



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»

Отчёт

по лабораторной работе № 3

Название: Исследование псевдослучайных чисел

Дисциплина: Моделирование

Студент

ИУ7-75Б

(Группа)

Д.В. Сусликов

(Подпись, дата)

(И.О. Фамилия)

Преподаватель

И.В. Рудаков

(Подпись, дата)

(И.О. Фамилия)

Москва, 2021

Содержание

| | |
|-----------------|----------|
| Задание | 3 |
| Теория | 4 |
| Примеры | 5 |
| Листинги | 8 |

Задание

Изучить методы генерирования псевдослучайных чисел, а также критерии оценки случайности последовательности. Реализовать критерий оценки случайной последовательности. Сравнить результаты работы данного критерия на одноразрядных, двухразрядных и трехразрядных последовательностях целых чисел. Последовательности получать алгоритмическим и табличным способами.

Теория

Псевдослучайная последовательность (ПСП) — последовательность чисел, которая была вычислена по некоторому правилу, но имеет свойства случайной последовательности чисел в рамках решаемой задачи.

Табличный способ генерация ПСП

Табличные генераторы в качестве источника случайных чисел используют заранее подготовленные таблицы, содержащие проверенные некоррелированные числа. Недостатки такого способа являются использование внешнего ресурса для хранения чисел и период, ограниченный размером таблицы.

Алгоритмический способ генерация ПСП

Алгоритмические генераторы случайных чисел для получения следующего псевдослучайного числа используют некоторое математическое правило, например, линейный конгруэнтный метод. В данной работе последовательность генерируется при помощи формулы:

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

Критерий оценки случайности последовательности

Для выполнения работы был выбран критерий «хи квадрат».

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

Это один из самых известных статистических критериев, также это основной метод, используемый в сочетании с другими критериями.

Примеры

Смотря на примеры работы программы для последовательности чисел в количестве 10000, сгенерированных линейным конгруэнтным и табличным методами, можно сделать вывод, что лучше себя в точности показал табличный метод.

Алгоритмический метод

| | 0..9 | 10..99 | 100..999 |
|----|------|--------|----------|
| 1 | 3 | 79 | 399 |
| 2 | 8 | 18 | 902 |
| 3 | 7 | 13 | 681 |
| 4 | 2 | 44 | 796 |
| 5 | 1 | 29 | 283 |
| 6 | 6 | 86 | 430 |
| 7 | 9 | 67 | 217 |
| 8 | 0 | 72 | 916 |
| 9 | 5 | 99 | 111 |
| 10 | 8 | 92 | 110 |

Вычислить

Мера случайности

0..9

6.604

10..99

79.37

100..999

855.08

Табличный метод

| | 0..9 | 10..99 | 100..999 |
|----|------|--------|----------|
| 1 | 3 | 55 | 347 |
| 2 | 1 | 49 | 696 |
| 3 | 9 | 33 | 905 |
| 4 | 1 | 55 | 506 |
| 5 | 0 | 44 | 711 |
| 6 | 9 | 77 | 755 |
| 7 | 8 | 76 | 191 |
| 8 | 6 | 87 | 302 |
| 9 | 8 | 33 | 358 |
| 10 | 2 | 12 | 324 |

Вычислить

Мера случайности

0..9

4.898

10..99

80.198

100..999

897.2

Ручной ввод (через пробел)

2 23 1 12 3 1 23 1

Вычислить

Мера случайности

38.0

Рисунок 1 – Пример 1

Алгоритмический метод

| | 0..9 | 10..99 | 100..999 |
|----|------|--------|----------|
| 1 | 5 | 23 | 603 |
| 2 | 0 | 32 | 566 |
| 3 | 3 | 43 | 605 |
| 4 | 0 | 84 | 372 |
| 5 | 5 | 47 | 495 |
| 6 | 2 | 22 | 134 |
| 7 | 5 | 77 | 369 |
| 8 | 6 | 66 | 684 |
| 9 | 3 | 19 | 383 |
| 10 | 0 | 56 | 426 |

Вычислить

Табличный метод

| | 0..9 | 10..99 | 100..999 |
|----|------|--------|----------|
| 1 | 5 | 74 | 808 |
| 2 | 8 | 24 | 170 |
| 3 | 2 | 21 | 463 |
| 4 | 8 | 45 | 499 |
| 5 | 4 | 58 | 876 |
| 6 | 3 | 95 | 648 |
| 7 | 4 | 55 | 191 |
| 8 | 2 | 23 | 828 |
| 9 | 3 | 85 | 777 |
| 10 | 2 | 69 | 105 |

Вычислить

Мера случайности

0..9

11.42

10..99

63.818

100..999

842.3

Мера случайности

0..9

9.402

10..99

70.586

100..999

854.54

Ручной ввод (через пробел)

1 1 1 1 1

Вычислить

Мера случайности

0.0

Рисунок 2 – Пример 2

Алгоритмический метод

| | 0..9 | 10..99 | 100..999 |
|----|------|--------|----------|
| 1 | 7 | 89 | 583 |
| 2 | 4 | 86 | 734 |
| 3 | 5 | 83 | 457 |
| 4 | 4 | 28 | 652 |
| 5 | 7 | 61 | 663 |
| 6 | 8 | 40 | 550 |
| 7 | 3 | 51 | 205 |
| 8 | 2 | 12 | 888 |
| 9 | 9 | 93 | 995 |
| 10 | 0 | 92 | 206 |

Вычислить

Мера случайности

0..9

5.98

10..99

75.338

100..999

806.48

Табличный метод

| | 0..9 | 10..99 | 100..999 |
|----|------|--------|----------|
| 1 | 7 | 97 | 499 |
| 2 | 4 | 26 | 449 |
| 3 | 3 | 37 | 940 |
| 4 | 4 | 95 | 693 |
| 5 | 4 | 74 | 924 |
| 6 | 1 | 80 | 445 |
| 7 | 8 | 19 | 587 |
| 8 | 4 | 54 | 888 |
| 9 | 5 | 49 | 942 |
| 10 | 3 | 42 | 313 |

Вычислить

Мера случайности

0..9

11.292

10..99

79.55

100..999

816.92

Ручной ввод (через пробел)

1 2 3 4 5

Вычислить

Мера случайности

0.0

Рисунок 3 – Пример 3

Листинг 1 – lab3.py

```
1  import sys
2  from PyQt5 import QtWidgets
3  from PyQt5 import uic, QtWidgets, QtGui
4  from PyQt5.QtWidgets import QApplication, QWidget,
5      QListWidgetItem, QTableWidgetItem, QMessageBox
6  import design
7
8  COUNT = 10000
9
10 class App(QtWidgets.QMainWindow, design.Ui_MainWindow):
11     def __init__(self):
12         self.current = 10
13         self.m = 2.**32
14         self.a = 1664525
15         self.c = 1013904223
16         self.count = 1
17
18     super().__init__()
19     self.setupUi(self)
20     self.initUI()
21
22     self.setFixedSize(858, 875)
23
24     def initUI(self):
25         self.calcAlgBtn.clicked.connect(self.calcAlgBtnPushed)
26         self.calcTableBtn.clicked.connect(self.calcTableBtnPushed)
27         self.calcManualBtn.clicked.connect(self.calcManualBtnPushed)
28
29     def get_number(self, low=0, high=100):
30         self.current = (self.a * self.current + self.c) % self.m
```



```

31         result = int(low + self.current % (high - low))
32         return result
33
34
35     def calcAlgBtnPushed(self):
36         one_alg, two_alg, three_alg = self.alg_rand()
37
38         for i in range(10):
39             self.tableAlg.setItem(i, 0, QTableWidgetItem(str(one_alg
40                 [10*(self.count-1):10*self.count][i])))
41             self.tableAlg.setItem(i, 1, QTableWidgetItem(str(two_alg
42                 [10*(self.count-1):10*self.count][i])))
43             self.tableAlg.setItem(i, 2, QTableWidgetItem(str(three_alg
44                 [10*(self.count-1):10*self.count][i])))
45
46             self.measureAlg1.clear()
47             self.measureAlg2.clear()
48             self.measureAlg3.clear()
49             self.measureAlg1.setText(str(round(self.calc_hi(one_alg,
50                 10000, 0, 10), 3)))
51             self.measureAlg2.setText(str(round(self.calc_hi(two_alg,
52                 10000, 10, 100), 3)))
53             self.measureAlg3.setText(str(round(self.calc_hi(three_alg,
54                 10000, 100, 1000), 3)))
55
56             self.count += 1
57
58     def calcTableBtnPushed(self):
59         one_tbl, two_tbl, three_tbl = self.table_rand()
60
61         for i in range(10):
62             self.tableTable.setItem(i, 0, QTableWidgetItem(str(one_tbl
63                 [:10][i])))
64             self.tableTable.setItem(i, 1, QTableWidgetItem(str(two_tbl
65                 [:10][i])))

```

```

58         self.tableTable.setItem(i, 2, QTableWidgetItem(str(
59             three_tbl[:10][i])))
60
61     self.measureTable1.clear()
62     self.measureTable2.clear()
63     self.measureTable3.clear()
64     self.measureTable1.setText(str(round(self.calc_hi(one_tbl,
65         10000, 0, 10), 3)))
66     self.measureTable2.setText(str(round(self.calc_hi(two_tbl,
67         10000, 10, 100), 3)))
68     self.measureTable3.setText(str(round(self.calc_hi(three_tbl,
69         10000, 100, 1000), 3)))
70
71     self.count += 1
72
73     def table_rand(self):
74         numbers = set()
75         with open('numbers.txt') as file:
76             line_num = 0
77             lines = islice(file, line_num, None)
78             for l in lines:
79                 numbers.update(set(l.split(" ")[1:-1]))
80                 line_num += 1
81                 if len(numbers) >= 3* COUNT + 1:
82                     break
83             numbers.remove("")
84             numbers = list(numbers)[:3*COUNT]
85             one_digit = [int(i)%10 for i in numbers[:COUNT]]
86             two_digits = [int(i)%90 + 10 for i in numbers[COUNT:COUNT *
87                 2]]
88             three_digits = [int(i)%900 + 100 for i in numbers[COUNT*2:3*
89                 COUNT]]
90             return one_digit, two_digits, three_digits
91
92     def alg_rand(self):

```

```

87     one_digit = [self.get_number(0, 10) for i in range(COUNT)]
88     two_digits = [self.get_number(10, 100) for i in range(COUNT)
89                    ]
90     three_digits = [self.get_number(100, 1000) for i in range(
91                       COUNT)]
92     return one_digit, two_digits, three_digits
93
94
95     def calc_hi(self, arr, n, start, end):
96         tab = [0 for i in range(start + end)]
97
98         for i in range(n):
99             tab[arr[i]] += 1
100
101         s = 0
102
103         for i in tab:
104             s += i * i
105
106         return s * (end - start) / n - n
107
108     def calcManualBtnPushed(self):
109         nums = self.manualText.toPlainText().split()
110         for i in range(len(nums)):
111             nums[i] = int(nums[i])
112
113         hi = self.calc_hi(nums, len(nums), min(nums), max(nums)+1)
114         self.measureManual.setText(str(round(hi, 3)))
115
116     def main():
117         app = QtWidgets.QApplication(sys.argv)
118         window = App()
119         window.show()
120         app.exec_()
121
122 if __name__ == '__main__':
123     main()

```