Министерство образования и науки Российской Федерации

федеральное государственное бюджетное образовательное

учреждение высшего образования

«Алтайский государственный технический университет им. И.И. Ползунова»

Факультет информационных технологий

Кафедра информатики, вычислительной техники и информационной безопасности

(наименование кафедры)

Отчет защищен с оценкой\_\_\_\_\_\_\_\_\_\_\_\_

Преподаватель Перепелкин Е.А.

(подпись) (и.о., фамилия)

«\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_2018 г.

(дата)

Отчет

по лабораторной работе №1

«Моделирование конечных цепей Маркова»

по дисциплине Моделирование информационных процессов

(наименование дисциплины)

ЛР 09.04.01.12.000 О

(обозначение документа)

Студенты группы 8ИВТ-71 Уваров К.А.

(фамилия и.о.)

Преподаватель  Перепелкин Е.А**.**

(должность, ученое звание) (фамилия и.о.)

Барнаул 2018

**1 Задание**.

Написать программу моделирования дискретной конечной цепи Маркова. По результатам моделирования оценить вероятности состояний системы при заданном числе шагов. Сравнить с теоретическими значениями. Рассчитать предельные вероятности.

**2 Теория**

Цепью Маркова называют такую последовательность случайных событий, в которой вероятность каждого события зависит только от состояния, в котором процесс находится в текущий момент и не зависит от более ранних состояний. Конечная дискретная цепь определяется:

1. **множеством состояний** *S* = {*s*1, …, *sn*}, событием является переход из одного состояния в другое в результате случайного испытания
2. **вектором начальных вероятностей** (начальным распределением) *p*(0) ={*p*(0)(1),…, *p*(0)(*n*)}, определяющим вероятности *p*(0)(*i*) того, что в начальный момент времени *t* = 0 процесс находился в состоянии *si*
3. **матрицей переходных вероятностей** *P* ={*pij*}, характеризующей вероятность перехода процесса с текущим состоянием *si* в следующее состояние *sj*, при этом сумма вероятностей переходов из одного состояния равна 1

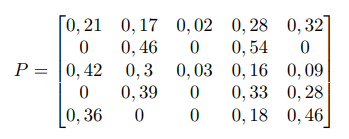
С помощью вектора начальных вероятностей и матрицы переходов можно вычислить стохастический вектор *p*(*n*) — вектор, составленный из вероятностей *p*(*n*)(*i*) того, что процесс окажется в состоянии *i* в момент времени *n*. Получить *p*(*n*) можно с помощью формулы:

*p*(*n*) = *p*(0)×*P* *n*

Векторы *p*(*n*) при росте *n* в некоторых случаях стабилизируются — сходятся к некоторому вероятностному вектору *ρ*, который можно назвать **стационарным распределением** цепи. Стационарность проявляется в том, что взяв *p*(0) = ρ, мы получим *p*(*n*) = ρ для любого *n*.

3 **Ход работы**

Вариант задания:



Марковская цепь изображена в виде графа переходов, вершины которого соответствуют состояниям цепи, а дуги — переходам между ними.

На написанной программе проведены испытания, результаты которых приведены ниже:

Первое испытание:

Условие: Начальная точка 1, количество переходов 10, количество экспериментов 1000:

Гистограмма результата:

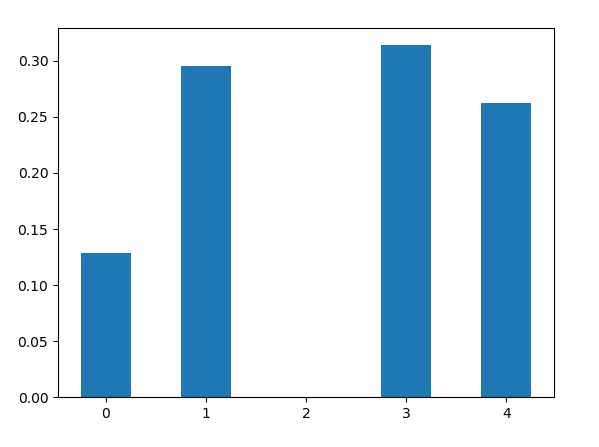


Рисунок 1 ‑ Гистограмма первого испытания

Второе испытание:

Условие: Начальная точка 2, количество переходов 20, количество экспериментов 1000:

Гистограмма результата:

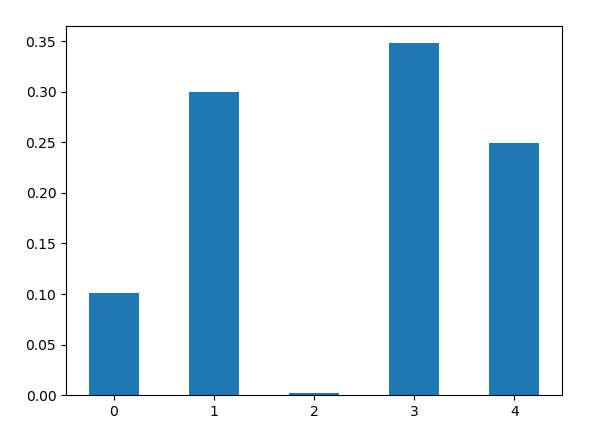


Рисунок 2 ‑ Гистограмма второго испытания

Третье испытание:

Условие: Начальная точка 3, количество переходов 15, количество экспериментов 1000:

Гистограмма результата:

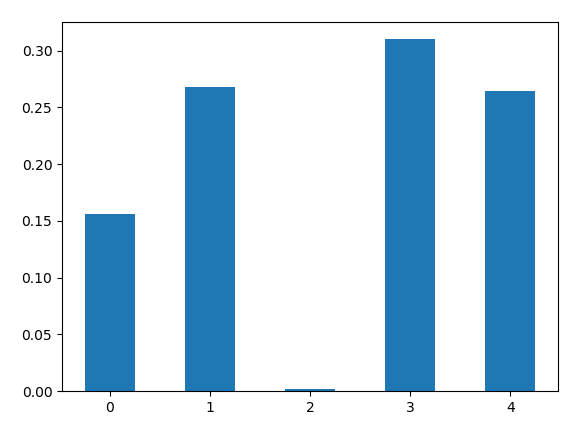


Рисунок 3 ‑ Гистограмма третьего испытания

Четвертое испытание:

Условие: Начальная точка 4, количество переходов 20, количество экспериментов 1000:

Гистограмма результата:

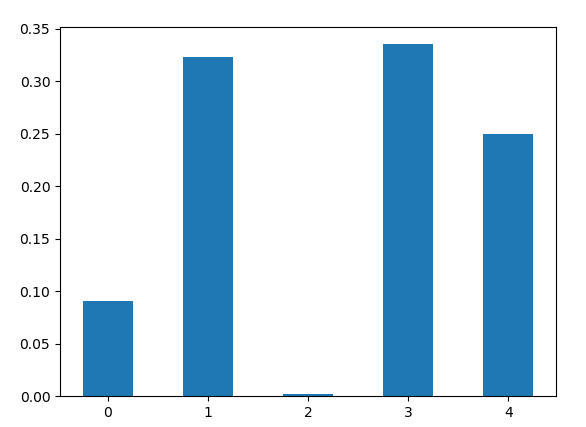


Рисунок 4 - Гистограмма четвертого испытания

Пятое испытание:

Условие: Начальная точка 5, количество переходов 20, количество экспериментов 1000:

Гистограмма результата:

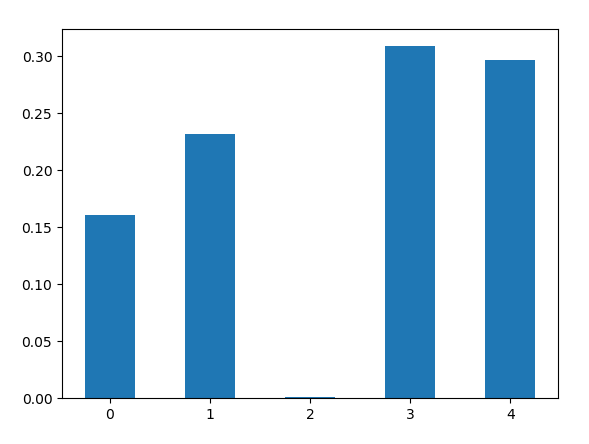
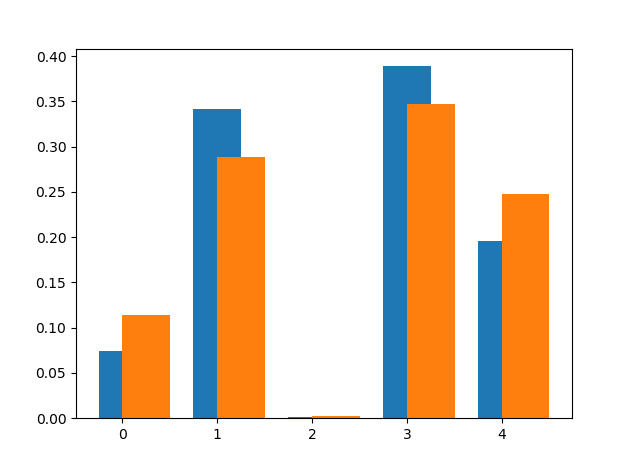


Рисунок 5 ‑ Гистограмма пятого испытания

Алгебраическое вычисление:

Условие: Начальная точка 3, количество переходов 9, количество экспериментов 100000:



**Вывод**: как видно из проведенных тестов, вероятность попадания в одно из состояний, при большом числе экспериментов, не зависит от начального состояния. Однако при малом числе переходов эти вероятности изменяются.

**Приложение А. Код программы:**

import random

import matplotlib.pyplot as plt

from numpy import array, linalg, matmul

def get\_next\_point(p\_list):

ver = random.randint(1, 100)

r\_ver = 0

for point in p\_list:

if point == 0:

continue

l\_ver = r\_ver

r\_ver += point \* 100

if l\_ver < ver <= r\_ver:

return p\_list.index(point)

raise ValueError

def main():

f\_count = 100

mpv\_size = 5

mpv = [0.21, 0.17, 0.02, 0.28, 0.32,

0, 0.46, 0, 0.54, 0,

0.42, 0.3, 0.03, 0.16, 0.09,

0, 0.39, 0, 0.33, 0.28,

0.36, 0, 0, 0.18, 0.46]

p\_list = [0] \* mpv\_size

print("Enter step count: ")

m\_step\_count = int(input())

print("First point: ")

m\_point = int(input()) - 1

for \_ in [x for x in range(f\_count)]:

point = m\_point

step\_count = m\_step\_count

print\_str = "start(p{}) -> ".format(point + 1)

while step\_count > 0:

mpv\_current\_point\_index = mpv\_size \* point

point = get\_next\_point(mpv[mpv\_current\_point\_index: mpv\_current\_point\_index + mpv\_size])

p\_list[point] += 1

print\_str += "p{} -> ".format(point + 1)

step\_count -= 1

print("{}end".format(print\_str))

mpv\_t = array(mpv)

# p2 = array([0, 0, 1, 0, 0])

# p3 = linalg.matrix\_power(mpv\_t, 3)

# p4 = matmul(p2, p3)

# print(p4)

print(p\_list)

p\_list = list(map(lambda p: p / (f\_count \* m\_step\_count), p\_list))

print(p\_list)

plt.bar([x for x in range(5)], p\_list, width=0.5)

plt.show()

if \_\_name\_\_ == '\_\_main\_\_':

main()

**Моделирование процесса цифровой обработки изображений облачного покрова**