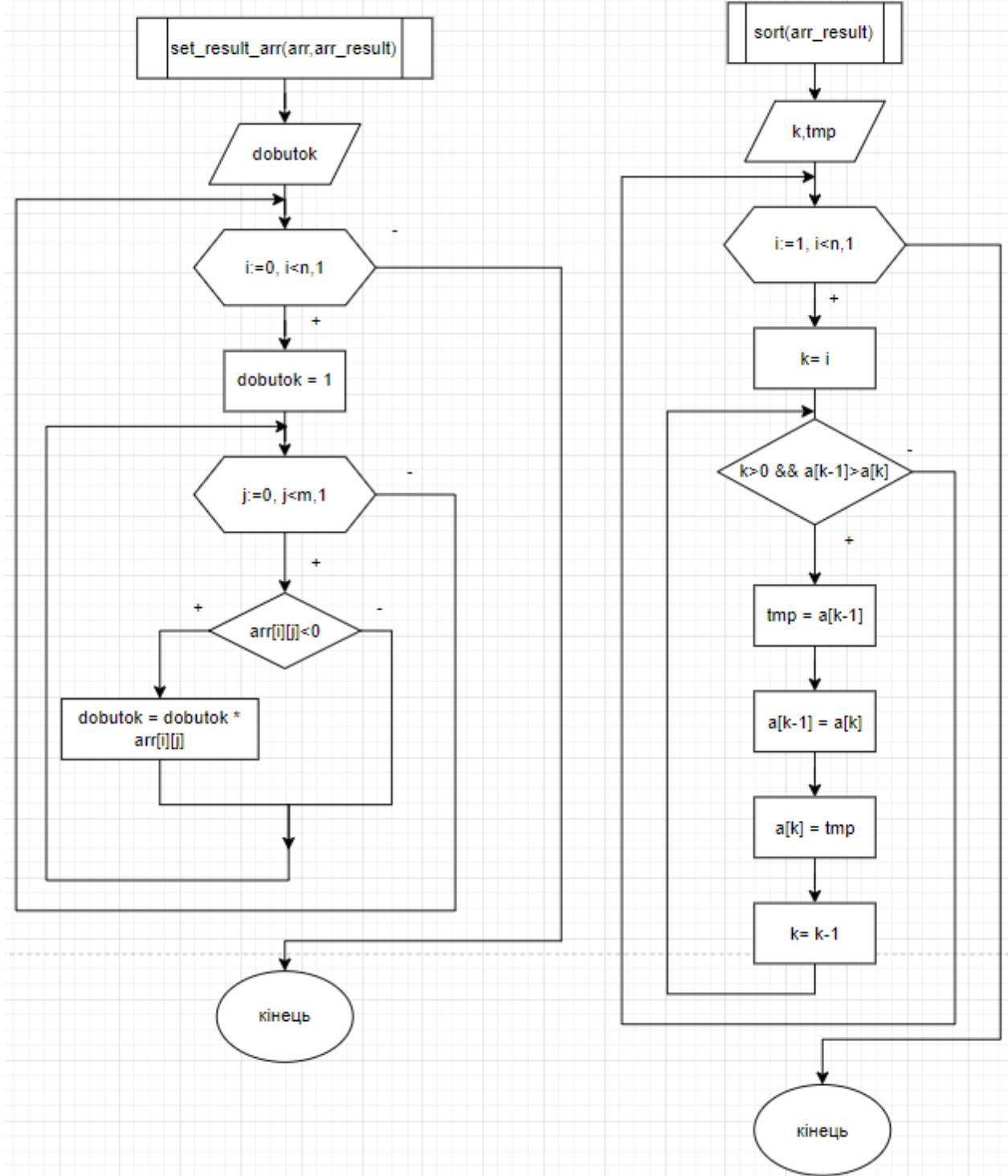
все повторити

кінець

Блок-схема



Підпрограми



Код

```
using System;

namespace Lab8
{
    internal class Program
    {
        public static void Main(string[] args)
        {
            int n = 8;
            int m = 4;
            double[,] arr = new double[n, m];
            double[] arr_result = new double[n];
            set_arr(arr);
            Console.WriteLine("Початковий масив:");
            for (int i = 0; i < n; i++)
            {
                for (int j = 0; j < m; j++)
                {
                    Console.Write(arr[i, j] + "\t");
                }
                Console.WriteLine();
            }
            set_result_arr(arr, arr_result);
            Console.WriteLine("Результатуючий масив:");
            for (int i = 0; i < n; i++)
            {
                Console.Write(arr_result[i] + "\t");
            }
            sort(arr_result);
            Console.WriteLine("\nВідсортований масив:");
            for (int i = 0; i < n; i++)
            {
                Console.Write(arr_result[i] + "\t");
            }
        }
    }
}
```

```
static void set_arr(double[,] arr)
{
    Random x = new Random();
    for (int i = 0; i < arr.GetLength( dimension: 0); i++)
    {
        for (int j = 0; j < arr.GetLength( dimension: 1); j++)
        {
            arr[i, j] = Math.Round(x.Next(-100, 100) + x.NextDouble(),2);
        }
    }
}

1 usage
static void set_result_arr(double[,] arr, double[] arr2)
{
    double dobutok = 1;
    for (int i = 0; i < arr.GetLength( dimension: 0); i++)
    {
        for (int j = 0; j < arr.GetLength( dimension: 1); j++)
        {
            if (arr[i, j] < 0) dobutok *= arr[i, j];
        }

        arr2[i] = Math.Round(dobutok,2);
        dobutok = 1;
    }
}

1 usage
static void sort(double[] a)
{
    for (int i = 1; i < a.Length; i++)
    {
        int k = i;
        while (k > 0 && a[k - 1] > a[k])
        {
            (a[k - 1], a[k]) = (a[k], a[k - 1]);
            k -= 1;
        }
    }
}
```

Тестування

Початковий масив:

```
-97    35,57   -6,5    -63,76
85,01  8,65    -60,12   -89,8
-39,94 41,63    -1,56     34
-74,77 -18,77   64,89   -16,08
37,79  -36,94   7,41    -13,29
86,09  -19,17   -75,85   68,81
10,76  -60,53   91,69    32,58
-33,32 -10,77   -80,05   -17,96
```

Результативний масив:

```
-40200,68      5398,78 62,31   -22567,2      490,93  1454,04 -60,53  515927,13
```

Відсортований масив:

```
-40200,68      -22567,2      -60,53  62,31   490,93  1454,04 5398,78 515927,13
```

```
Process finished with exit code 0.
```

Висновок: Ми дослідили алгоритми пошуку та сортування, набули практичних навичок використання цих алгоритмів під час складання програмних специфікацій.