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 № 6

Выполнил студент группы КС-36 Меркушов Даниил Игоревич

Ссылка на репозиторий: https://github.com/VooDooCry/Algorithms\_KS36.git

Приняли: Пысин Максим Дмитриевич

Краснов Дмитрий Олегович

Лобанов Алексей Владимирович

Крашенинников Роман Сергеевич

Дата сдачи: 07.04.2025

Оглавление

[Описание задачи. 2](#_Toc63548272)

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

[Выполнение задачи. 2](#_Toc63548274)

[Заключение. 2](#_Toc63548275)

# Описание задачи.

В рамках лабораторной работы необходимо изучить и реализовать бинарное дерево поиска и его самобалансирующийся вариант в лице AVL дерева.

# Структура.

Языки программирования: **Golang, Python**

**1. Структуры данных**

BST (Binary Search Tree):

Реализован через структуры BSTNode (узел) и BST (дерево).

Методы:

Insert(value) — вставка в цикле без балансировки.

Search(value) — поиск в цикле.

Delete(value) — удаление с обработкой трех случаев (лист, один потомок, два потомка).

AVL-дерево:

Реализован через структуры AVLNode (узел с высотой) и AVLTree.

Методы:

Insert(value) и Delete(value) — рекурсивные методы с балансировкой.

Балансировка через повороты (rotateRight, rotateLeft).

Автоматическое поддержание высоты узлов (updateHeight).

**2. Генерация данных**

generateArray(n, sorted):

Создает массив размером n.

Если sorted = false, массив перемешивается случайно.

**3. Поиск в массиве**

linearSearch(arr, value) — линейный поиск.

binarySearch(arr, value) — бинарный поиск (только для отсортированных массивов).

**4. Измерение производительности**

Структура Result

**5. Логика main**

Цикл по сериям:

Для series от 0 до 10:

Вычисляет n = 2^(10 + series) (начинается с 1024, заканчивается 524288).

Запускает 20 циклов (10 для случайных данных, 10 для отсортированных).

Операции для каждого цикла:

Вставка:

Замер времени заполнения BST и AVL.

Поиск:

1000 случайных поисковых запросов.

Для массива используется бинарный поиск (если отсортирован) или линейный.

Удаление:

Удаление 1000 случайных элементов.

Усреднение результатов:

Для каждого типа данных (random/sorted) усредняются результаты 10 запусков.

Сохранение в CSV:

Результаты записываются в файл results.csv в наносекундах.

# Заключение.

**Анализ сложности:**

Балансировка AVL: O (log n)

BST без балансировки: Может выражать линейную структуру, что ухудшает производительность до O (n)

Поиск – O (n)

Вставка, удаление – O (1)

:

***Вывод:***



