**Министерство науки и высшего образования Российской Федерации**

**Федеральное государственное бюджетное образовательное учреждение высшего образования**

**«Российский химико-технологический университет имени Д.И. Менделеева» Кафедра информационных компьютерных технологий**

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 10**

Выполнил студент группы ............ КС-36 ..................................Тернолуцкий Виктор Александрович

Ссылка на репозиторий: .... ... <https://github.com/MUCTR-IKT-CPP/VATernolutski_36>

Приняли: Крашенинников Роман Сергеевич

Дата сдачи: ................................................................................................................................. 07.05.2025

**Оглавление**

[Описание задачи 2](#_Toc4241)

[Описание метода/модели 2](#_Toc4242)

[Выполнение задачи 2](#_Toc4243)

[Заключение 5](#_Toc4244)

# Описание задачи.

В рамках лабораторной работы необходимо решить 1 из ниже приведенных задач.

1. Метод Монте Карло

Пусть 2 игрока A и B играют в следующую игру: у игроков есть монетка, где 0 это орел, а 1 это решка, каждый игрок выбирает комбинацию из 3 цифр 0/1(например 001), затем подбрасывается монетка и результат записывается в длинную строку, побеждает тот чъя комбинация будет на конце итоговой строки. Например: A - 001, B - 100, R - 01010101010100, победил B Необходимо смоделировать игру двух игроков и: Построить таблицу вероятности выигрыша одной из комбинаций, так что бы столбцы соответствовали игроку А, а строки игроку Б, а на пересечении была бы вероятность побед игрока А над игроком Б, при выбранных ими комбинациях. Так же по результатом всех попыток определить суммарный средний шанс выигрыша игрока А и игрока Б вне зависимости от выбранных комбинаций Проанализировать полученные результаты

# Описание метода/модели.

В данной задаче мы моделируем игру двух игроков A и B, каждый из которых заранее выбирает свою трёхсимвольную последовательность из {0,1}. После этого бесконечно подбрасывается честная монетка, и выпавшие результаты накапливаются в общую строку. Побеждает тот игрок, чей шаблон впервые появляется как суффикс текущей строки.

Для оценки вероятностей выигрыша каждой из 8×8 пар комбинаций мы применяем метод Монте-Карло, заключающийся в статистическом моделировании большого числа независимых «партий» и вычислении относительных частот побед.

# Выполнение задачи.

import random

import itertools

*# Все возможные комбинации из трёх цифр 0 и 1*

combinations = list(itertools.product([0, 1], repeat=3))

*# Функция для симуляции одной игры*

**def** simulate\_game(A, B, num\_flips=100):

*# Генерируем случайную строку из 0 и 1*

    flips = [random.randint(0, 1) for \_ in range(num\_flips)]

*# Преобразуем комбинации в строки*

    A\_str = ''.join(map(str, A))

    B\_str = ''.join(map(str, B))

*# Проверяем последние три символа*

    for i in range(2, num\_flips):

        current = ''.join(map(str, flips[i-2:i+1]))

        if current == A\_str:

            return 'A'

        elif current == B\_str:

            return 'B'

    return 'None'  *# Редкий случай, если ни одна комбинация не найдена*

*# Оценка вероятности победы A над B*

**def** estimate\_probability(A, B, num\_simulations=10000):

    wins\_A = 0

    for \_ in range(num\_simulations):

        winner = simulate\_game(A, B)

        if winner == 'A':

            wins\_A += 1

    return wins\_A / num\_simulations

*# Построение таблицы вероятностей*

**def** build\_probability\_table():

    table = {}

    for B in combinations:

        table[B] = {}

        for A in combinations:

            prob = estimate\_probability(A, B)

            table[B][A] = prob

    return table

*# Вывод таблицы*

**def** print\_table(table):

    print("Таблица вероятностей победы игрока A:")

    print("B \\ A", end='\t')

    for A in combinations:

        print(''.join(map(str, A)), end='\t')

    print()

    for B in combinations:

        print(''.join(map(str, B)), end='\t')

        for A in combinations:

            print(**f**"{table[B][A]**:.4f**}", end='\t')

        print()

*# Средний шанс выигрыша*

**def** calculate\_average\_probabilities(table):

    total\_prob\_A = 0

    num\_pairs = 0

    for B in table:

        for A in table[B]:

            total\_prob\_A += table[B][A]

            num\_pairs += 1

    avg\_prob\_A = total\_prob\_A / num\_pairs

    avg\_prob\_B = 1 - avg\_prob\_A  *# В каждой игре кто-то побеждает*

    return avg\_prob\_A, avg\_prob\_B

*# Основная функция*

**def** main():

    table = build\_probability\_table()

    print\_table(table)

    avg\_prob\_A, avg\_prob\_B = calculate\_average\_probabilities(table)

    print(**f**"\nСредний шанс выигрыша игрока A: {avg\_prob\_A**:.4f**}")

    print(**f**"Средний шанс выигрыша игрока B: {avg\_prob\_B**:.4f**}")

if \_\_name\_\_ == "\_\_main\_\_":

    main()

Результаты:

Таблица вероятностей победы игрока A:

B \ A 000 001 010 011 100 101 110 111

000 0.9997 0.5055 0.5878 0.6000 0.8741 0.5816 0.7023 0.4972

001 0.4901 1.0000 0.3327 0.3381 0.7469 0.3748 0.4945 0.3074

010 0.4046 0.6681 1.0000 0.4970 0.5027 0.5000 0.6307 0.4178

011 0.4009 0.6656 0.5068 1.0000 0.4993 0.4977 0.2541 0.1299

100 0.1267 0.2558 0.5000 0.5034 1.0000 0.4987 0.6629 0.4066

101 0.4086 0.6251 0.4966 0.5009 0.4973 1.0000 0.6650 0.3991

110 0.3021 0.5050 0.3671 0.7532 0.3375 0.3267 1.0000 0.4966

111 0.4979 0.6978 0.5891 0.8769 0.5980 0.5991 0.5001 0.9999

Средний шанс выигрыша игрока A: 0.5625

Средний шанс выигрыша игрока B: 0.4375

# Заключение.

Результат теста показал, что задача решена верно.

Источник:

<https://acm.timus.ru/problem.aspx?space=1&num=1005>