



Национальный исследовательский университет
«Высшая школа экономики» Специальность
«Компьютерная безопасность»

Отчет
По лабораторной работе №4
«Сортировки»
по дисциплине «Методы
программирования» направления
«Компьютерная безопасность»

Шаплавский Л.П.
СКБ-182

Москва, 2021

Практическое задание по теме

1. Модифицировать (предложить собственные) два метода генерации псевдослучайных чисел.
2. Получить не менее 10 выборок каждым методом (диапазон чисел в каждой выборке не менее 10000) объемом не менее 50 элементов каждая.
3. Для каждой выборки посчитать среднее, отклонение и коэффициент вариации. Сделать вывод об однородности выборки.
4. Каждую выборку проверить на равномерность распределения и случайность, используя критерий Хи-квадрат.
5. Засечь время генерации чисел от тысячи до миллиона элементов обоими предложенными методами и любым стандартным методом используемого языка программирования. Построить графики сравнения скоростей в зависимости от объема выборки.
6. В отчете обязательно отразить: код алгоритма генерации и критерия проверки гипотезы, скриншот с результатами выполнения, анализ полученных результатов и выводы.

main.py

```
from generator import *

m = (2**31) - 1
k = 1220703125
b = 7

generation_of_samples(m,k,b)

print('-----Тест скорости-----')
size = [1000,5000,10000,50000,100000,250000,500000,1000000]

test_of_speed(size,m,k,b)
```

generator.py

```
import datetime
from time import sleep,time
from testmodul import *
import random

def get_value():
    return datetime.datetime.now().microsecond

def generator_of_array_LKN(size,m,k,b):
    sleep(0.001)
    r0 = (get_value())%100
    array = []
    for i in range(size):
        r1 = ((k * r0 + b) % m) % 10000
        r0 = r1
        array.append(r1)
    return array

def generator_of_array(size,m,k1,k2,b):
    sleep(0.001)
    r0 = (get_value())%100
    array = []
    for i in range(size):
        r1 = ((k1 * r0**2 + k2 * r0 + b) % m) % 10000
        r0 = r1
        array.append(r1)
    return array

def generator_of_array_random(size):
    a = []
    for i in range(size):
        a.append(random.randint(0,10000))
    sleep(0.0001)
    return a

def generation_of_sampls(m, k, b):
    sempl1 = []
    sempl2 = []

    for i in range(10):
        sempl1.append(generator_of_array_LKN(50,m,k,b))
        sempl2.append(generator_of_array(50,m,k,k+7,b))
    print(sempl1)
    print(sempl2)

    print('\n\n-----1-----')
    for i in range(len(sempl1)):
        print('\nНомер выборки:', i+1)
        test_of_array(sempl1[i])

    print('\n\n-----2-----')
    for i in range(len(sempl2)):
        print('\nНомер выборки:', i + 1)
        test_of_array(sempl2[i])
    return 0

def test_of_speed(size,m,k,b):
    for i in size:
        start_time1 = time()
        generator_of_array_LKN(i,m,k,b)
        time1 = time() - start_time1
```

```
start_time2 = time()
generator_of_array(i, m, k, k+7, b)
time2 = time() - start_time2

start_time3 = time()
generator_of_array_random(i)
time3 = time() - start_time3

print('\n-----', i, '-----')
print('Время работы линейного конгруэнтного метода',
str(time1).replace('.', ', '))
print('\nВремя работы квадратичный конгруэнтного метода',
str(time2).replace('.', ', '))
print('\nВремя работы Python', str(time3).replace('.', ', '))
return 0
```

testmodul2.py

```
import math

def expected_value (array):
    r = 0
    for i in range(len(array)):
        r += array[i]
    return r/len(array)

def variance (array):
    expevalue = expected_value(array)
    d = 0
    for i in range(len(array)):
        d += (array[i]-expevalue)**2
    return d/len(array)

def standard_deviation (array):
    return variance(array)**(1/2)

def variation (array):
    return variance(array)**(1/2) / expected_value(array) * 100

def chia(array):
    b = sorted(array)
    k = math.ceil(math.log2(len(array))+1)
    step = 10000/k
    p = 1/k
    frequency = []
    for i in range(k):
        counter = 0
        for j in b:
            if (j > i*step) and (j <= (i+1)*step):
                counter += 1
            else:
                continue
        frequency.append(counter)
    chi = 0
    for i in range(k):
        chi += ((frequency[i] - p*len(b))**2) / (p*len(b))
    return chi

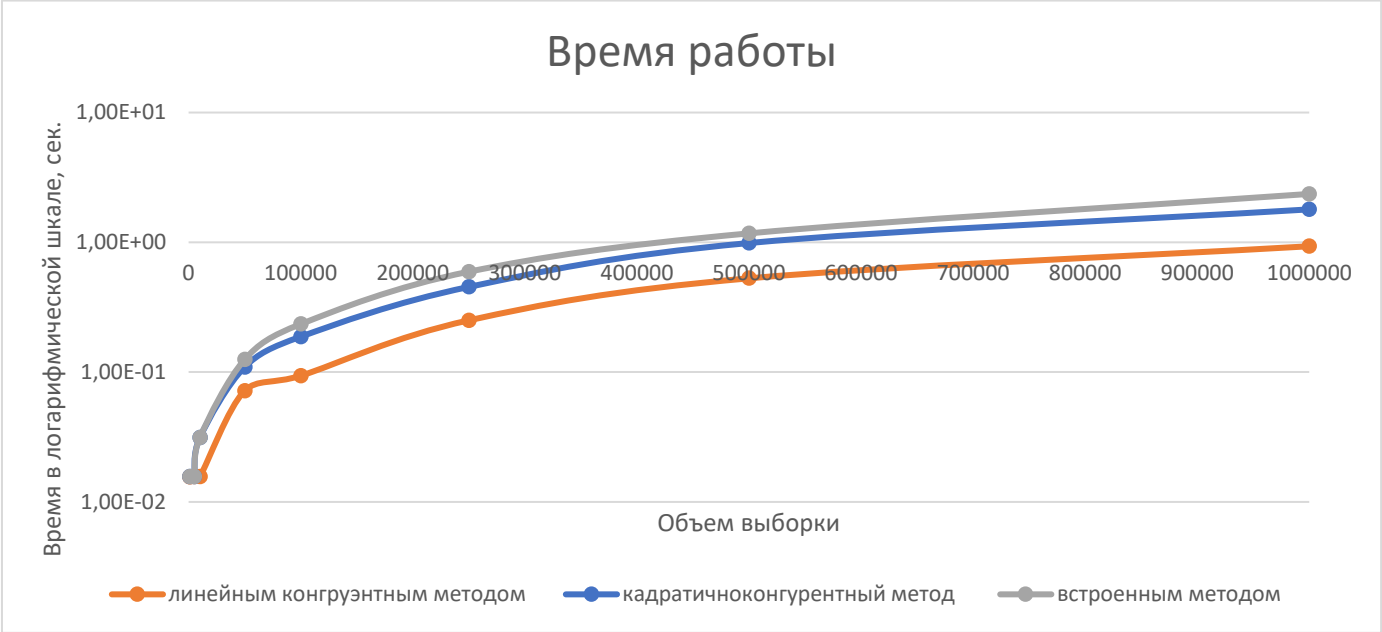
def test_of_array(array):
    print('-----')
    print('Матожидание: ', expected_value(array), ' | Дисперсия: ', variance(array),
'\nВариация: ', variation(array), ' | Хи-квадрат', chia(array))
    print('-----')
    if (chia(array) >= 0.18) and (chia(array) <=16.8):
        print("равномерное распределение")
    else:
        print("не равномерное распределение")

    if (variation(array) < 33):
        print("выборка однородна")
    else:
        print("выборка не однородна")
    return 0
```

Результат:

Размер выборки	Время		
	линейным конгруэнтным методом	кадратичноконгурентный метод	встроенным методом
1000	1,56E-02	1,56E-02	0,015626907
5000	1,56E-02	1,56E-02	0,015624285
10000	1,56E-02	3,12E-02	0,031253099
50000	7,18E-02	1,09E-01	0,124996185
100000	9,37E-02	1,88E-01	0,234371424
250000	2,51E-01	4,53E-01	0,593749523
500000	5,28E-01	9,87E-01	1,171888351
1000000	9,32E-01	1,79E+00	2,359407425

График:



Вывод программы

-----1-----

Номер выборки: 1

Матожидание: 4519.62 | Дисперсия: 8584962.035600001
Вариация: 64.82869325874165 | Хи-квадрат 11.320000000000002

равномерное распределение
выборка не однородна

Номер выборки: 2

Матожидание: 4497.2 | Дисперсия: 7480604.840000001
Вариация: 60.81716267240721 | Хи-квадрат 21.020000000000003

не равномерное распределение
выборка не однородна

Номер выборки: 3

Матожидание: 5252.84 | Дисперсия: 7308686.854400001
Вариация: 51.46660309070808 | Хи-квадрат 2.6399999999999997

равномерное распределение
выборка не однородна

Номер выборки: 4

Матожидание: 4933.42 | Дисперсия: 5739624.803600001
Вариация: 48.56167540746142 | Хи-квадрат 7.960000000000001

равномерное распределение
выборка не однородна

Номер выборки: 5

Матожидание: 4497.2 | Дисперсия: 7480604.840000001
Вариация: 60.81716267240721 | Хи-квадрат 21.020000000000003

не равномерное распределение
выборка не однородна

Номер выборки: 6

Матожидание: 5334.56 | Дисперсия: 7957110.286399999
Вариация: 52.87849482491035 | Хи-квадрат 5.44

равномерное распределение
выборка не однородна

Номер выборки: 7

Матожидание: 5402.96 | Дисперсия: 7909930.4384
Вариация: 52.05405658063454 | Хи-квадрат 5.72

равномерное распределение
выборка не однородна

Номер выборки: 8

Матожидание: 5256.22 | Дисперсия: 9492998.251600001
Вариация: 58.617617956377764 | Хи-квадрат 2.92

равномерное распределение
выборка не однородна

Номер выборки: 9

Матожидание: 4869.18 | Дисперсия: 7867527.1076
Вариация: 57.60541297964884 | Хи-квадрат 2.64

равномерное распределение
выборка не однородна

Номер выборки: 10

Матожидание: 4884.0 | Дисперсия: 7958183.92
Вариация: 57.76055141951307 | Хи-квадрат 3.7600000000000007

равномерное распределение
выборка не однородна

-----2-----

Номер выборки: 1

Матожидание: 4746.76 | Дисперсия: 7469239.062400002
Вариация: 57.57592260707829 | Хи-квадрат 3.1999999999999997

равномерное распределение
выборка не однородна

Номер выборки: 2

Матожидание: 4899.62 | Дисперсия: 8009677.315599997
Вариация: 57.762384257952895 | Хи-квадрат 4.04

равномерное распределение
выборка не однородна

Номер выборки: 3

Матожидание: 4876.68 | Дисперсия: 8730152.777600002
Вариация: 60.588006474024006 | Хи-квадрат 4.6000000000000005

равномерное распределение
выборка не однородна

Номер выборки: 4

Матожидание: 5106.04 | Дисперсия: 9337214.278399998
Вариация: 59.84452855878111 | Хи-квадрат 11.320000000000002

равномерное распределение
выборка не однородна

Номер выборки: 5

Матожидание: 4612.44 | Дисперсия: 8152631.006400001
Вариация: 61.9039217467016 | Хи-квадрат 1.5200000000000002

равномерное распределение
выборка не однородна

Номер выборки: 6

Матожидание: 4935.06 | Дисперсия: 8641615.216400003
Вариация: 59.56690359163079 | Хи-квадрат 2.64

равномерное распределение
выборка не однородна

Номер выборки: 7

Матожидание: 4659.16 | Дисперсия: 9173426.374400001
Вариация: 65.00670673175748 | Хи-квадрат 9.64

равномерное распределение
выборка не однородна

Номер выборки: 8

Матожидание: 4855.92 | Дисперсия: 8988626.1936
Вариация: 61.74121003468556 | Хи-квадрат 3.2

равномерное распределение
выборка не однородна

Номер выборки: 9

Матожидание: 4996.16 | Дисперсия: 6844556.6943999985
Вариация: 52.36442325528987 | Хи-квадрат 13.560000000000004

равномерное распределение
выборка не однородна

Номер выборки: 10

Матожидание: 4504.2 | Дисперсия: 7633788.239999998
Вариация: 61.341216992471914 | Хи-квадрат 2.64

равномерное распределение
выборка не однородна

Вывод:

Результаты показывают, что линейный конгруэнтный и квадратичноконгруэнтный методы работают быстрее чем встроенные методы языка программирования