**Міністерство Освіти І НАУКИ України**

**Національний університет "Львівська політехніка"**

Інститут **КНІТ**

Кафедра **ПЗ**

### ЗВІТ

До лабораторної роботи № 1

**З дисципліни:** *“Алгоритми та структури даних”*

**На тему:** *“Метод сортування бульбашкою”*

**Лектор:**

доц. каф. ПЗ

Коротєєва Т.О.

**Виконала:**

ст. групи ПЗ-26

Матолінець Л. А.

**Прийняв:**

асистент каф. ПЗ

Симець І.І.

« \_\_\_\_ » \_\_\_\_\_\_\_\_ 2022 р.

∑= \_\_\_\_\_ .

Львів – 2022

**Тема роботи:** Метод сортування бульбашкою

**Мета роботи:** Вивчити алгоритм сортування бульбашкою. Здійснити програмну реалізацію алгоритму сортування бульбашкою. Дослідити швидкодію алгоритму.

**TЕОРЕТИЧНІ ВІДОМОСТІ**

**Покроковий опис роботи алгоритму сортування бульбашкою з використанням прапорця.**

**Алгоритм FBS** реорганізує масив елементів, що містить n елементів, у зростаючому порядку, тобто для його елементів буде мати місце співвідношення Ri < Rj - для всіх i,j=1..n. flag – прапорець перестановок.

FВS1.Цикл за індексом проходження.

Повторювати кроки FВ2 і FВ3 при i=1..n-1.

FВS2. Ініціалізація прапорця перестановки: встановити flag=0.

FВS3. Виконання проходження. Повторювати при j=1,2,…,n-i : якщо Rj < Rj-1 , то встановити переставити місцями елементи Rj-1<->Rj, та flag=1 ; якщо flag=0, то завершити виконання алгоритму.

FВS4. Кінець. Вихід.

**ІНДИВІДУАЛЬНЕ ЗАВДАННЯ**

**Варіант 1**

Задано одномірний масив дійсних чисел. Виключити з нього мінімальний та максимальний елементи. Отриманий масив посортувати в порядку спадання.

**ХІД ВИКОНАННЯ**

Метод сортування бульбашкою та бульбашкою з прапорцем:

using System;

using System.Diagnostics;

using System.Threading;

public class HelloWorld

{

public static void Main(string[] args)

{

int Min = -50000;

int Max = 50000;

Console.WriteLine("Enter the size of your array: ");

int length = Convert.ToInt32(Console.ReadLine());

int[] unsortedArr = new int[length];

int[] unsortedArr2 = new int[length];

Random randNum = new Random();

for (int i = 0; i < length; i++)

{

unsortedArr[i] = randNum.Next(Min, Max);

unsortedArr2[i] = randNum.Next(Min, Max);

}

int[] newArray = flaggedBubbleSort(unsortedArr);

int[] newArray1 = bubbleSort(unsortedArr2);

}

public static int[] flaggedBubbleSort(int[] array){

Stopwatch stopwatch = new Stopwatch();

stopwatch.Start();

int flag;

for(int i = 0; i < array.Length-1; i++){

flag = 0;

for(int j = 1; j < array.Length-i; j++){

if(array[j] < array[j-1]){

swap(array, j-1, j);

flag = 1;

}

}

if(flag==0)break;

}

stopwatch.Stop();

Console.WriteLine("Time for flagged bubble sort is {0} ms", stopwatch.ElapsedMilliseconds);

return array;

}

public static int[] bubbleSort(int[] array){

Stopwatch stopwatch = new Stopwatch();

stopwatch.Start();

for(int i = 0; i < array.Length; i++){

for(int j = 1; j < array.Length; j++){

if(array[j] < array[j-1]){

swap(array, j-1, j);

}

}

}

stopwatch.Stop();

Console.WriteLine("Time for standart bubble sort is {0} ms", stopwatch.ElapsedMilliseconds);

return array;

}

public static void swap(int[] array, int x, int x1){

int temp = array[x1];

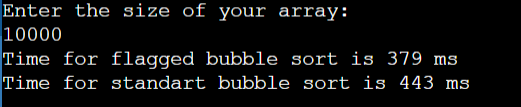
array[x1] = array[x];

array[x] = temp;

}

}

Метод сортування бульбашкою:



*Рис. 1 Результат роботи програми*

Код виконання індивідуального завдання:

// MainWindow.xaml.cs

using System;

using System.Collections.Generic;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Documents;

namespace SortingApp

{

/// <summary>

/// Interaction logic for MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

MyArray digitsArr = new MyArray();

public MainWindow()

{

InitializeComponent();

}

private void addArray\_Click(object sender, RoutedEventArgs e)

{

String arrField = arrayField.Text;

String[] strings = arrField.Split(" ; ");

foreach (var item in strings)

{

double x = Convert.ToDouble(item);

digitsArr.addElement(x);

}

}

public static double[] bubbleSort(double[] array, TextBox textBox)

{

textBox.Text = "";

int flag;

String line = "";

for (int i = 0; i < array.Length - 1; i++)

{

flag = 0;

line += i + ": ";

for (int j = 1; j < array.Length; j++)

{

if (j == 1)

{

foreach (var item in array)

{

line += Math.Round(item,2);

line += " ; ";

}

}

if (array[j] > array[j - 1])

{

swap(array, j - 1, j);

flag = 1;

}

}

line += "\n\n";

textBox.Text += line;

line = "";

if (flag == 0) break;

}

return array;

}

public static void swap(double[] array, int x, int x1)

{

double temp = array[x1];

array[x1] = array[x];

array[x] = temp;

}

private void Button\_Click(object sender, RoutedEventArgs e)

{

double minElem = digitsArr.findMinElement();

double maxElem = digitsArr.findMaxlement();

digitsArr.removeMaxAndMin(maxElem, minElem);

double[] arr = digitsArr.getList();

double[] sortedArray = bubbleSort(arr, stepsArea);

String result = "";

foreach (var item in sortedArray)

{

result += Convert.ToString(Math.Round(item, 1));

result += " ; ";

}

resultField.Content = result;

minElement.Content = minElem;

maxElement.Content = maxElem;

}

private void showSteps\_Click(object sender, RoutedEventArgs e)

{

}

}

}

//myArray.cs

using System;

using System.Collections.Generic;

using System.Linq;

namespace SortingApp

{

public class MyArray

{

protected double[] array = new double[0];

public MyArray() { }

public double[] getList() { return array; }

public void addElement(double item) {

Array.Resize(ref array, array.Length + 1);

array[array.Length-1] = item;

}

public double findMinElement()

{

double min = array[0];

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

{

if (array[i] < min) min = array[i];

}

return min;

}

public double findMaxlement()

{

double max = array[0];

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

{

if (array[i] > max) max = array[i];

}

return max;

}

public void removeMaxAndMin(double max, double min)

{

for(int i = 0; i < array.Length; i++)

{

if (array[i] == min || array[i] == max)

{

array = array.Where((source, index) => index != i).ToArray();

}

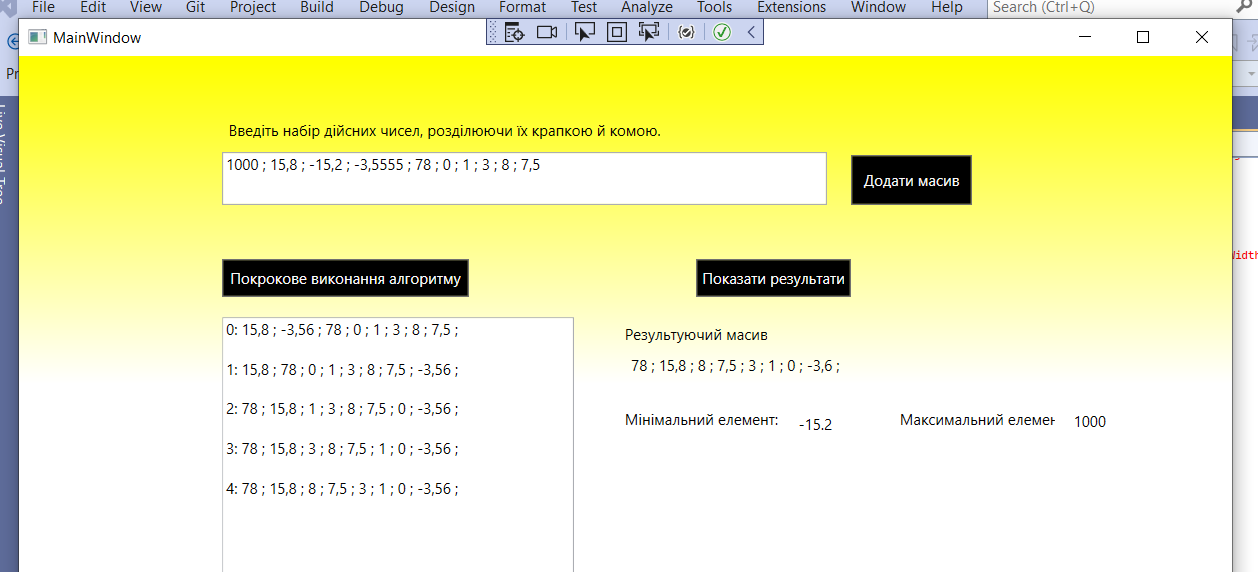
}

}

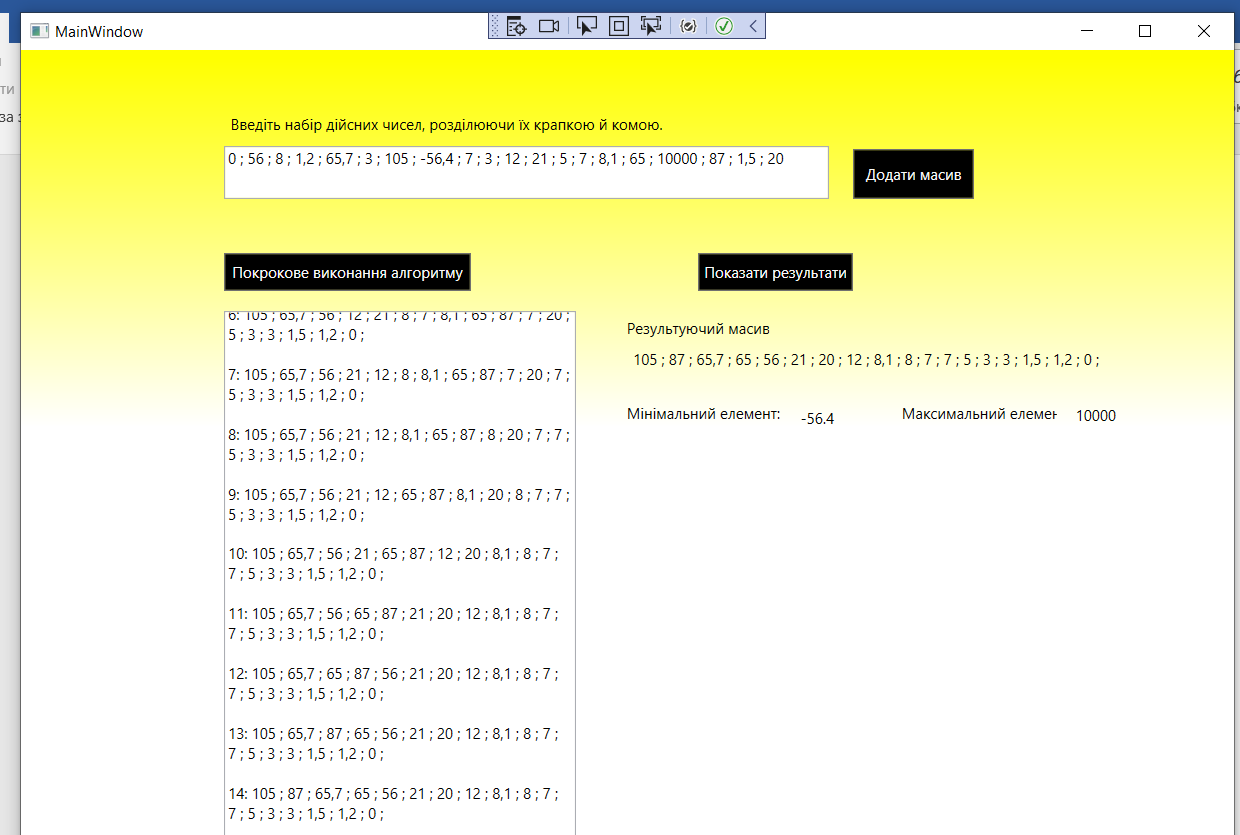
}

}

Результати виконання програми



*Рис. 2 Обробка та сортування десяти елементів*

**

*Рис. 3 Обробка та сортування 20ти елементів*

**Висновок**

На цьому лабораторному занятті я працювала з алгоритмом сортування бульбашкою (складність якого O(n2)) та його модифікацію - метод бульбашки з прапорцем. Створивши віконний застосунок, який дозволив мені використати цей алгоритм відносно дійсних чисел, я порівняла його швидкодію на декількох наборах даних. Як наслідок, я переконалася, що цей метод сортування значно зменшує свою швидкодію зі зростанням обсягу даних, які він має обробляти. Також стандартний алгоритм сортування бульбашкою є значно повільнішим, ніж його оптимізований варіант.