Федеральное агентство связи
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Сибирский государственный университет телекоммуникаций и информатики» (СибГУТИ)

Отчет

по лабораторной работе №3

по дисциплине «Теория массового обслуживания»

Тема: «Цепи Маркова и системы массового обслуживания»

Вариант 3

Выполнил:
студент гр. ИА-232
Московских Дмитрий Петрович

Новосибирск 2024

Цель работы

Цель лабораторной работы – изучить методы создания и анализа цепей Маркова.

Теоретические положения

Цепи Маркова

Цепь Маркова – последовательность случайных событий с конечным или счётным числом исходов, характеризующаяся тем свойством, что, говоря нестрого, при фиксированном настоящем будущее независимо от прошлого.

Матрица переходов

Переходы в цепях Маркова могут быть заданы при помощи матрицы переходов, в которой каждый элемент матрицы pij показывает вероятность перехода цепи из состоянияi в состояние j.

$$P = \begin{pmatrix} p_{11} & p_{12} & \cdots & p_{1N} \\ p_{21} & p_{22} & \cdots & p_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ p_{N1} & p_{N2} & \cdots & p_{NN} \end{pmatrix}$$

Системы массового обслуживания

Система массового обслуживания (СМО) – система, которая производит обслуживание поступающих в нее требований. Обслуживание требований в СМО осуществляется обслуживающими приборами. Классическая СМО содержит от одного до бесконечного числа приборов.

Многоканальная СМО с ограниченной длиной очереди. Имеется n каналов, на которые поступает поток заявок с интенсивностью λ. Поток обслуживании каждого канала имеет интенсивность μ. Длина очереди – m. Найти предельные вероятности состояний системы и показатели ее эффективности.

Показатели эффективности СМО

Ключевые показатели:

- Абсолютная пропускная способность системы (A) – среднее число заявок, обслуживаемых в единицу времени;

- Относительная пропускная способность (Q) – средняя доля поступивших заявок,обслуживаемых системой;

- Вероятность отказа (Pотк) – вероятность того, что заявка покинет СМО не обслуженной.

Другие показатели:

- Среднее количество занятых каналов (kзан);

- Среднее количество заявок в системе (Lсист);
- Среднее время пребывания заявки в системе (Tсист);

- Средняя длина очереди (Lоч);

- Среднее время ожидания заявки в очереди (Tоч).

Для многоканальной СМО с ограниченной длиной очереди эти характеристикирассчитываются следующим образом:

$$P_{\text{отк}} = p_{n+m} = \frac{\rho^{n+m}}{n^m n!} p_0$$

$$Q = 1 - P_{\text{отк}}$$

$$A = \lambda Q$$

$$\bar{k}_{\text{зан}} = \frac{A}{\mu} = \rho Q$$

$$L_{\text{оч}} = \frac{\rho^{n+1}}{nn!} \cdot \frac{1 - \left(\frac{\rho}{n}\right)^m \left(m + 1 - \frac{m}{n}\rho\right)}{\left(1 - \frac{\rho}{n}\right)^2} \; p_0$$

$$T_{\text{оч}} = \frac{L_{\text{оч}}}{\lambda}$$

$$L_{\text{сист}} = L_{\text{оч}} + \bar{k}_{\text{зан}}$$

$$T_{\text{сист}} = \frac{L_{\text{сист}}}{\lambda}$$

1. Создайте .m файл в MATLAB и сгенерируйте в нем матрицу переходов P в соответствии с вариантом в таблице 2.1.

Вариант	p_{11}	p_{12}	p_{13}	p_{14}	p_{21}	p_{22}	p_{23}	p_{24}	p_{31}	p_{32}	p_{33}	p_{34}	p_{41}	p_{42}	p_{43}	p_{44}
1	0.8	0.1	0.1	0	0.2	0.7	0.1	0	0.1	0.2	0.5	0.2	0.2	0.2	0.2	0.4

```
% Коэффициенты в матрице переходных вероятностей
P = [0.8, 0.1, 0.1, 0; 0.2, 0.7, 0.1, 0; 0.1, 0.2, 0.5, 0.2; 0.2 0.2 0.2 0.4];
```

2. Создайте цепь Маркова на основе полученной матрицы переходов, используя функцию ‘dtmc()’, задав названия состояний «Healthy», «Unwell», «Sick», «Very sick». Присвойте ее переменной MC.

```
% №2
MC = dtmc(P, "StateNames" , ["Healthy" "Unwell" "Sick" "Very sick"]);
```

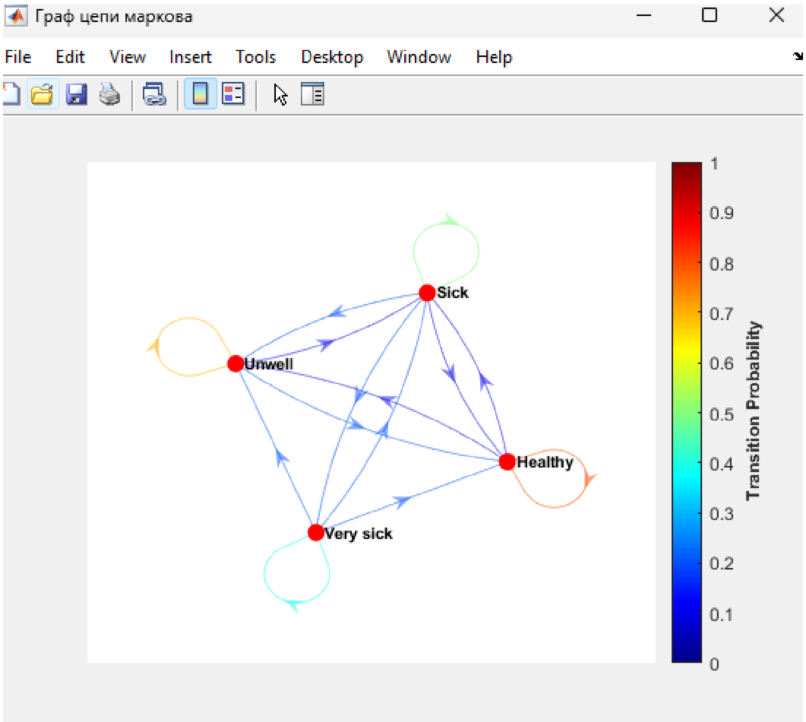
3. Выведите в консоль матрицу переходов полученной цепи, используя функцию ‘MC.P’. Обратите внимание, что полученная матрица является нормированной. Используя функцию ‘sum()’, убедитесь, что все строки матрицы дают в сумме 1. Сохраните полученную матрицу для отчета.

```
% №3
MCP = MC.P
S=sum(MCP,2)
```

```
MCP =

    0.8000    0.1000    0.1000         0
    0.2000    0.7000    0.1000         0
    0.1000    0.2000    0.5000    0.2000
    0.2000    0.2000    0.2000    0.4000


S =

    1.0000
    1.0000
    1.0000
    1.0000
```

4. Постройте граф матрицы при помощи функции ‘graphplot()’. Используя аргументы данной функции, покажите вероятности переходов различным цветом. Сохраните полученное изображение для отчета.

```
% №4
figure('Name','Граф цепи маркова','NumberTitle','off')
graphplot(MC,'ColorEdges',true);
```



5. Используя нормированную матрицу, постройте кумулятивную матрицу переходов, в которой каждое значения в последующем столбце матрицы являются суммой предыдущих.

Назовите эту матрицу P_cum.

```
% №5
P_cum = MCP;
for K = 1:4
    for L=1:3
        P_cum(K,L+1) = P_cum(K,L) + P_cum(K,L+1);
    end
end
```

```
P_cum =

    0.8000    0.9000    1.0000    1.0000
    0.2000    0.9000    1.0000    1.0000
    0.1000    0.3000    0.8000    1.0000
    0.2000    0.4000    0.6000    1.0000
```

6. Промоделируйте поведение цепи Маркова в течение 200 итераций, используя следующее выражение:

$$z_{t+1} = \sum_k (r > P\_cum(z_t, k)) + 1,$$

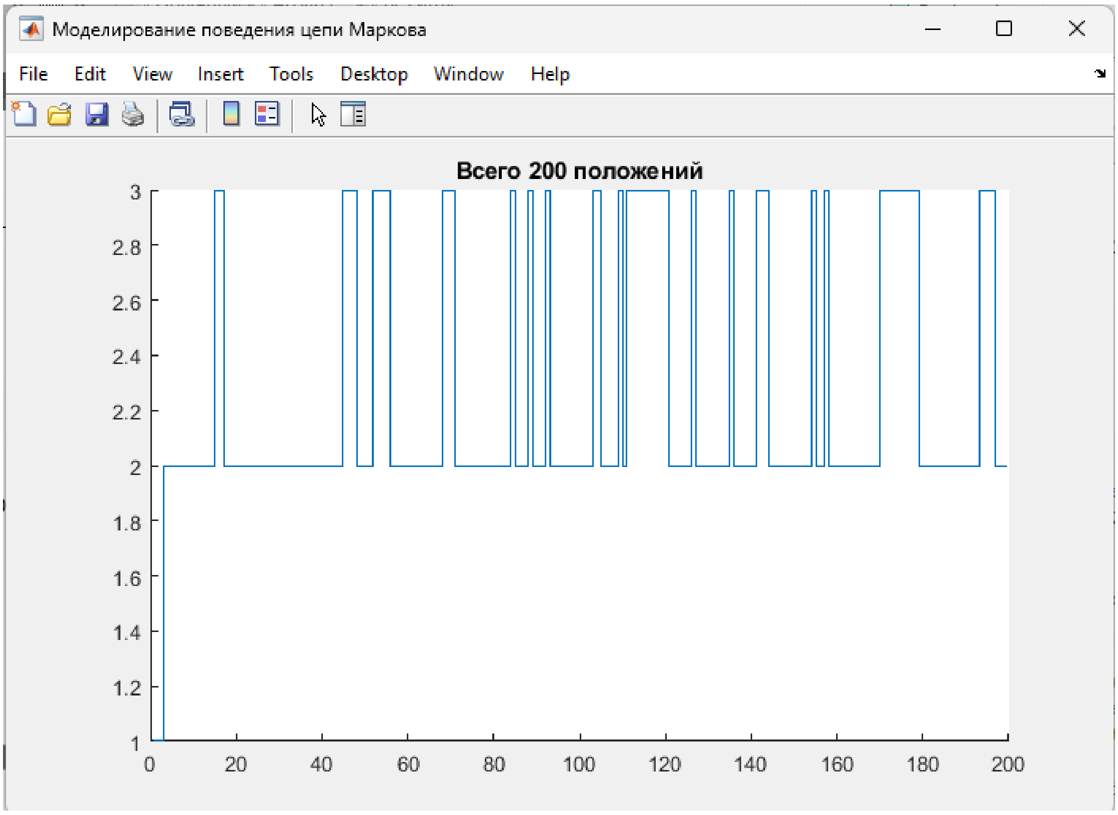
где r – случайное число, распределенное равномерно на интервале [0,1]; P_cum – кумулятивная матрица переходов, zt – состояние цепи в момент времени t.

Реализуем функцию st.m

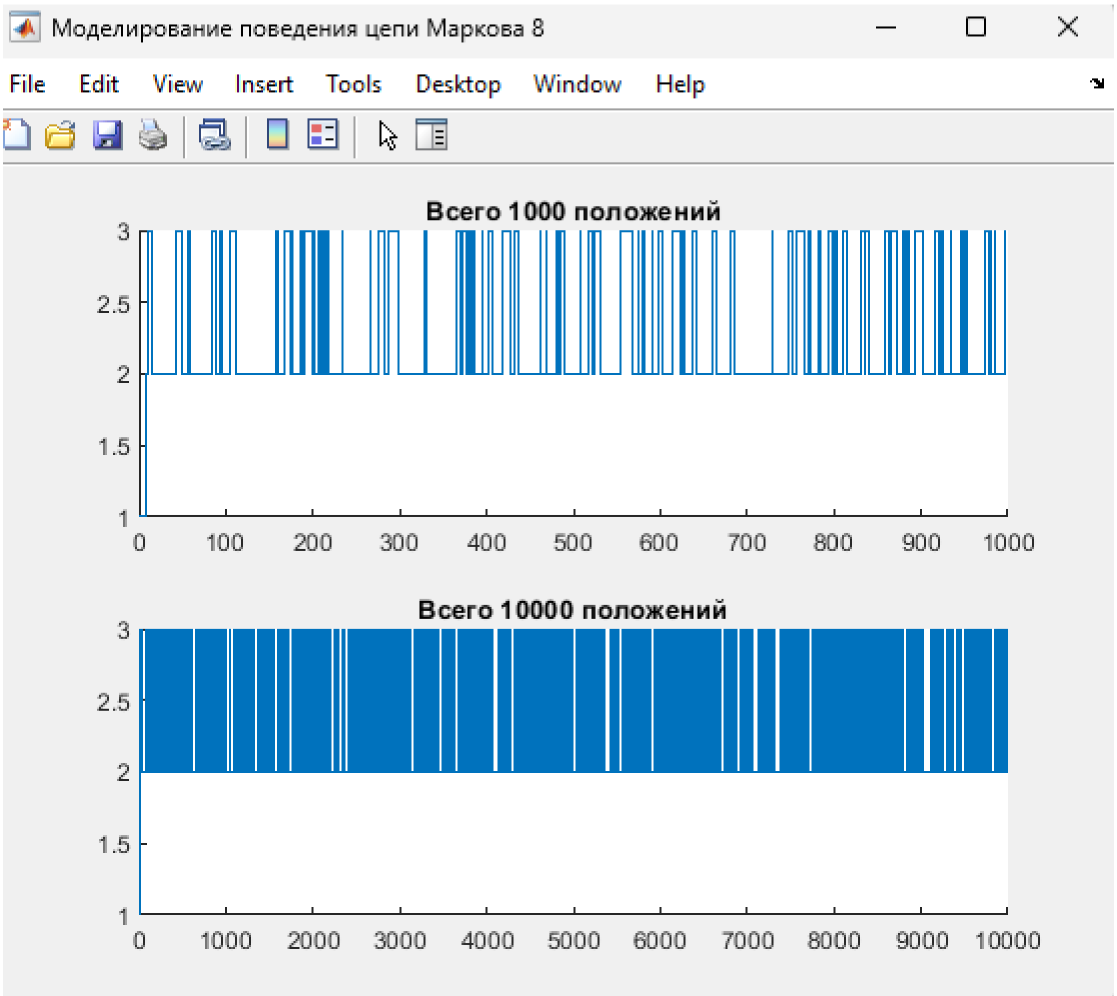
```
function z = st(N,P_cum)
    z=zeros(N,1);
    z(1,1)=1;
    r=rand(N,1);
    n=2;
    k=1; % Начальное состояние цепи
    while n~=N+1
        if (k==1)&&(r(n-1,1)>=P_cum(1,1))
            k=2;
            z(n,1)=k;
            n=n+1;
        elseif (k==1)&&(r(n-1,1)<P_cum(1,1))
            k=1;
            z(n,1)=k;
            n=n+1;
        end
        if (k==2)&&(r(n-1,1)>=P_cum(2,2))
            k=3;
            z(n,1)=k;
            n=n+1;
        elseif (k==2)&&(r(n-1,1)<=P_cum(2,2))
            k=2;
            z(n,1)=k;
            n=n+1;
        end
        if (k==3)&&(r(n-1,1)>=P_cum(3,2))
            k=3;
```

```
            z(n,1)=k;
            n=n+1;
        elseif (k==3)&&(r(n-1,1)<P_cum(3,2))
            k=2;
            z(n,1)=k;
            n=n+1;
        end
        if (k==4)&&(r(n-1,1)>=P_cum(4,3))
            k=4;
            z(n,1)=k;
            n=n+1;
        elseif (k==4)&&(r(n-1,1)<P_cum(4,3))
            k=3;
            z(n,1)=k;
            n=n+1;
        end
        if (n==N)
            z(n,1)=z(n-1,1);
            break
        end
    end
    x = 1:N;
    stairs(x,z);
end
```

7. Используя функцию ‘plot()’, постройте график, показывающий, как в течение 200 наблюдений менялось состояние, в котором находилась цепь. Сохраните этот график для отчета.



8. Повторите пункт 6 для 1000 и 10000 итераций.



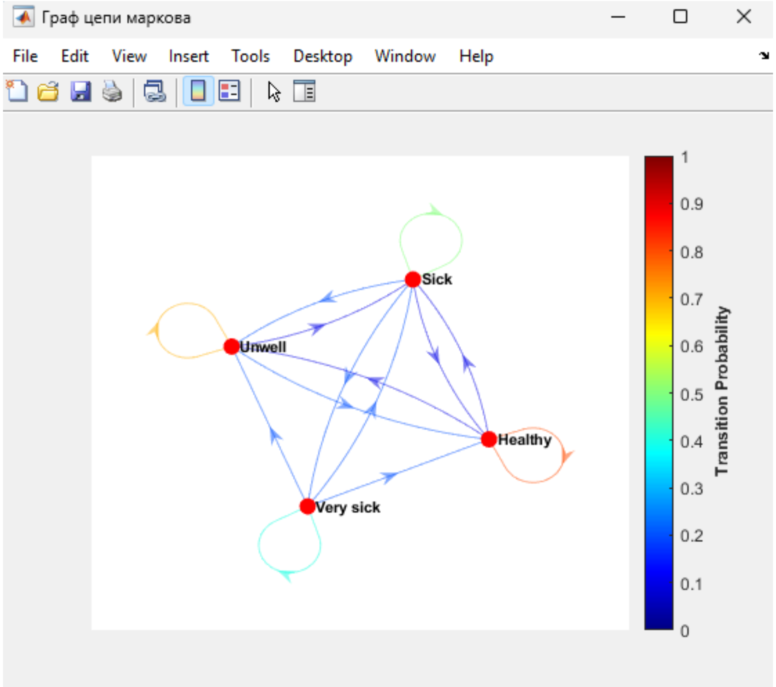
9. Рассчитайте оценку цепи Маркова по полученным наблюдениям для 200, 1000 и 10000 итераций, используя следующее выражение:

$$P\_obs(z_t, z_{t+1}) = \sum_t P\_obs(z_t, z_{t+1}) + 1$$

И затем нормализовав полученные матрицы переходов. Сохраните их для отчета. Повторите пункты 3 и 4 для каждой из полученных матриц.

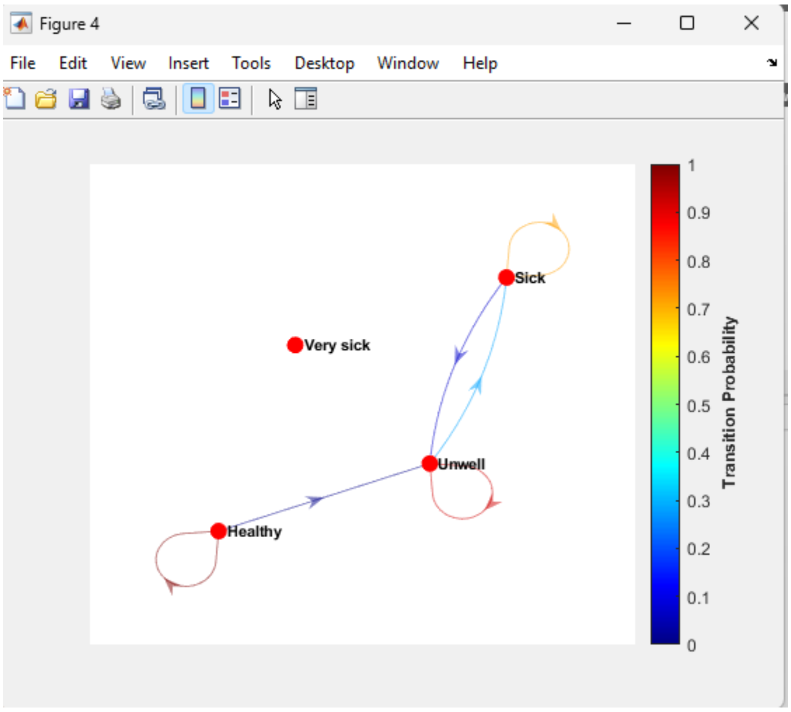
```
P_obs=zeros(4,4,3);
for k=1:3
    for n=2:N(k)
        for i=1:4
            if z(n-1,1)==i
                for j=1:4
                    if z(n,1)==j
                        P_obs(i,j,k)=P_obs(i,j,k)+1;
                    end
                end
            end
        end
    end
end
```

10. Сравните результаты, полученные в пунктах 4 и 9 (одна цепь Маркова, полученная в пункте 3, и еще три цепи, полученные в пункте 9)



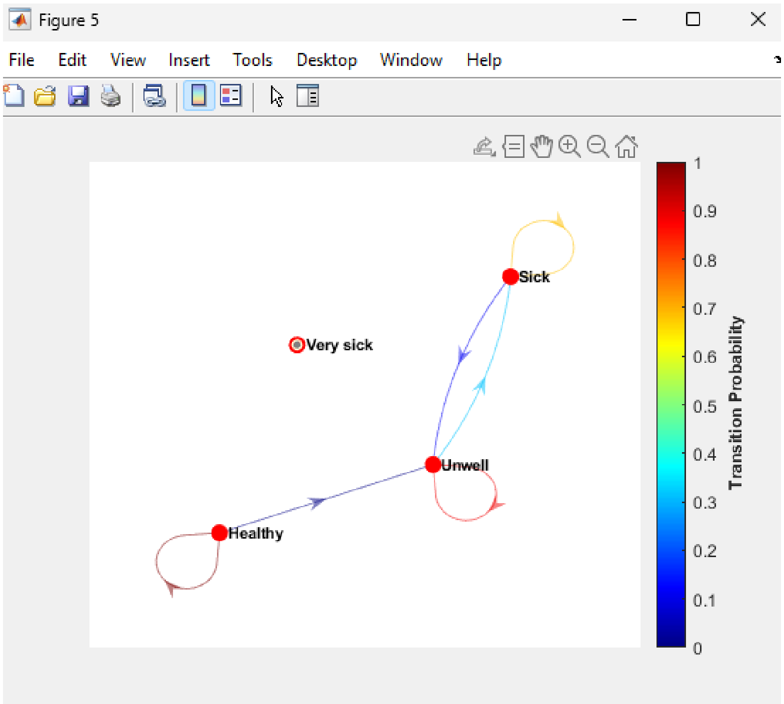

```
MCP =

    0.8000    0.1000    0.1000         0
    0.2000    0.7000    0.1000         0
    0.1000    0.2000    0.5000    0.2000
    0.2000    0.2000    0.2000    0.4000
```

200 наблюдений



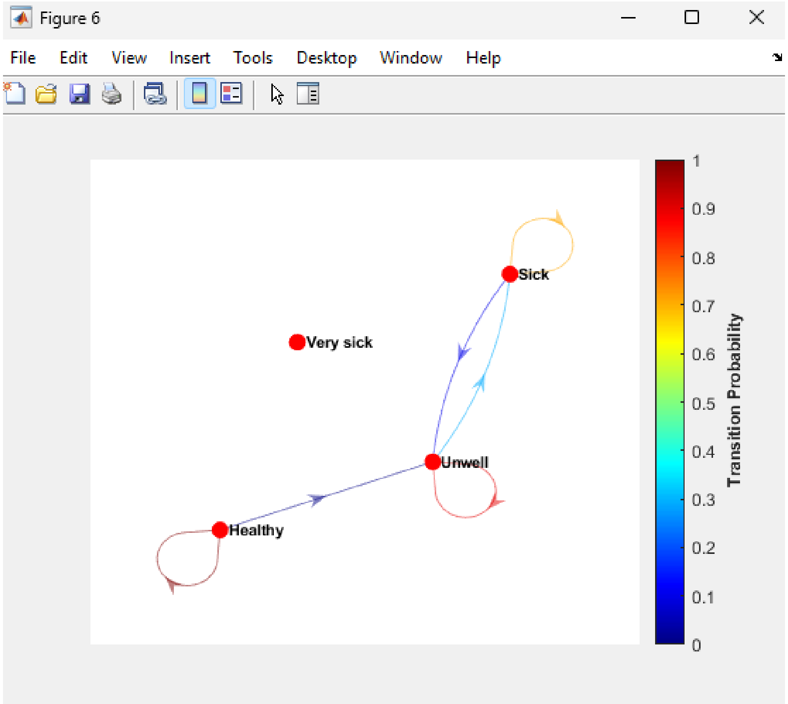
```
MC_Obs1T =

    1.0000    0.0070         0       NaN
         0    0.8671    0.3455       NaN
         0    0.1259    0.6545       NaN
         0         0         0       NaN
```

1000 наблюдений




```
MC_Obs2T =

    1.0000    0.0014         0       NaN
         0    0.8887    0.3142       NaN
         0    0.1099    0.6858       NaN
         0         0         0       NaN
```

10000 наблюдений




```
MC_Obs3T =

    1.0000    0.0001         0       NaN
         0    0.8998    0.2917       NaN
         0    0.1001    0.7083       NaN
         0         0         0       NaN
```

Сравнивая результаты этих цепей, можно отметить, что коэффициенты переходов для 2-го и 3-го состояний рассчитываются корректно и приближаются к исходным значениям по мере увеличения объёма выборки. В то же время, для 1-го и 4-го состояний наблюдаются отклонения, так как переходы в эти состояния возможны только из них самих. В результате, даже при увеличении выборки вероятность попадания в 4-е состояние остаётся крайне низкой, а система будет находиться в 1-м состоянии лишь несколько раз, с большой вероятностью всего один раз, поскольку это стартовое состояние.

11. Системы массового обслуживания. Рассмотрите СМО в соответствии с вашим вариантом. Напишите код для расчета всех показателей эффективности вашей СМО. Сведите все показатели в таблицу и добавьте ее к отчету вместе с характеристиками системы. Является ли данная система эффективной? Интерпретируйте результаты.

Вариант	λ	μ	n	m
1	5	1	1	5

Формулы, использованные для рассчётов

$$P_{\text{отк}} = p_{n+m} = \frac{\rho^{n+m}}{n^m n!} p_0$$

$$Q = 1 - P_{\text{отк}}$$

$$A = \lambda Q$$

$$\bar{k}_{\text{зан}} = \frac{A}{\mu} = \rho Q$$

$$L_{\text{оч}} = \frac{\rho^{n+1}}{n n!} \cdot \frac{1 - \left(\frac{\rho}{n}\right)^m \left(m + 1 - \frac{m}{n}\rho\right)}{\left(1 - \frac{\rho}{n}\right)^2} \; p_0$$

$$T_{\text{оч}} = \frac{L_{\text{оч}}}{\lambda}$$

$$L_{\text{сист}} = L_{\text{оч}} + \bar{k}_{\text{зан}}$$

[redacted]

$$T_{\text{сист}} = \frac{L_{\text{сист}}}{\lambda}$$

```
lam=5;
mu=1;
n=1;
m=5;

p=lam/mu;
Psum=0;
for i=0:m
    Psum=Psum+(p^i)/(factorial(i));
end
p0=Psum^(-1);
Potk=(p^(n+m))*p0/((n^m)*factorial(n));
Q=1-Potk;
A=lam*Q;

kzan=A/mu;
Loch=(p^(n+1))*(1-((p/n)^m)*(m+1-m*p/n))*p0/(n*factorial(n)*(1-p/n)^2);
Toch=Loch/lam;
Lsist=Loch+kzan;
Tsist=Lsist/lam;
```

Potk	170.9207
Q	-169.9207
A	-849.6035
Kzan	-849.6035
Loch	1.0149e+03
Toch	202.9717
Lsist	165.2552
Tsist	33.0510

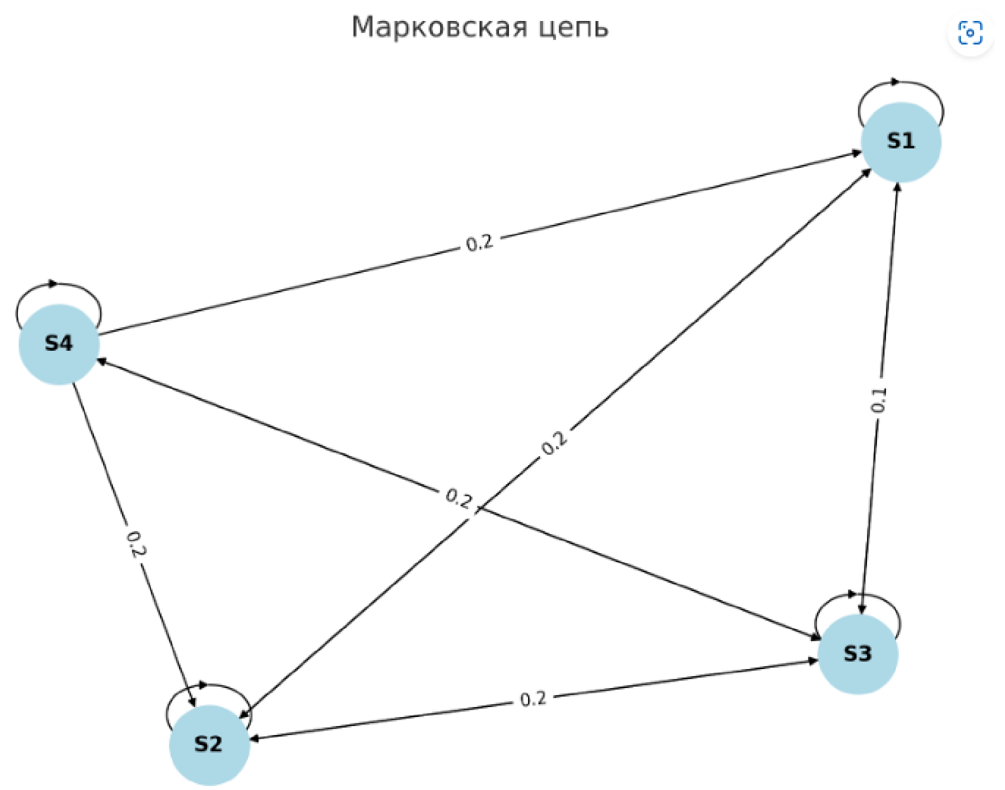
Результаты анализа системы массового обслуживания (СМО) показывают наличие серьёзных проблем. Некоторые ключевые параметры имеют отрицательные значения, что недопустимо для физически интерпретируемых величин. Вероятность отказа составляет 170.92, что превышает 100%. Это говорит о том, что модель имеет ошибки в расчётах или исходные данные не соответствуют реальности. Параметр Q, представляющий вероятность успешного обслуживания, оказался отрицательным, что также невозможно, так как вероятность должна находиться в диапазоне от 0 до 1. Это указывает на критическую ошибку в расчётах.Абсолютная пропускная способность равна -849.60, что означает, что система "теряет" больше заявок, чем может обслужить. Значение среднего количества занятых каналов также отрицательное, что физически невозможно, так как количество занятых каналов должно быть положительным.Среднее число заявок в очереди составляет 1014.9, что говорит о большой очереди и возможной перегрузке системы. Среднее время пребывания в очереди равно 202.97, что подтверждает идею о перегрузке. Среднее количество заявок в системе составляет 165.26, а среднее время пребывания заявки в системе равно 33.05. Это указывает на медленное обслуживание и перегруженность.В целом, результаты анализа указывают на значительные проблемы в работе СМО. Некорректные отрицательные значения вероятностей и пропускной способности указывают на ошибки в расчетах. Система испытывает сильную перегрузку, что делает её неэффективной. Необходимо пересмотреть параметры модели и исходные данные, так как такая система не сможет функционировать должным образом в реальных условиях.

Дополнительные задания:

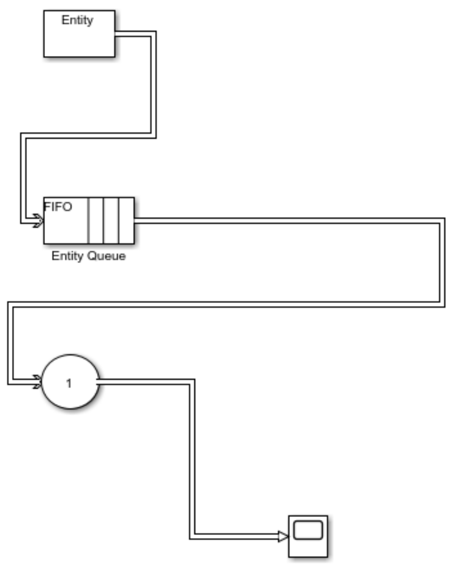
1. Создайте цепь Маркова при помощи языков программирования R, Python.

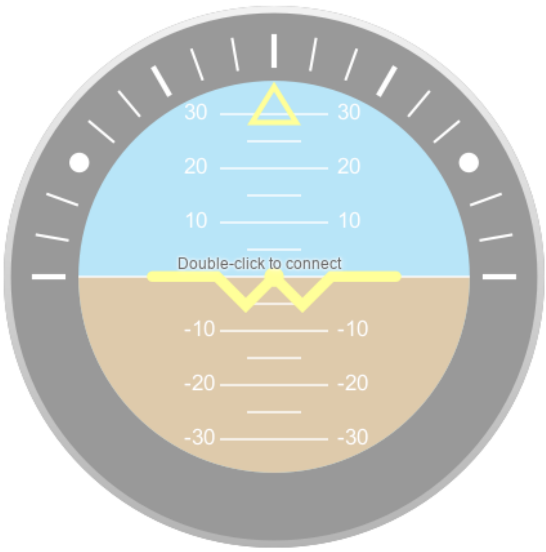
```
import numpy as np
import networkx as nx
import matplotlib.pyplot as plt


# Матрица переходов цепи Маркова
P = np.array([[0.8, 0.1, 0.1, 0],
           [0.2, 0.7, 0.1, 0],
           [0.1, 0.2, 0.5, 0.2],
           [0.2, 0.2, 0.2, 0.4]])
```

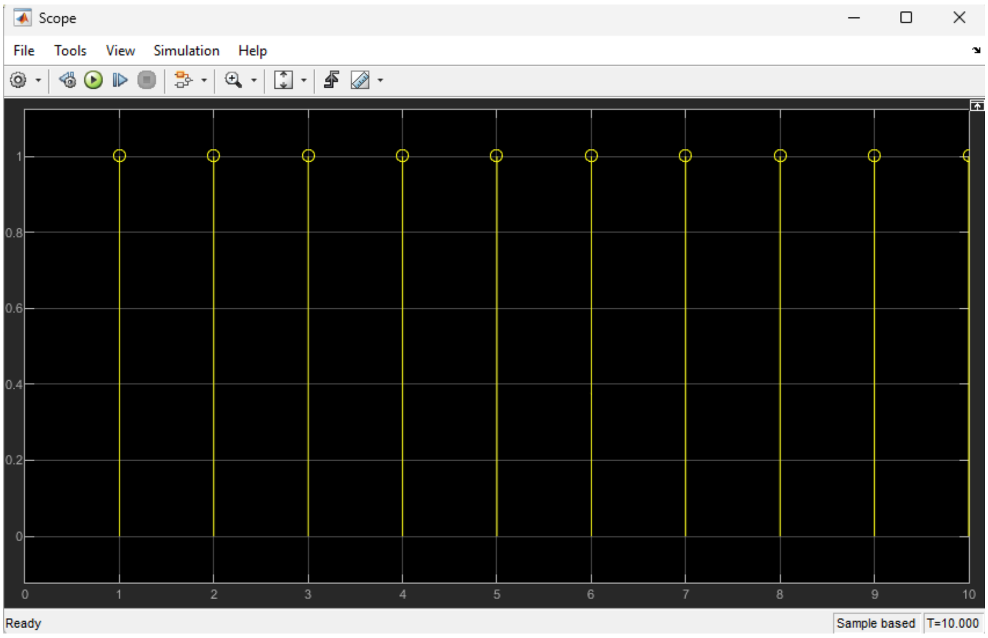
```
# Список состояний
states = ['S1', 'S2', 'S3', 'S4']

# Создание направленного графа (DiGraph) с помощью NetworkX
G = nx.DiGraph()

# Добавляем рёбра в граф с вероятностями переходов
for i, state_from in enumerate(states):
    for j, state_to in enumerate(states):
        if P[i, j] > 0:
            G.add_edge(state_from, state_to, weight=P[i, j])

# Позиции для узлов
pos = nx.spring_layout(G)

# Визуализация графа
plt.figure(figsize=(8, 6))
nx.draw(G, pos, with_labels=True, node_size=2000, node_color='lightblue', font_size=12,
  font_weight='bold')
edge_labels = {(state_from, state_to): f'{P[i, j]:.1f}'
               for i, state_from in enumerate(states)
               for j, state_to in enumerate(states) if P[i, j] > 0}
nx.draw_networkx_edge_labels(G, pos, edge_labels=edge_labels, font_size=10)

# Добавляем заголовок
plt.title("Марковская цепь")

# Показываем граф
plt.show()
```



2. Постройте систему Массового обслуживания при помощи элементов MATLAB Simulink.







Вывод

1) В процессе лабораторной работы были изучены методы создания и анализа цепей Маркова, а также систем массового обслуживания (СМО). Были получены матрицы переходов, построены графы и смоделированы изменения состояний системы при различном количестве итераций. Также была проведена оценка цепи Маркова. Для систем массового обслуживания были рассчитаны ключевые характеристики, а также сделан вывод об их эффективности.

2) Для систем массового обслуживания были определены ключевые и дополнительные показатели эффективности. Некоторые из них, такие как вероятность отказа, равная нулю, относительная пропускная способность, равная единице, и абсолютная пропускная способность, равная интенсивности потока, свидетельствуют о высокой эффективности смоделированной СМО. Время ожидания в очереди и длина очереди также оказались минимальными, что указывает на то, что заявки практически сразу поступают в канал.