**Федеральное агентство связи**

**Государственное бюджетное образовательное учреждение высшего образования Ордена Трудового Красного Знамени**

**«Московский технический университет связи и информатики»**

**Лабораторная работа № 1**

«Методы сортировки»

Выполнила: студентка группы БВТ1904

Хорикова Софья Гарегиновна

Проверил: Павликов Артём Евгеньевич

Москва, 2020

**Цель работы:**

Реализовать методы сортировки строк числовой матрицы в соответствии с индивидуальным заданием. Для всех вариантов добавить реализацию быстрой сортировки (quicksort). Оценить время работы каждого алгоритма сортировки и сравнить его со временем стандартной функции сортировки, используемой в выбранном языке программирования.

**Листинг программы:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Diagnostics;

namespace Lab\_number

{

static class Lab1

{

public static void Start\_Lab1()

{

while (true)

{

Console.WriteLine("Введите размер матрицы"); //приглашение

int size = int.Parse(Console.ReadLine()); //считываем с клавы

Random rnd = new Random(); //новый объект класса рандом

int[] matrix = new int[size \* size]; //выделяем память под матрицу

for (int i = 0; i < size \* size; i++)

{

matrix[i] = rnd.Next(-1000, 1001); //тут заполняем матрицу рандомными числами

}

Stopwatch sw = new Stopwatch(); //измерение затраченного времени

sw.Start();

var bubble = BubbleSort(matrix); //обменом

sw.Stop();

Console.WriteLine("Время выполнения сортировки обменом - " + sw.Elapsed); //получает общее затраченное время, измеренное текущим экземпляром

sw.Reset();

sw.Start();

var shell = ShellSort(matrix); //шелл

sw.Stop();

Console.WriteLine("Время выполнения сортировки Шелла - " + sw.Elapsed);

sw.Reset();

sw.Start();

var insertion = InsertionSort(matrix); //вставками

sw.Stop();

Console.WriteLine("Время выполнения сортировки вставками - " + sw.Elapsed);

sw.Reset();

sw.Start();

SelectionSort(matrix); //выбором

sw.Stop();

Console.WriteLine("Время выполения сортировки выбором - " + sw.Elapsed);

sw.Reset();

int[] baza = (int[])matrix.Clone();

sw.Start();

Array.Sort(baza);

sw.Stop();

Console.WriteLine("Время выполнения сортировки встроенной - " + sw.Elapsed);

sw.Reset();

int[] quick = (int[])matrix.Clone();

sw.Start();

Quicksort(quick, 0, quick.Length - 1);

sw.Stop();

Console.WriteLine("Время выполнения сортировки быстрой - " + sw.Elapsed);

sw.Reset();

Console.WriteLine("Нажмите любую кнопку для повторения");

Console.ReadLine();

}

}

static void Swap(ref int first, ref int second) //метод используется ниже

{

var temp = first;

first = second;

second = temp;

}

static int[] InsertionSort(int[] array) //сортировка вставками

{

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

{

var key = array[i];

var j = i;

while ((j > 1) && (array[j - 1] > key))

{

Swap(ref array[j - 1], ref array[j]);

j--;

}

array[j] = key;

}

return array;

}

static void Quicksort(int[] arr, int start, int end) //сортировка Хоара

{

{

int i = start;

int j = end;

int x = arr[(start + end) / 2];

while (i <= j)

{

while (arr[i] < x) i++;

while (arr[j] > x) j--;

if (i <= j)

{

int temp = arr[i];

arr[i] = arr[j];

arr[j] = temp;

i++;

j--;

}

}

if (start < j) Quicksort(arr, start, j);

if (i < end) Quicksort(arr, i, end);

}

}

static public void SelectionSort(int[] arr) //сортировка выбором

{

int min, temp;

int length = arr.Length;

for (int i = 0; i < length - 1; i++)

{

min = i;

for (int j = i + 1; j < length; j++)

{

if (arr[j] < arr[min])

{

min = j;

}

}

if (min != i)

{

temp = arr[i];

arr[i] = arr[min];

arr[min] = temp;

}

}

}

static int[] BubbleSort(int[] mas) //обменом (пузырек)

{

int temp;

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

{

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

{

if (mas[j + 1] < mas[j])

{

temp = mas[j + 1];

mas[j + 1] = mas[j];

mas[j] = temp;

}

}

}

return mas;

}

static int[] ShellSort(int[] array) //сортировка Шелла

{

var d = array.Length / 2;

while (d >= 1)

{

for (var i = d; i < array.Length; i++)

{

var j = i;

while ((j >= d) && (array[j - d] > array[j]))

{

Swap(ref array[j], ref array[j - d]);

j -= d;

}

}

d /= 2;

}

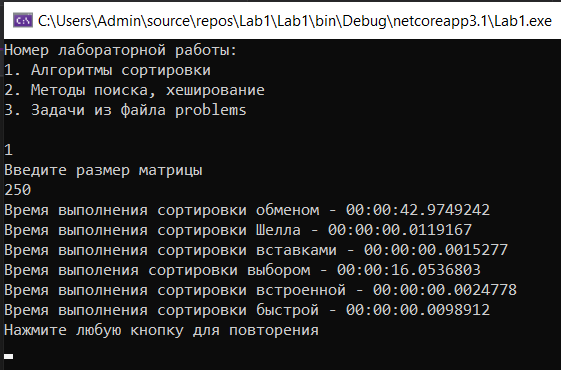
return array;

}

}

}

**Результат выполнения программы:**



**Выводы:**

Из написанных сортировок самая быстрая оказалась сортировка вставками, после неё идёт встроенная. Благодаря выполнению данной лабораторной работы, я сделала вывод, что следует навсегда забыть про «школьную» сортировку обменом (или по-другому: пузырьком), так как она долго справляется с большим объемом данных.