

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Российский химико-технологический университет имени Д.И.
Менделеева»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №6

Выполнил студент группы КС-36: Золотухин А.А.

Ссылка на репозиторий: [https://github.com/
MUCTR-IKT-CPP/
ZolotukhinAA_36_ALG](https://github.com/MUCTR-IKT-CPP/ZolotukhinAA_36_ALG)

Принял: Крашенников Роман Сергеевич

Дата сдачи: 07.04.2025

Москва
2025

Оглавление

Описание задачи	1
Описание метода/модели	2
Выполнение задачи	3

Описание задачи

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

Для проверки анализа работы структуры данных требуется провести 10 серий тестов:

- в каждой серии тестов требуется выполнять 20 циклов генерации и операций. При этом первые 10 работают с массивом заполненным случайным образом, во второй половине случаев, массив заполняется в порядке возрастания значений индекса, т.е. является отсортированным по умолчанию;
- требуется создать массив состоящий из 2^{10+i} элементов, где i это номер серии;
- массив должен быть помещен в оба варианта двоичных деревьев. При этом измеряется время затраченное на всю операцию вставки всего массива;
- после заполнения массива, требуется выполнить 1000 операций поиска по обоим вариантам дерева, случайного числа в диапазоне генерируемых значений, замерев время на все 1000 попыток и вычислив время 1 операции поиска;
- провести 1000 операций поиска по массиву, замерить требуемое время на все 1000 операций и найти время на 1 операцию;
- после, требуется выполнить 1000 операций удаления значений из двоичных деревьев, и замерить время затраченное на все операции, после чего вычислить время на 1 операцию;
- после выполнения всех серий тестов, требуется построить графики зависимости времени затрачиваемого на операции вставки, поиска, удаления от количества элементов. При этом требуется разделить графики для отсортированного набора данных и заполненных со случайным распределением. Так же, для операции поиска, требуется так же нанести для сравнения график времени поиска для обычного массива.

Описание метода/модели

Выполнение задачи

Бинарное дерево поиска и AVL-дерево реализованы на языке *C++*. Построение графиков проводились с помощью программы *GNUplot*.

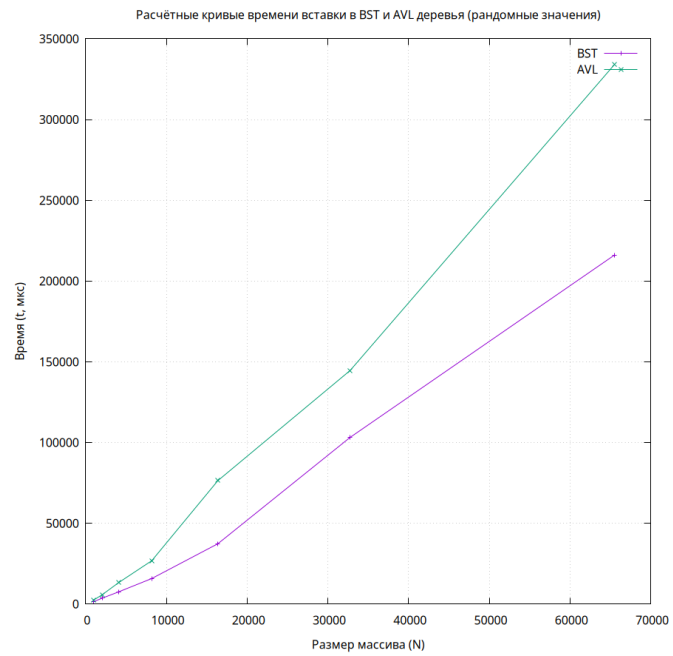


Рис. 1: Расчётные кривые времени вставки в BST и AVL деревья (рандомные данные).

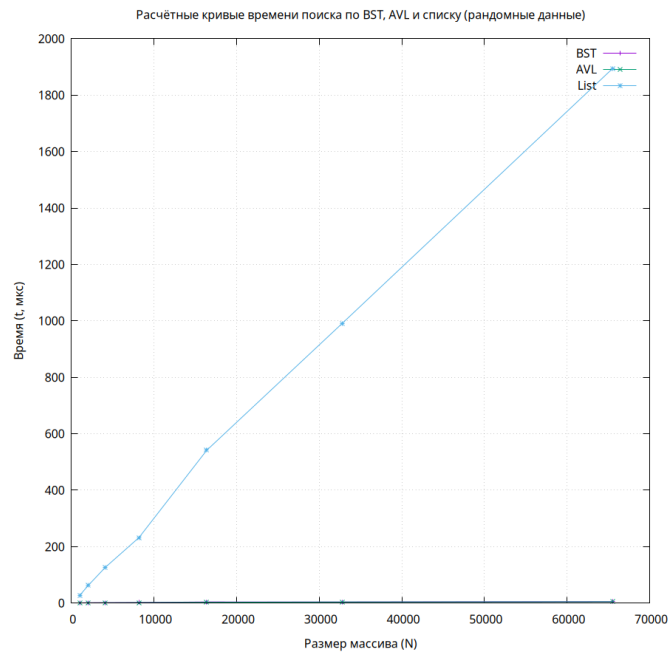


Рис. 2: Расчётные кривые времени поиска по BST, AVL и списку (рандомные данные).

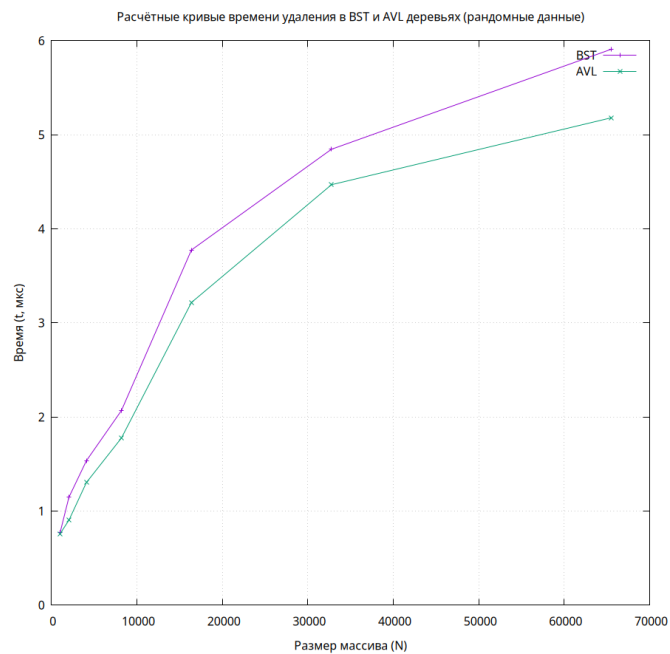


Рис. 3: Расчётные кривые времени удаления в BST и AVL деревьях (рандомные данные).

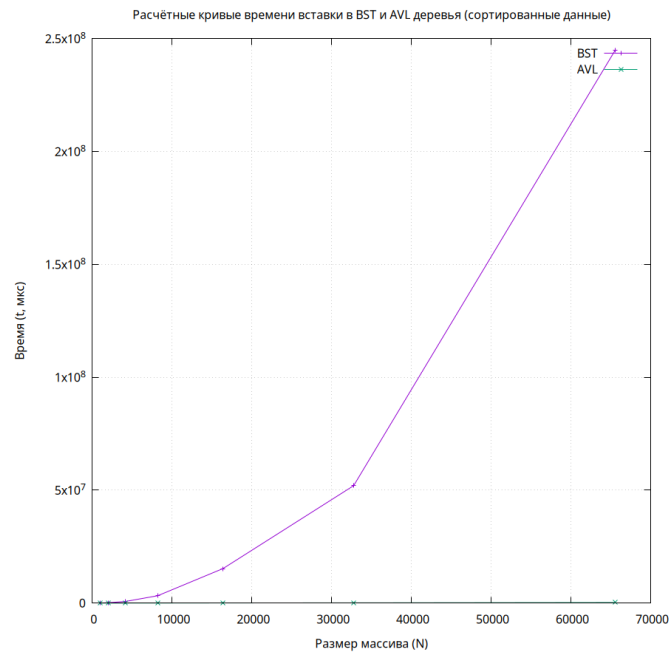


Рис. 4: Расчётные кривые времени вставки в BST и AVL деревья (сортированные данные).

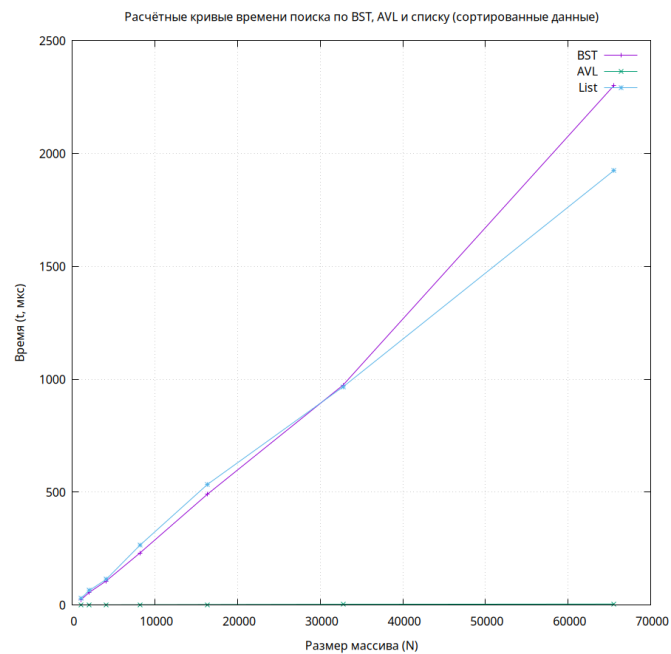


Рис. 5: Расчётные кривые времени поиска по BST, AVL и списку (сортированные данные).

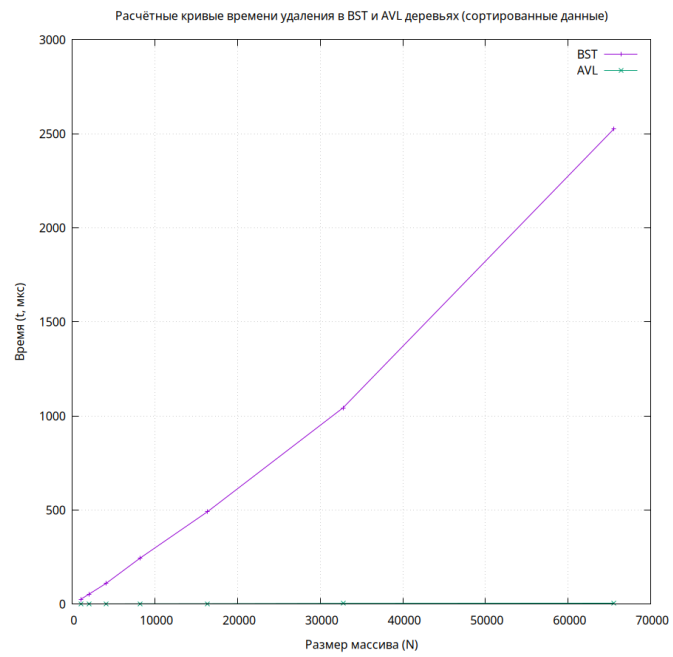


Рис. 6: Расчётные кривые времени удаления в BST и AVL деревьях (сортированные данные).