



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

По курсу: «Моделирование»

Тема: «Генерация псевдослучайных последовательностей»

Студент:

Пронин А. С.

Группа:

ИУ7-72Б

Преподаватель:

Рудаков И. В.

Оценка:

Москва

2022

Задание

Используя алгоритмический и табличный метод сгенерировать последовательность из 1000 одно-, дву- и трёхразрядных чисел. Отобразить первые 10 в графическом интерфейсе. Составить статистический критерий оценки случайности последовательностей и вывести его значение для каждой последовательности. Также предусмотреть возможность получения оценки для 10 чисел введённых пользователем.

1 Отчет

1.1 Теория

Существует три метода получения последовательности случайных чисел:

- аппаратный
- табличный
- алгоритмический

В данной лабораторной работе реализованы два последних из них

1.1.1 Алгоритмический генератор

Алгоритмический генератор использует одно или несколько значений для вычисления нового числа, после чего также используется для генерации последующих чисел.

В данной работе был использован линейный конгруэнтный метод. Последовательность в нём вычисляется по формуле

$$X_{n+1} = (aX_n + c) \bmod m, n \geq 1 \quad (1.1)$$

В данной лабораторной использованы значения из минимального стандартного генератора случайных чисел: $a = 16807$, $c = 2147483647$, $m = MAX_INT$. В качестве начального значения X_1 используется системное время.

1.1.2 Табличный генератор

Табличный генератор использует заранее подготовленные последовательности хранящиеся в памяти компьютера.

В лабораторной работе файлы были сгенерированы при помощи стандартной библиотеки языка *C#*. Начальное значение генерируется так же и обозначает смещение относительно начала файла.

1.1.3 Критерий оценки

В качестве статистического критерия оценки случайности был выбран критерий "хи-квадрат". В первую очередь вычисляется статистика:

$$V = \frac{1}{n} \sum_{s=min}^{max} \left(\frac{Y_i^2}{p} \right) - n \quad (1.2)$$

где n - длина последовательности, $[min, max)$ - диапазон чисел, Y_i - количество повторений числа i , $p = 1/(max - min)$.

После этого вычисляется критерий - квантиль хи-квадрат (степень свободы $max - min - 1$) от значения V . Полученное значение предлагается интерпретировать следующим образом:

- <0.1 или >0.9 - последовательность считается недостаточно случайной.
- >0.1 и <0.9 - последовательность считается случайной.

2 Текст программы

В листинге 2.1 представлен код программы, отвечающий за генерирование последовательностей и вычисление критерия оценки.

Листинг 2.1: Код программы

```
1 using System;
2 using System.IO;
3 using Accord.Math;
4 using MathNet.Numerics.Distributions;
5 using System.Collections.Generic;
6 using System.Linq;
7 using System.Text;
8 using System.Threading.Tasks;
9
10 namespace lab3
11 {
12     public abstract class IRndGen
13     {
14         public int MaxN;
15         public int MinN;
16         public abstract void SetSid(int s);
17         public abstract int Rand();
18         public abstract int[] RandArr(int size);
19     }
20
21     class RndAlgGen: IRndGen
22     {
23         private int seed;
24         public RndAlgGen(int minN, int maxN, int s=0)
25         {
26             MinN = minN;
27             MaxN = maxN;
28             SetSid(s);
29         }
30
31         public override void SetSid(int s)
32         {
33             seed = s;
```

```

34     }
35
36     public override int Rand()
37     {
38         seed = seed * 16807 + 2147483647;
39         var res = ((uint)seed >> 16) % (MaxN - MinN) + MinN;
40         return (int)res;
41     }
42
43     public override int[] RandArr(int size)
44     {
45         int[] res = new int[size];
46         for (int i = 0; i < size; i++)
47             res[i] = Rand();
48         return res;
49     }
50 }
51
52 class RndTabGen : IRndGen
53 {
54     private int seed;
55     private StreamReader f;
56     private string fpath;
57
58     public static void FillFile(string fpath_, int min, int
max)
59     {
60         var rand = new Random();
61         using (StreamWriter sw = File.CreateText(fpath_))
62         {
63             for (int i = 0; i < (max - min) * 1000; i++)
64                 sw.WriteLine(rand.Next(min, max));
65         }
66     }
67     public RndTabGen(int minN, int maxN, string fp, int s =
0)
68     {
69         MinN = minN;
70         MaxN = maxN;
71         fpath = fp;
72         SetSid(s);

```

```

73         var chi = new ChiSquared(1.0);
74     }
75
76     private void OpenFile(string fp)
77     {
78         fpath = fp;
79         if (f != null)
80             f.Close();
81         f = File.OpenText(fpath);
82         for (int i = 0; i < seed; i++)
83             f.ReadLine();
84     }
85
86     public override void SetSid(int s)
87     {
88         seed = s;
89         OpenFile(fpath);
90     }
91
92     public override int Rand()
93     {
94         string fileStr = f.ReadLine();
95         if (fileStr == null)
96         {
97             SetSid(0);
98             fileStr = f.ReadLine();
99         }
100         seed++;
101
102         var res = (UInt32.Parse(fileStr)) % (MaxN - MinN) +
MinN;
103         return (int)res;
104     }
105
106     public override int[] RandArr(int size)
107     {
108         int[] res = new int[size];
109         for (int i = 0; i < size; i++)
110             res[i] = Rand();
111         return res;
112     }

```

```

113     }
114
115     class Crit
116     {
117         public int[] arr;
118         public Crit(int[] arr_)
119         {
120             arr = arr_;
121         }
122
123         public double Value(int min, int max)
124         {
125             int n = arr.Length;
126             double p = 1.0 / (max - min);
127             double acc = 0;
128             for (int i = min; i < max; i++)
129                 acc += Math.Pow(arr.Count(x => x == i), 2) / p;
130             acc = acc / (double)n - n;
131
132             double res = ChiSquared.CDF(max - min - 1, acc);
133             return res;
134         }
135     }
136 }

```


3 Результаты

Для демонстрации работы критерия предоставлены 10 полей ввода. Оценка производится для целых чисел от 0 до 9. Примеры работы программы приведены на рисунках 3.1–3.4.

Генерация псевдослучайных последовательностей

Собственный ввод	Алгоритмический метод			Табличный метод		
0	5	55	785	5	88	823
0	5	24	629	2	31	532
0	5	98	630	7	10	306
0	3	57	448	5	42	935
0	7	50	126	0	68	692
0	5	24	335	1	71	683
0	3	87	186	2	42	161
0	8	18	718	6	51	506
0	2	65	148	9	98	407
0	2	25	784	2	24	987
0,9999999999999998	0,0547037	0,0143055	0,1247252	0,5737282	0,7174990	0,6988404

Рассчитать

Сгенерировать последовательности

Рис. 3.1: Последовательность из одинаковых чисел

Генерация псевдослучайных последовательностей

Собственный ввод	Алгоритмический метод			Табличный метод		
6	5	55	785	5	88	823
2	5	24	629	2	31	532
0	5	98	630	7	10	306
1	3	57	448	5	42	935
5	7	50	126	0	68	692
7	5	24	335	1	71	683
4	3	87	186	2	42	161
8	8	18	718	6	51	506
9	2	65	148	9	98	407
3	2	25	784	2	24	987
0	0,0547037	0,0143055	0,1247252	0,5737282	0,7174990	0,6988404

Рассчитать Сгенерировать последовательности

Рис. 3.2: Последовательность из чисел от 0 до 9

Генерация псевдослучайных последовательностей

Собственный ввод	Алгоритмический метод			Табличный метод		
0	5	55	785	5	88	823
1	5	24	629	2	31	532
2	5	98	630	7	10	306
3	3	57	448	5	42	935
4	7	50	126	0	68	692
5	5	24	335	1	71	683
6	3	87	186	2	42	161
7	8	18	718	6	51	506
8	2	65	148	9	98	407
9	2	25	784	2	24	987
0	0,0547037	0,0143055	0,1247252	0,5737282	0,7174990	0,6988404

Рассчитать Сгенерировать последовательности

Рис. 3.3: Возрастающая последовательность

Генерация псевдослучайных последовательностей

Собственный ввод	Алгоритмический метод			Табличный метод		
1	5	55	785	5	88	823
1	5	24	629	2	31	532
3	5	98	630	7	10	306
3	3	57	448	5	42	935
4	7	50	126	0	68	692
6	5	24	335	1	71	683
2	3	87	186	2	42	161
0	8	18	718	6	51	506
3	2	65	148	9	98	407
4	2	25	784	2	24	987
0,649514787676641	0,0547037	0,0143055	0,1247252	0,5737282	0,7174990	0,6988404

Рассчитать

Сгенерировать последовательности

Рис. 3.4: Введена случайная последовательность