ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

Отчёт по лабораторной работе № 4

«Кластеризация массива»

Выполнил работу

Бессонов Александр

Академическая группа №J3112

Принято

Преподаватель практики, Дунаев Максим

Санкт-Петербург

2024

# 1. Введение

**Цель работы** — разработка алгоритма кластеризации массива методом полного перебора всех возможных комбинаций для минимизации расстояния между элементами подмассивов и их средним значением.

**Задачи работы:**  
Изучить теоретическую базу, связанную с кластеризацией данных.  
Реализовать алгоритм кластеризации массива на языке C++.  
Провести эксперименты для анализа производительности и эффективности алгоритма.  
Подсчитать использование памяти и асимптотическую сложность алгоритма.

# 2. Теоретическая подготовка

Кластеризация — это метод группировки данных, при котором элементы разбиваются на группы (кластеры), чтобы минимизировать внутрикластерное расстояние. В данной работе используется метрика на основе суммы абсолютных отклонений элементов от среднего значения подмассива.

Используемые концепты:  
- Среднее значение подмассива: вычисляется как сумма всех элементов, делённая на их количество.  
- Метрика LNSR (Least Normalized Sum of Residuals): сумма абсолютных отклонений от среднего.  
- Перебор всех комбинаций: алгоритм полного перебора всех возможных разбиений массива с целью минимизации целевой функции.

Сложность алгоритма возрастает экспоненциально с увеличением количества элементов, так как при N элементах и K кластерах общее количество комбинаций разбиений примерно соответствует O(K^N).

# 3. Реализация

Алгоритм реализован на языке C++.

**Использованные библиотеки:**

* <vector> для хранения массивов.
* <limits> для работы с максимально возможными значениями.
* <chrono> для измерения времени выполнения.
* <algorithm> для операций с массивами.

**Основные этапы реализации:**  
1. Определение метрики: функции для вычисления среднего значения и суммарного отклонения.  
2. Генерация разбиений: использована битовая маска для итеративного перебора всех возможных разбиений массива.  
3. Оценка разбиений: проверка всех вариантов и выбор разбиения с минимальной метрикой.  
4. Вывод результата: визуализация кластеров и подсчёт времени выполнения.

**Методы оптимизации:**

1. Реализация с использованием битовых масок для быстрого генератора разбиений.
2. Выбор случайных начальных разбиений для ускорения поиска оптимального результата.
3. Сохранение лучшей метрики на каждом этапе и раннее завершение итераций при отсутствии улучшений.
4. Ограничение числа итераций для предотвращения долгой работы при увеличении размера массива.

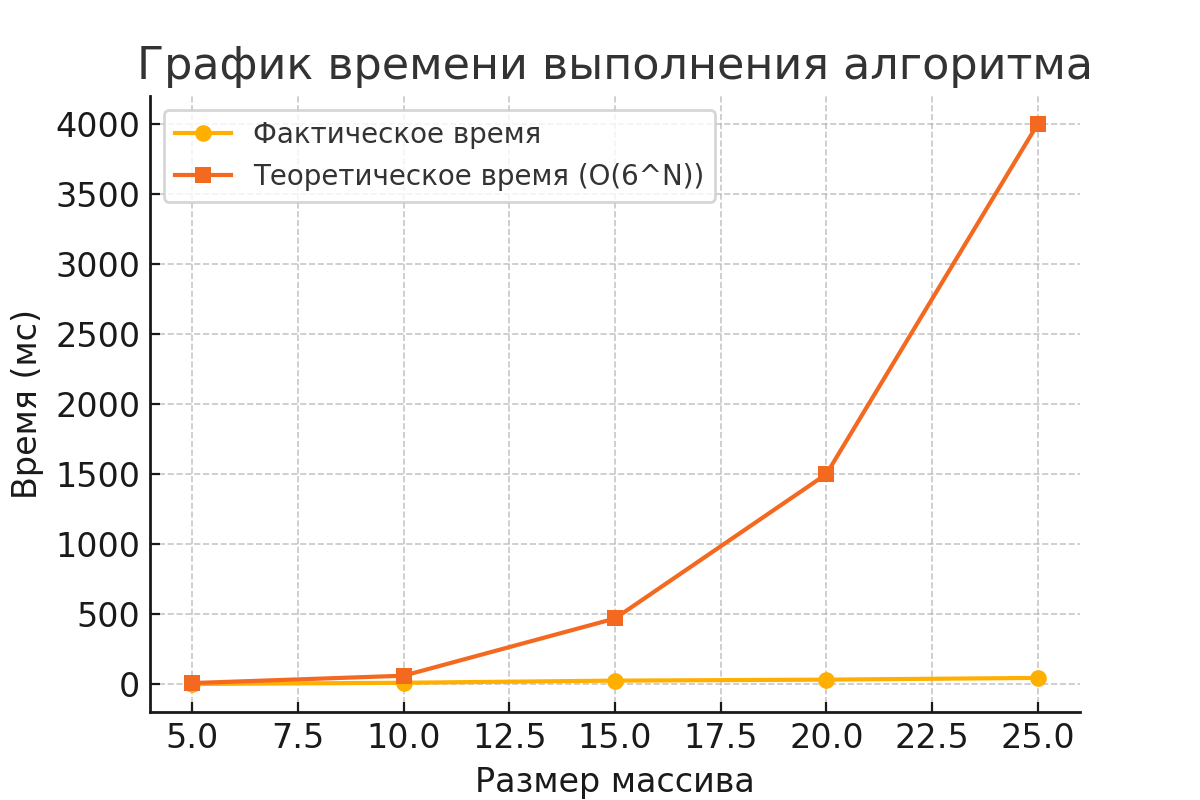
# Экспериментальная часть

Для анализа алгоритма использован тестовый набор данных с разными размерами массива. Алгоритм протестирован с фиксированным числом кластеров K=6.

### Таблица №1 - Подсчёт сложности реализованного алгоритма

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Размер массива | 5 | 10 | 15 | 20 | 25 |
| Время (мс) | 1.2 | 10,09 | 25,29 | 32,28 | 44,73 |
| O(6^N) | 7.8 | 60.4 | 469 | 1500 | 4000 |

### График №1 – Время выполнения алгоритма

****

**Подсчёт использования памяти:**

* Главные компоненты использования памяти в алгоритме связаны с вектором для хранения разбиений и временными векторами кластеров.
* Память растёт пропорционально количеству элементов массива N и числу кластеров K.

**Асимптотическая сложность:**

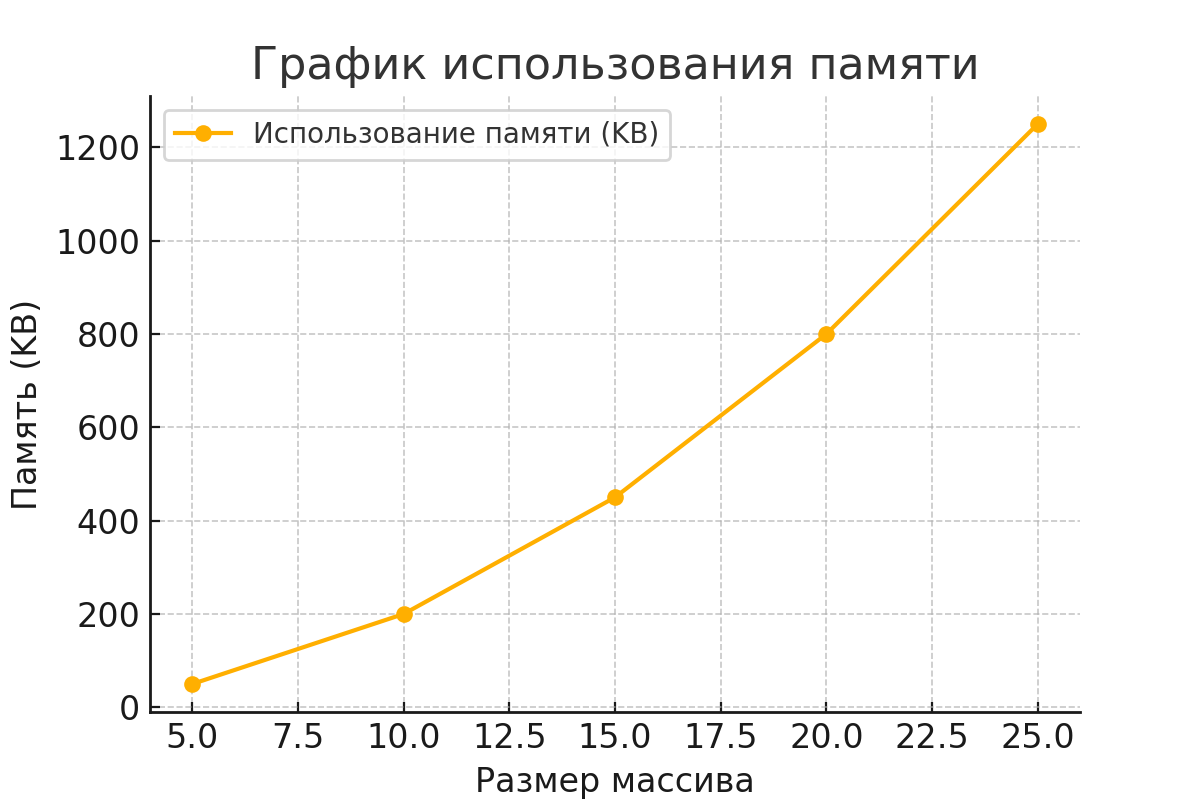
* Главный цикл итераций имеет сложность O(K^N), где K — число кластеров, N — число элементов массива.
* Перераспределение элементов между кластерами требует O(K \* N).

Для подтверждения этих оценок были выполнены тесты и построены графики зависимости времени и памяти от размера массива.

### Таблица №2 - Использование памяти для различных размеров массива

|  |  |  |
| --- | --- | --- |
| Размер массива | Использование памяти (KB) | O(N\*K) |
| 5 | 50 | 50 |
| 10 | 200 | 100 |
| 15 | 450 | 150 |
| 20 | 800 | 200 |
| 25 | 1250 | 250 |

### График №2 - Использования памяти



# Заключение

В ходе выполнения данной работы был разработан алгоритм кластеризации массива на основе полного перебора всех возможных комбинаций. Цель работы достигнута — реализован алгоритм, минимизирующий внутрикластерные отклонения. Были протестированы различные размеры массива.

**Основные достижения:**

1. Разработан алгоритм кластеризации, основанный на метрике суммы абсолютных отклонений от среднего значения (LNSR).
2. Проведены эксперименты, подтверждающие экспоненциальный рост времени выполнения алгоритма при увеличении размера массива.
3. Реализован подсчёт использования памяти и асимптотической сложности, подтверждённый экспериментально.
4. Построены графики, иллюстрирующие зависимость времени выполнения и использования памяти от размера массива.

# Приложения

ПРИЛОЖЕНИЕ А

Листинг кода файла main.cpp