



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Московский государственный технический университет имени
Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

Отчет по лабораторной работе №3

Название Случайные числа

Дисциплина Моделирование

Студент Егорова П. А.

Группа ИУ7-74Б

Оценка (баллы) _____

Преподаватель Рудаков И. В.

Москва — 2023 г.

1 Задание

Написать программу, которая генерирует алгоритмическим способом и получает табличным способом случайные последовательности одноразрядных, двухразрядных и трехразрядных целых чисел.

Разработать количественный критерий оценки случайности последовательности чисел. Для каждой полученной последовательности вычислить и вывести значение критерия.

Предусмотреть возможность ввода последовательности чисел и оценки их случайности с помощью критерия.

2 Теоретические сведения

2.1 Способы получения случайных чисел

На практике наиболее распространены 3 способа получения случайных чисел.

Аппаратный способ

При использовании аппаратного способа случайные числа вырабатываются специальной электронной приставкой (генератором случайных чисел). Реализация данного способа не требует дополнительных вычислений, необходима только одна операция – обращение к вычислительному устройству.

В качестве физического эффекта, лежащего в основе генерации случайных чисел, может использоваться, например, шум в электронных приборах. Для генерации необходимы источник шума, ключевая схема, формирователь импульсов и пересчетная схема.

Табличный способ

Случайные числа берутся из заранее подготовленной таблицы, которая находится во внешней или оперативной памяти. Числа в таблице

проверены на случайность и некоррелированы.

Алгоритмический способ

Алгоритмический способ основан на использовании специальных алгоритмов. К таким алгоритмам, например, относятся следующие:

- алгоритм Фон-Неймана (метод серединных квадратов);
- метод перемешивания (сдвигов);
- вихрь Мерсенна;
- линейный конгруэнтный генератор;

Последний выбран для реализации в данной лабораторной работе.

В этом методе каждое следующее число рассчитывается на основе предыдущего по формуле (1).

$$R_{n+1} = (a \cdot R_n + b) \bmod N, n \geq 1 \quad (1)$$

где a , b – коэффициенты, N – модуль.

Для качественного генератора требуется подобрать подходящие коэффициенты. Например, в таблице 1 приведены некоторые из них.

Таблица 1 – Примеры коэффициентов

| a | b | N |
|------------|----------|----------|
| 106 | 1283 | 6075 |
| 430 | 2531 | 11979 |
| 84589 | 15989 | 217728 |
| 1103515245 | 12345 | 2^{31} |
| ... | ... | ... |

2.2 Критерий оценки случайности последовательности

В рамках лабораторной работы в качестве статистической оценки используется критерий «хи-квадрат» (χ^2 -критерий).

Привлекая этот критерий можно оценить, удовлетворяет ли генератор требованию равномерного распределения или нет.

Используется статистика, представленная формулой (2).

$$V = \frac{1}{n} \sum_{s=\min}^{\max} \left(\frac{Y_s^2}{p_s} \right) - n \quad (2)$$

где n - длина последовательности, \min/\max - границы, в пределах которых находятся элементы последовательности, Y_s - число повторений числа s ,
$$p = \frac{1}{\max - \min}.$$

Вычисляется квантиль хи-квадрат от вычисленного V . Если полученное значение будет меньше 0.1 или больше 0.9, то эти числа считаются недостаточно случайными. Если же нет, то последовательность принимается как случайная.

3 Результаты работы программы

Интерфейс предполагает вывод первых 10 элементов из 1000 каждой последовательности (за исключением той, которую введёт пользователь, так как в этом случае длина фиксирована и равна 10).

Результат работы программы приведен на рисунках 1.. Можно заметить, что значения критерия в любом из столбцов находится в интервале от 0.1 до 0.9, что позволяет считать последовательности случайными.



Рисунок 1 – Результаты работы

4 Код программы

Код основной программы, которая инициирует генерацию последовательностей и рассчитывает критерии, приведен в листинге 1 (используемый язык – Swift).

Листинг 1 – Код основной программы

```

0 class CongruentMethod {
1     private let a: Int = 1103515245
2     private let b: Int = 12345
3     private let n: Int = Int(pow(Double(2), 31))
4     private var r: Int
5
6     init(seed: Int) {
7         self.r = seed
8     }
9
10    func getNum(start: Int, end: Int) -> Int {
11        r = (a * r + b) % n

```

```

12     return r % (end - start) + start
13 }
14 }
15
16 class AlgorithmMethod {
17     let helpers = Helpers()
18
19     func generateAlgorithmMethod(filename: String, start: Int, end: Int) {
20         var arr = [String]()
21         var nums = [Int]()
22         let algorithmMethod = CongruentMethod(seed: Int.random(in:
0...100));
23
24         for _ in 0..<1000 {
25             let num = algorithmMethod.getNum(start: start, end: end)
26             nums.append(num)
27             arr.append("\(num)")
28         }
29
30         helpers.write_to_file(text: arr.joined(separator: "\n"), filename:
filename)
31     }
32 }
33
34 func generateTableMethod(filename: String, start: Int, end: Int) {
35     let helpers = Helpers()
36     var arr = Array(repeating: 0, count: 1000)
37     let num = Int.random(in: 0...1000)
38     var j = num
39
40     var string = table1
41     if start == 10 {
42         string = table10
43     }
44     if start == 100 {
45         string = table100
46     }
47     var nums = [Int]()
48     let array = string.components(separatedBy: ",")
49     for el in array { nums.append(Int(el)!) }

```

```

50
51     for i in 0..

```