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

к лабораторной работе №3

По курсу: «Моделирование»

Студент ***ИУ7И-76Б*** _____ **Нгуен Ф. С.**
(Группа) (Подпись, дата) (И.О. Фамилия)

Преподаватель _____ **Рудаков И.В.**
(Подпись, дата) (И.О. Фамилия)

Москва, 2021 г.

Оглавление

I. Теоретическая часть

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.

Генераторы случайных чисел по способу получения чисел делятся на:

- Аппаратные;
- Табличные;
- Алгоритмические;

1. **Аппаратные**:

   Аппаратные генераторы случайных чисел – это устройства, использующие для создания случайных чисел замеры параметров некоторых физических процессов.

   Случайные числа вырабатываются специальной электронной приставкой, то есть генератором случайных чисел. Как правило это практически любое внешнее устройство компьютера. Реализация этого способа не требует дополнительных вычислительных операций по выработке чисел, а необходима только операция обращения к этому внешнему устройству.
2. **Табличные**:

   Формируется таблица и записывается в память. Математики проверили её на случайность и мы можем многократно её использовать.

   Недостаток - использование внешнего ресурса для хранения чисел, количество чисел ограничено.
3. **Алгоритмические**:

   Достоинства:

   - Однократная проверка,
   - Можно многократно воспроизводить последовательность, относительно малое место в оперативной памяти,
   - Не используются внешние устройства.

   Недостаток:

   - Запас чисел ограничен периодом,
   - Требуются затраты машинного времени.
4. **Линейный конгруэнтный метод**

   Генераторы псевдослучайных чисел могут работать по разным алгоритмам.

Одним из простейших генераторов является так называемый линейный конгруэнтный генератор, который для вычисления очередного числа ki использует формулу:

$X_i = (a * X_{i-1} + c) \, mod \, m$	*(1)*

Где m > 0 - модуль,
a, c - константа,
X_{i-1} — предыдущее псевдослучайное число.
X_0 -начальное значение.

5. **Критерий равномерности:**

Был выбран критерий χ^2.

Это один из самых известных статистических критериев, также это основной метод, используемый в сочетании с дргуими критериями. С помощью этого критерия можно узнать, удовлетворяет ли генератор случайных чисел требованию равномерного распределения или нет.

Для оценки по этому критерию необходимо вычислить статистику

$Z = \frac{1}{n} \sum_{k=1}^{s} \frac{Y_k^2}{p_k} - n \sim \chi^2(s-1)$	*(2)*

Где n – количество независимых испытаний,
S – количество категорий (для задания лабораторной s = 10, 90, 900),
Yk — число наблюдений действительно относятся к категории K,
Pk — вероятность того, что каждое наблюдение относится к категории k.

Если вычисленное Z окажется меньше 1%-й точки или больше 99%-й точки, можно сделать вывод, что эти числа недостаточно случайные.

Если Z лежит между 1% и 5% точками или между 95% и 99% точками, то эти числа «подозрительны».

Если Z лежит между 5% и 10% точками или 90%-95% точками, то числа можно считать «почти подозрительными».

Обычно необходимо произвести проверку три раза и более с разными данными. Если по крайней мере два из трех результатов оказываются подозрительными, то числа рассматриваются как недостаточно случайные

N-1	P = 1%	P=5%	P=25%	P=50%	P=75%	P=95%	P=99%
9	2.088	3.325	5.899	8.343	11.39	1692	21.67
89	60.93	68.25	79.68	88.33	97.60	112.02	122.94
899	803.31	830.41	870.05	898.33	927.23	969.86	1000.57

II. Экспериментальная часть

Результаты работы программы (N = 10000)
(m = 2e31, a = 12345, c = 101234323, $X_0 = 10$)

	Алгоритмический метод			Табличный метод		
	1 разряд	2 разряд	3 разряд	1 разряд	2 разряд	3 разряд
0	3	45	717	7	75	995
1	2	92	992	3	77	301
2	3	61	871	0	94	384
3	0	46	766	6	87	683
4	7	21	121	3	43	901
5	8	58	160	6	12	105
6	1	93	527	9	36	222
7	6	36	126	7	37	606
8	1	99	785	2	99	119
9	8	96	116	5	20	556
10	5	81	251	1	10	182
11	2	96	430	2	92	132
12	9	27	265	5	31	525
13	2	52	568	4	19	824
14	7	13	775	6	30	745
...	...	...	...	...	...	...
Chiquare	48.113 %	52.273 %	51.548 %	76.283 %	3.085 %	4.673 %
Chiquare	9.140	90.098	900.980	12.764	66.644	830.060

III. Код программы:

```
from scipy.stats import chi2

class LinearCong:
    def __init__(self):
        self.current = 10
        self.m = 2.**31
        self.a = 12345
        self.c = 101234323

    def next(self, low=0, high=10):
        self.current = (self.a * self.current + self.c) % self.m
        result = int(low + self.current % (high - low))
        return result

class FileTable:
    def __init__(self):
        self.nums = None
        with open('nums.txt', 'r') as f:
            a = list(f.read().split('\n'))
            nums = [list(i.split()) for i in a]

            self.nums = nums
            self.columns = len(self.nums[0])
```

```
            self.rows = len(self.nums)

            self.cur_x = 0
            self.cur_y = 0

    def next(self):
        self.cur_x += 1
        if self.cur_x == self.columns:
            self.cur_x = 0
            self.cur_y += 1
        if self.cur_y == self.rows:
            self.cur_y = 0

        return self.nums[self.cur_y][self.cur_x]

def ChiSquare(arr, a = None, b = None):
    if (a == None):
        a = min(arr)
    if (b == None):
        b = max(arr)
    tmp = [0 for i in range(b - a+1)]
    for i in arr:
        tmp[i - a] += 1

    n = len(arr)
    k = len(tmp)

    p = 1.0 / k

    chi = 0

    for i in tmp:
        chi += i * i
    chi = chi / p / n - n

    return (chi2.cdf(chi, k)) * 100, chi,


def output(RandomAlg, RandomTab, ChiAlg, ChiTab, isPrint = False):

    ...

def main():
    algGen = LinearCong()
    tabGen = FileTable()
    n = 10000

    a, b = 0, 10
    _1DigitAlg = [algGen.next(a, b) for i in range(n)]
    a, b = 10, 100
    _2DigitAlg = [algGen.next(a, b) for i in range(n)]
    a, b = 100, 1000
    _3DigitAlg = [algGen.next(a, b) for i in range(n)]
    RandomAlg = [_1DigitAlg, _2DigitAlg, _3DigitAlg]

    _1DigTab = [int(tabGen.next()) % 10 for i in range(n)]
    _2DigTab = [int(tabGen.next()) % 90 + 10 for i in range(n)]
    _3DigTab = [int(tabGen.next()) % 900 + 100 for i in range(n)]
    RandomTab = [_1DigTab, _2DigTab, _3DigTab]


    ChiAlg = [ChiSquare(RandomAlg[0], 0, 9),  ChiSquare(RandomAlg[1], 10, 99),
ChiSquare(RandomAlg[2], 100, 999)]
```

```
    ChiTab = [ChiSquare(RandomTab[0], 0, 9),  ChiSquare(RandomTab[1], 10, 99),
ChiSquare(RandomTab[2], 100, 999)]

    output(RandomAlg, RandomTab, ChiAlg, ChiTab, 1)

main()
```