**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра САПР.**

ОТЧЁТ

**по лабораторной работе №1**

**по дисциплине «Алгоритмы и структуры данных»**

Тема: «Реализация и сравнение эффективности красно-черного, бинарного и авл-деревьев».

|  |  |  |
| --- | --- | --- |
| Студент гр. 1301 |  | Ахметзянов Д.А. |
| Преподаватель |  | Родионова Е. А. |

Санкт-Петербург

2023

Оглавление

[Формулировка задания. 3](#_Toc1069)

[Описание методов и оценка временной сложности. 3](#_Toc11877)

[Пример работы программы. 4](#_Toc28194)

[Текст программы. 5](#_Toc29095)

# Формулировка задания.

Реализовать двоичное дерево поиска, красно-черное дерево и АВЛ-дерево. Сравнить длины деревьев на случайном наборе входных данных, распределенных равномерно и экспоненциально. Сравнить временные затраты на балансировку для красно-черного и АВЛ-дерева. Отчёт должен содержать графики.

# Теоретические сведения.

**Двоичное дерево поиска** (binary search tree, BST) — двоичное дерево, для которого выполняются следующие дополнительные условия (свойства дерева поиска):

1. оба поддерева — левое и правое — являются двоичными деревьями поиска;
2. у всех узлов левого поддерева произвольного узла X значения ключей данных меньше либо равны, нежели значение ключа данных самого узла X;
3. у всех узлов правого поддерева произвольного узла X значения ключей данных больше, нежели значение ключа данных самого узла X.

**Красно-чёрное дерево** (англ. red-black tree, RB tree) — один из видов самобалансирующихся двоичных деревьев поиска, гарантирующих логарифмический рост высоты дерева от числа узлов и позволяющее быстро выполнять основные операции дерева поиска: добавление, удаление и поиск узла. Сбалансированность достигается за счёт введения дополнительного атрибута узла дерева — «цвета». Этот атрибут может принимать одно из двух возможных значений — «чёрный» или «красный».

**АВЛ-дерево** — это прежде всего двоичное дерево поиска, ключи которого удовлетворяют стандартному свойству: ключ любого узла дерева не меньше любого ключа в левом поддереве данного узла и не больше любого ключа в правом поддереве этого узла. Это значит, что для поиска нужного ключа в АВЛ-дереве можно использовать стандартный алгоритм.

# Тесты по времени.

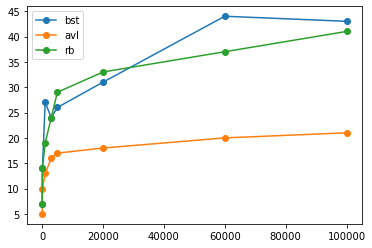


рис. 1 График высоты дерева в зависимости от количества входных данных

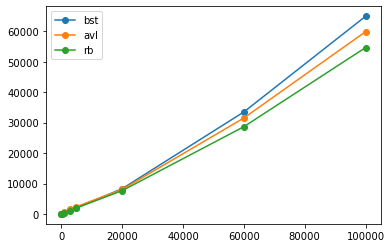


рис. 2 График времени балансировки деревьев

Таким образом, в получившейся реализации красно черное дерево балансируется быстрее всего, но АВЛ-дерево наиболее оптимально по высоте.

# Текст программы.

[Ссылка на github](https://raw.githubusercontent.com/damirahm/AADS/master/2__lab1)