Отчет

по лабораторной работе № 2

«Теоретическая оценка среднего количества операций переприсваивания в алгоритме поиска минимума»

по дисциплине

ОСНОВЫ ТЕОРИИ АЛГОРИТМОВ

Выполнил студент гр. ИС/б-22о

Горбенко К.Н.

Проверил:

Карлусов В.Ю.

* 1. Цель работы

Лабораторная работа посвящена экспериментальной проверке теоретической оценки трудоемкости алгоритма поиска минимума и включает ознакомление с принципами использования генератора случайных чисел для создания наборов исходных данных.

* 1. Индивидуальное задание для варианта № 17

Наибольшее случайное число в последовательности – 750.

Количество элементов в массиве случайных чисел – 100, 500, 1000.

* 1. Текст программы на языке C#

Метод Main()

public static void Main()

{

using (var streamWriter = new StreamWriter("file.txt"))

{

var arraysWithRandomValues = new[]

{

CreateArrayWithRandomValues(100),

CreateArrayWithRandomValues(500),

CreateArrayWithRandomValues(1000),

};

foreach (var array in arraysWithRandomValues)

{

streamWriter.WriteLine($"Array with {array.Length} random values:");

streamWriter.WriteLine($"Number of assignments: {array.GetNumberOfAssignments()}");

streamWriter.WriteLine($"Mathematical expectation: {Math.Log(array.Length) + 0.57}");

}

}

Console.Read();

}

Класс NumberOfAssignments:

public static class NumberOfAssignments

{

const int arrayMaxValue = 750;

public static int[] CreateArrayWithRandomValues(int arrayLength)

{

var randomGenerator = new Random();

return new int[arrayLength].Select(x => randomGenerator.Next(arrayMaxValue)).ToArray();

}

public static int GetNumberOfAssignments(this int[] array)

{

var minimalValue = array[0];

var assignmentsCounter = 0;

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

{

if (array[i] < minimalValue)

{

minimalValue = array[i];

assignmentsCounter += 1;

}

}

return assignmentsCounter;

}

}

* 1. Блок-схема программы

Блок-схема основной части программы:

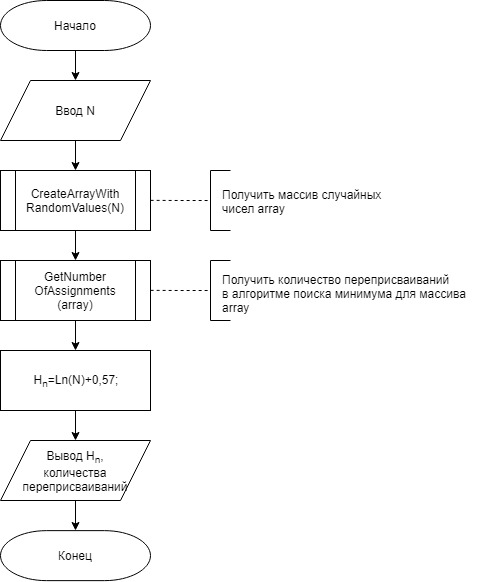


Рис. 1 – Блок-схема основной части программы

Блок-схема метода CreateArrayWithRandomValues(int N):

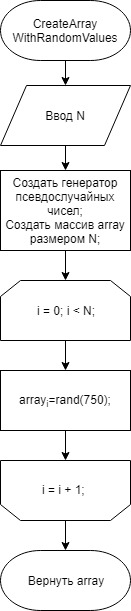


Рис. 2 – Блок-схема метода CreateArrayWithRandomValues

Блок-схема метода GetNumberOfAssignment(int[] array):

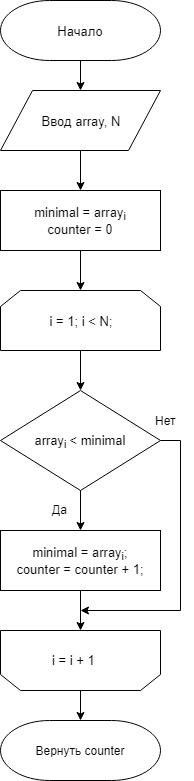


Рис. 3 – Блок-схема метода GetNumberOfAssignments

* 1. Описание функций, используемых в программе

В программе используется стандартный генератор псевдослучайных чисел – класс Random. Его экземпляр инициализируется начальным состоянием seed через конструктор класса. При каждом вызове метода Next(), он возвращает число, выглядящее случайным на фоне возвращенных ранее чисел этим генератором.

* 1. Результаты работы программы

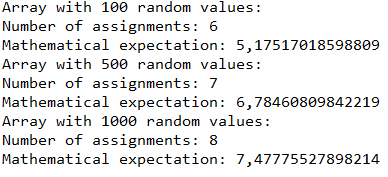


Рис. 4 – Содержимое файла с результатами работы программы

Для массива из 100 случайных чисел фактическое количество присваиваний составило 6. Значение n-го гармонического числа (математическое ожидание количества присваиваний) составило 5,17517018598809. Разность между практически полученным числом присваиваний и теоретическим составила: .

Для массива из 500 случайных чисел фактическое количество присваиваний составило 7. Значение n-го гармонического числа (математическое ожидание количества присваиваний) составило 6,78460809842219. Разность между практически полученным числом присваиваний и теоретическим составила: .

Для массива из 1000 случайных чисел фактическое количество присваиваний составило 8. Значение n-го гармонического числа (математическое ожидание количества присваиваний) составило 6,78460809842219. Разность между практически полученным числом присваиваний и теоретическим составила: .

* 1. Вывод

В ходе лабораторной работы была проведена оценка теоретической трудоемкости алгоритма поиска минимума. Для создания последовательности псевдослучайных чисел использовался генератор Random языка C#. Каждое число, возвращенное генератором Random, при статистическом анализе ведет себя как случайное по отношению к числам, возращенным этим генератором ранее.

В результате получили следующие разности между ожидаемым числом присваиваний и полученным практически:

* для массива из 100 случайных чисел разность составила 0,825;
* для массива из 500 случайных чисел разность составила 0,215;
* для массива из 100 случайных разность составила 0,522.

Эти результаты подтверждают корректность использования оценки для оценки количества переприсваиваний алгоритма поиска минимума.