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 Акулинин Андрей Игоревич

Ссылка на репозиторий: <https://github.com/MUCTR-IKT-CPP/AkulininAI_36/tree/main/algorithms>

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.

Два игрока (A и B) выбирают комбинации из трех битов (например, 001 или 110). Затем многократно подбрасывается монетка (0 — орёл, 1 — решка), и результаты записываются в строку. Побеждает тот игрок, чья комбинация первой появится в конце строки.

**Требуется:**

1. Построить таблицу вероятностей выигрыша игрока A против всех возможных комбинаций игрока B.
2. Определить средний шанс выигрыша игроков A и B по всем возможным парам комбинаций.

# Описание метода/модели.

**Метод Монте-Карло**

* Используется для оценки вероятностей методом статистического моделирования.
* Проводится большое количество симуляций (10 000), чтобы приблизить вероятность выигрыша.

**Алгоритм**

1. **Генерация всех комбинаций** (000, 001, ..., 111).
2. **Симуляция игры:**
   * Последовательно генерируются случайные биты (0/1).
   * Проверяются последние 3 бита на совпадение с комбинациями A или B.
3. **Расчет вероятности:**
   * Для каждой пары комбинаций (A, B) проводится N симуляций.
   * Вероятность выигрыша A = (число побед A) / N.

**Сложность**

* Время: O(N × M), где N — число симуляций, M — средняя длина игры.
* Память: O(1) (хранятся только счетчики).

**Преимущества:**

* Простота реализации.
* Точность растет с увеличением числа симуляций.

**Недостатки:**

* Требует большого числа итераций для высокой точности.
* Зависит от качества генератора случайных чисел.

# Выполнение задачи.

**Язык и реализация**

* Язык: **C++** (быстрый, удобные структуры данных).
* Использованы:
  + std::unordered\_map — хранение вероятностей.
  + std::mt19937 — генератор случайных чисел.
  + std::vector — перебор всех комбинаций.

**Организация программы**

1. **Генерация комбинаций** (000–111).
2. **Симуляция игр** для всех пар (A, B).
3. **Построение таблицы** вероятностей.
4. **Расчет средних вероятностей** для A и B.

**Тестирование и результаты**

* Число симуляций: **10 000**.
* Вывод:
  + Таблица 8×8 (вероятности выигрыша A).
  + Средний шанс A: **~50%**, B: **~50%** (симметричность задачи).

# Заключение.

1. Метод Монте-Карло эффективен для оценки вероятностей в подобных задачах.
2. Результаты подтверждают симметричность игры (в среднем шансы равны).
3. Основная сложность — выбор достаточного числа симуляций для точности.
4. **Вывод:** алгоритм успешно решает задачу, но требует оптимизации при увеличении длины комбинаций.