

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

02.03.02 Фундаментальная информатика
и информационные технологии
Профиль: Системное программное
обеспечение
(очная форма обучения)

ОТЧЕТ ПО ПРОИЗВОДСТВЕННОЙ ПРАКТИКЕ

в/на ФИВТ СибГУТИ
(наименование профильной организации/структурного подразделения СибГУТИ)

«ГЕНЕРАТОР И АНАЛИЗАТОР ЦЕПИ МАРКОВА»

Выполнил:

студент ФИВТ

гр. ИС-841

«29» мая 2021г.

_____/_____
(подпись)

Проверил:

Руководитель от СибГУТИ

«29» мая 2021г.

_____/_____
(подпись)

Новосибирск 2021

План-график проведения _____ производственной _____ практики

Вид практики

Горюнова Диана Николаевна

Фамилия Имя Отчество студента

факультета Информатики и вычислительной техники, 3 курса, гр. ИС-841

Направление: 02.03.02 Фундаментальная информатика и информационные технологии

Профиль: Системное программное обеспечение

Место прохождения практики: ФИВТ СибГУТИ

Объем практики: **216/6** часов/ЗЕ

Вид практики: ***Производственная***

Тип практики: ***Производственная практика - Производственная практика по получению профессиональных умений и опыта профессиональной деятельности***

Срок практики с «01» февраля 2021 г.

по «29» мая 2021 г.

Содержание практики*:

Наименование видов деятельности	Дата (начало – окончание)
1. Общее ознакомление со структурным подразделением предприятия, вводный инструктаж по технике безопасности	01.02.2021–13.02.2021
2. Выдача задания на практику, деление студентов на группы (если необходимо), определение конкретной индивидуальной темы, формирование плана работ	15.02.2021–20.02.2021
3. Работа с библиотечными фондами структурного подразделения или предприятия, сбор и анализ материалов по теме практики	22.02.2021–20.03.2021
4. Выполнение работ в соответствии с составленным планом: – Ознакомление с цепью Маркова – Реализация генератора цепи Маркова – Реализация анализатора цепи Маркова	22.03.2021 – 22.05.2021
5. Анализ полученных результатов и произведенной работы Составление отчета по практике, защита отчета	24.05.2021–29.05.2021

*В соответствии с программой практики

Руководитель от СибГУТИ

«_____» _____ 2021г.

_____/_____/

(подпись)

ЗАДАНИЕ НА ПРАКТИКУ

Реализация генератора и анализатора цепи Маркова.

ВВЕДЕНИЕ

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

Процесс в каждый момент времени находится в одном из n состояний. При этом, если он находится в состоянии с номером i , то он перейдет в состояние j с вероятностью p_{ij} .

Матрицу $P = \|p_{ij}\|$ называют матрицей переходов.

На такую матрицу переходов накладываются условия:

1. $p_{ij} \geq 0$
2. $\forall_i \sum_j p_{ij} = 1$

В ходе прохождения практики был реализован генератор матрицы переходов случайными значениями, который не нарушает условия.

```
vladxyk@DESKTOP-922BP0L:/mnt/c/Users/Diana/Desktop/п.практика$ ./markov
Enter the number of events: 2

Enter the transition matrix or get a random transition matrix (e/g): g

Transition matrix:
0.15 0.85
0.45 0.55
```

Также реализована возможность ввести матрицу переходов вручную.

```
vladxyk@DESKTOP-922BP0L:/mnt/c/Users/Diana/Desktop/п.практика$ ./markov
Enter the number of events: 2

Enter the transition matrix or get a random transition matrix (e/g): e
0.2
0.8
0.7
0.3

Transition matrix:
0.2 0.8
0.7 0.3
```

В случае ввода матрицы, нарушающей условия, будет выведена ошибка.

```
vladxyk@DESKTOP-922BP0L:/mnt/c/Users/Diana/Desktop/п.практика$ ./markov
Enter the number of events: 2

Enter the transition matrix or get a random transition matrix (e/g): e
0.9
2
Error: sum of the elements of the row must be equal to 1
```

Реализован анализатор, который позволяет:

1. получить вероятность перехода из одного состояния в другое.

Для этого необходимо ввести вектор, описывающий исходное распределение вероятностей (начальный вектор состояния).

```
vladxyk@DESKTOP-922BP0L:/mnt/c/Users/Diana/Desktop/п.практика$ ./markov
Enter the number of events: 2

Enter the transition matrix or get a random transition matrix (e/g): g

Transition matrix:
0.39 0.61
0.62 0.38

If you want to get the probability from one state to another, enter 1
If you want to get this probability in a few steps, enter 2
1

Enter initial state vector: 0.2 0.8

Result vector:
0.566 0.428
```

Сумма элементов этого вектора должна быть равна 1, иначе ошибка.

```

vladxyk@DESKTOP-922BP0L:/mnt/c/Users/Diana/Desktop/п.практика$ ./markov
Enter the number of events: 2

Enter the transition matrix or get a random transition matrix (e/g): g

Transition matrix:
0.55 0.45
0.04 0.96

If you want to get the probability from one state to another, enter 1
If you want to get this probability in a few steps, enter 2
1

Enter initial state vector: 2 3
Error: sum of the elements must be equal to 1

```

2. получить вероятность перехода из одного состояния в другое за несколько шагов.

Для этого необходимо ввести количество шагов.

```

vladxyk@DESKTOP-922BP0L:/mnt/c/Users/Diana/Desktop/п.практика$ ./markov
Enter the number of events: 2

Enter the transition matrix or get a random transition matrix (e/g): g

Transition matrix:
0.07 0.93
0.79 0.21

If you want to get the probability from one state to another, enter 1
If you want to get this probability in a few steps, enter 2
2

Enter the number of steps: 2

Transition matrix in 2 steps
0.7396 0.2604
0.2212 0.7788

```

ЗАКЛЮЧЕНИЕ

В результате прохождения практики изучена цепь Маркова, выполнена реализация генератора и анализатора марковской цепи. Получена вероятность перехода из одного состояния в другое при начальном векторе состоянии $V = \begin{pmatrix} 0.2 \\ 0.8 \end{pmatrix}$ и матрице переходов $M = \begin{pmatrix} 0.07 & 0.93 \\ 0.79 & 0.21 \end{pmatrix}$

```
Transition matrix:
```

```
0.07 0.93
```

```
0.79 0.21
```

```
If you want to get the probability from one state to another, enter 1
```

```
If you want to get this probability in a few steps, enter 2
```

```
1
```

```
Enter initial state vector: 0.2 0.8
```

```
Result vector:
```

```
0.758 0.326
```

Эта вероятность равна $\begin{pmatrix} 0.758 \\ 0.326 \end{pmatrix}$

Также получена вероятность перехода из одного состояния в другое за 2 шага.

```
Transition matrix:
```

```
0.07 0.93
```

```
0.79 0.21
```

```
If you want to get the probability from one state to another, enter 1
```

```
If you want to get this probability in a few steps, enter 2
```

```
2
```

```
Enter the number of steps: 2
```

```
Transition matrix in 2 steps
```

```
0.7396 0.2604
```

```
0.2212 0.7788
```

Она равна $\begin{pmatrix} 0.7396 & 0.2604 \\ 0.2212 & 0.7788 \end{pmatrix}$

Цепь Маркова в виде графа:



СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Романовский И. В. Дискретный анализ. – СПб.: Невский Диалект; БХВ-Петербург, 2003. – 338 с.
2. Чернова Н.И. Теория вероятностей: Учебное пособие. – СибГУТИ. – Новосибирск, 2009. – 128 с.

ПРИЛОЖЕНИЯ

```
#include <iostream>
#include <string>
#include "string.h"

using namespace std;

void mult(float **a, int n, float **res){

    for (int i = 0; i < n; i++){
        res[i] = new float[n];
        for (int j = 0; j < n; j++){
            res[i][j] = 0;
            for (int k = 0; k < n; k++){
                res[i][j] = res[i][j] + a[i][k] * a[k][j];
            }
        }
    }

    for(int i = 0 ; i < n ; i++){
        for(int j = 0 ; j < n; j++){
            a[i][j]=res[i][j];
        }
    }
}

/*
    cout << "Mult matrix" << endl;
    for (int i = 0; i < n; i++){
        for (int j = 0; j < n; j++){
            cout << res[i][j] << " ";
        }
        cout << endl;
    }
*/

}

int main(){

    cout << "Enter the number of events: ";
    int n;
    cin >> n;
    cout << endl;

    float** matrix = new float*[n];
    srand(time(NULL));

    float sum;
    cout << "Enter the transition matrix or get a random transition matrix (e/g): ";
    string c;
    cin >> c;
```

```

    if(c == "e"){
        for(int i = 0; i < n; i++){
            matrix[i] = new float[n];
            sum = 0;
            for(int j = 0; j < n; j++){
                cin >> matrix[i][j];
                sum = sum + matrix[i][j];
            }
            if(sum < 1 || sum > 1){
                cout << "Error: sum of the ele-
ments of the row must be equal to 1" << endl;
                return -1;
            }
        }
    }

    else if(c == "g"){
        for(int i = 0; i < n; i++){
            matrix[i] = new float[n];
            sum = 0;
            for(int j = 0; j < n; j++){
                matrix[i][j] = rand()%101;
                matrix[i][j] = matrix[i][j] / 100;
                if(i != j){
                    sum = sum + matrix[i][j];
                }
            }
            if(sum < 1){
                matrix[i][i] = 1 - sum;
            }
            else if(sum == 1){
                matrix[i][i] = 0;
            }
            else if(sum > 1){
                for(int j = 0; j < n; j++){
                    matrix[i][j] = 0;
                }
                int j = rand()%n;
                matrix[i][j] = 1;
            }
        }
    }
}

else{
    cout << "Error: enter e/g" << endl;
    return -1;
}

cout << endl << "Transition matrix: " << endl;
for(int i = 0; i < n; ++i){
    for(int j = 0; j < n; ++j){
        cout << matrix[i][j] << " ";
    }
}

```

```

    }
    cout << endl;
}

/*
float k;
k = matrix[0][0];
int events;
int t = 0;

while(t<10){
    for(int i = 0; i < n; i++){
        cout << "verhina " << i + 1 << endl;
        for(int j = 0; j < n; j++){
            events = rand()%n;
            k = matrix[i][events];
            cout << "events = " << events << " k = " << k << " ";
        }
        cout << endl;
    }
    cout << endl;
    t++;
}

*/
cout << endl;
cout << "If you want to get the probability from one state to another, enter 1" << endl;
cout << "If you want to get this probability in a few steps, enter 2" << endl;

int choice;
cin >> choice;

if(choice == 1){
    float *vector = new float[n];
    float tmp;
    float *res_vector = new float[n];
    sum = 0;

    cout << endl << "Enter initial state vector: ";
    for(int i = 0; i < n; i++){
        cin >> vector[i];
        sum = sum + vector[i];
    }

    if(sum < 1 || sum > 1){
        cout << "Error: sum of the elements must be equal to 1" << endl;
        return -1;
    }
}

/*
cout << endl << "Initial state vector: " << endl;
for(int i = 0; i < n; i++){

```

```

        cout << vector[i] << " ";
    }
*/
    for(int i = 0; i < n; i++){
        tmp = 0;
        for(int j = 0; j < n; j++){
            tmp = tmp + matrix[i][j] * vector[j];
        }
        res_vector[i] = tmp;
    }

    cout << endl << "Result vector: " << endl;
    for(int i = 0; i < n; i++){
        cout << res_vector[i] << " ";
    }
    cout << endl;

    delete[] vector;
    delete[] res_vector;
}
else if(choice == 2){
    cout << endl << "Enter the number of steps: ";
    int steps;
    cin >> steps;
    int tsteps = steps;
    float **res = new float*[n];

    while(tsteps > 1){
        mult(matrix, n, res);
        tsteps--;
    }

    cout << endl << "Transition matrix in "<< steps << " steps" << endl;
    for(int i = 0; i < n; ++i){
        for(int j = 0; j < n; ++j){
            cout << res[i][j] << " ";
        }
        cout << endl;
    }
    delete[] res;
}

cout << endl;

delete[] matrix;
return 0;
}

```

Горюнова Диана Николаевна

[illegible]

Уровень освоения компетенций

Горюнова Диана Николаевна

(ФИО студента)

Компетенции	Уровень сформированности компетенций
<i>ОК-7 – способностью к самоорганизации и самообразованию</i>	
<i>ПК-1 - способностью собирать, обрабатывать и интерпретировать данные современных научных исследований, необходимые для формирования выводов по соответствующим научным исследованиям</i>	
<i>ПК-4 - способностью решать задачи профессиональной деятельности в составе научно-исследовательского и производственного коллектива</i>	

отметка о зачете _____

Руководитель практики от СибГУТИ:

Должность руководителя

подпись

ФИО руководителя

«___» _____ 2021 г.