**Міністерство Освіти І НАУКИ України**

**Національний університет "Львівська політехніка"**

**IКНІТ**

Кафедра **ПЗ**

### ЗВІТ

До лабораторної роботи № 6

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

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

**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 підрахунком**.**

**Мета роботи**:  Вивчити [алгоритм](http://vns.lp.edu.ua/mod/glossary/showentry.php?courseid=472&eid=19102&displayformat=dictionary" \o "Глосарій: Алгоритм) сортування підрахунком. Здійснити програмну реалізацію [алгоритм](http://vns.lp.edu.ua/mod/glossary/showentry.php?courseid=472&eid=19102&displayformat=dictionary" \o "Глосарій: Алгоритм)у сортування підрахунком. Дослідити швидкодію [алгоритм](http://vns.lp.edu.ua/mod/glossary/showentry.php?courseid=472&eid=19102&displayformat=dictionary" \o "Глосарій: Алгоритм)у сортування підрахунком.

**TЕОРЕТИЧНІ ВІДОМОСТІ**

***Покроковий опис роботи алгоритму сортування підрахунком.***

**Алгоритм СS** Задано одновимірний масив елементів *R1 ,R2 ,…,Rn.* Даний алгоритм реорганізує масив у висхідному порядку таким чином, що створює додатковий масив countArray для збереження кількості входжень елемента в масив, сортує значення та записує їх у endArray.

CS1. Отримуємо значення мінімального та максимального елемента масиву – min, max, спектр їх значень range = max-min +1 – довжина масиву для countArray.

CS2. Ініціалізація countArray, endArray.

CS3. Проходження по масиву countArray та запис кількості входжень кожного елемента: countArray[arr[i] - min]++, поки i < arr.Length;

CS4. Проходження по масиву countArray, визначення останньої можливої позиції елемента: countArray[i] += countArray[i - 1];

CS5. Запис масиву endArray, визначення правильної позиції кожного елемента, починаючи визначення з кінця початкового масиву:

endArray[countArray[arr[i] - minValue] - 1] = arr[i];

CS6. Присвоєння нових значень початковому масиву: arr[i] = endArray[i];

CS7. Кінець функції.

***Покроковий опис роботи алгоритму індивідуального завдання.***

Алгоритм TransformNedative полягає у тому, щоб до всіх від’ємних елементів масиву застосувати функцію y = sin(x), де arr – початковий масив.

TN1. Ініціалізація індексу i;

TN1. Цикл проходження по масиву за індексом i поки i < arr.Length;

TN2. Перевірка значення елемента масиву: якщо arr[i] < 0, -> TN3. Якщо arr[i] > 0, i++ -> TN1;

TN3. arr[i] = Math.Sin(arr[i]), і++; Перехід на крок TN1.

TN4. Кінець функції.

**ІНДИВІДУАЛЬНЕ ЗАВДАННЯ**

**Варіант 6**

Задано одномірний масив дійсних чисел. До від’ємних елементів масиву застосувати функцію sin(x) . Отриманий масив посортувати в порядку спадання.

**ХІД ВИКОНАННЯ**

Метод сортування підрахунком:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Diagnostics;

using System.Threading;

class CountingSort

{

static void countSort(int[] arr)

{

int maxValue = arr.Max();

int minValue = arr.Min();

int rangeOfValues = maxValue - minValue + 1;

int []countArray = new int[rangeOfValues];

int []endArray = new int[arr.Length];

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

countArray[i] = 0;

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

countArray[arr[i] - minValue]++;

}

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

countArray[i] += countArray[i - 1];

}

for (int i = arr.Length - 1; i >= 0; i--) {

endArray[countArray[arr[i] - minValue] - 1] = arr[i];

countArray[arr[i] - minValue]--;

}

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

arr[i] = endArray[i];

}

}

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[] arr = new int[length];

Random randNum = new Random();

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

{

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

}

Stopwatch stopwatch = new Stopwatch();

stopwatch.Start();

countSort(arr);

stopwatch.Stop();

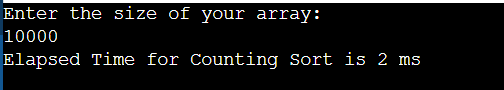
Console.WriteLine("Elapsed Time for Counting Sort is {0} ms", stopwatch.ElapsedMilliseconds);

}

}Console.WriteLine("Elapsed Time for Counting Sort is {0} ms", stopwatch.ElapsedMilliseconds);

}

}



*Рис.1 Результат виконання програми*

Вихідний код виконання індивідуального завдання:

//myArray.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Windows.Documents;

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 void setArray(double[] arr)

{

this.array = arr;

}

public void transformValues()

{

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

{

if (array[i] < 0)

{

double temp = Math.Sin(array[i]);

array[i] = temp;

}

}

}

}

}

//MainWindow.xaml.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Numerics;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

namespace SortingApp

{

/// <summary>

/// Interaction logic for MainWindow.xaml

/// </summary>

///

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

}

private static int[] countSort(int[] arr)

{

int maxValue = arr.Max();

int minValue = arr.Min();

int rangeOfValues = maxValue - minValue + 1;

int[] countArray = new int[rangeOfValues];

int[] endArray = new int[arr.Length];

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

{

countArray[arr[i] - minValue]++;

}

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

{

countArray[i] += countArray[i - 1];

}

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

endArray[countArray[arr[i] - minValue] - 1] = arr[i];

countArray[arr[i] - minValue]++;

}

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

{

arr[i] = endArray[i];

}

return endArray;

}

MyArray array = new MyArray();

int[] arrForSorting;

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

{

String arrField = column\_field.Text;

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

arrForSorting = new int[strings.Length];

foreach (var item in strings)

{

double x = Convert.ToDouble(item);

array.addElement(x);

}

}

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

{

array.transformValues();

double[] list = array.getList();

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

{

arrForSorting[i] = (int)(list[i] \* 100);

}

int[] sortedArr = countSort(arrForSorting);

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

{

double val = sortedArr[i];

list[i] = val/100;

}

array.setArray(list);

String result = "";

foreach (var item in list)

{

result += Convert.ToString(item);

result += " ; ";

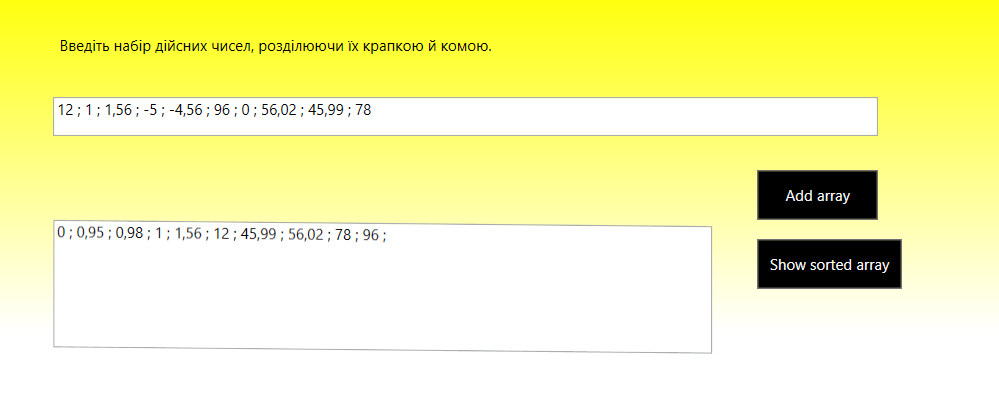
}

showedArray.Text = result;

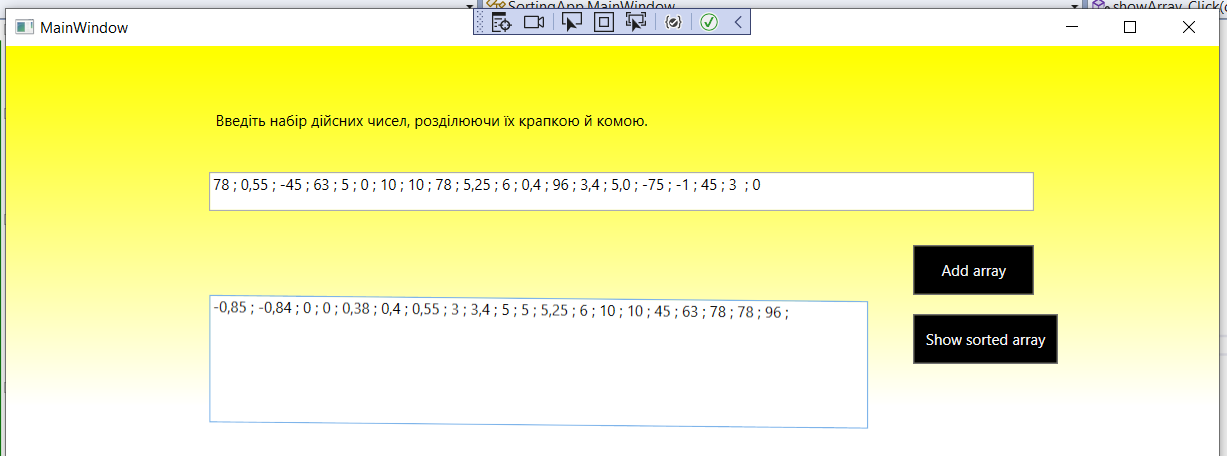
}

}

}



*Рис. 1 Результат сортування масиву з 10ти елементів*



*Рис. 2 Результат сортування масиву з 20ти елементів*

**Висновок**

На цій лабораторній роботі я познайомилася з алгоритмом сортування підрахунком, складність якого – O(N + K), де N – довжина масиву, K – величина діапазону. В результаті я реалізувала алгоритм, який працює з масивом цілих значень як від’ємних так і додатних і виконала індивідуальне завдання, в якому до всіх від’ємних чисел застосувала функцію y = sin(x).