

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В. Г. ШУХОВА»
(БГТУ им. В.Г. Шухова)

Кафедра программного обеспечения вычислительной техники и автоматизированных
систем

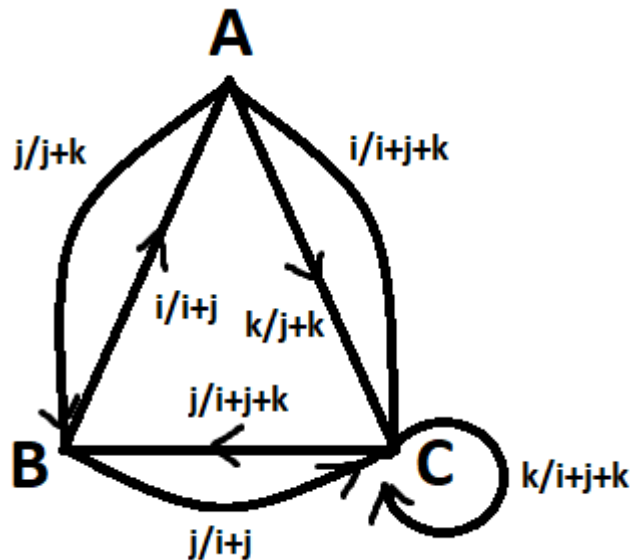
Лабораторная работа №3
по дисциплине «Теория надежности»
тема: «Муха в треугольнике»

Выполнил: ст. группы ВТ-32
Воскобойников И. С.
Проверил: Кабелянц П. С.

Белгород 2021 г.

Задание:

Имеется треугольник с вершинами A, B, C. В начальный момент времени муха находится в вершине A. Каждую секунду муха перелетает в другую вершину или остается на месте. Ее поведение задается графом марковского процесса (смотри приложенный рисунок). 1) Необходимо определить среднее время, через которое муха вернется в вершину A. 2) Написать программу, которая имитирует поведение мухи и выводит среднее количество переходов до первого возвращения в точку A. Сравните результаты.



Выполнение:

	A	B	C
A	0	1/11	1/21
B	10/20	0	10/21
C	10/20	10/11	10/21

$i = 1$
 $J = 10$
 $K = 10$

$$\begin{cases} y/11 + z/21 = x \\ x/2 + z \cdot 10/21 = y \\ x/2 + y \cdot 10/11 + z \cdot 10/21 = z \end{cases}$$

$$x + y + z = 1$$

$$x = \frac{1}{17}$$

$$y = \frac{11}{34}$$

$$z = \frac{21}{34}$$

$x = 17$ среднее время до взрыва
в A

```
from random import random  
i = 1  
j = 10  
k = 10
```

```

def step(state):
    random_value = random()

    if state == "A":
        if random_value < (10. / 20.):
            state = "B"
        else:
            state = "C"

        return state

    if state == "B":
        if random_value < (1. / 11.):
            state = "A"
        else:
            state = "C"

        return state

    if state == "C":
        if random_value < (1. / 21.):
            state = "A"
        elif random_value < (10. / 21.):
            state = "B"
        else:
            state = "C"

        return state

# возвращает время до возврата в начальное положение

def run():
    state = 'A'
    count_step = 0

    while True:
        count_step += 1
        state=step(state)

        if (state == 'A'):
            return count_step

n = 100000

generate_result = [run() for _ in range(n)]
avg_ret_A = sum(generate_result)/len(generate_result)
print(f"Эмпирическое среднее время до возврата в A = {avg_ret_A}")
print("Теоретическое среднее время до возврата в A = 17")
print(f"Разница между полученными данными и посчитанными = {avg_ret_A - 17}")
print("Разница не большая значит все расчеты были выполнены верно.")

```

```

D:\Users\500a5\PycharmProjects\ti_myha\venv\Scripts\python.exe D:/Users/500a5/PycharmProjects/ti_myha/main.py
Эмпирическое среднее время до возврата в A = 17.15073
Теоретическое среднее время до возврата в A = 17
Разница между полученными данными и посчитанными = 0.15072999999999936
Разница не большая значит все расчеты были выполнены верно.

```

```
Process finished with exit code 0
```