

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 35

Виконав студент ІП-15, Шабанов Метін Шаміль огли
(шифр, прізвище, ім'я, по батькові)

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

Лабораторна робота 8

Дослідження складних циклічних алгоритмів

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

Індивідуальне завдання

Варіант 35

Умова задачі

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

8x4	Дійсний	Із добутку від'ємних значень елементів рядків двовимірного масиву. Відсортувати методом вставки за зростанням.
-----	---------	--

Постановка задачі

Відсортувати масив, який буде створено з добутку від'ємних значень з кожного рядка матриці.

Побудова математичної моделі

Змінна	Тип	Ім'я	Призначення
Генератор матриці	Підпрограма	matrixGenerator(matrix[,])	Функція
Перетворювач матриці в масив	Підпрограма	transformToArray(matrix[,], array[])	Функція
Сортування масиву методом вставки	Підпрограма	insertionSort (array[])	Функція
Кількість рядків матриці (відповідно – кількість елементів одновимірного масиву)	Ціле	size1	Проміжні дані
Кількість стовпців матриці	Ціле	size2	Проміжні дані

Матриця	Індексований	matrix[size1, size2]	Проміжні дані
Одновимірний масив	Індексований	array[size1]	Проміжні дані
Генерація випадкового числа	Метод	rand.Next()	Отримати випадкове значення
Довжина масиву	Метод	.Length	Отримати довжину масиву
Довжина матриці у відповідному вимірі	Метод	.GetLength()	Отримати довжину матриці у відповідному вимірі
Лічильники циклів для обробки масиву	Ціле додатне	i, j	Проміжні дані
Буфер	Дійсний	temp	Проміжні дані
Поточний елемент масиву у функції transformToArray	Ціле додатне	result	Проміжні дані

Для генерації матриці скористаємося `rand()`, яке вставимо у вкладені цикли для перебору рядків та стовпці матриці. Після цього ініціалізуємо змінну `result` як 1 і будемо множити від'ємні елементи кожного рядка матриці на нього, вкладаючи при цьому елементи у масив розміром `size1`. Далі відсортуємо отриманий одновимірний масив методом вставки за зростанням.

Розв'язання

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

Крок 1. Визначимо основні дії;

Крок 2. Ініціалізуємо `size1` і `size2`;

Крок 3. Деталізуємо створення матриці розмірності `[size1, size2]` та ініціалізацію елементів матриці випадковими значеннями у підпрограмі `matrixGenerator`;

Крок 4. Деталізуємо роботу та виклик функції `transformToArray`;

Крок 5. Деталізуємо роботу та виклик функції `insertionSort`;

Псевдокод

Основна програма

size1 = 8

size2 = 4

matrix [size1, size2]

matrixGenerator(matrix)

array[size1]

transformToArray(matrix, array)

insertionSort(array)

Підпрограма matrixGenerator (matrix[,])

повторити

для i від 0 до matrix.GetLength(0)

повторити

для j від 0 до matrix.GetLength(1)

matrix[i, j] = rand.Next(-100, 101) / 10

все повторити

все повторити

повернути matrix

Підпрограма transformToArray (matrix[,], array[])

повторити

для i від 0 до matrix.GetLength(0)

result = 1

повторити

для j від 0 до matrix.GetLength(1)

якщо matrix[i, j] < 0

ТО

result = result * matrix[i, j]

все якщо

все повторити

array[i] = result

все повторити

повернути array

Підпрограма insertionSort(array[])

повторити

для i від 1 до array.Length

temp = array[i]

j = i - 1

поки j >= 0 та array[j] > temp

array[j + 1] = array[j]

j--

все поки

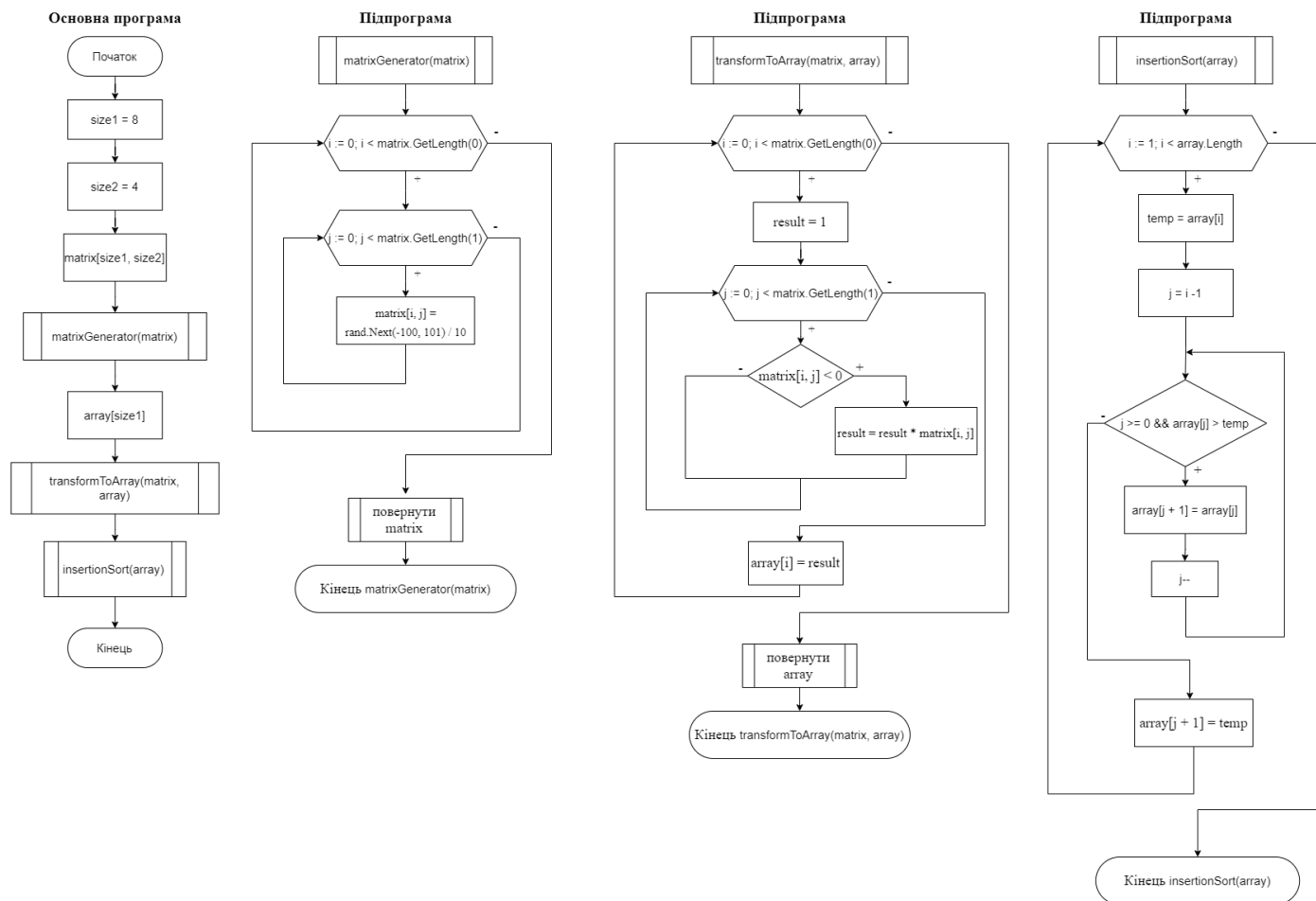
array[j + 1] = temp

все повторити

повернути array

Кінець

Блок схема



Код програми (C#)

```

1  using System;
2
3  namespace MKR
4  {
5      Ссылка: 0
6      class Program
7      {
8          Ссылка: 0
9          static void Main(string[] args)
10         {
11             int size1 = 8;
12             int size2 = 4;
13             double[,] matrix = new double[size1, size2];
14             Console.WriteLine("Matrix: ");
15             matrixGenerator(matrix);
16             printMatrix(matrix);
17             Console.WriteLine();
18
19             double[] array = new double[size1];
20             transformToArray(matrix, array);
21             Console.WriteLine("Unsorted array: ");
22             printArray(array);
23             Console.WriteLine();
24
25             insertionSort(array);
26             Console.WriteLine("Sorted array: ");
27             printArray(array);
28         }
29     }
30
31     Ссылка: 1
32     static double[,] matrixGenerator(double[,] matrix)
33     {
34         var rand = new Random();
35         for (int i = 0; i < matrix.GetLength(0); i++)
36         {
37             for (int j = 0; j < matrix.GetLength(1); j++)
38             {
39                 matrix[i, j] = Convert.ToDouble(rand.Next(-100, 101)) / 10;
40             }
41         }
42         return matrix;
43     }
44
45     Ссылка: 1
46     static void printMatrix(double[,] matrix)
47     {
48         for (int i = 0; i < matrix.GetLength(0); i++)
49         {
50             for (int j = 0; j < matrix.GetLength(1); j++)
51             {
52                 Console.Write("{0, 10}", matrix[i, j]);
53             }
54             Console.WriteLine("\n");
55         }
56     }
57
58     Ссылка: 1
59     static double[] transformToArray(double[,] matrix, double[] array)
60     {
61         double result;
62         for (int i = 0; i < matrix.GetLength(0); i++)
63         {
64             result = 1;

```

```

59         for (int j = 0; j < matrix.GetLength(1); j++)
60         {
61             if (matrix[i, j] < 0)
62             {
63                 result = result * matrix[i, j];
64             }
65         }
66         array[i] = Math.Round(result, 1);
67     }
68
69     return array;
70 }
71
72 Ссылка: 2
73 static void printArray(double[] array)
74 {
75     for (int i = 0; i < array.Length; i++)
76     {
77         Console.Write("{0, 8}", array[i]);
78     }
79     Console.WriteLine();
80 }
81
82 Ссылка: 1
83 static double[] insertionSort(double[] array)
84 {
85     for (int i = 1; i < array.Length; i++)
86     {
87         double temp = array[i];
88         int j = i - 1;
89         while (j >= 0 && array[j] > temp)

```

```

89     {
90         array[j + 1] = array[j];
91         j--;
92     }
93     array[j + 1] = temp;
94 }
95 return array;
96 }
97 }
98 }
99

```

Випробовування

Консоль отладки Microsoft Visual Studio

Matrix:

8,2	-5,3	6,7	5,1
-0,3	2,1	-2,1	6,3
-1,6	-1,8	3,5	5
-1,9	4,4	9,8	-3,9
-3,6	4,1	-2	6,4
-9,9	7,3	2,1	-3,7
1,4	-6,6	2,7	4,5
-1,5	-7,7	7,9	-9,7

Unsorted array:

-5,3	0,6	2,9	7,4	7,2	36,6	-6,6	-112
------	-----	-----	-----	-----	------	------	------

Sorted array:

-112	-6,6	-5,3	0,6	2,9	7,2	7,4	36,6
------	------	------	-----	-----	-----	-----	------

C:\Users\mitya\source\repos\ConsoleApp1\MKR\bin\Debug\net5.0\MKR.exe (процесс 19348) завершил работу с кодом 0.
 Чтобы автоматически закрывать консоль при остановке отладки, включите параметр "Сервис" -> "Параметры" -> "Отладка" -> "Автоматически закрыть консоль при остановке отладки".
 Нажмите любую клавишу, чтобы закрыть это окно...

Висновки

Ми дослідили алгоритми пошуку та сортування, набути практичних навичок використання цих алгоритмів під час складання програмних специфікацій.. Як результат, ми отримали алгоритм сортування масиву, який буде створено з добутку від'ємних значень з кожного рядка матриці, розділивши задачу на п'ять кроків: визначення основних дій, ініціалізація `size1` і `size2`, деталізація створення матриці розмірності `[size1, size2]` та ініціалізація елементів матриці випадковими значеннями у підпрограмі `matrixGenerator`, деталізація роботи та виклику функції `transformToArray`, деталізація роботи та виклику функції `insertionSort`. В процесі випробовування ми отримали відсортований масив.