Министерство науки и высшего образования Российской Федерации

Федеральное государственное образовательное учреждение

высшего образования

**ВЯТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

Институт математики и информационных систем

Факультет автоматики и вычислительной техники

Кафедра систем автоматизации управления

**Дисциплина: Алгоритмы и структуры данных**

**Отчёт по лабораторной работе №6**

**Тема: «Алгоритмы сортировки, поиска и выборки»**

**Вариант 2**

|  |  |  |  |
| --- | --- | --- | --- |
| Выполнил студент  группы ИТб 2302-02-20 |  | Ердяков Роман Александрович |  |
|  |  |
| Проверила |  | Кашина Елена Вячеславовна |  |

СОДЕРЖАНИЕ

[1 Задание 3](#_Toc200883298)

[2 Тестирование (задача 1) 4](#_Toc200883299)

[3 Программа на C# (задача 1): 5](#_Toc200883300)

[4 Тестирование (задача 2) 8](#_Toc200883301)

[5 Программа на C# (задача 2): 10](#_Toc200883302)

[6 Вывод 12](#_Toc200883303)

# Задание

Задача 1.

Задать массив из 8-12 элементов. Разобрать на примере этого массива и подробно описать сортировку

1. методом выбора;
2. методом обмена (пузырька);
3. методом вставки.

Подсчитать сложность каждого метода для вашего конкретного примера, то есть подсчитать количество операций сравнения и записи (присвоения). Вывести на экран.

Задача 2.

Изменить решение в методе выбора, что бы осуществлялась сортировка элементов, записанных на нечетных местах

# Тестирование (задача 1)

Примеры выполнения программы представлены на рисунке 1, 2.

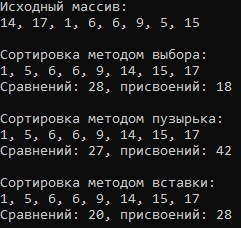


Рисунок 1 – Экранная форма программы

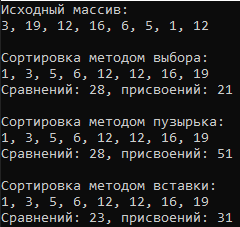


Рисунок 2 – Экранная форма программы

# Программа на C# (задача 1):

using System;

class Program

{

static void Main()

{

Random rnd = new Random();

int[] array = new int[8];

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

{

array[i] = rnd.Next(1, 21);

}

Console.WriteLine("Исходный массив:");

PrintArray(array);

int[] arrSelection = (int[])array.Clone();

int[] arrBubble = (int[])array.Clone();

int[] arrInsertion = (int[])array.Clone();

Console.WriteLine("\nСортировка методом выбора:");

SortSelection(arrSelection);

Console.WriteLine("\nСортировка методом пузырька:");

SortBubble(arrBubble);

Console.WriteLine("\nСортировка методом вставки:");

SortInsertion(arrInsertion);

}

static void PrintArray(int[] arr)

{

Console.WriteLine(string.Join(", ", arr));

}

static void SortSelection(int[] arr)

{

int n = arr.Length;

int comparisons = 0;

int assignments = 0;

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

{

int minIndex = i;

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

{

comparisons++;

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

{

minIndex = j;

}

}

if (minIndex != i)

{

int temp = arr[i];

arr[i] = arr[minIndex];

arr[minIndex] = temp;

assignments += 3;

}

}

PrintArray(arr);

Console.WriteLine($"Сравнений: {comparisons}, присвоений: {assignments}");

}

static void SortBubble(int[] arr)

{

int n = arr.Length;

int comparisons = 0;

int assignments = 0;

bool swapped;

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

{

swapped = false;

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

{

comparisons++;

if (arr[j] > arr[j + 1])

{

int temp = arr[j];

arr[j] = arr[j + 1];

arr[j + 1] = temp;

assignments += 3;

swapped = true;

}

}

if (!swapped)

break;

}

PrintArray(arr);

Console.WriteLine($"Сравнений: {comparisons}, присвоений: {assignments}");

}

static void SortInsertion(int[] arr)

{

int n = arr.Length;

int comparisons = 0;

int assignments = 0;

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

{

int key = arr[i];

assignments++;

int j = i - 1;

while (j >= 0)

{

comparisons++;

if (arr[j] > key)

{

arr[j + 1] = arr[j];

assignments++;

j--;

}

else

{

break;

}

}

arr[j + 1] = key;

assignments++;

}

PrintArray(arr);

Console.WriteLine($"Сравнений: {comparisons}, присвоений: {assignments}");

}

}

# Тестирование (задача 2)

Примеры выполнения программы представлены на рисунке 3, 4.

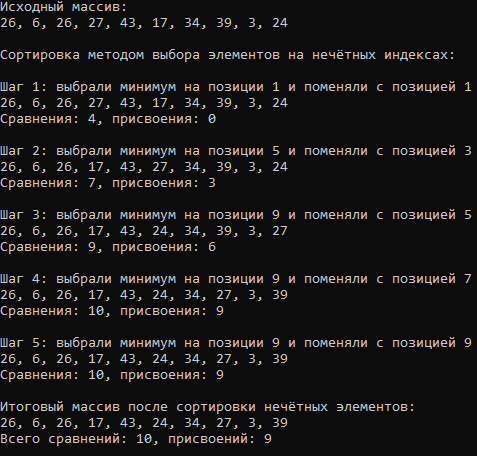


Рисунок 3 – Экранная форма программы

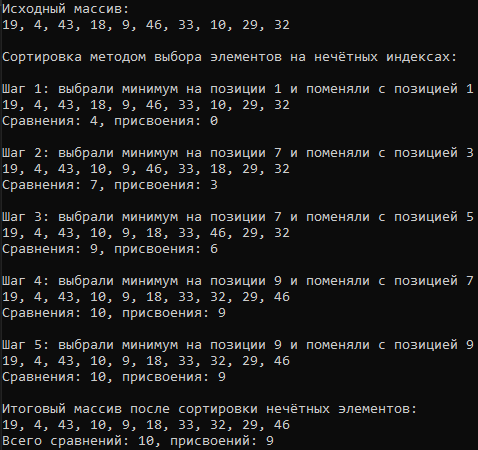


Рисунок 4 – Экранная форма программы

# Программа на C# (задача 2):

using System;

class Program

{

static void Main()

{

int[] array = new int[10];

Random rnd = new Random();

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

array[i] = rnd.Next(1, 51);

Console.WriteLine("Исходный массив:");

PrintArray(array);

Console.WriteLine("\nСортировка методом выбора элементов на нечётных индексах:\n");

SelectionSortOddPositions(array);

}

static void SelectionSortOddPositions(int[] arr)

{

int comparisons = 0;

int assignments = 0;

int n = arr.Length;

int oddCount = (n % 2 == 0) ? n / 2 : n / 2 + 1;

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

{

int minIndex = i;

for (int j = i + 2; j < n; j += 2)

{

comparisons++;

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

minIndex = j;

}

if (minIndex != i)

{

int temp = arr[i];

arr[i] = arr[minIndex];

arr[minIndex] = temp;

assignments += 3;

}

Console.WriteLine($"Шаг {((i + 1) / 2)}: выбрали минимум на позиции {minIndex} и поменяли с позицией {i}");

PrintArray(arr);

Console.WriteLine($"Сравнения: {comparisons}, присвоения: {assignments}\n");

}

Console.WriteLine("Итоговый массив после сортировки нечётных элементов:");

PrintArray(arr);

Console.WriteLine($"Всего сравнений: {comparisons}, присвоений: {assignments}");

}

static void PrintArray(int[] arr)

{

Console.WriteLine(string.Join(", ", arr));

}

}

# Вывод

В ходе выполнения этих задач научились реализовывать сортировки методом выбора, пузырька, вставки. Рассмотрели адаптацию сортировки выбором только элементов на нечётных позициях массива, а также выполнять пошаговый разбор с подсчётом операций сравнения и присвоения.