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-30 Джотян Давид Араикович

Ссылка на репозиторий: https://github.com/MUCTR-IKT-CPP/DzhotyanDA\_30/tree/main/lab10\_algo

Приняли: Пысин Максим Дмитриевич

Краснов Дмитрий Олегович

Лобанов Алексей Владимирович

Крашенинников Роман Сергеевич

Дата сдачи: 05.05.2025

Оглавление

[Описание задачи. 2](#_Toc63548272)

[Описание метода/модели. 2](#_Toc63548273)

[Выполнение задачи. 2](#_Toc63548274)

[Заключение. 2](#_Toc63548275)

# Описание задачи.

В рамках лабораторной работы нужно решить задачу с помощью метода 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 результатам всех попыток определить суммарный средний шанс выигрыша игрока А и игрока Б вне зависимости от выбранных комбинаций.

# Описание метода/модели.

Метод Монте-Карло — это эвристический алгоритм, основанный на **случайном (стохастическом)** моделировании процессов. Он применяется для приближённого решения задач, которые трудно или невозможно решить аналитически.

**Суть метода**

Метод Монте-Карло использует **генерацию большого количества случайных данных** для оценки результата. Вместо того чтобы точно вычислять, алгоритм «пробует» возможные варианты и **делает статистический вывод**.

**Общие шаги метода**

1. Определить область поиска.
2. Для i = 1 до N:
   1. Сгенерировать случайное решение.
   2. Оценить "качество" или результат этого решения (например, входит ли точка в область, или каков выигрыш).
   3. Накопить данные (сумму, счёт, частоту и т.д.).
3. После N итераций — усреднить/обработать результаты.
4. Вернуть приближённый ответ.

# Выполнение задачи.

Файл Program.cs

using System;

namespace lab10\_algo

{

class Program

{

static void Main()

{

const int GAMES\_COUNT = 10000;

string[] allCombinations = ["000", "001", "010", "011", "100", "101", "110", "111"];

double[,] winTable = new double[allCombinations.Length, allCombinations.Length];

Random rnd = new();

double totalA = 0;

double totalB = 0;

int totalPairs = allCombinations.Length \* allCombinations.Length;

for (int i = 0; i < allCombinations.Length; i++)

{

for (int j = 0; j < allCombinations.Length; j++)

{

string combinationA = allCombinations[i];

string combinationB = allCombinations[j];

int winsA = 0;

for (int game = 0; game < GAMES\_COUNT; game++)

{

string sequence = "";

while (true)

{

sequence += FlipCoin(rnd);

if (sequence.Length >= 3)

{

string last3 = sequence.Substring(sequence.Length - 3, 3);

if (last3 == combinationA)

{

if (last3 != combinationB)

winsA++;

break;

}

else if (last3 == combinationB)

{

break;

}

}

}

}

double winProbability = (double)winsA / GAMES\_COUNT;

winTable[i, j] = winProbability;

totalA += winProbability;

totalB += (1.0 - winProbability);

}

}

Console.WriteLine("Вероятности выигрыша Игрока A против Игрока B");

Console.Write(" ");

foreach (var col in allCombinations)

Console.Write($"{col} ");

Console.WriteLine();

for (int i = 0; i < allCombinations.Length; i++)

{

Console.Write($"{allCombinations[i]} | ");

for (int j = 0; j < allCombinations.Length; j++)

{

Console.Write($"{winTable[i, j]:0.000} ");

}

Console.WriteLine();

}

double avgA = totalA / totalPairs;

double avgB = totalB / totalPairs;

Console.WriteLine($"\nСредний шанс победы игрока A: {avgA:0.000}");

Console.WriteLine($"Средний шанс победы игрока B: {avgB:0.000}");

}

static string FlipCoin(Random rnd)

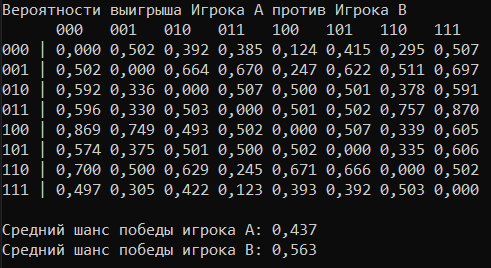
{

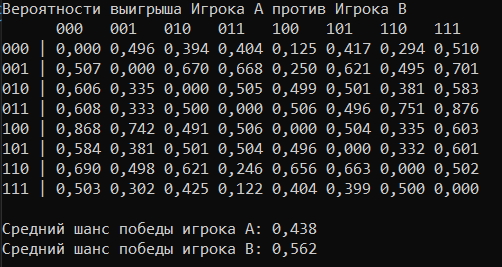
return rnd.Next(2) == 0 ? "0" : "1";

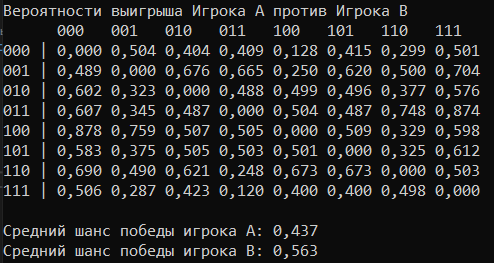
}

}

}







# Заключение.

Метод Монте – Карло показывает, что даже при нескольких независимых запусках он выдает почти идентичные результаты, что подтверждает его применение в сложных ситуациях, которые трудны для понимания и решения с помощью аналитических методов, как например в этой задаче.