Лабораторная работа №3

Сортировка массивов

Составить программу, проводящую сравнительную характеристику методов сортировки массивов.

Программа должна выполнять следующие действия:

1. Производить сортировку массива соответствующими методами.

2. Иллюстрировать работу каждого метода на небольших массивах (размером до 10 элементов).

3. Производить сортировку каждым из методов случайного массива, уже отсортированного массива,

массива, отсортированного в обратном порядке. Засечь время. Размер массива при этом должен выбираться

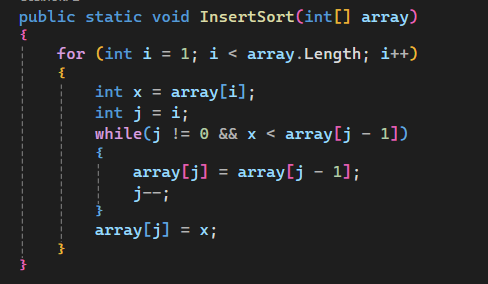
пользователем. После проведения сортировки, вывести данные о скорости работы методов.

Вариант №5

5.Метод прямого включения, шейкерная сортировка, метод Шелла.

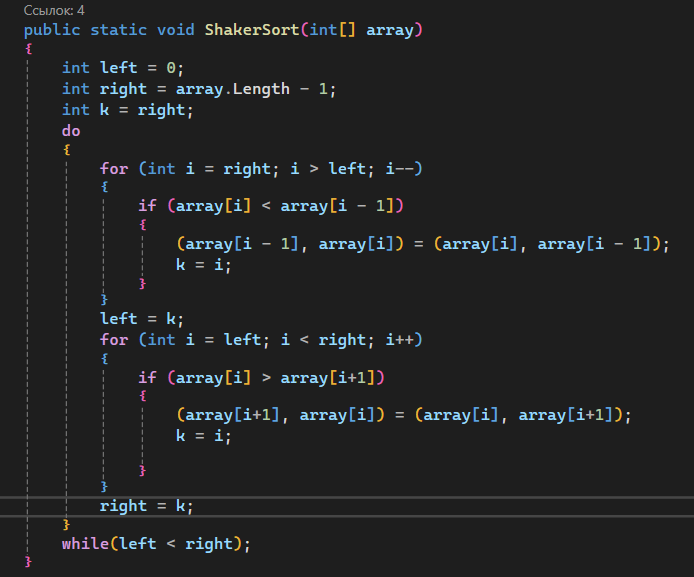
Алгоритм:

Метод прямого включения:



I-тый элемент вставляется среди предыдущих на подходящее для него место. Эта процедура проводится для всех элементов массива, начиная со второго и до последнего.

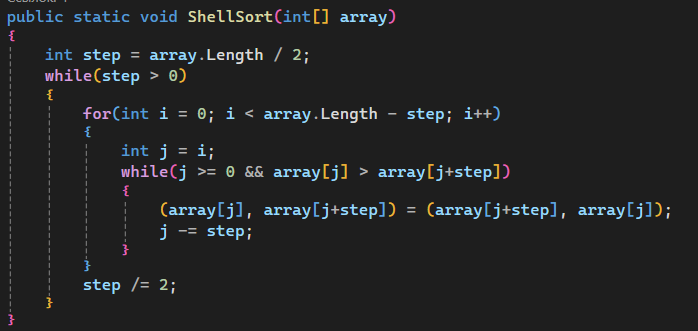
Шейкерная сортировка:



Данный метод является улучшением метода пузырька:

1. Запоминает, были или нет перестановки в процессе некоторого прохода. Если не было - сортировку можно завершить
2. Запоминает не только сам факт перестановки, но и место последнего обмена. Ясно, что после этого места массив отсортирован.
3. Проводить сортировку последовательно в 2 направлениях по массиву - сначала от конца к началу, а затем от начала к концу

Сортировка Шелла:



Сначала отдельно группируются и сортируются элементы, отстоящие друг от друга на расстоянии length/2m, Затем на расстояние length/4 и так далее, пока не дойдем до обычной одинарной сортировки. На каждом проходе сортировка программируется как сортировка вставками, поэтому если какая-то последовательность уже отсортирована, происходит переход в следующий.

Код:

using System.Diagnostics;

namespace Sort

{

public static class Sort

{

public static void InsertSort(int[] array)

{

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

{

int x = array[i];

int j = i;

while(j != 0 && x < array[j - 1])

{

array[j] = array[j - 1];

j--;

}

array[j] = x;

}

}

public static void ShakerSort(int[] array)

{

int left = 0;

int right = array.Length - 1;

int k = right;

do

{

for (int i = right; i > left; i--)

{

if (array[i] < array[i - 1])

{

(array[i - 1], array[i]) = (array[i], array[i - 1]);

k = i;

}

}

left = k;

for (int i = left; i < right; i++)

{

if (array[i] > array[i+1])

{

(array[i+1], array[i]) = (array[i], array[i+1]);

k = i;

}

}

right = k;

}

while(left < right);

}

public static void ShellSort(int[] array)

{

int step = array.Length / 2;

while(step > 0)

{

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

{

int j = i;

while(j >= 0 && array[j] > array[j+step])

{

(array[j], array[j+step]) = (array[j+step], array[j]);

j -= step;

}

}

step /= 2;

}

}

public static void PrintArray(int[] array)

{

foreach (int i in array)

{

Console.Write($"{i} ");

}

Console.WriteLine();

}

public static void SetTimer(int[] ints, Action<int[]> action, string nameSort)

{

var sw = new Stopwatch();

sw.Start();

action(ints);

sw.Stop();

Console.WriteLine($"{nameSort}: {sw.Elapsed.TotalSeconds} seconds");

}

}

}

Тест:

