

Министерство науки и высшего образования Российской Федерации  
Муромский институт (филиал)  
Федерального государственного бюджетного образовательного учреждения  
высшего образования  
«Владимирский государственный университет  
имени Александра Григорьевича и Николая Григорьевича Столетовых»

Факультет \_\_\_\_\_ ИТР \_\_\_\_\_

Кафедра \_\_\_\_\_ ПИН \_\_\_\_\_

# ЛАБОРАТОРНАЯ РАБОТА № 1

По \_\_\_\_\_ Структуры и алгоритмы обработки данных \_\_\_\_\_

Тема \_\_\_\_\_ Алгоритмы сортировки линейных коллекций данных \_\_\_\_\_

Руководитель

Привезенцев Д.Г.  
\_\_\_\_\_  
(фамилия, инициалы)

\_\_\_\_\_  
(подпись) \_\_\_\_\_ (дата)

Студент \_\_\_\_\_ ПИН - 121 \_\_\_\_\_  
(группа)

Мочалин Н.А.  
\_\_\_\_\_  
(фамилия, инициалы)

\_\_\_\_\_  
(подпись) \_\_\_\_\_ (дата)

Муром 2023

## Лабораторная работа №2

**Тема:** Алгоритмы сортировки линейных коллекций данных

**Цель работы:** Реализовать и протестировать алгоритмы сортировки

**Ход работы:**

### Задание на лабораторную работу

1. Реализовать три метода сортировки коллекций согласно таблице 1.
2. Реализовать управляющую программу(ы), включающую:
  - a. ввод исходных данных: из файла, генерацией случайных чисел (способа ввода данных предусмотреть в программе путем ввода выбора пользователя, при этом размер массива тоже указывается во время выполнения программы);
  - b. ввод на экран исходных данных;
  - c. выбор направления сортировки;
  - d. сортировку коллекции;
  - e. вывод на экран результата работы;
  - f. замер времени выполнения сортировки.
3. Реализовать вывод отладочной информации:
  - a. количество сравнений двух элементов;
  - b. количество перестановок двух элементов.
4. Выполнить исследование реализованных трех алгоритмах на разных коллекциях:
  - a. Выполнить замеры времени для коллекций размера 10, 100, 1000, 10000, 100000, 1000000 элементов.
  - b. Заполнить таблицы 2-4 для каждого анализируемого алгоритма.
  - c. Построить графики зависимости времени сортировки, числа перестановок, числа сравнений элементов от размера коллекции для каждого анализируемого алгоритма.
  - d. Сделать выводы по таблицам и графику.

					МИВУ 09.03.04 - 18.003			
Изм.	Лист	№ докум.	Подпись	Дата				
Разраб.		Мочалин Н.А.			Алгоритмы сортировки линейных коллекций данных	Лит.	Лист	Листов
Провер.		Привезенцев Д.Г.					2	18
Реценз.						МИ ВлГУ ПИН-121		
Н. Контр.								
Утверд.								

## Класс BubbleSort:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Laba1_class_library_
{
    public class BubbleSort : IStrategy
    {
        private ulong _comparisonCount = 0;
        private ulong _replacementCount = 0;
        public int[] Array { get; set; }
        public int[] Sort(int[] array)
        {
            for (int i = 1; i < array.Length; i++)
            {
                for (int j = 0; j < array.Length - i; j++)
                {
                    _comparisonCount++;
                    if (array[j] > array[j + 1])
                    {
                        Swap(ref array[j], ref array[j + 1]);
                        _replacementCount++;
                    }
                }
            }
            return array;
        }
        static void Swap(ref int value1, ref int value2)
        {
            int swap = value1;
            value1 = value2;
            value2 = swap;
        }
        public ulong OutComparisonCount()
        {
            return _comparisonCount;
        }
        public ulong OutReplacementCount()
        {
            return _replacementCount;
        }
    }
}
```

## Класс InsertionSort:

```
using System;
using System.Collections.Generic;
using System.Globalization;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Laba1_class_library_
{
    public class InsertionSort : IStrategy
    {
        private ulong _comparisonCount = 0;
        private ulong _replacementCount = 0;
        public int[] Array { get; set; }
        public int[] Sort(int[] array)
        {

```

					МИВУ 09.03.04 – 18.001	Лист
						3
Изм.	Лист	№ докум.	Подпись	Дата		

```

        for (int i = 1; i < array.Length; i++)
        {
            int key = array[i];
            int j = i;

            while (j > 1 && array[j - 1] > key)
            {
                Swap(ref array[j - 1], ref array[j]);
                j--;
                _comparisonCount++;
                _replacementCount++;
            }
            array[j] = key;
        }
        return array;
    }
    static void Swap(ref int value1, ref int value2)
    {
        int swap = value1;
        value1 = value2;
        value2 = swap;
    }
    public ulong OutComparisonCount()
    {
        return _comparisonCount;
    }
    public ulong OutReplacementCount()
    {
        return _replacementCount;
    }
}
}

```

### Класс GnomeSort:

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Laba1_class_library_
{
    public class GnomeSort : IStrategy
    {
        private ulong _comparisonCount = 0;
        private ulong _replacementCount = 0;

        public int[] Array { get; set; }
        public int[] Sort(int[] array)
        {
            int i = 1;
            while (i < array.Length)
            {
                _comparisonCount++;
                if (i > 0 && array[i] < array[i - 1])
                {
                    Swap(ref array[i], ref array[i - 1]);
                    _replacementCount++;
                    i--;
                }
                else
                    i++;
            }
            return array;
        }
    }
}

```

					МИВУ 09.03.04 – 18.001	Лист
Изм.	Лист	№ докум.	Подпись	Дата		4

```

static void Swap(ref int value1, ref int value2)
{
    int swap = value1;
    value1 = value2;
    value2 = swap;
}
public ulong OutComparisonCount()
{
    return _comparisonCount;
}
public ulong OutReplacementCount()
{
    return _replacementCount;
}
}
}

```

### Класс MergeSort:

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Laba1_class_library_
{
    public class MergeSort : IStrategy
    {
        static ulong _comparisonCount = 0;
        static ulong _replacementCount = 0;

        public int[] Array { get; set; }
        public int[] Sort(int[] arrayNoSort)
        {
            return MergeSorting(arrayNoSort);
        }
        static void Merge(int[] array, int lowIndex, int middleIndex, int highIndex)
        {
            int left = lowIndex;
            int right = middleIndex + 1;
            int[] tempArray = new int[highIndex - lowIndex + 1];
            int index = 0;

            while ((left <= middleIndex) && (right <= highIndex))
            {
                _comparisonCount++;
                if (array[left] < array[right])
                {
                    tempArray[index] = array[left];
                    _replacementCount++;
                    left++;
                }
                else
                {
                    tempArray[index] = array[right];
                    _replacementCount++;
                    right++;
                }
                index++;
            }

            for (int i = left; i <= middleIndex; i++)
            {
                tempArray[index] = array[i];
                _replacementCount++;
            }
        }
    }
}

```

					МИВУ 09.03.04 – 18.001	Лист
Изм.	Лист	№ докум.	Подпись	Дата		5

```

        index++;
    }

    for (int i = right; i <= highIndex; i++)
    {
        tempArray[index] = array[i];
        _replacementCount++;
        index++;
    }

    for (int i = 0; i < tempArray.Length; i++)
    {
        array[lowIndex + i] = tempArray[i];
        _replacementCount++;
    }
}

static int[] MergeSorting(int[] array, int lowIndex, int highIndex)
{
    if (lowIndex < highIndex)
    {
        int middleIndex = (lowIndex + highIndex) / 2;
        MergeSorting(array, lowIndex, middleIndex);
        MergeSorting(array, middleIndex + 1, highIndex);
        Merge(array, lowIndex, middleIndex, highIndex);
    }

    return array;
}

static int[] MergeSorting(int[] array)
{
    return MergeSorting(array, 0, array.Length - 1);
}

public ulong OutComparisonCount ()
{
    return _comparisonCount;
}

public ulong OutReplacementCount()
{
    return _replacementCount;
}
}
}

```

### Класс SelectionSort:

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Laba1_class_library_
{
    public class SelectionSort : IStrategy
    {
        private ulong _comparisonCount = 0;
        private ulong _replacementCount = 0;

        public int[] Array { get; set; }
        public int[] Sort(int[] array)
        {
            for (int i = 0; i < array.Length - 1; i++)
            {
                int k = i;

```

					МИВУ 09.03.04 – 18.001	Лист
Изм.	Лист	№ докум.	Подпись	Дата		6

```

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

            if (array[j] < array[k])
            {
                k = j;

                _replacementCount++;
            }

            _comparisonCount++;

            if(i != k)
            {
                Swap(ref array[j], ref array[k]);

                _replacementCount++;
            }
        }
        return array;
    }
    static void Swap(ref int value1, ref int value2)
    {
        int swap = value1;
        value1 = value2;
        value2 = swap;
    }
    public ulong OutComparisonCount()
    {
        return _comparisonCount;
    }
    public ulong OutReplacementCount()
    {
        return _replacementCount;
    }
}
}

```

### Класс CocktailSort:

```

namespace Laba1_class_library_
{
    public class CocktailSort : IStrategy
    {
        private ulong _comparisonCount = 0;
        private ulong _replacementCount = 0;
        public int[] Array { get; set; }
        public int[] Sort(int[] array)
        {
            int left = 0;
            int right = array.Length - 1;
            int count = 0;

            while (left < right)
            {
                for (int i = left; i < right; i++)
                {
                    _comparisonCount++;
                    if (array[i] > array[i + 1])
                    {
                        Swap(ref array[i], ref array[i + 1]);
                        count++;
                        _replacementCount++;
                    }
                }
            }
        }
    }
}

```

```

    }
    right--;
    for (int i = right; i > left; i--)
    {
        _comparisonCount++;
        if (array[i] < array[i - 1])
        {
            Swap(ref array[i], ref array[i - 1]);
            count++;
            _replacementCount++;
        }
    }
    left++;
}
return array;
}
static void Swap(ref int value1, ref int value2)
{
    int swap = value1;
    value1 = value2;
    value2 = swap;
}
public ulong OutComparisonCount()
{
    return _comparisonCount;
}
public ulong OutReplacementCount()
{
    return _replacementCount;
}
}
}

```

### Класс CombSort:

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Laba1_class_library_
{
    public class CombSort : IStrategy
    {
        private ulong _comparisonCount = 0;
        private ulong _replacementCount = 0;

        public int[] Array { get; set; }

        public int[] Sort(int[] array)
        {
            int currentStep = array.Length - 1;

            while (currentStep > 1)
            {
                for(int i = 0; i + currentStep < array.Length; i++)
                {
                    _comparisonCount++;

                    if (array[i] > array[i + currentStep])
                    {
                        Swap(ref array[i], ref array[i + currentStep]);
                        _replacementCount++;
                    }
                }
            }
        }
    }
}

```

					МИВУ 09.03.04 – 18.001	Лист
Изм.	Лист	№ докум.	Подпись	Дата		8



```

        currentStep = GetNextStep(currentStep);
    }

    for (int i = 1; i < array.Length; i++)
    {
        bool swapFlag = false;

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

            if (array[j] > array[j + 1])
            {
                Swap(ref array[j], ref array[j + 1]);
                swapFlag = true;
            }

            _comparisonCount++;

            if (!swapFlag)
            {
                break;
            }
        }
        return array;
    }
}

static void Swap(ref int value1, ref int value2)
{
    int swap = value1;
    value1 = value2;
    value2 = swap;
}

static int GetNextStep(int s)
{
    s = s * 1000 / 1247;
    return s > 1 ? s : 1;
}

public ulong OutComparisonCount()
{
    return _comparisonCount;
}

public ulong OutReplacementCount()
{
    return _replacementCount;
}
}
}

```

### Класс ShellSort:

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Runtime.CompilerServices;
using System.Text;
using System.Threading.Tasks;

namespace Laba1_class_library_
{
    public class ShellSort : IStrategy
    {
        private ulong _comparisonCount = 0;
        private ulong _replacementCount = 0;
        public int[] Array { get; set; }
        public int[] Sort(int[] array)
        {

```

					МИВУ 09.03.04 – 18.001	Лист
Изм.	Лист	№ докум.	Подпись	Дата		9

```

        int step = array.Length / 2;
        while(step >= 1)
        {
            for (int i = 0; i < array.Length; i++)
            {
                int j = i;
                while((j >= step) && (array[j - step] > array[j]))
                {
                    _comparisonCount++;
                    _replacementCount++;

                    Swap(ref array[j], ref array[j - step]);
                    j = j - step;
                }

                _comparisonCount++;
                step = step / 2;
            }
            return array;
        }
        static void Swap(ref int value1, ref int value2)
        {
            int swap = value1;
            value1 = value2;
            value2 = swap;
        }
        public ulong OutComparisonCount()
        {
            return _comparisonCount;
        }
        public ulong OutReplacementCount()
        {
            return _replacementCount;
        }
    }
}

```

### Класс CountingSort:

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Laba1_class_library_
{
    public class CountingSort : IStrategy
    {
        private ulong _comparisonCount = 0;
        private ulong _replacementCount = 0;
        public int[] Array { get; set; }
        public int[] Sort(int[] array)
        {
            int min = array[0];
            int max = array[0];

            foreach (int element in array)
            {
                _comparisonCount++;
                if (element < min)
                {
                    min = element;
                    _replacementCount++;
                }
                else if (element > max)

```

					МИВУ 09.03.04 – 18.001	Лист
Изм.	Лист	№ докум.	Подпись	Дата		10

```

        {
            max = element;
            _replacementCount++;
        }
    }

    int correctionFactor = min != 0 ? -min : 0;
    max += correctionFactor;

    int[] count = new int[max + 1];
    for (int i = 0; i < array.Length; i++)
    {
        count[array[i] + correctionFactor]++;
    }

    int index = 0;
    for (int i = 0; i < count.Length; i++)
    {
        for (int j = 0; j < count[i]; j++)
        {
            array[index] = i - correctionFactor;
            index++;
            _replacementCount++;
        }
    }
    return array;
}

public ulong OutComparisonCount()
{
    return _comparisonCount;
}

public ulong OutReplacementCount()
{
    return _replacementCount;
}
}
}

```

### Интерфейс IStrategy:

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Laba1_class_library_
{
    public interface IStrategy
    {
        int[] Sort(int[] array);
        ulong OutComparisonCount();
        ulong OutReplacementCount();
    }
}

```

### Класс Context:

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Laba1_class_library_
{
    public class Context

```

					МИВУ 09.03.04 – 18.001	Лист
Изм.	Лист	№ докум.	Подпись	Дата		11

```

    {
        public IStrategy Strategy { get; set; }
        public int[] Request(int[] array)
        {
            return Strategy.Sort(array);
        }
    }
}

```

### Класс Timer:

```

using System;
using System.Collections.Generic;
using System.Diagnostics;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Laba1_AlgorithmsForSortingLinearDataCollections_
{
    internal class Timer
    {
        Stopwatch timeWatch = new Stopwatch();
        public void Start()
        {
            timeWatch.Start();
        }
        public void Stop()
        {
            timeWatch.Stop();
        }
        public override string ToString()
        {
            TimeSpan timeSpend = timeWatch.Elapsed;
            string elapsedTime = String.Format("{0:00}:{1:00}:{2:00}.{3:00}",
timeSpend.Hours, timeSpend.Minutes, timeSpend.Seconds, timeSpend.Milliseconds / 10);
            return elapsedTime;
        }
    }
}

```

### Класс Timer:

```

using Laba1_class_library_;
using System;
using System.Windows.Forms;

namespace Laba1_AlgorithmsForSortingLinearDataCollections_
{
    public partial class Form1 : Form
    {
        int[] arrayNoSort;
        int quantity;

        public Form1()
        {
            InitializeComponent();
        }

        private void Clean()
        {
            textBox1.Clear();
            textBox4.Clear();
            textBox5.Clear();
            textBox6.Clear();
        }
    }
}

```

					МИВУ 09.03.04 – 18.001	Лист
Изм.	Лист	№ докум.	Подпись	Дата		12

```

    }

    private void PrintOnTextBox(int[] array)
    {
        for (int i = 0; i < array.Length; i++)
        {
            textBox1.Text += array[i] + " ";
            progressBar1.Value = i;
        }
    }

    private void ProgressBarBoundaries()
    {
        progressBar1.Value = 0;
        progressBar1.Minimum = 0;
        progressBar1.Maximum = arrayNoSort.Length - 1;
    }

    private void TimerPrint(Timer timer, IStrategy typeSort)
    {
        textBox4.Text = timer.ToString();
        textBox5.Text = Convert.ToString(typeSort.OutComparisonCount());
        textBox6.Text = Convert.ToString(typeSort.OutReplacementCount());
    }

    //Random number generator
    private void button4_Click(object sender, EventArgs e)
    {
        textBox2.Clear();
        quantity = int.Parse(textBox3.Text);
        arrayNoSort = new int[quantity];
        Random random = new Random();

        progressBar1.Minimum = 0;
        progressBar1.Maximum = arrayNoSort.Length - 1;

        for (int i = 0; i < arrayNoSort.Length; i++)
        {
            arrayNoSort[i] = random.Next(0, 100);
            progressBar1.Value = i;
        }
    }

    //Output of the generated array
    private void button11_Click(object sender, EventArgs e)
    {
        progressBar1.Minimum = 0;
        progressBar1.Maximum = arrayNoSort.Length - 1;

        for (int i = 0; i < arrayNoSort.Length; i++)
        {
            textBox2.Text += arrayNoSort[i] + " ";
            progressBar1.Value = i;
        }
    }

    private void button12_Click(object sender, EventArgs e)
    {
    }

    //Cocktail Sort
    private void button1_Click(object sender, EventArgs e)
    {
        Clean();
    }

```

```

        Timer timer = new Timer();
        CocktailSort cocktailSort = new CocktailSort();

        ProgressBarBoundaries();

        timer.Start();
        int[] cloneArray = (int[])arrayNoSort.Clone();
        int[] returnArray = cocktailSort.Sort(cloneArray);
        PrintOnTextBox(returnArray);
        timer.Stop();

        TimerPrint(timer, cocktailSort);
    }

    //Gnome Sort
    private void button2_Click(object sender, EventArgs e)
    {
        Clean();

        Timer timer = new Timer();
        GnomeSort gnomeSort = new GnomeSort();

        ProgressBarBoundaries();

        timer.Start();
        int[] cloneArray = (int[])arrayNoSort.Clone();
        int[] returnArray = gnomeSort.Sort(cloneArray);
        PrintOnTextBox(returnArray);
        timer.Stop();

        TimerPrint(timer, gnomeSort);
    }

    //Merge Sort
    private void button3_Click(object sender, EventArgs e)
    {
        Clean();

        MergeSort mergeSort = new MergeSort();
        Timer timer = new Timer();

        ProgressBarBoundaries();

        timer.Start();
        int[] cloneArray = (int[])arrayNoSort.Clone();
        int[] returnArray = mergeSort.Sort(cloneArray);
        PrintOnTextBox(returnArray);
        timer.Stop();

        TimerPrint(timer, mergeSort);
    }

    //Bubble Sort
    private void button5_Click(object sender, EventArgs e)
    {
        Clean();

        BubbleSort bubbleSort = new BubbleSort();
        Timer timer = new Timer();

        ProgressBarBoundaries();

        timer.Start();
        int[] cloneArray = (int[])arrayNoSort.Clone();
        int[] returnArray = bubbleSort.Sort(cloneArray);

```

					МИВУ 09.03.04 – 18.001	Лист
Изм.	Лист	№ докум.	Подпись	Дата		14

```

        PrintOnTextBox(returnArray);
        timer.Stop();

        TimerPrint(timer, bubbleSort);
    }

    //Insertion Sort
    private void button7_Click(object sender, EventArgs e)
    {
        Clean();

        InsertionSort insertionSort = new InsertionSort();
        Timer timer = new Timer();

        ProgressBarBoundaries();

        timer.Start();
        int[] cloneArray = (int[])arrayNoSort.Clone();
        int[] returnArray = insertionSort.Sort(cloneArray);
        PrintOnTextBox(returnArray);
        timer.Stop();

        TimerPrint(timer, insertionSort);
    }

    //Selection Sort
    private void button9_Click(object sender, EventArgs e)
    {
        Clean();

        SelectionSort selectionSort = new SelectionSort();
        Timer timer = new Timer();

        ProgressBarBoundaries();

        timer.Start();
        int[] cloneArray = (int[])arrayNoSort.Clone();
        int[] returnArray = selectionSort.Sort(cloneArray);
        PrintOnTextBox(returnArray);
        timer.Stop();

        TimerPrint(timer, selectionSort);
    }

    //Comb Sort
    private void button6_Click(object sender, EventArgs e)
    {
        Clean();

        CombSort combSort = new CombSort();
        Timer timer = new Timer();

        ProgressBarBoundaries();

        timer.Start();
        int[] cloneArray = (int[])arrayNoSort.Clone();
        int[] returnArray = combSort.Sort(cloneArray);
        PrintOnTextBox(returnArray);
        timer.Stop();

        TimerPrint(timer, combSort);
    }

    //Shell Sort
    private void button8_Click(object sender, EventArgs e)
    {

```

					МИВУ 09.03.04 – 18.001	Лист
Изм.	Лист	№ докум.	Подпись	Дата		15

```

Clean();

ShellSort shellSort = new ShellSort();
Timer timer = new Timer();

ProgressBarBoundaries();

timer.Start();
int[] cloneArray = (int[])arrayNoSort.Clone();
int[] returnArray = shellSort.Sort(cloneArray);
PrintOnTextBox(returnArray);
timer.Stop();

TimerPrint(timer, shellSort);
}

//Counting Sort
private void button10_Click(object sender, EventArgs e)
{
    Clean();

    CountingSort countingSort = new CountingSort();
    Timer timer = new Timer();

    ProgressBarBoundaries();

    timer.Start();
    int[] cloneArray = (int[])arrayNoSort.Clone();
    int[] returnArray = countingSort.Sort(cloneArray);
    PrintOnTextBox(returnArray);
    timer.Stop();

    TimerPrint(timer, countingSort);
}
}

```

Form1

Количество элементов массива для генерации  
1000

Сгенерировать массив случайных чисел

Вывести сгенерированный массив на экран  
(Не рекомендуется для больших массивов)

Не отсортированный массив

25	41	90	73	43	67	7	27	4	3	73	81	75	62	32	25	74	70	59	84	39	19	49	38
57	49	77	33	42	37	47	49	14	50	7	96	30	1	76	65	20	52	61	64	80	19	94	
54	47	34	68	61	9	95	65	6	91	52	16	93	90	73	84	66	55	23	67	24	55	35	
95	72	94	94	64	86	57	85	57	38	55	22	85	7	69	86	39	18	84	1	4	77	85	82
74	46	67	94	9	57	52	70	99	12	39	95	67	15	31	79	48	6	94	78	38	67	88	
77	97	68	83	48	39	53	30	17	86	11	19	18	18	62	76	0	86	85	23	73	67	73	
19	23	87	5	52	98	82	7	44	7	57	90	97	40	73	3	30	45	81	22	8	55	96	12
85	30	94	91	53	44	14	10	23	49	59	61	34	0	32	35	79	66	11	60	28	65	35	
27	83	54	77	91	45	64	63	27	74	45	67	67	88	15	91	62	42	80	74	90	13	41	
96	44	48	87	29	88	84	52	31	20	94	66	8	98	66	23	64	58	85	32	86	88	33	
64	46	63	75	49	17	48	90	44	14	41	58	32	42	58	35	74	42	22	13	46	73	21	
6	92	95	19	15	16	5	80	55	7	11	84	40	65	13	35	34	83	29	49	52	67	39	34
80	6	22	11	90	63	75	86	91	25	57	82	53	97	32	74	39	33	86	51	21	58	18	

Сортировка перемешивани

Сортировка пузырьком

Сортировка расчёской

Гномья сортировка

Сортировка вставками

Сортировка Шелла

Сортировка слиянием

Сортировка выбором

Сортировка подсчётом

Отсортированный массив

0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
5	5	5	5	6	6	6	6	6	6	6	7	7	7	7	7	7	7	7	7	7	7	7	7
10	10	10	10	10	10	10	10	10	10	10	11	11	11	11	11	11	11	11	11	11	11	11	11
12	12	12	12	13	13	13	13	13	13	13	14	14	14	14	14	14	14	14	14	14	14	14	14
15	15	15	15	15	15	15	15	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16
17	17	17	17	17	17	17	17	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18
19	19	19	19	19	19	19	19	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20
22	22	22	22	22	22	22	22	22	22	22	23	23	23	23	23	23	23	23	23	23	23	23	23
24	24	24	24	24	24	24	24	24	24	24	25	25	25	25	25	25	25	25	25	25	25	25	25
26	26	26	26	26	26	26	26	26	26	26	27	27	27	27	27	27	27	27	27	27	27	27	27
28	28	28	28	28	28	28	28	28	28	28	28	28	28	28	28	28	28	28	28	28	28	28	28
30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30
33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33
35	35	35	35	35	35	35	35	35	35	35	35	35	35	35	35	35	35	35	35	35	35	35	35

Вывести отсортированный массив  
(Все также не рекомендуется для больших массивов)

Время выполнения сортировки: 00:00:02.12

Кол-во сравнений двух элементов: 499500

Кол-во перестановок двух элементов: 252917

Прогресс выполнения



ВРЕМЯ	10	100	1000	10000	100000	1000000
Сортировка перемешиванием	0	0	0	0	6	1358
Гномья сортировка	0	0	0	0	54	1403
Сортировка слиянием	0	0	0	0	0	11
Мин. значение	0	0	0	0	0	11
Макс. значение	0	0	0	0	54	1403
Ср. значение	0	0	0	0	20	924

ПЕРЕСТАНОВКИ	10	100	1000	10000	100000	1000000
Сортировка перемешиванием	38	2611	241621	24541390	2464919218	246456721834
Гномья сортировка	85	2611	241621	24541390	2479765213	247936521374
Сортировка слиянием	1412	2756	22708	267232	3605088	43507936
Мин. значение	38	2611	22708	267232	3605088	43507936
Макс. значение	1412	2756	241621	24541390	2479765213	247936521374
Ср. значение	511,66667	2659,3333	168650	16450004	1649429840	164812250381

СРАВНЕНИЕ	10	100	1000	10000	100000	1000000
Сортировка перемешиванием	45	4950	499500	49995000	4999950000	499995000000
Гномья сортировка	85	5321	484241	49092779	4959630425	495464342543
Сортировка слиянием	554	1100	9776	120302	1652676	20269606
Мин. значение	45	1100	9776	120302	1652676	20269606
Макс. значение	554	5321	484241	49092779	4959630425	495464342543
Ср. значение	228	3790,3333	331172,33	33069360,3	3320411034	331826537383

### Время в секундах



## Число сравнений



## Число перестановок



**Вывод:** Я Реализовал и протестировал алгоритмы сортировки.