

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ
УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра КСУП

Отчет по лабораторной работе
по дисциплине «Структуры данных»
Тема: «Анализ красно-черного и AVL дерева»

Студент гр. 582-1

Полушвайко Константин Николаевич

___ декабря 2023 г.

Оглавление

1 Введение.....	3
2 Основная часть	5
3 Заключение	11

1 Введение

В качестве шестой лабораторной работы нужно создать красно-черное дерево, (далее КЧД), и АВЛ-дерево, а также проанализировать их работу и сравнить. Данные деревья являются сбалансированными бинарными деревьями поиска.

Красно-черное дерево

Рассмотрим поближе КЧД. Оно удовлетворяет следующим свойствам:

1. Каждый узел является либо красным, либо черным;
2. Корень дерева является черным узлом;
3. Каждый лист дерева (NIL) является черным узлом;
4. Если узел красный, то оба его дочерних узла черные;
5. Для каждого узла все простые пути от него до листьев, являющихся потомками данного узла, содержат одно и то же количество черных узлов.

КЧД с n внутренними узлами имеет высоту, не превышающую:

$$h \leq 2\lg(n + 1) \quad (1.1)$$

Балансировка осуществляется поворотами:

- Левый малый (выполняется за время $O(1)$);
- Правый малый (выполняется за время $O(1)$);

Вставка узла в дерево выполняется за время $O(\lg n)$, причем в балансировке может произойти максимум 2 поворота. Удаление узла из дерева также выполняется за время $O(\lg n)$, а поворотов в балансировке может произойти максимум 3. Также стоит отметить, что узел КЧД занимает всего 1 бит дополнительной памяти для хранения цвета вершины.

АВЛ-дерево

Теперь рассмотрим АВЛ-дерево. Оно поддерживает следующее свойство: для каждой его вершины высота её двух поддеревьев различается не более чем на 1.

АВЛ-дерево с n внутренними узлами имеет высоту, не превышающую:

$$h < 1,44 \lg(n + 2) - 0,328 \quad (1.2)$$

Балансировка осуществляется поворотами:

- Левый малый (выполняется за время $O(1)$);
- Правый малый (выполняется за время $O(1)$);
- Левый большой (который содержит в себе 2 малых поворота);
- Правый большой (который содержит в себе 2 малых поворота).

Вставка узла в дерево выполняется за время $O(\lg n)$, в балансировке может произойти максимум 2 поворота. Удаление узла из дерева также выполняется за время $O(\lg n)$. Так как в удалении используется рекурсия до корня, то поворотов может произойти не больше $\lg n$. Узел АВЛ-дерева занимает 1 байт дополнительной информации для хранения высоты.

2 Основная часть

Анализ деревьев

Чтобы проанализировать работу каждого дерева проведем исследование:

- Создадим КЧД и АВЛ-дерево и заполним их 10^k случайными элементами;
- Будем добавлять в каждое дерево по 10^{k-1} случайных элементов и замерять суммарное время работы вставок и количество поворотов, а также максимумы времени и поворотов на одну операцию;
- Затем, по такому же принципу будем удалять по 10^{k-1} случайных элементов.

Примечание: порядок вставки и удаления случайных чисел рассматривается абсолютно одинаковый для двух деревьев. Это осуществляется при помощи общего seed для генерации случайных чисел.

Проведем наше исследование: выберем стартовое количество узлов – 10^8 , шаг измерений будет 10^7 . Получили таблицу. Построили графики, которые приведены на рисунках 2.1–2.8.

Анализ графиков показывает, что КЧД работает быстрее АВЛ-дерева:

- В 4 раза при вставке элемента;
- В 2 раза при удалении элемента.

Различия между суммами поворотов получилась не сильно критичными:

- При вставке элемента у КЧД на 600.000 поворотов больше, чем у АВЛ-дерева;
- При удалении элемента у АВЛ-дерева на 100.000 поворотов больше, чем у КЧД.

Стоит отметить, что сбалансированность КЧД ниже, чем у АВЛ-дерева. Но тем не менее для балансировки КЧД производится меньше действий, чем с АВЛ-деревом. Также у АВЛ дополнительно занимается целый байт памяти, а у красно-черного дерева всего бит. Преимуществом же АВЛ является то, что его намного проще реализовать, логика балансировки АВЛ рассматривает всего пару случаев, а придерживаться надо одного свойства. КЧД же достаточно трудно в реализации, так как необходимо поддерживать все 5 свойств, а в ходе балансировки могут возникать множество случаев, которые не факт, что взаимоисключающие.

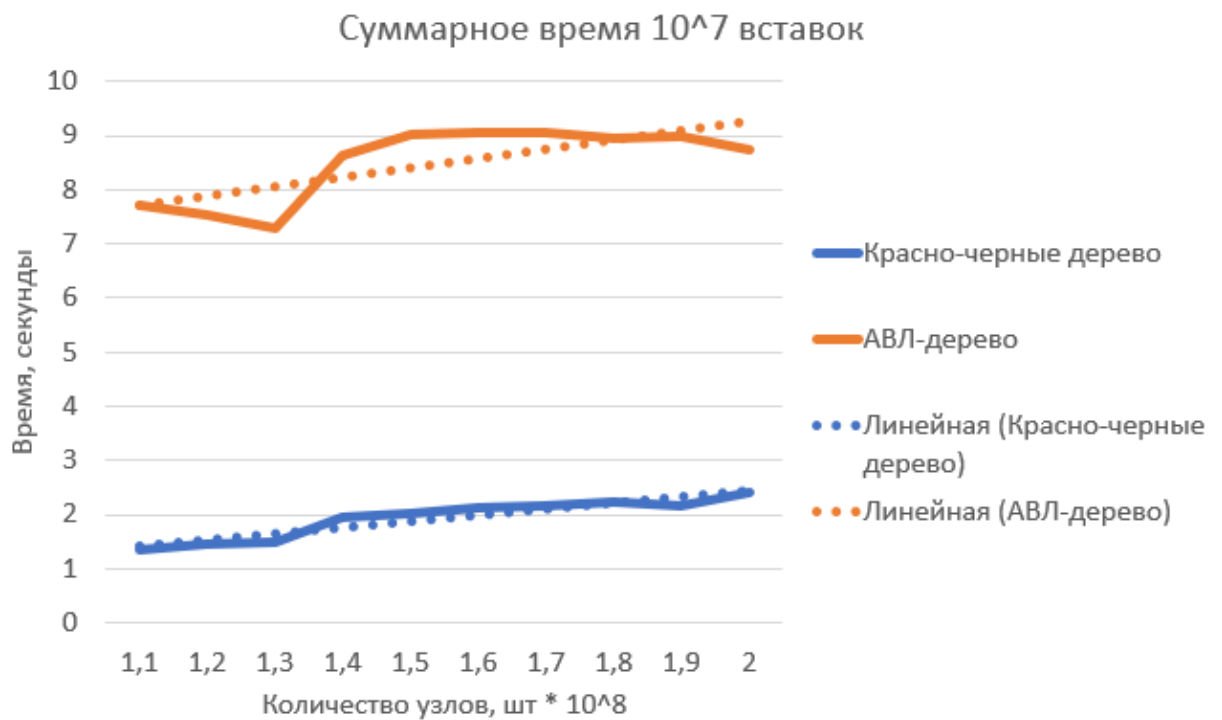


Рисунок 2.1 – График суммарного времени выполнения 10^7 вставок

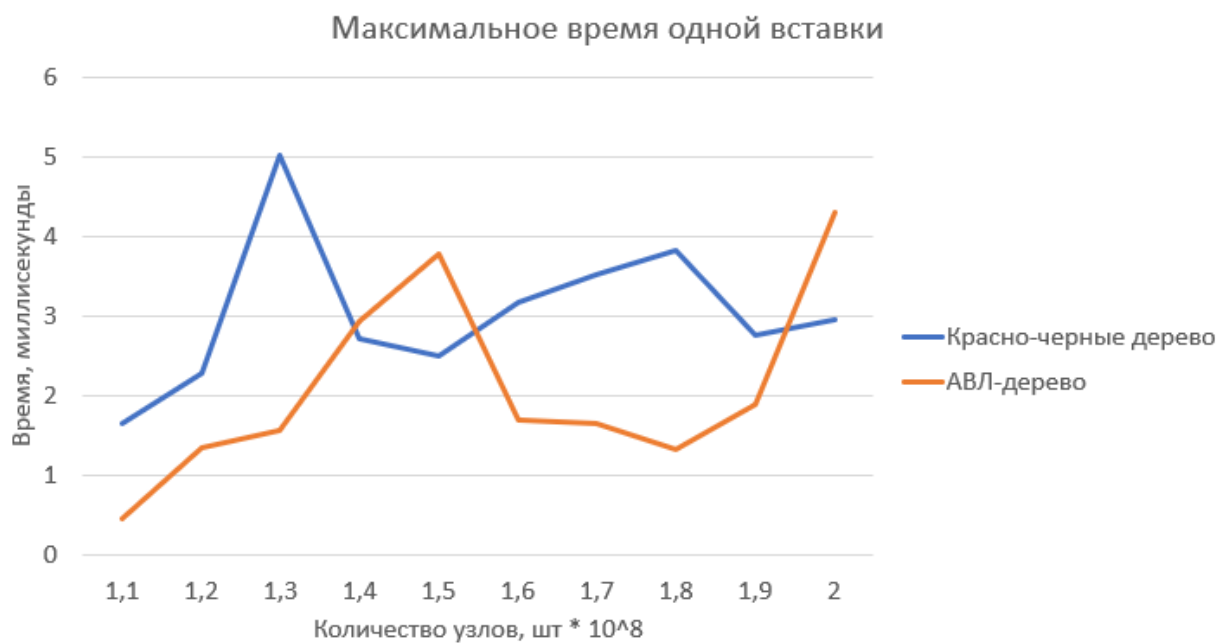


Рисунок 2.2 – График максимального времени одной вставки из 10^7

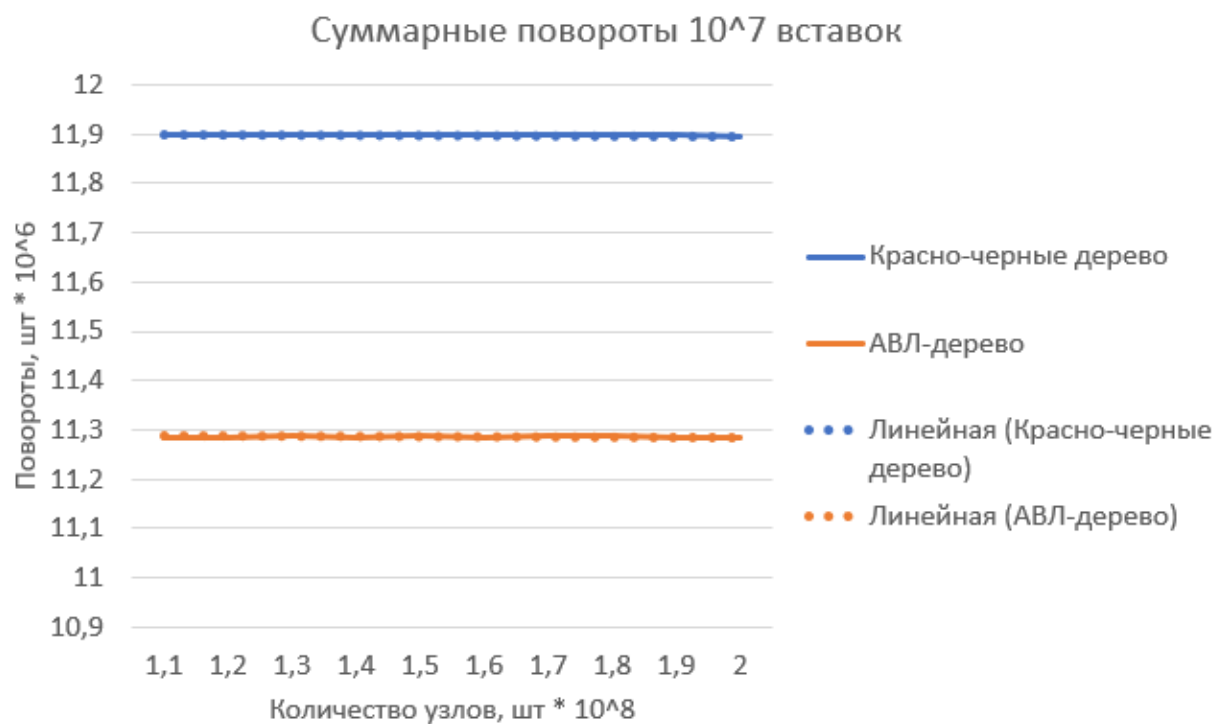


Рисунок 2.3 – График суммарных поворотов при выполнении 10^7 вставок



Рисунок 2.4 – График максимального кол-во поворотов одной вставки из 10^7

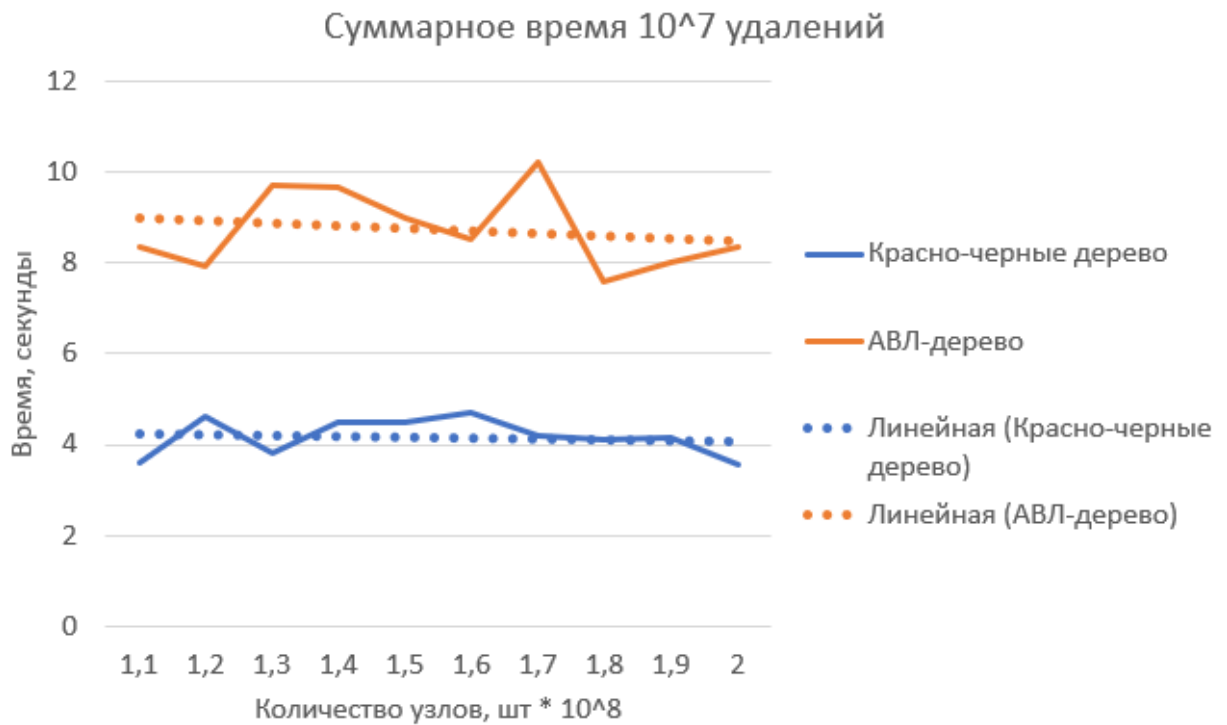


Рисунок 2.5 – График суммарного времени при выполнении 10^7 удалений

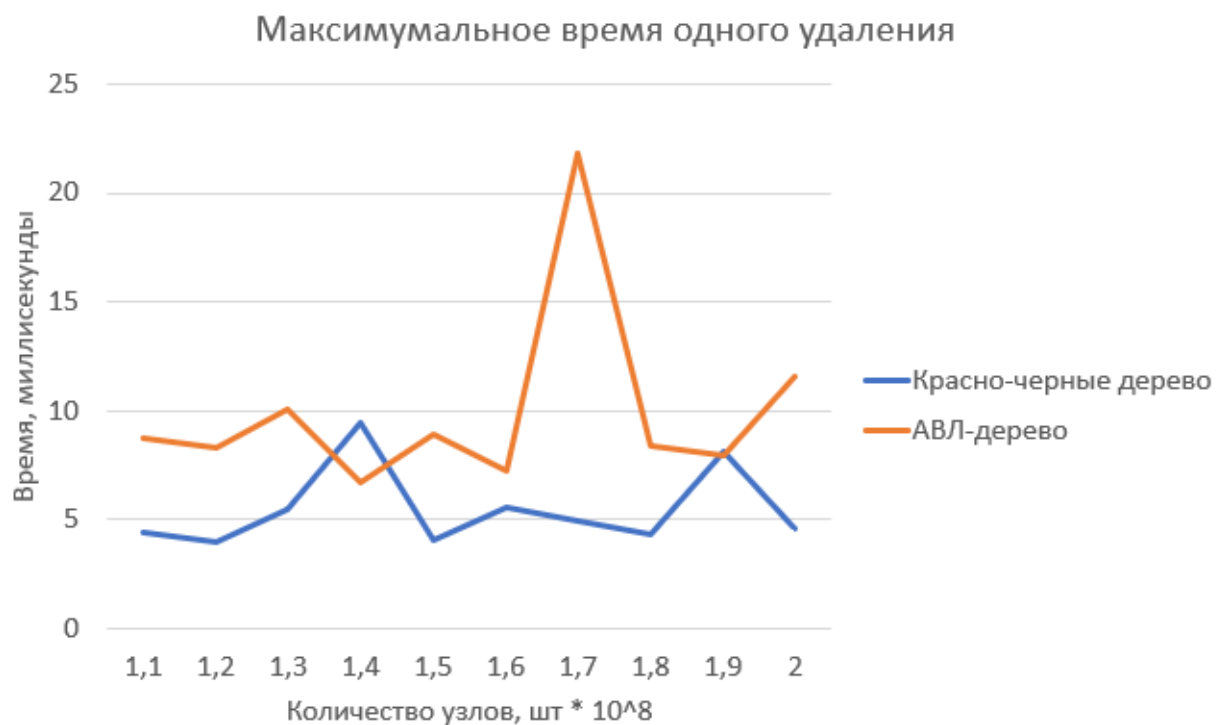


Рисунок 2.6 – График максимального времени одного удаления из 10^7

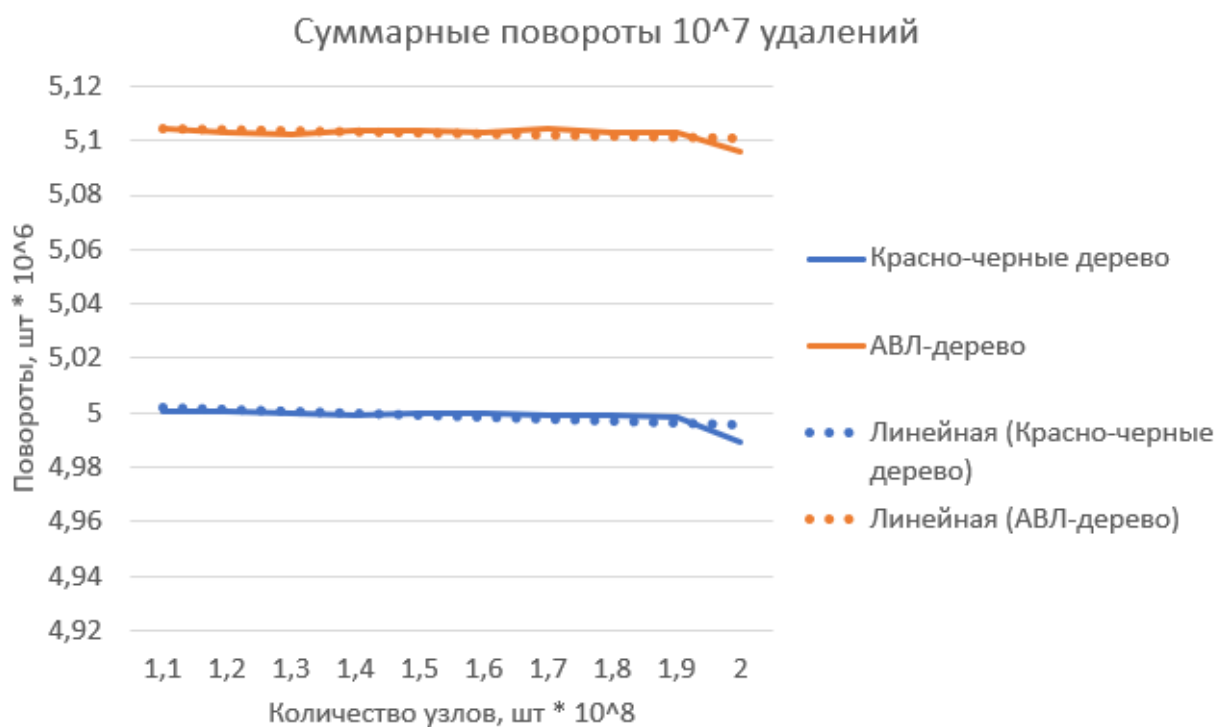


Рисунок 2.7 – График суммарных поворотов при выполнении 10^7 удалений

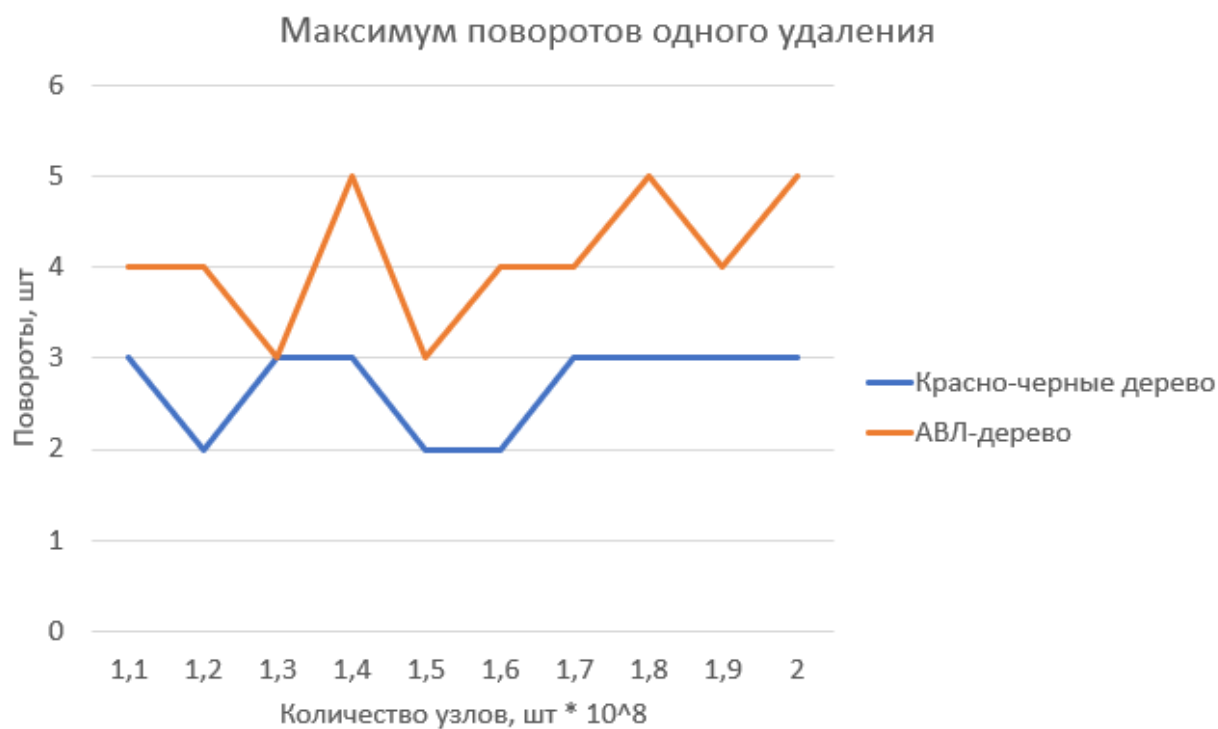


Рисунок 2.8 – График максимума поворотов одного удаления из 10^7

3 Заключение

Из всей вышеперечисленной информации можно сделать вывод, что красно-черное дерево эффективнее в скорости и памяти, а AVL выигрывает в сбалансированности дерева и простоте алгоритма.