

ФЕДЕРАЛЬНОЕ АГЕНТСТВО СВЯЗИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«СИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ТЕЛЕКОММУНИКАЦИЙ И ИНФОРМАТИКИ»

ФАКУЛЬТЕТ ИНФОРМАТИКИ И ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

Кафедра вычислительных систем

**ОТЧЁТ**

по лабораторной работе №3  
по дисциплине «Моделирование»  
на тему «Генерация и исследование рандомизированных  
цепей Маркова»

Выполнил:  
студент группы ИВ-222  
Терешков Р. В.

Проверил:  
д.т.н., доцент  
Родионов А. С.

Новосибирск – 2016

## Задание

В данной лабораторной работе исследуется поведение моделей рандомизированных цепей Маркова, заданных двустохастическими матрицами переходов. Матрицы переходов между состояниями в цепях представлены ниже.

### Цель 1:

0.1	0.4	0.5	0	0	0	0	0	0	0
0	0.1	0.4	0.5	0	0	0	0	0	0
0	0	0.1	0.4	0.5	0	0	0	0	0
0	0	0	0.1	0.4	0.5	0	0	0	0
0	0	0	0	0.1	0.4	0.5	0	0	0
0	0	0	0	0	0.1	0.4	0.5	0	0
0	0	0	0	0	0	0.1	0.4	0.5	0
0	0	0	0	0	0	0	0.1	0.4	0.5
0.5	0	0	0	0	0	0	0	0.1	0.4
0.4	0.5	0	0	0	0	0	0	0	0.1

Цепь 2:

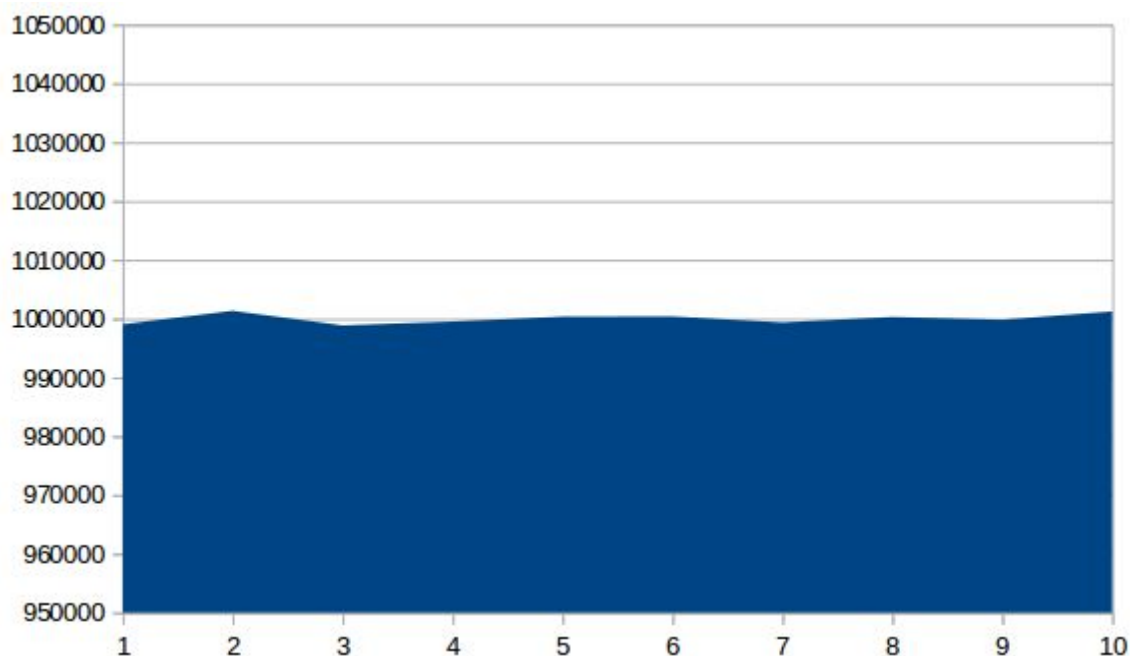
	0.5	0.5	0	0	0	0	0	0	0
	0	0.5	0.5	0	0	0	0	0	0
	0	0	0.5	0.5	0	0	0	0	0
	0	0	0	0.5	0.5	0	0	0	0
	0	0	0	0	0.5	0.5	0	0	0
	0	0	0	0	0	0.5	0.5	0	0
	0	0	0	0	0	0	0.5	0.5	0
	0	0	0	0	0	0	0	0.5	0.5
	0.5	0	0	0	0	0	0	0	0.5

## Ход работы и результаты моделирования

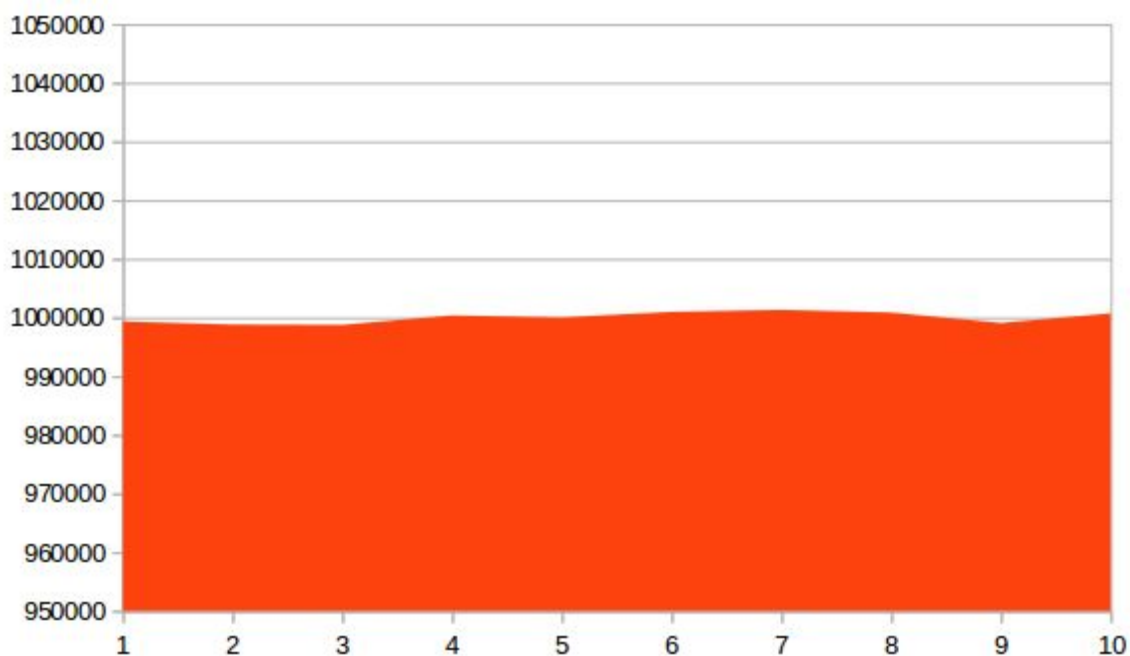
В ходе выполнения лабораторной работы была реализована программа на языке программирования С, выполняющая моделирование рандомизированных цепей Маркова.

Условно разобьём ход работы на три части. В первой части построим гистограмму состояний в цепи Маркова.

Цель 1:



Цель 2:



Во второй части работы необходимо построить гистограмму распределения значений функции. Функция экспоненциального распределения:

$$F(x) = 1 - e^{-\lambda x};$$

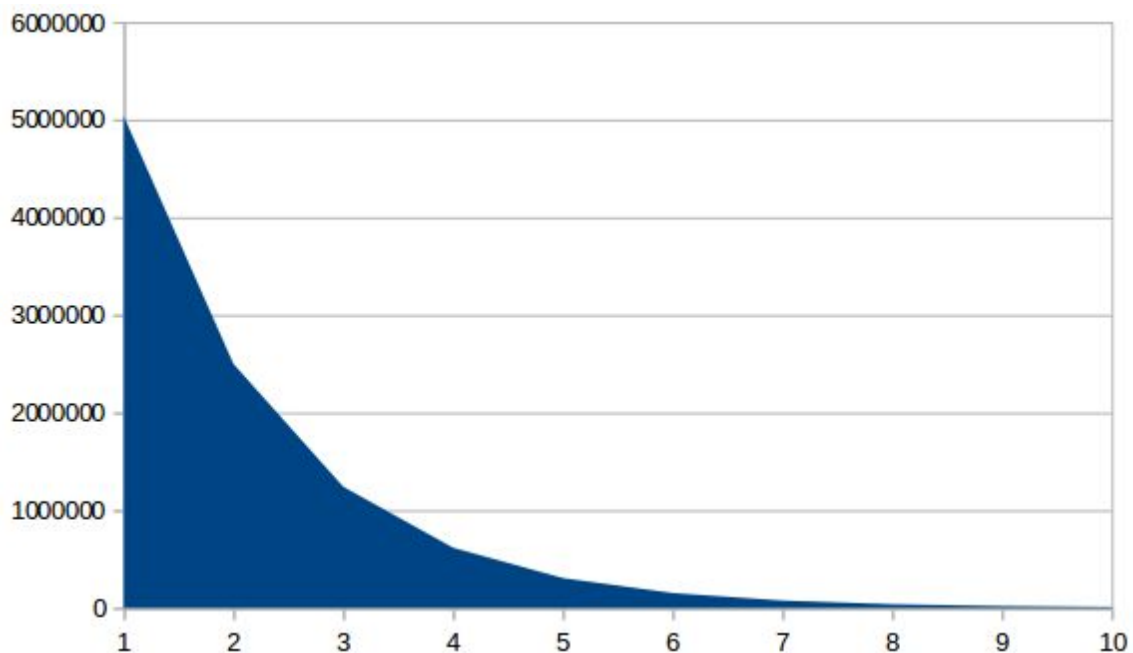
находим экспоненциальное распределение:

$$f(x) = \lambda e^{-\lambda x};$$

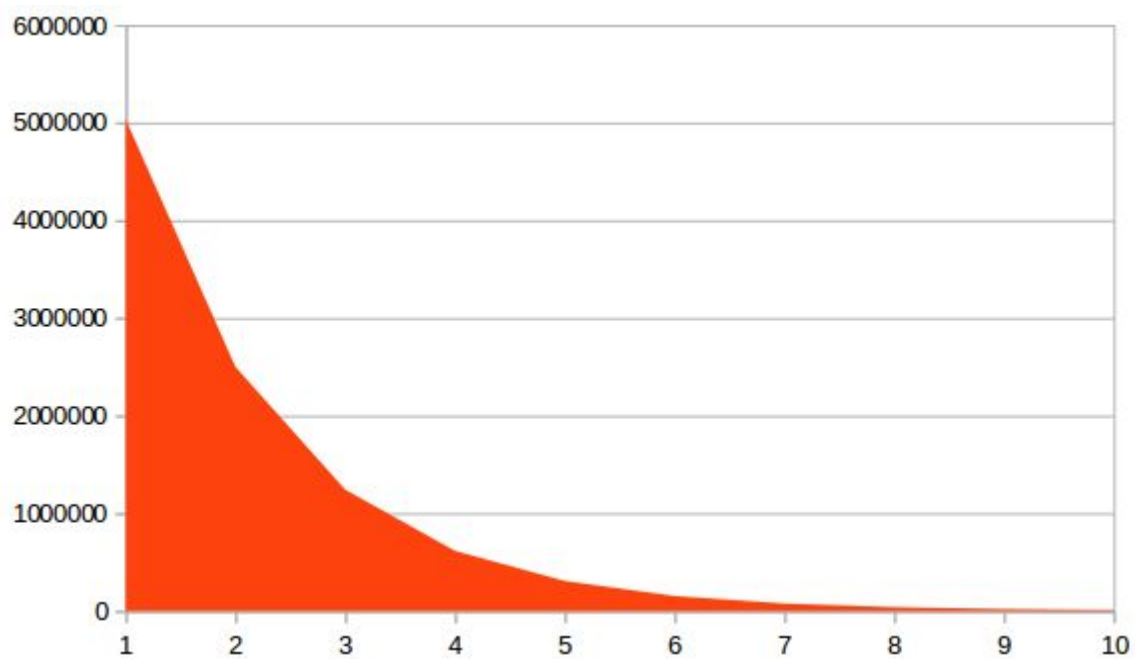
тогда случайная величина находится по формуле:

$$\xi = -\lambda * \log(x).$$

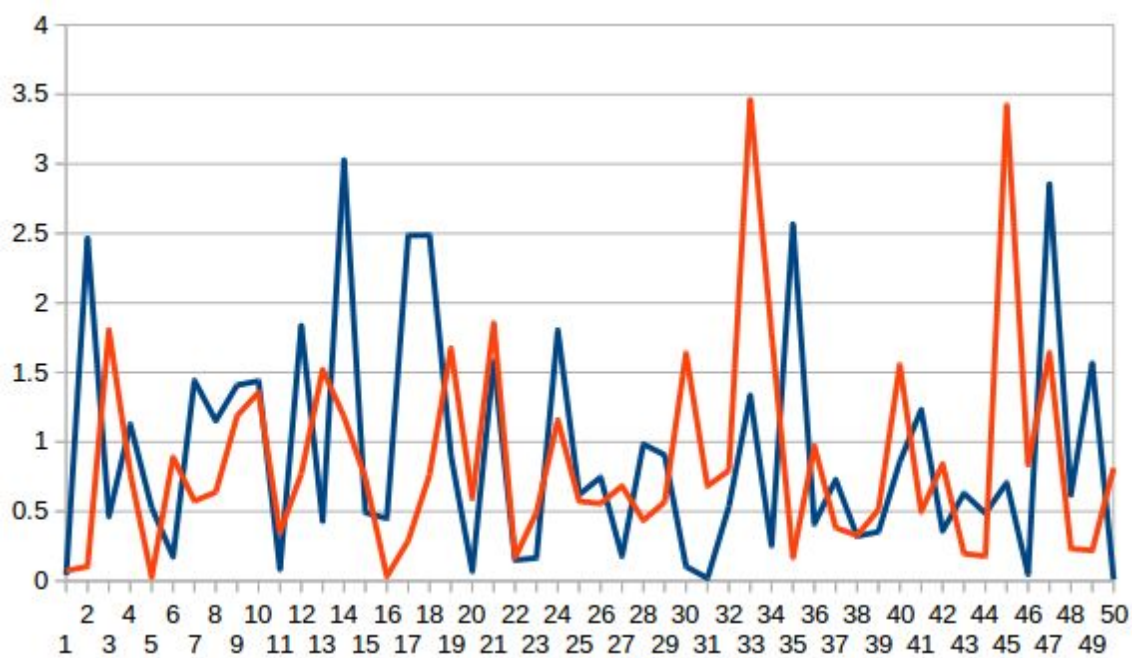
Гистограмма для цепи 1 при параметре  $\lambda = 1$  примет следующий вид:



Для цепи 2:



В заключительной части работы оценим поведение моделируемой случайной величины для двух рандомизированных цепей Маркова.



## Листинг программы

```
#include <math.h>
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

#define N 10000000
#define n 10

double p[n][n] =
{
    0.1, 0.4, 0.5, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
    0.0, 0.1, 0.4, 0.5, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
    0.0, 0.0, 0.1, 0.4, 0.5, 0.0, 0.0, 0.0, 0.0, 0.0,
    0.0, 0.0, 0.0, 0.1, 0.4, 0.5, 0.0, 0.0, 0.0, 0.0,
    0.0, 0.0, 0.0, 0.0, 0.1, 0.4, 0.5, 0.0, 0.0, 0.0,
    0.0, 0.0, 0.0, 0.0, 0.0, 0.1, 0.4, 0.5, 0.0, 0.0,
    0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.1, 0.4, 0.5, 0.0,
    0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.1, 0.4, 0.5,
    0.5, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.1, 0.4,
    0.4, 0.5, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.1
};

double p2[n][n] =
{
    0.5, 0.5, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
    0.0, 0.5, 0.5, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
    0.0, 0.0, 0.5, 0.5, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
    0.0, 0.0, 0.0, 0.5, 0.5, 0.0, 0.0, 0.0, 0.0, 0.0,
    0.0, 0.0, 0.0, 0.0, 0.5, 0.5, 0.0, 0.0, 0.0, 0.0,
    0.0, 0.0, 0.0, 0.0, 0.0, 0.5, 0.5, 0.0, 0.0, 0.0,
    0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.5, 0.5, 0.0, 0.0,
    0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.5, 0.5, 0.0,
    0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.5, 0.5,
    0.5, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.5
};

int histo[n] = { 0 };
int histo_x[n] = { 0 };

int main()
{
    srand((unsigned)time(NULL));

    int cur = 0, prev, idx;
    int cnt_mtx[n][n];
    double pp[n][n], ksi_mass[50];
    double ksi;
```

```

FILE *fp = fopen("res", "w");

for (int i = 0; i < n; ++i) {
    pp[i][0] = p[i][0];
    for (int j = 1; j < n; ++j) {
        pp[i][j] = p[i][j] + pp[i][j - 1];
    }
}

for (int z = 0; z < N; ++z) {
    prev = cur;
    double tmp = drand48();
    for (int i = 0; i < n; ++i) {
        if (tmp < pp[cur][i]) {
            cur = i;
            ++histo[cur];

            ksi = drand48();
            ksi = -1.0 * (log(ksi));
            idx = (int)(ksi / 0.7);
            ++histo_x[idx];
            break;
        }
    }
    if (z < 50) ksi_mass[z] = ksi;
}

fprintf(fp, "\n*****\n\n");
for (int i = 0; i < n; ++i) fprintf(fp, "%d\n", histo[i]);
fprintf(fp, "\n*****\n\n");
for (int i = 0; i < n; ++i) fprintf(fp, "%d\n", histo_x[i]);
fprintf(fp, "\n*****\n\n");
for (int i = 0; i < 50; ++i) fprintf(fp, "%.6lf\n", ksi_mass[i]);

fclose(fp);
return 0;
}

```