Федеральное государственное автономное

образовательное учреждение

высшего образования

«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий

институт

Кафедра «Информатика»

кафедра

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №3**

«Сортировка массивов»

тема

Преподаватель \_\_\_\_\_\_\_\_\_ Р.Ю. Царев

подпись, дата инициалы, фамилия

Студент КИ16-16б, №031620303 \_\_\_\_\_\_\_\_\_ В.В. Кокташев

номер группы, зачетной книжки подпись, дата инициалы, фамилия

Красноярск 2018

**СОДЕРЖАНИЕ**

1 Цель работы…………………………………………………………...3

2 Задачи …………………………………………………………………3

3 Ход работы……………………………………………………………3

4 Вывод…………………………………………...……………………..6

ПРИЛОЖЕНИЕ А ……………………………………………….……..7

ПРИЛОЖЕНИЕ Б…………………………………………………..…..9

ПРИЛОЖЕНИЕ В……………………………………………………..11

ПРИЛОЖЕНИЕ Г……………………………………………………..12

**1 Цель работы**

# Изучение различных алгоритмов сортировки массивов.

# 2 Задание

1. реализовать в программе два из следующих алгоритмов: сортировка с помощью прямого включения, сортировка с помощью двоичного включения, сортировка с помощью прямого выбора;
2. сравнить эффективность реализованных алгоритмов по числу перестановок и количеству сравнений.

**3 Ход работы**

1. С использованием языка программирования C# было создано графическое приложение, осуществляющее сортировку массивов с помощью алгоритмов сортировки прямого включения и прямого выбора. В первом случае каждый элемент массива, начиная со второго, сравнивается с предыдущими элементами и вставляется в необходимое место. Во втором случае в массиве ищется минимальный элемент и этот элемент меняется местами с первым элементом оставшейся последовательности. Примеры работы программы приведены на рисунках 1-2.

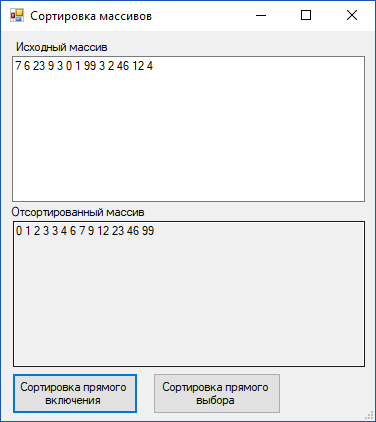


Рисунок 1 – Пример работы программы 1

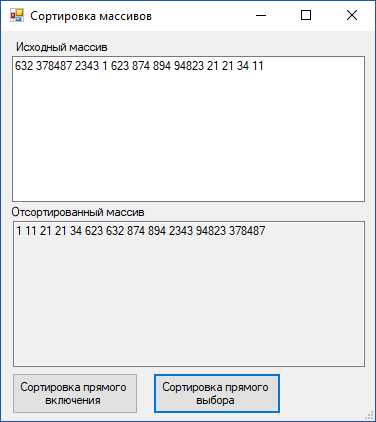


Рисунок 2 – Пример работы программы 2

1. Число сравнений и перестановок в сортировкаx указано на рисунках 3-4, где C – количество сравнений, M – количество перестановок. Из этих данных можно сделать вывод, что в среднем количество сравнений в сортировке прямого выбора больше, что подтверждает графики соответствующих функций (рисунок 5). Количество же перестановок больше в сортировке прямыми вставками (рисунок 6)

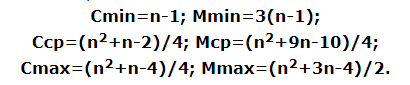


Рисунок 3 – количество сравнений и перестановок в сортировке прямыми включениями

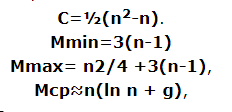


Рисунок 4 – количество сравнений и перестановок в сортировке прямого выбора

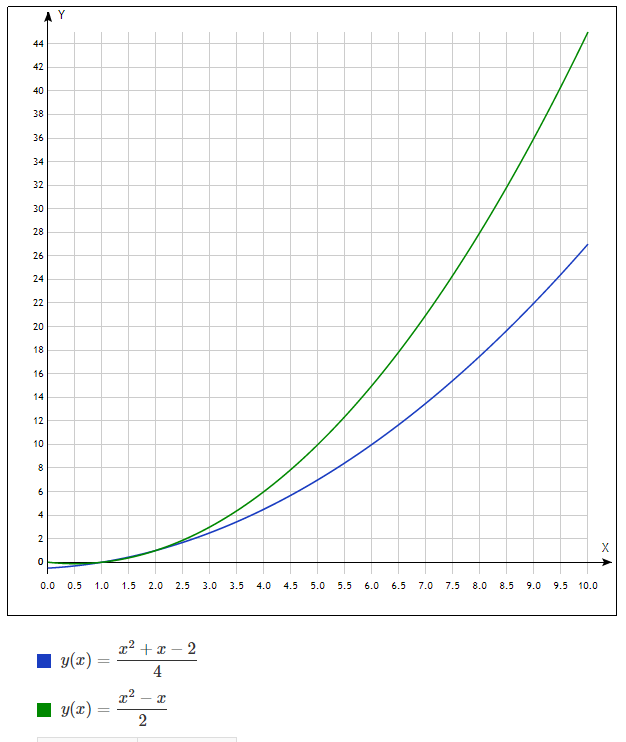


Рисунок 5 – График функций Cср и С

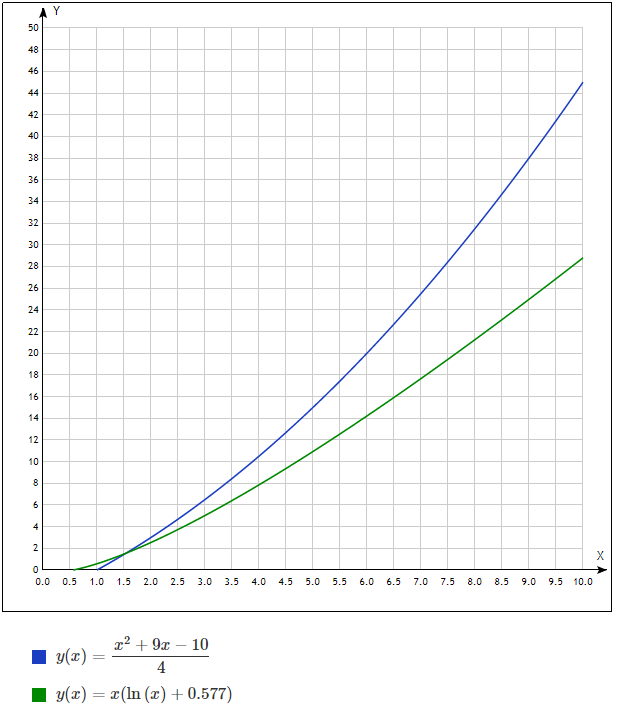


Рисунок 6 – График функций Mср и M

# 4 Вывод

В ходе выполнения лабораторной работы приобретен опыт реализации алгоритмов сортировки массивов, а также оценки их эффективности в зависимости от количества перестановок и сравнений в алгоритмах. Закреплены навыки программирования на языке C# и платформе .NET Framework.

**ПРИЛОЖЕНИЕ А**

**Листинг исходного кода приложения**

Algorithms.cs

|  |
| --- |
| using System;  using System.Text.RegularExpressions;  using System.Collections.Generic;  namespace ALabs  {  static class Algorithms  {  public static int[] inclusionSort(int[] mas)//Сортировка прямого включения  {  for (int i = 1; i < mas.Length; i++)  {  int value = mas[i]; // запоминаем значение текущего элемента  int index = i; // и его индекс  while ((index > 0) && (mas[index - 1] > value))  { // смещаем другие элементы к концу массива пока они меньше index  mas[index] = mas[index - 1];  index--; // смещаем индекс к началу массива  }  mas[index] = value; // рассматриваемый элемент помещаем на освободившееся место  }  return mas;  }  public static int[] selectionSort(int[] mas)//Сортировка прямого выбора  {  int min, temp; // для поиска минимального элемента и для обмена  for (int i = 0; i < mas.Length - 1; i++)  {  min = i; // запоминаем индекс текущего элемента  // ищем минимальный элемент чтобы поместить на место i-ого  for (int j = i + 1; j < mas.Length; j++) // для остальных элементов после i-ого  {  if (mas[j] < mas[min]) // если элемент меньше минимального,  min = j; // запоминаем его индекс в min  }  temp = mas[i]; // меняем местами i-ый и минимальный элементы  mas[i] = mas[min];  mas[min] = temp;  }  return mas;  }  public static int[] StringToIntArray(string value)//Преобразование строки в целочисленный массив  {  string text = string.Empty;  text = Regex.Replace(value.Trim(' '), @"\s+", " ");  List<int> list = new List<int>();  foreach (string i in text.Split(' '))  if (int.TryParse(i, out int Number)) list.Add(Number);  return list.ToArray();  }  public static string IntArrayToString(int[] value)//Перевод целочисленного массива в строку  {  string text = string.Empty;  for (int index = 0; index < value.Length; index++)  text += value[index] + " ";  return text;  }  }  } |

Form3.cs:

|  |
| --- |
| using System;  using System.Windows.Forms;  namespace ALabs  {  public partial class Form3 : Form  {  public Form3() //Точка вxода в программу  {  InitializeComponent();  }    private void inclusionSortButton\_Click\_1(object sender, EventArgs e)//Сортировка прямого включения  {  textBox2.Text = Algorithms.IntArrayToString(Algorithms.inclusionSort(Algorithms.StringToIntArray(textBox1.Text)));  }  private void selectionSortButton\_Click\_1(object sender, EventArgs e)//Сортировка прямого выбора  {  textBox2.Text = Algorithms.IntArrayToString(Algorithms.selectionSort(Algorithms.StringToIntArray(textBox1.Text)));  }  }  } |