**Міністерство Освіти І НАУКИ України**

**Національний університет "Львівська політехніка"**

**КНІТ**

Кафедра **ПЗ**

### ЗВІТ

До лабораторної роботи № 2

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

**На тему:** *“Метод сортування простою вибіркою”*

**Лектор:**

доц. каф. ПЗ

Коротєєва Т.О.

**Виконала:**

ст. групи ПЗ-26

Матолінець Л. А.

**Прийняв:**

доц. каф. ПЗ

Симець І.І.

« \_\_\_\_ » \_\_\_\_\_\_\_\_ 2022 р.

∑= \_\_\_\_\_ .

Львів – 2022

**Тема роботи:** Метод сортування простою вибіркою**.**

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

**TЕОРЕТИЧНІ ВІДОМОСТІ**

***Покроковий опис роботи алгоритму сортування простою вибіркою.***

**Алгоритм SS** Задано масив елементів *R1 ,R2 ,…,Rn.* Даний алгоритм реорганізує масив у висхідному порядку, таким чином для його елементів буде мати місце співвідношення *Ri < Rj -* для всіх *i,j=1..n.*

S1. Цикл за індексом проходження. Повторювати кроки S2-S4 при i=1..n-1.

S2. Зафіксувати перший поточний елемент: встановити R0 = Ri.

S3.Пошук найменшого значення min Rj для елементів з індексом j=i+1,i+2,…,n.

S4. Перестановка елементів. Якщо min Rj < R0 та j !=i , то min Rj <-> R0.

S5. Кінець. Вихід.

**ІНДИВІДУАЛЬНЕ ЗАВДАННЯ**

**Варіант 2**

Задано перелік товарів у супермаркеті та їх ціни. Упорядкувати за алфавітом лише ті товари, вартість яких не перевищує середню.

**ХІД ВИКОНАННЯ**

Метод сортування простою вибіркою:

using System;

using System.Diagnostics;

using System.Threading;

class HelloWorld {

static void Main() {

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];

Random randNum = new Random();

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

{

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

}

int[] newArray = selectionSort(unsortedArr);

}

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

Stopwatch stopwatch = new Stopwatch();

stopwatch.Start();

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

{

int minElement = i;

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

if (array[j] < array[minElement])

minElement = j;

int temp = array[minElement];

array[minElement] = array[i];

array[i] = temp;

}

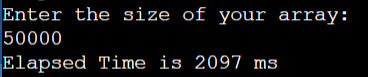
stopwatch.Stop();

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

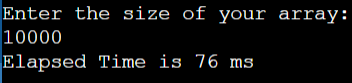
return array;

}

}



*Рис.1 Результат виконання програми*

**

*Рис.2 Результат виконання програми*

Вихідний код виконання індивідуального завдання:

//MainWindow.xaml.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

namespace ProductsSorting

{

/// <summary>

/// </summary>

///

public partial class MainWindow : Window

{

public class Product : IComparable

{

public string Column1 { get; set; }

public double Column2 { get; set; }

public int CompareTo(object incomingobject)

{

Product inputItem = incomingobject as Product;

return this.Column1.CompareTo(inputItem.Column1);

}

}

public List<Product> products = new List<Product>();

public List<Product> sortableItems = new List<Product>();

public MainWindow()

{

InitializeComponent();

}

public static double getAveragePrice(List<Product> list)

{

double averagePrice = 0;

foreach (Product item in list) averagePrice += item.Column2;

averagePrice /= list.Count;

return averagePrice;

}

public static Product[] selectionSort(Product[] array)

{

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

{

int minElement = i;

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

{

if (array[j].CompareTo(array[minElement]) < 0)

minElement = j;

Product temp = array[minElement];

array[minElement] = array[i];

array[i] = temp;

}

}

return array;

}

private void unordered\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

}

private void addProduct\_Copy1\_Click(object sender, RoutedEventArgs e)

{

var dg = dataGrid1;

for (int i = 1; i <= 2; ++i)

{

var column = new DataGridTextColumn();

column.Header = "Column" + i;

column.Binding = new Binding("Column" + i);

dg.Columns.Add(column);

}

products.Add(new Product()

{

Column1 = "Banana",

Column2 = 25.8

});

products.Add(new Product()

{

Column1 = "Oil",

Column2 = 85.7

});

products.Add(new Product()

{

Column1 = "Bread",

Column2 = 12.8

});

products.Add(new Product()

{

Column1 = "Sugar",

Column2 = 43

});

products.Add(new Product()

{

Column1 = "Chocolate",

Column2 = 27.7

});

products.Add(new Product()

{

Column1 = "Grapes",

Column2 = 30

});

foreach(var item in products)

{

dg.Items.Add(item);

}

}

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

{

var dg = dataGrid1;

products.Add(new Product { Column1 = productName.Text, Column2 = Convert.ToDouble(price.Text)});

dg.Items.Add(new Product { Column1=productName.Text, Column2 = Convert.ToDouble(price.Text) });

}

private void addProduct\_Copy\_Click(object sender, RoutedEventArgs e)

{

var sortedDG = sortedDataGrid;

double avPrice = getAveragePrice(products);

avPriceField.Content += Math.Round(avPrice, 2).ToString();

foreach (Product Item in products){

if (Item.Column2 <= avPrice) sortableItems.Add(Item);

}

for (int i = 1; i <= 2; ++i)

{

var column = new DataGridTextColumn();

column.Header = "Column" + i;

column.Binding = new Binding("Column" + i);

sortedDG.Columns.Add(column);

}

Product[] itemsArray = selectionSort(sortableItems.ToArray());

foreach (Product Item in itemsArray)

{

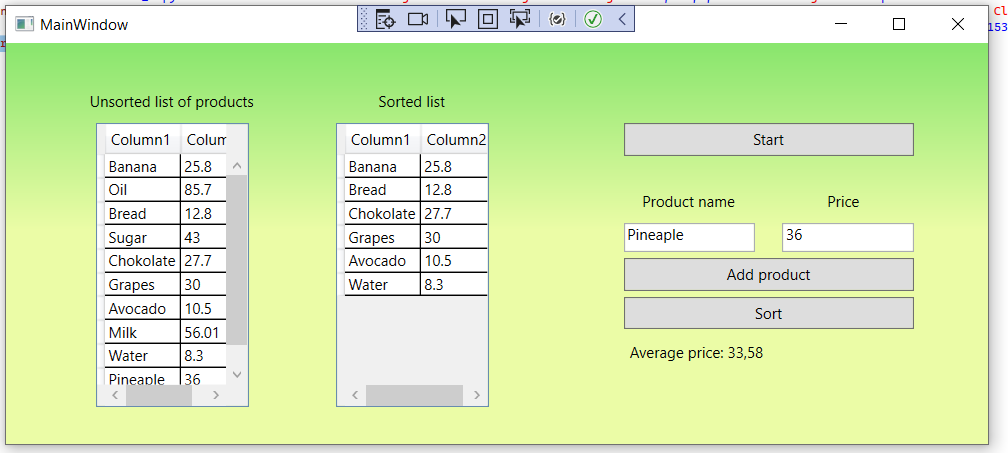
sortedDG.Items.Add(Item);

}

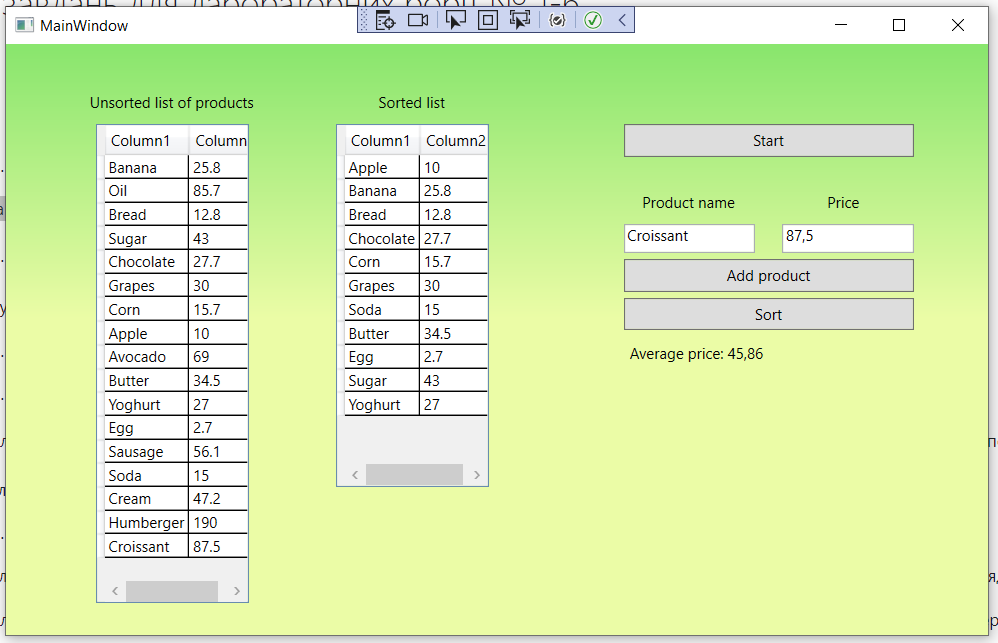
}

}

}



*Рис. 1 Результат виконання програми*

**

*Рис. 2 Результат виконання програми*

**Висновок**

На цій лабораторній роботі я познайомилася з алгоритмом сортування простою вибіркою, складність якого – О(0.5\*n\*(n-1)). Також я виконала індивідуальне завдання, де застосувала алгоритм сортування простою вибіркою до масиву об’єктів, зокрема сортувала ці об’єкти за їхньою назвою, якщо значення поля “Ціна” є меншим за середнє.