МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования

**«Национальный исследовательский**

**Нижегородский государственный университет им. Н.И. Лобачевского»**

**(ННГУ)**

Институт информационных технологий, математики и механики

Кафедра информатики и автоматизации научных исследований

**Курсовая работа**

Использование цепей Маркова в моделировании

социально-экономических процессов

**Выполнил:** студент группы 381807-2

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Ионов М.В

*подпись*

**Научный руководитель:**

Заведующий кафедрой ИАНИ,  
 доктор технических наук

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Прилуцкий М.Х

*подпись*

Нижний Новгород  
2021

**Содержание:**

Оглавление

[Введение: 3](#_Toc72779650)

[Актуальность 3](#_Toc72779651)

[Цель: Экспериментальным путем исследовать поведение цепи Маркова при разных входных значениях и научиться выбирать лучшее решение в цепях Маркова с управлением. 4](#_Toc72779652)

[Задачи: 4](#_Toc72779653)

[Основная часть: 5](#_Toc72779654)

[Математическая модель 5](#_Toc72779655)

[Экспериментальное исследование в рамках поставленной задачи 7](#_Toc72779656)

[Эксперимент 1 8](#_Toc72779657)

[Задача 8](#_Toc72779658)

[Программа. 9](#_Toc72779659)

[Вычислительный эксперимент – решение примеров с разными характеристиками. 11](#_Toc72779660)

[Эксперимент 2. 18](#_Toc72779661)

[Задача 18](#_Toc72779662)

[Программа. 19](#_Toc72779663)

[Вычислительный эксперимент – решение примеров с разными характеристиками. 20](#_Toc72779664)

[Эксперимент 3. 32](#_Toc72779665)

[Задача 33](#_Toc72779666)

[Программа 33](#_Toc72779667)

[Вычислительный эксперимент – решение примеров с разными характеристиками. 36](#_Toc72779668)

[Заключение: 45](#_Toc72779669)

[Источники 46](#_Toc72779670)

[Приложение 47](#_Toc72779671)

# Введение:

Цепью Маркова называют такую последовательность случайных событий, в которой вероятность каждого события зависит только от предыдущего, но не зависит от более ранних событий.

Цепь Маркова это один из самых простых случаев последовательности случайных событий. Но, несмотря н

а свою простоту, она часто может быть полезной даже при описании довольно сложных явлений.

Цепи Маркова служат хорошим введением в теорию случайных процессов, т.е. теорию простых последовательностей семейств случайных величин, обычно зависящих от параметра, который в большинстве приложений играет роль времени. Она предназначена, главным образом, для полного описания как долговременного, так и локального поведения процесса.

Актуальность**:**

В последнее время можно услышать о применении цепей Маркова в самых разных областях: современных веб-технологиях, при анализе литературных текстов или даже при разработке тактики игры футбольной команды . Можем увидеть , что разнообразие форм применения Цепей Маркова обширно, поэтому тяжело усомниться в актуальности изучения данной темы в наше время.

# Цель: Экспериментальным путем исследовать поведение цепи Маркова при разных входных значениях и научиться выбирать лучшее решение в цепях Маркова с управлением.

# Задачи:

1. Понаблюдать, как система будет себя вести в зависимости от начального

состояния, с которого мы будем проводить наблюдения.

2) Понаблюдать, как будет себя вести система, при введении в него учета «доходов».

3) Смоделировать работу предприятия, используя Марковские цепи.

4) Найти лучшую стратегию для этого предприятия.

# Основная часть:

## Математическая модель

Пусть { , , ..., } - множество возможных состояний некоторой физической системы. В любой момент времени система может находиться только в одном состоянии. С течением времени система переходит последовательно из одного состояния в другое. Каждый такой переход называется шагом процесса.

Для описания эволюции этой системы введем последовательность дискретных случайных величин , ,..., ,... Индекс n играет роль времени. Если в момент времени n система находилась в состоянии , то мы будем считать, что = j. Таким образом, случайные величины являются номерами состояний системы.

Последовательность , ,..., ,... образует цепь Маркова, если для любого n и любых , , ..., ,...

P(=j / = , ..., =i)=P(=j / =i).

Для цепей Маркова вероятность в момент времени n попасть в состояние , если известна вся предыдущая история изучаемого процесса, зависит только от того, в каком состоянии находился процесс в момент n-1. То есть при фиксированном "настоящем" "будущее" не зависит от "прошлого". Свойство независимости "будущего" от "прошлого" при фиксированном "настоящем" называется *марковским свойством*.

Вероятности ( =j / =i), i, j=1,2,..., r называются вероятностями перехода из состояния в состояние за один шаг.

Цепь Маркова называется однородной, если вероятности перехода не зависят от n, т.е. если вероятности перехода не зависят от номера шага, а зависят только от того, из какого состояния и в какое осуществляется переход. Для однородных цепей Маркова вместо будем писать .

Вероятности перехода удобно располагать в виде квадратной матрицы



Матрица P называется матрицей вероятностей перехода однородной цепи Маркова за один шаг. Она обладает следующими свойствами:

а) ;

б) для всех i: 

Квадратные матрицы, для которых выполняются условия а) и б), называются *стохастическими*.

Вектор , где =P(), i=1,2...,r называется вектором начальных вероятностей.

Свойства однородных цепей Маркова полностью определяются вектором начальных вероятностей и матрицей вероятностей перехода.

Приведем пример: Завод выпускает телевизоры определенного типа. В зависимости от того, находит ли данный тип телевизора спрос у населения, завод в конце каждого года может находиться в одном из состояний: состояние 1 – спрос есть, состояние 2 – спроса нет. Пусть вероятность сохранить состояние 1 в в следующем году с учетом возможного изменения спроса равна , а вероятность изменить состояние 2 с учетом мероприятий по улучшению выпускаемой модели равна . Тогда процесс производства на данном заводе можно описать цепью Маркова с матрицей переходов:



Можно показать, что матрица вероятностей перехода цепи Маркова за n шагов равняется n-ой степени матрицы P вероятностей перехода за один шаг. Для однородной цепи Маркова при любом m выполняется равенство

P()=P().

Но последняя вероятность есть вероятность перехода из состояния  в состояние за n шагов.

**На основании изученной теории можем перейти к проведению экспериментов в рамках поставленной задачи.**

# Экспериментальное исследование в рамках поставленной задачи

Для описания функционирования Маковской цепи необходимо задать:

1. Состояния цепи
2. Матрицу вероятностных переходов
3. Количество тактов функционирования Марковской цепи

## Эксперимент 1

Понаблюдать, как система будет себя вести в зависимости от начального

состояния, с которого мы будем проводить наблюдения.

**А)Матрица переходных вероятностей марковской цепи**



**B) Состояния цепи**

1)Начальное состояние системы задается вектором начальных состояний

**(0)=(1,0 )**

Т.е. система с вероятностью 1 находится в состоянии s1, и с вероятностью 0 в состоянии s2.

2)Начальное состояние системы ~~s2~~ задается вектором начальных состояний

**(0)=(0,1 )**

Т.е. система с вероятностью 0 находится в состоянии s1, и с вероятностью 1 в состоянии s2.

**C) Количество тактов функционирования марковской цепи**.

Будет подобрано экспериментально в ходе эксперимента.

## Задача

Написать программу, которая при заданных величинах:

n−количество состояний , m − количество тактов ,

P=‖pij‖n×n− матрица переходных вероятностей

находит

πi(m ) - вероятность того, что система через m шагов окажется в состоянии si, при i=.

Если будет найдена некоторая закономерность – использовать ее для анализа

поведения системы.

Существует много приложений для такой задачи – задача управления парком

машин, задача выпуска новых изделий, в зависимости от условий продаж и др.

## Программа.

**Блок 1. Генерация задач.**

Данный блок предназначен для генерирования задач с помощью ручного.

Генерирование задач с помощью ручного ввода включает в себя функции :

* Считывания количества состояний – с помощью ручного ввода задаем целое число состояний в цепи.
* Считывания количество тактов цепи – с помощью ручного ввода задаем необходимое целое число тактов марковской цепи.
* Генерации матрицы переходных вероятностей марковской цепи – формируем ,с помощью ручного ввода, двумерный массив ,состоящий из вероятностей перехода для всех состояний.  
  Также здесь присутствует ограничение на то, чтобы сумма всех вероятностей для одного состояния была не больше 1.
* Генерации вектора начального состояния – формируем, с помощью ручного ввода , массив, состоящий из вероятностей нахождения в начальный момент времени в каждом состоянии.
* Вывода введённых данных в виде таблицы.

**Блок 2. Функциональный блок.**

Данный блок предназначен для на основании исходных данных провести необходимые вычисления.

Функциональный блок включает в себя:

* Цикл, отвечающий за исполнение определенного числа тактов – Цикл 1- имеет в своем составе цикл 1 . Они вместе реализуют подсчёт вероятностей попасть в каждое состояние через заданное количество шагов.
* Подсчёт попасть в определенное состояние для каждого состояния на следующем такте – Цикл 2 -рассчитывает вероятность попасть каждое состояние на следующем такте, записывает эти вероятности в массив, находится в Цикле 1.

**Блок 3. Отображение решений.**

Предназначен для вывода решений в виде «Вероятность попасть в состояние N = M , через P шагов», где

N = 1,2 … K , где К- заданное пользователем количество состояний.

M – полученный результат

P – заданное пользователем количество тактов.

**Программа реализована средствами языка С++ и его встроенных библиотек.**

## Вычислительный эксперимент – решение примеров с разными характеристиками.

**Задачи 1.1**

**Количество состояний в системе: 2**

**Рассмотрим начальные состояния (1,0) и (0,1)**

**Система исходно находится в состоянии (1,0)**

|  |  |  |
| --- | --- | --- |
| Шаг | Вероятность попасть в 1 состояние | Вероятность попасть в 2-е состояние |
| 1 | 0,5 | 0,5 |
| 2 | 0,45 | 0,55 |
| 3 | 0,445 | 0,555 |
| 4 | 0,4445 | 0,5555 |
| 5 | 0,44445 | 0,55555 |
| 6 | 0,444445 | 0,555555 |
| 7 | 0,444444 | 0,555556 |
| 8 | 0,444444 | 0,555556 |
| 9 | 0,444444 | 0,555556 |
| 10 | 0,444444 | 0,555556 |
| 11 | 0,444444 | 0,555556 |
| 12 | 0,444444 | 0,555556 |
| 13 | 0,444444 | 0,555556 |
| 14 | 0,444444 | 0,555556 |
| 15 | 0,444444 | 0,555556 |
| 16 | 0,444444 | 0,555556 |
| 17 | 0,444444 | 0,555556 |
| 18 | 0,444444 | 0,555556 |
| 19 | 0,444444 | 0,555556 |
| 20 | 0,444444 | 0,555556 |

**Задачи 1.2**

Система исходно находится в состоянии (0,1)

|  |  |  |
| --- | --- | --- |
| Шаг | Вероятность попасть в 1 состояние | Вероятность попасть в 2-е состояние |
| 1 | 0,4 | 0,6 |
| 2 | 0,44 | 0,56 |
| 3 | 0,444 | 0,556 |
| 4 | 0,4444 | 0,5556 |
| 5 | 0,44444 | 0,55556 |
| 6 | 0,444444 | 0,555556 |
| 7 | 0,444444 | 0,555556 |
| 8 | 0,444444 | 0,555556 |
| 9 | 0,444444 | 0,555556 |
| 10 | 0,444444 | 0,555556 |
| 11 | 0,444444 | 0,555556 |
| 12 | 0,444444 | 0,555556 |
| 13 | 0,444444 | 0,555556 |
| 14 | 0,444444 | 0,555556 |
| 15 | 0,444444 | 0,555556 |
| 16 | 0,444444 | 0,555556 |
| 17 | 0,444444 | 0,555556 |
| 18 | 0,444444 | 0,555556 |
| 19 | 0,444444 | 0,555556 |
| 20 | 0,444444 | 0,555556 |

Начиная с 6 такта вероятности переходов становятся достаточно близки .

Можем наглядно увидеть, что уже при прошествии 20 недель, со дня начала работы , оказаться в состояниях 1 или 2 практически одинаковы, несмотря на отсутствии или наличии спроса на игрушки .

**1 набор характеристик:**

Количество состояний в системе: 3

Три эксперимента :

1. Начальное состояние (1,0,0)
2. Начальное состояние (0,1,0)
3. Начальное состояние (0,0,1)

Показать, что начиная с некоторого такта финальные вероятности будут близки или нет.

**Матрица вероятностей перехода**

|  |  |  |  |
| --- | --- | --- | --- |
|  | В 1-ое состояние | Во 2-ое состояние | В 3-ое состояние |
| Из 1-го состояния | 0.5 | 0.3 | 0.2 |
| Из 2-го состояния | 0.4 | 0.4 | 0.2 |
| Из 3-го состояния | 0.2 | 0.2 | 0.6 |

**Таблица начальных вероятностей.**

|  |  |  |  |
| --- | --- | --- | --- |
| Нахождение в состоянии | 1 | 2 | 3 |
| Вероятность | 0.3 | 0.4 | 0.3 |

**Результаты**

|  |  |  |  |
| --- | --- | --- | --- |
| **Нахождение в данном состоянии через 3 шагов** | 1 | 2 | 3 |
| Вероятность для 1 первого эксперимента | 0.385 | 0.303 | 0.312 |
| Вероятность для 2 первого эксперимента | 0.384 | 0.304 | 0.312 |
| Вероятность для 3 первого эксперимента | 0.342 | 0.282 | 0.376 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Нахождение в данном состоянии через 8 шагов** | 1 | 2 | 3 |
| Вероятность для 1 первого эксперимента | 0.370516 | 0.296369 | 0.333115 |
| Вероятность для 2 первого эксперимента | 0.370516 | 0.296369 | 0.333115 |
| Вероятность для 3 первого эксперимента | 0.370079 | 0.296151 | 0.33377 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Нахождение в данном состоянии через 20 шагов** | 1 | 2 | 3 |
| Вероятность для 1 первого эксперимента | 0.37037 | 0.296296 | 0.333333 |
| Вероятность для 2 первого эксперимента | 0.37037 | 0.296296 | 0.333333 |
| Вероятность для 3 первого эксперимента | 0.37037 | 0.296296 | 0.333333 |

**2 набор характеристик**

Количество состояний в системе: 3

Количество шагов : 2000

**Матрица вероятностей перехода**

|  |  |  |  |
| --- | --- | --- | --- |
|  | В 1-ое состояние | Во 2-ое состояние | В 3-ое состояние |
| Из 1-го состояния | 0.5 | 0.3 | 0.2 |
| Из 2-го состояния | 0.4 | 0.4 | 0.2 |
| Из 3-го состояния | 0.2 | 0.2 | 0.4 |

Таблица начальных вероятностей.

|  |  |  |  |
| --- | --- | --- | --- |
| Нахождение в состоянии | 1 | 2 | 3 |
| Вероятность | 0.4 | 0.2 | 0.4 |

**Результаты**

|  |  |  |  |
| --- | --- | --- | --- |
| Нахождение в данном состоянии через 2000 шагов | 1 | 2 | 3 |
| Вероятность | 0.37037 | 0.296296 | 0.333333 |

**3 набор характеристик**

Количество состояний в системе:4

Три эксперимента :

1. Начальное состояние (1,0,0,0)
2. Начальное состояние (0,1,0,0)
3. Начальное состояние (0,0,1,0)
4. Начальное состояние (0,0,0,1)

**Матрица вероятностей перехода**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | В 1-ое состояние | Во 2-ое состояние | В 3-ое состояние | В 4-ое состояние |
| Из 1-го состояния | 0.5 | 0.3 | 0.1 | 0.1 |
| Из 2-го состояния | 0.3 | 0.2 | 0.3 | 0.2 |
| Из 3-го состояния | 0.2 | 0.2 | 0.5 | 0.1 |
| Из 4-го состояния | 0.2 | 0.2 | 0.2 | 0.4 |

**Результаты**

|  |  |  |  |
| --- | --- | --- | --- |
| **Нахождение в данном состоянии через 20 шагов** | 1 | 2 | 3 |
| Вероятность для 1 первого эксперимента | 0.37037 | 0.296296 | 0.333333 |
| Вероятность для 2 первого эксперимента | 0.37037 | 0.296296 | 0.333333 |
| Вероятность для 3 первого эксперимента | 0.37037 | 0.296296 | 0.333333 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Нахождение в данном состоянии через 20 шагов** | 1 | 2 | 3 |
| Вероятность для 1 первого эксперимента | 0.37037 | 0.296296 | 0.333333 |
| Вероятность для 2 первого эксперимента | 0.37037 | 0.296296 | 0.333333 |
| Вероятность для 3 первого эксперимента | 0.37037 | 0.296296 | 0.333333 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Нахождение в данном состоянии через 3 шагов** | 1 | 2 | 3 | 4 |
| Вероятность для 1 эксперимента | 0.339 | 0.238 | 0.25 | 0.173 |
| Вероятность для 2 эксперимента | 0.316 | 0.231 | 0.276 | 0.177 |
| Вероятность для 3 эксперимента | 0.306 | 0.228 | 0.299 | 0.167 |
| Вероятность для 4 эксперимента | 0.306 | 0.228 | 0.272 | 0.194 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Нахождение в данном состоянии через 8 шагов** | 1 | 2 | 3 | 4 |
| Вероятность для 1 эксперимента | 0.31892 | 0.231908 | 0.273166 | 0.176006 |
| Вероятность для 2 эксперимента | 0.318829 | 0.231881 | 0.273308 | 0.175982 |
| Вероятность для 3 эксперимента | 0.31879 | 0.231869 | 0.273399 | 0.175942 |
| Вероятность для 4 эксперимента | 0.31879 | 0.231869 | 0.273333 | 0.176008 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Нахождение в данном состоянии через 20 шагов** | 1 | 2 | 3 | 4 |
| Вероятность для 1 эксперимента | 0.318841 | 0.231884 | 0.273292 | 0.175983 |
| Вероятность для 2 эксперимента | 0.318841 | 0.231884 | 0.273292 | 0.175983 |
| Вероятность для 3 эксперимента | 0.318841 | 0.231884 | 0.273292 | 0.175983 |
| Вероятность для 4 эксперимента | 0.318841 | 0.231884 | 0.273292 | 0.175983 |

**Вывод:** Исходя из трёх экспериментов можем увидеть, что при одной и той же матрице вероятностей , вероятность попадания в определенное состояние после некоторого шага примерно одинакова , независимо от начального состояния.

## Эксперимент 2.

Для описания однородной марковской цепи должны быть определены

состояния цепи, такты функционирования и матрицы переходных

вероятностей цепи.

Пусть при переходе системы из одного состояния в

другое приобретается «доход».

Доход является величиной, определяемой

вероятностными связями марковской цепи. Тогда марковская цепь при

функционировании порождает последовательность соответствующих переходам марковской цепи.

Пусть  – полный ожидаемый доход, который получит мастер, если он начинает работать в состоянии i и ему остается работать еще *n*-шагов.

Тогда , , *n=1,2…* или

,

где  – средний одношаговый доход для состояния *i*, 

## Задача

Написать программу, которая при заданных величинах:

n−количество состояний , m − количество тактов ,

P=− матрица переходных вероятностей

R= ­− матрица доходов

находит

(n) – полный ожидаемый доход, который будет получен в результате работы алгоритма, если он начинает работать в состоянии i и ему остается работать еще n-шагов.

## Программа.

**Блок 1. Генерация задач.**

Данный блок предназначен для генерирования задач с помощью ручного.

Генерирование задач с помощью ручного ввода включает в себя функции :

* Считывания количества состояний – с помощью ручного ввода задаем целое число состояний в цепи.
* Считывания количество тактов цепи – с помощью ручного ввода задаем необходимое целое число тактов марковской цепи.
* Генерации матрицы переходных вероятностей марковской цепи – формируем ,с помощью ручного ввода, двумерный массив ,состоящий из вероятностей перехода для всех состояний.  
  Также здесь присутствует ограничение на то, чтобы сумма всех вероятностей для одного состояния была не больше 1.
* Вывода введённых данных в виде таблицы.

**Блок 2. Функциональный блок.**

Данный блок предназначен для на основании исходных данных провести необходимые вычисления.

Функциональный блок включает в себя:

* Цикл, отвечающий за исполнение определенного числа тактов – Цикл 1- имеет в своем составе Цикл 2 и Цикл 3 . Они вместе реализуют подсчёт вероятностей попасть в каждое состояние через заданное количество шагов.
* Подсчёт попасть в определенное состояние для каждого состояния на следующем такте – Цикл 2 -рассчитывает вероятность попасть каждое состояние на следующем такте, записывает эти вероятности в массив, находится в Цикле 1.
* Подсчёт полного ожидаемого дохода – Цикл 3 -рассчитывает полный ожидаемый доход на текущем такте системы для конкретного состояния, записывает результаты в массив, находится в Цикле 1.
* Цикл 4– вычисление среднего одношагового дохода для каждого состояния

**Блок 3. Отображение решений.**

Предназначен для вывода решений в виде «Полученный доход через P тактов N состояния составляет : M», где

N = 1,2 … K , где К- заданное пользователем количество состояний.

M – полученный результат

P – заданное пользователем количество тактов.

**Программа реализована средствами языка С++ и его встроенных библиотек.**

## Вычислительный эксперимент – решение примеров с разными характеристиками.

**1 набор характеристик:**

**Количество состояний в системе: 2**

**Матрица вероятностей перехода**

|  |  |  |
| --- | --- | --- |
|  | В 1-ое состояние | Во 2-ое состояние |
| Из 1-го состояния | 0.5 | 0.5 |
| Из 2-го состояния | 0.4 | 0.6 |

**Матрица доходов**

|  |  |  |
| --- | --- | --- |
|  | В 1-ое состояние | Во 2-ое состояние |
| Из 1-го состояния | 9 | 3 |
| Из 2-го состояния | 3 | -7 |

**Полученные результаты**

|  |  |  |
| --- | --- | --- |
| n | (n) | (n) |
| 0 | 0 | 0 |
| 1 | 6 | -3 |
| 2 | 7.5 | -2.4 |
| 3 | 8.55 | -1.44 |
| 4 | 9.555 | -0.444 |
| 5 | 10.5555 | 0.5556 |
| 6 | 11.5556 | 1.55556 |
| 7 | 12.5556 | 2.55556 |
| 8 | 13.5556 | 3.55556 |
| 9 | 14.5556 | 4.55556 |
| 10 | 15.5556 | 5.55556 |
| 11 | 16.5556 | 6.55556 |
| 12 | 17.5556 | 7.55556 |
| 13 | 18.5556 | 8.55556 |
| 14 | 19.5556 | 9.55556 |
| 15 | 20.5556 | 10.5556 |
| 16 | 21.5556 | 11.5556 |
| 17 | 22.5556 | 12.5556 |
| 18 | 23.5556 | 13.5556 |
| 19 | 24.5556 | 14.5556 |
| 20 | 25.5556 | 15.5556 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| n | 0 | 1 | 2 | 3 | 4 | 5 |
|  | 0 | 6 | 15/2 | 8.55 | 9.55 | 10.55 |
|  | 0 | -3 | -12/5 | -1.44 | -0.44 | 0.55 |

**2 набор характеристик:**

**Количество состояний в системе: 3**

**Матрица вероятностей перехода**

|  |  |  |  |
| --- | --- | --- | --- |
|  | В 1-ое состояние | Во 2-ое состояние | В 3-ое состояние |
| Из 1-го состояния | 0.5 | 0.3 | 0.2 |
| Из 2-го состояния | 0.4 | 0.4 | 0.2 |
| Из 3-го состояния | 0.2 | 0.2 | 0.6 |

**Матрица доходов**

|  |  |  |  |
| --- | --- | --- | --- |
|  | В 1-ое состояние | Во 2-ое состояние | В 3-ое состояние |
| Из 1-го состояния | 9 | 3 | -6 |
| Из 2-го состояния | 4 | 3 | 2 |
| Из 3-го состояния | -5 | -2 | 2 |

**Полученные результаты**

|  |  |  |  |
| --- | --- | --- | --- |
| **n** | **(n)** | **(n)** | **(n)** |
| **0** | 0 | 0 | 0 |
| **1** | 4.2 | 3.2 | -0.2 |
| **2** | 7.22 | 6.12 | 1.16 |
| **3** | 9.878 | 8.768 | 3.164 |
| **4** | 12.4022 | 11.2912 | 5.4276 |
| **5** | 14.874 | 13.7629 | 7.79524 |
| **6** | 17.3249 | 16.2138 | 10.2045 |
| **7** | 19.7675 | 18.6564 | 12.6304 |
| **8** | 22.2067 | 21.0956 | 15.063 |
| **9** | 24.6447 | 23.5336 | 17.4983 |
| **10** | 27.0821 | 25.971 | 19.9346 |
| **11** | 29.5192 | 28.4081 | 22.3714 |
| **12** | 31.9563 | 30.8452 | 24.8083 |
| **13** | 34.3934 | 33.2823 | 27.2453 |
| **14** | 36.8304 | 35.7193 | 29.6823 |
| **15** | 39.2675 | 28.1564 | 32.1193 |
| **16** | 41.7045 | 40.5934 | 34.5564 |
| **17** | 44.1416 | 43.0305 | 36.9934 |
| **18** | 46.5786 | 45.4675 | 39.4305 |
| **19** | 49.0156 | 47.9045 | 41.8675 |
| **20** | 51.4527 | 50.3416 | 44.3045 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| n | 0 | 1 | 2 | 3 | 4 | 5 |
| (n) | 0 | 4.2 | 7.22 | 9.878 | 12.4022 | 14.874 |
| (n) | 0 | 3.2 | 6.12 | 8.768 | 11.2912 | 13.7629 |
| (n) | 0 | -0.2 | 1.16 | 3.164 | 5.4276 | 7.79524 |

**3 набор характеристик:**

**Количество состояний в системе: 5**

**Матрица вероятностей перехода**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | В 1-ое состояние | Во 2-ое состояние | В 3-ое состояние | В 4-ое состояние | В 5-ое состояние |
| Из 1-го состояния | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 |
| Из 2-го состояния | 0.1 | 0.1 | 0.1 | 0.1 | 0.6 |
| Из 3-го состояния | 0.3 | 0.3 | 0.3 | 0.05 | 0.05 |
| Из 4-го состояния | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 |
| Из 5-го состояния | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 |

**Матрица доходов**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | В 1-ое состояние | Во 2-ое состояние | В 3-ое состояние | В 4-ое состояние | В 5-ое состояние |
| Из 1-го состояния | 1 | 2 | 3 | 2 | 1 |
| Из 2-го состояния | -1 | 1 | -1 | 1 | 2 |
| Из 3-го состояния | 2 | 2 | 2 | 2 | -8 |
| Из 4-го состояния | -1 | 1 | 2 | 3 | 1 |
| Из 5-го состояния | 2 | -4 | 3 | 3 | 2 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| n | 0 | 1 | 2 | 3 | 4 | 5 |
| (n) | 0 | 1.8 | 3.18 | 4.56 | 5.94 | 7.32 |
| (n) | 0 | 1.2 | 2.49 | 3.87 | 5.25 | 6.63 |
| (n) | 0 | 1.5 | 2.97 | 4.35 | 5.73 | 7.11 |
| (n) | 0 | 1.2 | 2.58 | 3.96 | 5.34 | 6.72 |
| (n) | 0 | 1.2 | 2.58 | 3.96 | 5.34 | 6.72 |

**4 набор характеристик:**

**Количество состояний в системе: 10**

**Матрица вероятностей перехода**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | В 1-ое состояние | Во 2-ое состояние | В 3-ое состояние | В 4-ое состояние | В 5-ое состояние |
| Из 1-го состояния | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 |
| Из 2-го состояния | 0.05 | 0.05 | 0.05 | 0.05 | 0.05 |
| Из 3-го состояния | 0.05 | 0.2 | 0.2 | 0.2 | 0.1 |
| Из 4-го состояния | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 |
| Из 5-го состояния | 0.05 | 0.05 | 0.05 | 0.05 | 0.05 |
| Из 6-го состояния | 0.05 | 0.2 | 0.2 | 0.2 | 0.1 |
| Из 7-го состояния | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 |
| Из 8-го состояния | 0.05 | 0.05 | 0.05 | 0.05 | 0.05 |
| Из 9-го состояния | 0.05 | 0.2 | 0.2 | 0.2 | 0.1 |
| Из 10-го состояния | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 |
|  | В 6-ое состояние | Во 7-ое состояние | В 8-ое состояние | В 9-ое состояние | В 10-ое состояние |
| Из 1-го состояния | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 |
| Из 2-го состояния | 0.05 | 0.2 | 0.2 | 0.2 | 0.1 |
| Из 3-го состояния | 0.05 | 0.05 | 0.05 | 0.05 | 0.05 |
| Из 4-го состояния | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 |
| Из 5-го состояния | 0.05 | 0.2 | 0.2 | 0.2 | 0.1 |
| Из 6-го состояния | 0.05 | 0.05 | 0.05 | 0.05 | 0.05 |
| Из 7-го состояния | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 |
| Из 8-го состояния | 0.05 | 0.2 | 0.2 | 0.2 | 0.1 |
| Из 9-го состояния | 0.05 | 0.05 | 0.05 | 0.05 | 0.05 |
| Из 10-го состояния | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 |

**Матрица доходов**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | В 1-ое состояние | Во 2-ое состояние | В 3-ое состояние | В 4-ое состояние | В 5-ое состояние |
| Из 1-го состояния | 2 | 1 | -3 | 4 | -1 |
| Из 2-го состояния | 1 | 2 | 2 | 3 | 1 |
| Из 3-го состояния | 2 | 1 | -4 | 2 | 1 |
| Из 4-го состояния | 1 | 1 | 1 | 1 | -5 |
| Из 5-го состояния | 2 | 1 | -3 | 4 | -1 |
| Из 6-го состояния | 2 | 1 | -3 | 4 | -1 |
| Из 7-го состояния | 2 | 1 | -3 | 4 | -1 |
| Из 8-го состояния | 2 | 2 | -3 | 4 | -1 |
| Из 9-го состояния | 1 | 2 | 3 | 4 | -10 |
| Из 10-го состояния | 2 | 1 | -3 | 4 | -1 |
|  | В 6-ое состояние | Во 7-ое состояние | В 8-ое состояние | В 9-ое состояние | В 10-ое состояние |
| Из 1-го состояния | 1 | 2 | 2 | 2 | 1 |
| Из 2-го состояния | 1 | 1 | 1 | 1 | -5 |
| Из 3-го состояния | 2 | 1 | -3 | 1 | -1 |
| Из 4-го состояния | 2 | 3 | -3 | 4 | -1 |
| Из 5-го состояния | -10 | -2 | -3 | 2 | 1 |
| Из 6-го состояния | 2 | 1 | -3 | 4 | -1 |
| Из 7-го состояния | -1 | -2 | -2 | 3 | -1 |
| Из 8-го состояния | 2 | 1 | -3 | 4 | -1 |
| Из 9-го состояния | 2 | 1 | -3 | 4 | -1 |
| Из 10-го состояния | 2 | 1 | -3 | 4 | -1 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| n | 0 | 1 | 2 | 3 | 4 | 5 |
| (n) | 0 | 1.1 | 1.55875 | 2.01956 | 2.48098 | 2.94241 |
| (n) | 0 | 1.1875 | 1.68687 | 2.14769 | 2.60911 | 3.07053 |
| (n) | 0 | 0.2 | 0.425 | 0.887844 | 1.34927 | 1.81069 |
| (n) | 0 | 0.4 | 0.85875 | 1.31956 | 1.78098 | 2.24241 |
| (n) | 0 | -0.85 | -0.350625 | 0.110187 | 0.571609 | 1.03303 |
| (n) | 0 | 0.55 | 0.975 | 1.43784 | 1.89927 | 2.36069 |
| (n) | 0 | 5.5 | 0.45875 | 0.919563 | 1.38098 | 1.84241 |
| (n) | 0 | 0.6 | 1.09937 | 1.56019 | 2.02161 | 2.48303 |
| (n) | 0 | 1 | 1.425 | 1.88784 | 2.34927 | 2.81069 |
| (n) | 0 | 0.6 | 1.05872 | 1.51956 | 1.98098 | 2.44241 |

## Эксперимент 3.

Для описания однородной марковской цепи определим состояния цепи, такты функционирования, управления, которые можно применять к системе, матрицы переходных вероятностей при выбранных управлениях, матрицы доходов при выбранных управлениях.

Нам требуется определиться какие управления нужно применять в каких состояниях, с тем, чтобы математическое ожидание дохода было максимально.

Их предыдущей задачи мы имеем данные рекуррентные соотношения , для поиска ожидаемого дохода.

, , *n=1,2…* или

,

где  – средний одношаговый доход для состояния *i*, 

Можем заметить, что , начиная с первого шага , математическое ожидание дохода зависит только от среднего одношагового дохода , то бишь от характеристик матрицы вероятностей и доходов.

Исходя из этого можем сделать вывод, что при наличии элементов управления , нам необходимо выбирать для каждого состояния такое управление, чтобы средний одношаговый доход был максимальным.

Исходя из этого получаем:

где – максимальный средний одношаговый доход для состояния *i*, среди всех управлений применяемых к нему. 

*,*

k- количество управлений для состояния,

– средний одношаговый доход, для состояния i при использовании управления s, s.

## Задача

Написать программу, которая при заданных величинах:

n−количество состояний , m − количество тактов , k – количество управлений для каждого состояния .

P=− матрица переходных вероятностей

R= ­− матрица доходов

находит

(n) – максимальное математическое ожидание дохода, который будет получен в результате работы алгоритма, если он начинает работать в состоянии i и ему остается работать еще n-шагов .

Программа.

**Блок 1. Генерация задач.**

Данный блок предназначен для генерирования задач с помощью ручного.

Генерирование задач с помощью ручного ввода включает в себя функции :

* Считывания количества состояний – с помощью ручного ввода задаем целое число состояний в цепи.
* Считывания количество тактов цепи – с помощью ручного ввода задаем необходимое целое число тактов марковской цепи.
* Считывания количества управлений для состояний – с помощью ручного ввода задаем необходимое целое число количества управлений, единое для каждого состояния.
* Генерации матрицы переходных вероятностей марковской цепи – формируем ,с помощью ручного ввода, трехмерный массив ,состоящий из вероятностей перехода для всех состояний.
* Генерации матрицы изменения «дохода »марковской цепи – формируем ,с помощью ручного ввода, трехмерный массив ,состоящий из изменения «дохода »марковской цепи для всех состояний.  
    
  Также здесь присутствует ограничение на то, чтобы сумма всех вероятностей для одного состояния была не больше 1.
* Вывода введённых данных в виде таблицы.

**Блок 2. Функциональный блок.**

Данный блок предназначен для на основании исходных данных провести необходимые вычисления.

Функциональный блок включает в себя:

* Цикл, отвечающий за исполнение определенного числа тактов – Цикл 1- имеет в своем составе Цикл 2 и Цикл 3 . Они вместе реализуют подсчёт вероятностей попасть в каждое состояние через заданное количество шагов.
* Подсчёт попасть в определенное состояние для каждого состояния на следующем такте – Цикл 2 -рассчитывает вероятность попасть каждое состояние на следующем такте, записывает эти вероятности в массив, находится в Цикле 1.
* Подсчёт полного ожидаемого дохода – Цикл 3 -рассчитывает максимальный ожидаемый доход на текущем такте системы для конкретного состояния, записывает результаты в массив, находится в Цикле 1.
* Поиск наибольшего среднего одношагового дохода для каждого состояния- Цикл 4 – ведет поиск среди массива средних одношаговых доходов конкретного состояния для всех управлений , **максимальное значение** одношагового среднего дохода при применении определенного управления.
* Цикл 5 – вычисление среднего одношагового дохода для каждого состояния и каждого управления.

**Блок 3. Отображение решений.**

Предназначен для вывода решений в виде «Максимальный полученный доход через P тактов N состояния составляет : M», где

N = 1,2 … K , где К- заданное пользователем количество состояний.

M – полученный результат

P – заданное пользователем количество тактов.

**Программа реализована средствами языка С++ и его встроенных библиотек.**

## Вычислительный эксперимент – решение примеров с разными характеристиками.

**1 набор характеристик:**

**Количество состояний в системе: 2**

**Матрица вероятностей перехода**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Состояния | Управления | Вероятность перехода | | Изменение дохода | |
| В 1 | Во 2 |  |  |
| Из 1 | u =1 | 0,5 | 0,5 | 9 | -3 |
| u =2 | 0,8 | 0,2 | 4 | 2 |
| Из 2 | u =1 | 0,4 | 0,6 | 3 | -7 |
| u =2 | 0,7 | 0,3 | 1 | -19 |

**Полученные результаты**

|  |  |  |
| --- | --- | --- |
| n | (n) | (n) |
| 0 | 0 | 0 |
| 1 | 3.6 | -3 |
| 2 | 5.88 | -3.36 |
| 3 | 7.632 | -2.664 |
| 4 | 9.1728 | -1.5456 |
| 5 | 10.6291 | -0.25824 |
| 6 | 12.0516 | 1.0967 |
| 7 | 12.0516 | 2.47868 |
| 8 | 13.4607 | 3.55556 |
| 9 | 16.2657 | 5.26859 |
| 10 | 17.6663 | 6.66744 |
| 11 | 19.0065 | 8.06697 |
| 12 | 20.4666 | 9.46679 |
| 13 | 21.8666 | 10.8667 |
| 14 | 23.2667 | 12.2667 |
| 15 | 24.6667 | 13.6667 |
| 16 | 26.0667 | 15.0667 |
| 17 | 27.4667 | 16.4667 |
| 18 | 28.8667 | 17.8667 |
| 19 | 30.2667 | 19.2667 |
| 20 | 31.6667 | 20.6667 |

|  |  |  |
| --- | --- | --- |
| Для состояния n наибольший одношаговый доход | Управление | Наибольший одношаговый доход |
| 1 | 2 | 3.6 |
| 2 | 1 | -3 |

**2 набор характеристик:**

**Количество состояний в системе: 3**

**Количество управлений : 4**

**Матрица вероятностей перехода**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Состояния | Управления | Вероятность перехода | | |
| В 1 | Во 2 | В 3 |
| Из 1 | u =1 | 0,5 | 0,3 | 0,2 |
| u =2 | 0,5 | 0,2 | 0,3 |
| u =3 | 0,2 | 0,6 | 0,2 |
| u =4 | 0,5 | 0,2 | 0,3 |
| Из 2 | u =1 | 0,4 | 0,3 | 0,3 |
| u =2 | 0,7 | 0,1 | 0,2 |
| u =3 | 0,2 | 0,3 | 0,5 |
| u =4 | 0,7 | 0,1 | 0,2 |
| Из 3 | u =1 | 0,7 | 0,2 | 0,1 |
| u =2 | 0,2 | 0,1 | 0,7 |
| u =3 | 0,1 | 0,4 | 0,5 |
| u =4 | 0,4 | 0,1 | 0,5 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Состояния | Управления | Изменение дохода | | |
| В 1 | Во 2 | В 3 |
| Из 1 | u =1 | 9 | -3 | 3 |
| u =2 | 4 | 2 | 2 |
| u =3 | -8 | 3 | -7 |
| u =4 | 2 | 3 | -4 |
| Из 2 | u =1 | 3 | -7 | 1 |
| u =2 | 1 | -19 | -3 |
| u =3 | -7 | 2 | 4 |
| u =4 | 4 | -1 | 2 |
| Из 3 | u =1 | 4 | -7 | 1 |
| u =2 | 2 | -8 | -7 |
| u =3 | -7 | 2 | 4 |
| u =4 | 2 | -1 | 3 |

**Полученные результаты**

|  |  |  |  |
| --- | --- | --- | --- |
| **n** | **(n)** | **(n)** | **(n)** |
| **0** | 0 | 0 | 0 |
| **1** | 4.2 | 3.4 | 2.2 |
| **2** | 7.76 | 7.12 | 5.35 |
| **3** | 11.28 | 10.608 | 8.676 |
| **4** | 14.7576 | 14.092 | 12.1108 |
| **5** | 18.2286 | 17.5617 | 15.5676 |
| **6** | 21.6963 | 21.0297 | 19.0314 |
| **7** | 25.1633 | 24.4967 | 22.4972 |
| **8** | 28.6301 | 27.9634 | 25.9636 |
| **9** | 32.0968 | 31.4301 | 29.4302 |
| **10** | 35.5635 | 34.8968 | 32.8968 |
| **11** | 39.0302 | 38.3635 | 36.3635 |
| **12** | 42.4968 | 41.8302 | 39.8302 |
| **13** | 45.9635 | 45.2968 | 43.2968 |
| **14** | 49.4302 | 48.7635 | 46.7635 |
| **15** | 52.8968 | 52.2302 | 50.2302 |
| **16** | 56.3635 | 55.6968 | 53.6968 |
| **17** | 59.8302 | 59.1635 | 57.1635 |
| **18** | 63.2968 | 62.6302 | 60.6302 |
| **19** | 66.7635 | 66.0968 | 64.0968 |
| **20** | 70.2302 | 69.5635 | 67.5635 |

**Для состояния 1 наибольший одношаговый доход 4.2 при условии выбора управления: 1**

**Для состояния 2 наибольший одношаговый доход 3.4 при условии выбора управления: 2**

**Для состояния 3 наибольший одношаговый доход 2.2 при условии выбора управления: 4**

3 **набор характеристик:**

**Количество состояний в системе: 10**

**Количество управлений : 5**

**Все приведенные данные**

**Матрица вероятностей перехода**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Состояния | Управления | Вероятность перехода умноженная на 2 | | | | |
| В 1 | Во 2 | В 3 | В 4 | В 5 |
| Из 1 | u =1 | 0,3 | 0,3 | 0,2 | 0,1 | 0,1 |
| u =2 | 0,5 | 0,2 | 0,1 | 0,1 | 0,1 |
| u =3 | 0,2 | 0,2 | 0,2 | 0,2 | 0,2 |
| u =4 | 0,1 | 0,2 | 0,3 | 0,2 | 0,2 |
| Из 2 | u =1 | 0,4 | 0,3 | 0,1 | 0,1 | 0,1 |
| u =2 | 0,5 | 0,1 | 0,2 | 0,1 | 0,1 |
| u =3 | 0,2 | 0,3 | 0,1 | 0,2 | 0,2 |
| u =4 | 0,1 | 0,1 | 0,2 | 0,3 | 0,3 |
| Из 3 | u =1 | 0,2 | 0,2 | 0,1 | 0,2 | 0,3 |
| u =2 | 0,2 | 0,2 | 0,2 | 0,2 | 0,2 |
| u =3 | 0,5 | 0,1 | 0,2 | 0,1 | 0,1 |
| u =4 | 0,1 | 0,1 | 0,2 | 0,3 | 0,3 |
| Из 4 | u =1 | 0,1 | 0,2 | 0,3 | 0,2 | 0 |
| u =2 | 0,1 | 0,1 | 0,2 | 0,3 | 0,3 |
| u =3 | 0,1 | 0,2 | 0,3 | 0,2 | 0,2 |
| u =4 | 0,5 | 0,1 | 0,2 | 0,1 | 0,1 |
| Из 5 | u =1 | 0,1 | 0,1 | 0,2 | 0,3 | 0,3 |
| u =2 | 0,5 | 0,1 | 0,2 | 0,1 | 0,1 |
| u =3 | 0,1 | 0,1 | 0,2 | 0,3 | 0,3 |
| u =4 | 0,5 | 0,1 | 0,2 | 0,1 | 0,1 |
| Из 6 | u =1 | 0,1 | 0,2 | 0,3 | 0,2 | 0,2 |
| u =2 | 0,1 | 0,1 | 0,2 | 0,3 | 0,3 |
| u =3 | 0,4 | 0,3 | 0,1 | 0,1 | 0,1 |
| u =4 | 0,5 | 0,1 | 0,2 | 0,1 | 0,1 |
| Из 7 | u =1 | 0,5 | 0,1 | 0,2 | 0,1 | 0,1 |
| u =2 | 0,1 | 0,1 | 0,2 | 0,3 | 0,3 |
| u =3 | 0,4 | 0,3 | 0,1 | 0,1 | 0,1 |
| u =4 | 0,1 | 0,2 | 0,3 | 0,2 | 0,2 |
| Из 8 | u =1 | 0,5 | 0,1 | 0,2 | 0,1 | 0,1 |
| u =2 | 0,1 | 0,1 | 0,2 | 0,3 | 0,3 |
| u =3 | 0,4 | 0,3 | 0,1 | 0,1 | 0,1 |
| u =4 | 0,1 | 0,2 | 0,3 | 0,2 | 0,2 |
| Из 9 | u =1 | 0,5 | 0,1 | 0,2 | 0,1 | 0,1 |
| u =2 | 0,1 | 0,1 | 0,2 | 0,3 | 0,3 |
| u =3 | 0,1 | 0,2 | 0,3 | 0,2 | 0,2 |
| u =4 | 0,5 | 0,1 | 0,2 | 0,1 | 0,1 |
| Из 10 | u =1 | 0,1 | 0,1 | 0,1 | 0,2 | 0,3 |
| u =2 | 0,5 | 0,1 | 0,2 | 0,1 | 0,1 |
| u =3 | 0,4 | 0,3 | 0,1 | 0,1 | 0,1 |
| u =4 | 0,5 | 0,1 | 0,2 | 0,1 | 0,1 |
| Состояния | Управления | Вероятность перехода | | | | |
| В 6 | Во 7 | В 8 | В 9 | В 10 |
| Из 1 | u =1 | 0,3 | 0,3 | 0,2 | 0,1 | 0,1 |
| u =2 | 0,4 | 0,2 | 0,2 | 0,1 | 0,1 |
| u =3 | 0,2 | 0,2 | 0,2 | 0,2 | 0,2 |
| u =4 | 0,1 | 0,2 | 0,3 | 0,2 | 0,2 |
| Из 2 | u =1 | 0,4 | 0,3 | 0,1 | 0,1 | 0,1 |
| u =2 | 0,5 | 0,1 | 0,2 | 0,1 | 0,1 |
| u =3 | 0,2 | 0,3 | 0,1 | 0,2 | 0,2 |
| u =4 | 0,1 | 0,1 | 0,2 | 0,3 | 0,3 |
| Из 3 | u =1 | 0,2 | 0,2 | 0,1 | 0,2 | 0,3 |
| u =2 | 0,4 | 0,2 | 0,1 | 0,1 | 0,1 |
| u =3 | 0,4 | 0,3 | 0,1 | 0,1 | 0,1 |
| u =4 | 0,2 | 0,2 | 0,1 | 0,2 | 0,3 |
| Из 4 | u =1 | 0,4 | 0,2 | 0,1 | 0,2 | 0,3 |
| u =2 | 0,5 | 0,1 | 0,2 | 0,1 | 0,1 |
| u =3 | 0,5 | 0,2 | 0,1 | 0,1 | 0,1 |
| u =4 | 0,1 | 0,1 | 0,2 | 0,3 | 0,3 |
| Из 5 | u =1 | 0,2 | 0,2 | 0,1 | 0,2 | 0,3 |
| u =2 | 0,4 | 0,3 | 0,1 | 0,1 | 0,1 |
| u =3 | 0,5 | 0,1 | 0,2 | 0,1 | 0,1 |
| u =4 | 0,5 | 0,2 | 0,1 | 0,1 | 0,1 |
| Из 6 | u =1 | 0,1 | 0,1 | 0,2 | 0,3 | 0,3 |
| u =2 | 0,4 | 0,3 | 0,1 | 0,1 | 0,1 |
| u =3 | 0,2 | 0,2 | 0,1 | 0,2 | 0,3 |
| u =4 | 0,5 | 0,1 | 0,2 | 0,1 | 0,1 |
| Из 7 | u =1 | 0,5 | 0,2 | 0,1 | 0,1 | 0,1 |
| u =2 | 0,1 | 0,1 | 0,2 | 0,3 | 0,3 |
| u =3 | 0,2 | 0,2 | 0,1 | 0,2 | 0,3 |
| u =4 | 0,4 | 0,3 | 0,1 | 0,1 | 0,1 |
| Из 8 | u =1 | 0,5 | 0,1 | 0,2 | 0,1 | 0,1 |
| u =2 | 0,5 | 0,2 | 0,1 | 0,1 | 0,1 |
| u =3 | 0,1 | 0,1 | 0,2 | 0,3 | 0,3 |
| u =4 | 0,4 | 0,3 | 0,1 | 0,1 | 0,1 |
| Из 9 | u =1 | 0,5 | 0,1 | 0,2 | 0,1 | 0,1 |
| u =2 | 0,5 | 0,2 | 0,1 | 0,1 | 0,1 |
| u =3 | 0,1 | 0,1 | 0,2 | 0,3 | 0,3 |
| u =4 | 0,2 | 0,2 | 0,1 | 0,2 | 0,3 |
| Из 10 | u =1 | 0 | 0,8 | 0,2 | 0,1 | 0,1 |
| u =2 | 0,2 | 0,2 | 0,1 | 0,2 | 0,3 |
| u =3 | 0,1 | 0,1 | 0,2 | 0,3 | 0,3 |
| u =4 | 0,5 | 0,2 | 0,1 | 0,1 | 0,1 |

**Матрица доходов**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Состояния | Управления | Вероятность перехода | | | | |
| В 1 | Во 2 | В 3 | В 4 | В 5 |
| Из 1 | u =1 | 1 | 2 | 3 | 1 | 2 |
| u =2 | 5 | 2 | 1 | 1 | 1 |
| u =3 | 2 | 2 | 2 | 2 | 2 |
| u =4 | 1 | 2 | 3 | 2 | 2 |
| Из 2 | u =1 | 4 | 3 | 1 | 1 | 1 |
| u =2 | 5 | 1 | 2 | 1 | 1 |
| u =3 | 2 | 3 | 1 | 2 | 2 |
| u =4 | 1 | 1 | 2 | 3 | 3 |
| Из 3 | u =1 | 2 | 2 | 1 | 2 | 3 |
| u =2 | 2 | 2 | 2 | 2 | 2 |
| u =3 | 5 | 1 | 2 | 1 | 1 |
| u =4 | 1 | 1 | 2 | 3 | 3 |
| Из 4 | u =1 | 1 | 2 | 3 | 2 | 2 |
| u =2 | 1 | 1 | 2 | 3 | 3 |
| u =3 | 1 | 2 | 3 | 2 | 2 |
| u =4 | 5 | 1 | 2 | 1 | 1 |
| Из 5 | u =1 | 1 | 1 | 2 | 3 | 3 |
| u =2 | 5 | 1 | 2 | 1 | 1 |
| u =3 | 1 | 1 | 2 | 3 | 3 |
| u =4 | 5 | 1 | 2 | 1 | 1 |
| Из 6 | u =1 | 1 | 2 | 3 | 2 | 2 |
| u =2 | 1 | 1 | 2 | 3 | 3 |
| u =3 | 4 | 3 | 1 | 1 | 1 |
| u =4 | 5 | 1 | 2 | 1 | 1 |
| Из 7 | u =1 | 5 | 1 | 2 | 1 | 1 |
| u =2 | 1 | 1 | 2 | 3 | 3 |
| u =3 | 4 | 3 | 1 | 1 | 1 |
| u =4 | 1 | 2 | 3 | 2 | 2 |
| Из 8 | u =1 | 5 | 1 | 2 | 1 | 1 |
| u =2 | 1 | 1 | 2 | 3 | 3 |
| u =3 | 4 | 3 | 1 | 1 | 1 |
| u =4 | 1 | 2 | 3 | 2 | 2 |
| Из 9 | u =1 | 5 | 1 | 2 | 1 | 1 |
| u =2 | 1 | 1 | 2 | 3 | 3 |
| u =3 | 1 | 2 | 3 | 2 | 2 |
| u =4 | 5 | 1 | 2 | 1 | 1 |
| Из 10 | u =1 | 1 | 1 | 2 | 3 | 3 |
| u =2 | 5 | 1 | 2 | 1 | 1 |
| u =3 | 4 | 3 | 1 | 1 | 1 |
| u =4 | 5 | 1 | 2 | 1 | 1 |
| Состояния | Управления | Вероятность перехода | | | | |
| В 6 | Во 7 | В 8 | В 9 | В 10 |
| Из 1 | u =1 | 1 | 2 | 3 | 1 | 2 |
| u =2 | 5 | 2 | 1 | 1 | 1 |
| u =3 | 2 | 2 | 2 | 2 | 2 |
| u =4 | 1 | 2 | 3 | 2 | 2 |
| Из 2 | u =1 | 4 | 3 | 1 | 1 | 1 |
| u =2 | 5 | 1 | 2 | 1 | 1 |
| u =3 | 2 | 3 | 1 | 2 | 2 |
| u =4 | 1 | 1 | 2 | 3 | 3 |
| Из 3 | u =1 | 2 | 2 | 1 | 2 | 3 |
| u =2 | 2 | 2 | 2 | 2 | 2 |
| u =3 | 5 | 1 | 2 | 1 | 1 |
| u =4 | 1 | 1 | 2 | 3 | 3 |
| Из 4 | u =1 | 1 | 2 | 3 | 2 | 2 |
| u =2 | 4 | 3 | 1 | 1 | 1 |
| u =3 | 5 | 1 | 2 | 1 | 1 |
| u =4 | 1 | 1 | 2 | 3 | 3 |
| Из 5 | u =1 | 2 | 3 | 1 | 2 | 2 |
| u =2 | 1 | 2 | 3 | 1 | 2 |
| u =3 | 4 | 3 | 1 | 1 | 1 |
| u =4 | 5 | 1 | 2 | 1 | 1 |
| Из 6 | u =1 | 2 | 3 | 1 | 2 | 2 |
| u =2 | 1 | 1 | 2 | 3 | 3 |
| u =3 | 2 | 2 | 2 | 2 | 2 |
| u =4 | 4 | 3 | 1 | 1 | 1 |
| Из 7 | u =1 | 2 | 3 | 1 | 2 | 2 |
| u =2 | 1 | 2 | 3 | 1 | 2 |
| u =3 | 5 | 1 | 2 | 1 | 1 |
| u =4 | 1 | 1 | 2 | 3 | 3 |
| Из 8 | u =1 | 2 | 2 | 2 | 2 | 2 |
| u =2 | 2 | 3 | 1 | 2 | 2 |
| u =3 | 4 | 3 | 1 | 1 | 1 |
| u =4 | 5 | 1 | 2 | 1 | 1 |
| Из 9 | u =1 | 1 | 2 | 3 | 1 | 2 |
| u =2 | 1 | 1 | 2 | 3 | 3 |
| u =3 | 2 | 3 | 1 | 2 | 2 |
| u =4 | 2 | 2 | 2 | 2 | 2 |
| Из 10 | u =1 | 5 | 1 | 2 | 1 | 1 |
| u =2 | 4 | 3 | 1 | 1 | 1 |
| u =3 | 2 | 3 | 1 | 2 | 2 |
| u =4 | 1 | 1 | 2 | 3 | 3 |

**Таблица всех возможных средних одношаговых доходов**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Состояние\Управление | 1 | 2 | 3 | 4 |
| 1 | 1.8 | **2.95** | 2.05 | 2.15 |
| 2 | **3** | 2.25 | 1.95 | 2.45 |
| 3 | 2 | 1.95 | **2.4** | 2.4 |
| 4 | 2.2 | 2.55 | **2.95** | 1.95 |
| 5 | **2.8** | 2.65 | 2.65 | 2.05 |
| 6 | **3.2** | 2.8 | 2.1 | 2.6 |
| 7 | 2.2 | 2.25 | **2.35** | 1.65 |
| 8 | 2.4 | 2.4 | 1.85 | **2.6** |
| 9 | 2.2 | 2.55 | **2.6** | 2.35 |
| 10 | 2 | **3.15** | 2.25 | 2.35 |

**Для состояния 1 наибольший одношаговый доход 2.95 при условии выбора управления: 2**

**Для состояния 2 наибольший одношаговый доход 3 при условии выбора управления: 1**

**Для состояния 3 наибольший одношаговый доход 2.4 при условии выбора управления: 3**

**Для состояния 4 наибольший одношаговый доход 2.95 при условии выбора управления: 3**

**Для состояния 5 наибольший одношаговый доход 2.8 при условии выбора управления: 1**

**Для состояния 6 наибольший одношаговый доход 3.2 при условии выбора управления: 1**

**Для состояния 7 наибольший одношаговый доход 2.35 при условии выбора управления: 3**

**Для состояния 8 наибольший одношаговый доход 2.6 при условии выбора управления: 4**

**Для состояния 9 наибольший одношаговый доход 2.6 при условии выбора управления: 3**

**Для состояния 10 наибольший одношаговый доход 3.15 при условии выбора управления: 2**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| n | 0 | 1 | 2 | 3 | 4 | 5 |
| (n) | 0 | 2.95 | 5.775 | 8.63275 | 11.4907 | 14.3485 |
| (n) | 0 | 3 | 5.8675 | 8.72375 | 11.5814 | 14.4393 |
| (n) | 0 | 2.4 | 5.265 | 8.1245 | 10.9823 | 13.8401 |
| (n) | 0 | 2.95 | 5.83 | 8.68687 | 11.5446 | 14.4024 |
| (n) | 0 | 2.8 | 5.6575 | 8.5175 | 11.3752 | 14.233 |
| (n) | 0 | 3.2 | 6.08 | 8.93687 | 11.7946 | 14.6524 |
| (n) | 0 | 2.35 | 5.215 | 8.0745 | 10.9323 | 13.7901 |
| (n) | 0 | 2.6 | 5.4325 | 8.28975 | 11.1478 | 14.0056 |
| (n) | 0 | 2.6 | 5.48 | 8.33687 | 11.1946 | 14.0524 |
| (n) | 0 | 3.15 | 6.0175 | 8.876 | 11.7338 | 14.5917 |

**В итоге можем увидеть на нескольких примерах, использование какой стратегии является наиболее рациональным для каждого случая и состояния.**

# Заключение:

**В результате работы было исследовано поведение характеристик Марковской Цепи при вводе разных входных данных. Была смоделирована работа в рамках отдельного предприятия и выбрана наиболее оптимизированная стратегия для его видения.**

# Источники

1. «Теория выбора и принятия решений»: учебное пособие. И.М. Макаров, Т.М.Виноградская, А.А. Рубчинский, В.Б. Соколов. Москва, изд. «Наука», 1982.

2. «Теория вероятностей» Е.С. Вентцель. Москва, изд. «Наука», 1969.

3.[**https://ru.wikipedia.org/wiki/Цепь\_Маркова#Цепь\_Маркова\_с\_дискретным\_временем**](https://ru.wikipedia.org/wiki/Цепь_Маркова#Цепь_Маркова_с_дискретным_временем)

# Приложение

**Код программы для задачи 1:**

#include <iostream>

#include<ctime>

#include<string>

#include<algorithm>

#include<cmath>

using namespace std;

int main()

{

setlocale(LC\_ALL, "Russian");

int n, m, p;

double r1, re;

cout << "Введите количество состояний в системе" << endl;

cin >> n;

cout << "Введите количество шагов" << endl;

cin >> m;

double\* chance = new double[n];

double\* chancehelp = new double[n];

double\*\* arr = new double\* [n]; // две строки в массиве

for (int count = 0; count < n; count++)

arr[count] = new double[n];

cout << "Сумма всех вероятностей перехода для одного состояния должна равняться единице и каждая вероятность должна быть в границах от 0 до 1" << endl;

for (int count\_row = 0; count\_row < n; count\_row++)

{

double sum = 0;

while (sum != 1)

{

sum = 0;

for (int count\_column = 0; count\_column < n; count\_column++)

{

do

{

cout << "\n Введите веротность перехода из " << count\_row + 1 << " состояния в " << count\_column + 1 << " состояние" << endl;

cin >> arr[count\_row][count\_column];

sum += arr[count\_row][count\_column];

} while ((arr[count\_row][count\_column] > 1) || ((arr[count\_row][count\_column]) < 0));

}

}

}

for (int count\_row = 0; count\_row < n; count\_row++)

{

cout << "\n";

for (int count\_column = 0; count\_column < n; count\_column++)

cout << arr[count\_row][count\_column] << "\t";

}

double sum = 0;

while (sum != 1)

{

sum = 0;

for (int i = 0; i < n; i++)

{

cout << "\n Введите вероятность начального нахождения в " << i + 1 << "-ом состоянии ";

cin >> chance[i];

chancehelp[i] = 0;

sum += chance[i];

}

}

double sum1 = 0;

for (int number = 1; number <= m; number++)

{

for (int count\_row = 0; count\_row < n; count\_row++)

{

for (int count\_column = 0; count\_column < n; count\_column++)

{

sum1 = arr[count\_row][count\_column] \* chance[count\_row];

chancehelp[count\_column] += sum1;

sum1 = 0;

}

sum1 = 0;

}

for (int i = 0; i < n; i++)

{

chance[i] = chancehelp[i];

chancehelp[i] = 0;

}

for (int i = 0; i < n; i++)

{

cout << "\n Вероятоность попасть через " << number << "тактов в " << i + 1 << "состояние составляет :\t" << chance[i] << "\t";

}

cout << endl;

}

system("pause");

}

**Код программы для задачи 2:**

#include <iostream>

#include<ctime>

#include<string>

#include<algorithm>

#include<cmath>

using namespace std;

int main()

{

setlocale(LC\_ALL, "Russian");

int n, m, p;

double r1, re;

cout << "Введите количество состояний в системе" << endl;

cin >> n;

cout << "Введите количество шагов" << endl;

cin >> m;

double\*\* dohod = new double\* [n]; // две строки в массиве

for (int count = 0; count < n; count++)

dohod[count] = new double[n];

double\* income = new double[n];

double\* incomehelp = new double[n];

double\* q = new double[n];

double\* v0 = new double[n];

double\* chance = new double[n];

double\* chancehelp = new double[n];

double\*\* arr = new double\* [n]; // две строки в массиве

for (int count = 0; count < n; count++)

arr[count] = new double[n];

cout << "Сумма всех вероятностей перехода для одного состояния должна равняться единице и каждая вероятность должна быть в границах от 0 до 1" << endl;

for (int count\_row = 0; count\_row < n; count\_row++)

{

double sum = 0;

// while (sum != 1)

// {

sum = 0;

for (int count\_column = 0; count\_column < n; count\_column++)

{

do

{

cout << "\n Введите веротность перехода из " << count\_row + 1 << " состояния в " << count\_column + 1 << " состояние" << endl;

cin >> arr[count\_row][count\_column];

sum += arr[count\_row][count\_column];

} while ((arr[count\_row][count\_column] > 1) || ((arr[count\_row][count\_column]) < 0));

}

}

for (int count\_row = 0; count\_row < n; count\_row++)

{

for (int count\_column = 0; count\_column < n; count\_column++)

{

do

{

cout << "\n Введите полученный доход перехода из " << count\_row + 1 << " состояния в " << count\_column + 1 << " состояние" << endl;

cin >> dohod[count\_row][count\_column];

} while ((arr[count\_row][count\_column] > 1) || ((arr[count\_row][count\_column]) < 0));

}

}

for (int count\_row = 0; count\_row < n; count\_row++)

{

cout << "\n";

for (int count\_column = 0; count\_column < n; count\_column++)

cout << arr[count\_row][count\_column] << "\t";

}

cout << endl;

for (int count\_row = 0; count\_row < n; count\_row++)

{

cout << "\n";

for (int count\_column = 0; count\_column < n; count\_column++)

cout << dohod[count\_row][count\_column] << "\t";

}

double sum = 0;

//while (sum != 1)

//{

sum = 0;

for (int i = 0; i < n; i++)

{

cout << "\n Введите вероятность начального нахождения в " << i + 1 << "-ом состоянии ";

cin >> chance[i];

chancehelp[i] = 0;

sum += chance[i];

}

//}

double sum1 = 0;

for (int count\_row = 0; count\_row < n; count\_row++)

{

q[count\_row] = 0;

}

for (int count\_row = 0; count\_row < n; count\_row++)

{

for (int count\_column = 0; count\_column < n; count\_column++)

{

q[count\_row] += arr[count\_row][count\_column] \* dohod[count\_row][count\_column];

}

}

for (int count\_row = 0; count\_row < n; count\_row++)

{

income[count\_row] = q[count\_row];

}

for (int number = 1; number <= m; number++)

{

for (int count\_row = 0; count\_row < n; count\_row++)

{

incomehelp[count\_row] = q[count\_row];

for (int count\_column = 0; count\_column < n; count\_column++)

{

sum1 = arr[count\_row][count\_column] \* chance[count\_row];

chancehelp[count\_column] += sum1;

sum1 = 0;

incomehelp[count\_row] += income[count\_column] \* arr[count\_row][count\_column];

}

sum1 = 0;

}

for (int i = 0; i < n; i++)

{

chance[i] = chancehelp[i];

chancehelp[i] = 0;

}

for (int i = 0; i < n; i++)

{

cout << "\n Вероятоность попасть через " << number << "тактов в " << i + 1 << "состояние составляет :\t" << chance[i] << "\t";

}

cout << endl << endl;

for (int i = 0; i < n; i++)

{

cout << "\n Полученный доход через " << number << "тактов " << i + 1 << "состояния составляет :\t" << income[i] << "\t";

}

for (int i = 0; i < n; i++)

{

income[i] = incomehelp[i];

incomehelp[i] = 0;

}

}

system("pause");

}

**Код программы для задачи 3:**

#include <iostream>

#include<ctime>

#include<string>

#include<algorithm>

#include<cmath>

using namespace std;

int main()

{

setlocale(LC\_ALL, "Russian");

int n, m, p;

double r1, re;

cout << "Введите количество состояний в системе" << endl;

cin >> n;

cout << "Введите количество шагов" << endl;

cin >> m;

int control; // Выбор управления

cout << "Введите <<1>> если хотите использовать управление и <<2>> - если нет" << endl;

cin >> control;

if (control == 2)

{

double\*\* dohod = new double\* [n]; // две строки в массиве

for (int count = 0; count < n; count++)

dohod[count] = new double[n];

double\* income = new double[n];

double\* incomehelp = new double[n];

double\* q = new double[n];

double\* v0 = new double[n];

double\* chance = new double[n];

double\* chancehelp = new double[n];

double\*\* arr = new double\* [n]; // две строки в массиве

for (int count = 0; count < n; count++)

arr[count] = new double[n];

cout << "Сумма всех вероятностей перехода для одного состояния должна равняться единице и каждая вероятность должна быть в границах от 0 до 1" << endl;

for (int count\_row = 0; count\_row < n; count\_row++)

{

double sum = 0;

// while (sum != 1)

// {

sum = 0;

for (int count\_column = 0; count\_column < n; count\_column++)

{

do

{

cout << "\n Введите веротность перехода из " << count\_row + 1 << " состояния в " << count\_column + 1 << " состояние" << endl;

cin >> arr[count\_row][count\_column];

sum += arr[count\_row][count\_column];

} while ((arr[count\_row][count\_column] > 1) || ((arr[count\_row][count\_column]) < 0));

}

}

for (int count\_row = 0; count\_row < n; count\_row++)

{

for (int count\_column = 0; count\_column < n; count\_column++)

{

do

{

cout << "\n Введите полученный доход перехода из " << count\_row + 1 << " состояния в " << count\_column + 1 << " состояние" << endl;

cin >> dohod[count\_row][count\_column];

} while ((arr[count\_row][count\_column] > 1) || ((arr[count\_row][count\_column]) < 0));

}

}

for (int count\_row = 0; count\_row < n; count\_row++)

{

cout << "\n";

for (int count\_column = 0; count\_column < n; count\_column++)

cout << arr[count\_row][count\_column] << "\t";

}

cout << endl;

for (int count\_row = 0; count\_row < n; count\_row++)

{

cout << "\n";

for (int count\_column = 0; count\_column < n; count\_column++)

cout << dohod[count\_row][count\_column] << "\t";

}

double sum = 0;

//while (sum != 1)

//{

sum = 0;

for (int i = 0; i < n; i++)

{

cout << "\n Введите вероятность начального нахождения в " << i + 1 << "-ом состоянии ";

cin >> chance[i];

chancehelp[i] = 0;

sum += chance[i];

}

//}

double sum1 = 0;

for (int count\_row = 0; count\_row < n; count\_row++)

{

q[count\_row] = 0;

}

for (int count\_row = 0; count\_row < n; count\_row++)

{

for (int count\_column = 0; count\_column < n; count\_column++)

{

q[count\_row] += arr[count\_row][count\_column] \* dohod[count\_row][count\_column];

}

}

for (int count\_row = 0; count\_row < n; count\_row++)

{

income[count\_row] = q[count\_row];

}

for (int number = 1; number <= m; number++)

{

for (int count\_row = 0; count\_row < n; count\_row++)

{

incomehelp[count\_row] = q[count\_row];

for (int count\_column = 0; count\_column < n; count\_column++)

{

sum1 = arr[count\_row][count\_column] \* chance[count\_row];

chancehelp[count\_column] += sum1;

sum1 = 0;

incomehelp[count\_row] += income[count\_column] \* arr[count\_row][count\_column];

}

sum1 = 0;

}

for (int i = 0; i < n; i++)

{

chance[i] = chancehelp[i];

chancehelp[i] = 0;

}

for (int i = 0; i < n; i++)

{

cout << "\n Вероятоность попасть через " << number << "тактов в " << i + 1 << "состояние составляет :\t" << chance[i] << "\t";

}

cout << endl << endl;

for (int i = 0; i < n; i++)

{

cout << "\n Полученный доход через " << number << "тактов " << i + 1 << "состояния составляет :\t" << income[i] << "\t";

}

for (int i = 0; i < n; i++)

{

income[i] = incomehelp[i];

incomehelp[i] = 0;

}

}

}

else

{

int controlcount;

cout << "Введите количество управлений для каждого элемента" << endl;

cin >> controlcount;

double\*\*\* dohod = new double\*\* [controlcount]; // Создание трехмерного массива доходов

for (int i = 0; i < controlcount; i++) {

dohod[i] = new double\* [n]; //выделяем память на столбцы

for (int j = 0; j < n; j++) {

dohod[i][j] = new double[n]; //выделяем память на строки.

}

}

double\* incomeindex = new double[n];

double\* income = new double[n];

double\* incomehelp = new double[n];

int\* maxincomeindex = new int[n];// Индекс этой величины

double\* maxincome = new double[n];// Величина для поиска максимального одношагового дохода для каждой величины

double\*\* q = new double\* [controlcount]; // две строки в массиве

for (int count = 0; count < controlcount; count++)

q[count] = new double[n];

//double\* q = new double[n];

double\* v0 = new double[n];

double\* chance = new double[n];

double\* chancehelp = new double[n];

double\*\*\* arr = new double\*\* [controlcount]; // Создание трехмерного массива вероятностей

for (int i = 0; i < controlcount; i++) {

arr[i] = new double\* [n]; //выделяем память на столбцы

for (int j = 0; j < n; j++) {

arr[i][j] = new double[n]; //выделяем память на строки.

}

}

cout << "Сумма всех вероятностей перехода для одного состояния должна равняться единице и каждая вероятность должна быть в границах от 0 до 1" << endl;

for (int count\_control = 0; count\_control < controlcount; count\_control++)

{

for (int count\_row = 0; count\_row < n; count\_row++)

{

double sum = 0;

// while (sum != 1)

// {

sum = 0;

for (int count\_column = 0; count\_column < n; count\_column++)

{

//do

//{

cout << "\n Введите веротность при использовании \n" << count\_control+1 << " управления \nперехода из " << count\_row + 1 << " состояния в " << count\_column + 1 << " состояние" << endl;

cin >> arr[count\_control][count\_row][count\_column];

sum += arr[count\_control][count\_row][count\_column];

//} while ((arr[count\_control][count\_row][count\_column] > 1) || ((arr[count\_control][count\_row][count\_column]) < 0));

}

}

}

for (int count\_control = 0; count\_control < controlcount; count\_control++)

{

for (int count\_row = 0; count\_row < n; count\_row++)

{

for (int count\_column = 0; count\_column < n; count\_column++)

{

do

{

cout << "\n Введите полученный доход при использовании \n" << count\_control+1 << " управления \n перехода из " << count\_row + 1 << " состояния в " << count\_column + 1 << " состояние" << endl;

cin >> dohod[count\_control][count\_row][count\_column];

} while ((arr[count\_control][count\_row][count\_column] > 1) || ((arr[count\_row][count\_column]) < 0));

}

}

}

for (int count\_control = 0; count\_control < controlcount; count\_control++)

{

cout << "\nИспользование " << count\_control << " управления \n" << "Матрица вероятностей\n";

for (int count\_row = 0; count\_row < n; count\_row++)

{

cout << "\n";

for (int count\_column = 0; count\_column < n; count\_column++)

cout << arr[count\_control][count\_row][count\_column] << "\t";

}

cout << endl;

cout << "\nИспользование " << count\_control << " управления \n" << "Матрица дохода\n";

for (int count\_row = 0; count\_row < n; count\_row++)

{

cout << "\n";

for (int count\_column = 0; count\_column < n; count\_column++)

cout << dohod[count\_control][count\_row][count\_column] << "\t";

}

}

double sum = 0;

//while (sum != 1)

//{

sum = 0;

for (int i = 0; i < n; i++)

{

cout << "\n Введите вероятность начального нахождения в " << i + 1 << "-ом состоянии ";

cin >> chance[i];

chancehelp[i] = 0;

sum += chance[i];

}

//}

double sum1 = 0;

for (int count\_control = 0; count\_control < controlcount; count\_control++)

{

for (int count\_row = 0; count\_row < n; count\_row++)

{

q[count\_control][count\_row] = 0;

}

}

for (int count\_control = 0; count\_control < controlcount; count\_control++)

{

for (int count\_row = 0; count\_row < n; count\_row++)

{

for (int count\_column = 0; count\_column < n; count\_column++)

{

q[count\_control][count\_row] += arr[count\_control][count\_row][count\_column] \* dohod[count\_control][count\_row][count\_column];

cout << "count\_control " <<count\_control << "\tcount\_row " << count\_row << "\tcount\_column " <<count\_column << " q "<<q[count\_control][count\_row] << endl;

}

}

}

for (int i = 0; i < n; i++)

{

maxincome[i] = -9999;

maxincomeindex[i] = 0;

}

for (int count\_row = 0; count\_row < n; count\_row++) // Вычисляем максимальный ожидаемый одношаговый доход для каждого состояния

{

for (int count\_control = 0; count\_control < controlcount; count\_control++)

{

if (q[count\_control][count\_row] > maxincome[count\_row])

{

maxincome[count\_row] = q[count\_control][count\_row];

maxincomeindex[count\_row] = count\_control;

}

}

income[count\_row] = maxincome[count\_row];

incomeindex[count\_row] = maxincomeindex[count\_row];

}

for (int number = 1; number <= m; number++) // Цикл шагов

{

for (int count\_row = 0; count\_row < n; count\_row++)

{

incomehelp[count\_row] = maxincome[count\_row];

for (int count\_column = 0; count\_column < n; count\_column++)

{

sum1 = arr[maxincomeindex[count\_row]][count\_row][count\_column] \* chance[count\_row];

chancehelp[count\_column] += sum1;

sum1 = 0;

incomehelp[count\_row] += income[count\_column] \* arr[maxincomeindex[count\_row]][count\_row][count\_column];

}

sum1 = 0;

}

for (int i = 0; i < n; i++)

{

chance[i] = chancehelp[i];

chancehelp[i] = 0;

}

cout << endl << endl;

for (int i = 0; i < n; i++)

{

cout << "\n Вероятоность попасть через " << number << "тактов в " << i + 1 << " состояние составляет :\t" << chance[i] << endl;

}

cout << endl;

for (int i = 0; i < n; i++)

{

cout << "Полученный доход через " << number << "тактов " << i + 1 << " состояния составляет :\t" << income[i] << endl;

}

for (int i = 0; i < n; i++)

{

income[i] = incomehelp[i];

incomehelp[i] = 0;

}

}

cout << endl << endl;

for (int i = 0; i < n; i++)

{

cout << "\nДля состояния " << i + 1 << " наибольший одношаговый доход " << maxincome[i] << " при условии выбора управления: " << maxincomeindex[i]+1 << endl;

}

}

system("pause");

}