

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»

Работу выполнил:

студент гр. 582-1

Полушвайко Константин Николаевич

Проверил:

к. т. н., доцент каф. КСУП

Калентьев Алексей Анатольевич

2023

## Оглавление

1 Введение .....	3
2 Основная часть .....	5
3 Заключение.....	12

## 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:

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

Балансировка осуществляется поворотами:

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

Вставка узла в дерево выполняется за время  $O(\log n)$ , причем в балансировке может произойти максимум 2 поворота. Удаление узла из дерева также выполняется за время  $O(\log n)$ , 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:

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

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  $O(\log n)$ , в балансировке может произойти максимум 2 поворота. Удаление узла из дерева также выполняется за время  $O(\log n)$ . Так как в удалении используется рекурсия до корня, то поворотов может произойти не больше  $\log n$ . 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 таблицы: 2.1, 2.2, 2.3, 2.4. Построили графики, которые приведены на рисунках 2.1–2.8.

Таблица 2.1 – Вставка по времени.

Вставка (время) шт * $10^8$	КЧД (суммарное) сек	КЧД (максимум) мс	АВЛ (суммарное) сек	АВЛ (максимум) мс
1,1	1,342421	1,643	7,726619	0,466
1,2	1,45197	2,291	7,548376	1,355
1,3	1,494345	5,018	7,274328	1,566
1,4	1,944915	2,714	8,64503	2,944
1,5	2,017105	2,489	9,001547	3,784
1,6	2,108273	3,173	9,056084	1,698
1,7	2,165038	3,512	9,04729	1,653
1,8	2,215432	3,815	8,938346	1,333
1,9	2,163229	2,752	8,977362	1,894
2	2,404743	2,951	8,723432	4,304

Таблица 2.2 – Вставка по поворотам.

Вставка (повороты) шт * 10 <sup>8</sup>	КЧД (суммарное) шт * 10 <sup>6</sup>	КЧД (максимум) шт	АВЛ (суммарное) шт * 10 <sup>6</sup>	АВЛ (максимум) шт
1,1	11,89809	2	11,285993	2
1,2	11,897546	2	11,28626	2
1,3	11,899242	2	11,287453	2
1,4	11,897016	2	11,285947	2
1,5	11,899775	2	11,286395	2
1,6	11,898301	2	11,285907	2
1,7	11,898887	2	11,286755	2
1,8	11,897406	2	11,286924	2
1,9	11,899135	2	11,286029	2
2	11,896757	2	11,28449	2

Таблица 2.3 – Удаление по времени.

Удаление (время) шт * 10 <sup>8</sup>	КЧД (суммарное) сек	КЧД (максимум) мс	АВЛ (суммарное) сек	АВЛ (максимум) мс
1,1	3,600812	4,428	8,345177	8,776
1,2	4,620549	3,92	7,930852	8,254
1,3	3,801078	5,492	9,724957	10,027
1,4	4,48389	9,472	9,646939	6,717
1,5	4,501903	4,01	8,977212	8,886
1,6	4,693199	5,573	8,506081	7,262
1,7	4,196871	4,948	10,208134	21,826
1,8	4,111817	4,338	7,581153	8,349
1,9	4,129347	8,131	8,02057	7,951
2	3,539308	4,555	8,358947	11,55

Таблица 2.4 – Удаление по поворотам.

Удаление (повороты) шт * 10 <sup>8</sup>	КЧД (суммарное) шт * 10 <sup>6</sup>	КЧД (максимум) шт	АВЛ (суммарное) шт * 10 <sup>6</sup>	АВЛ (максимум) шт
1,1	5,000453	3	5,10457	4
1,2	5,000374	2	5,103162	4
1,3	4,99991	3	5,102233	3
1,4	4,99914	3	5,103803	5
1,5	5,000001	2	5,103347	3
1,6	4,999959	2	5,103263	4
1,7	4,998798	3	5,10449	4
1,8	4,999296	3	5,102735	5
1,9	4,998113	3	5,103173	4
2	4,989176	3	5,095963	5

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.

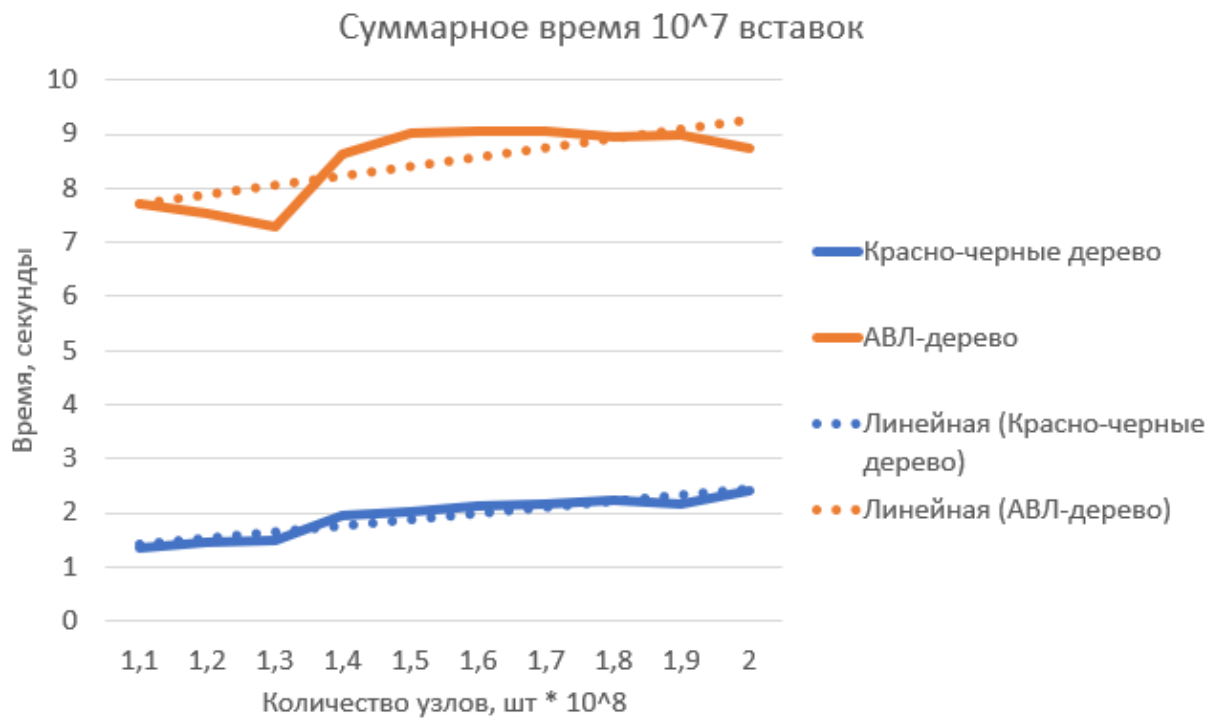


Рисунок 2.1 – График суммарного времени выполнения  $10^7$  вставок

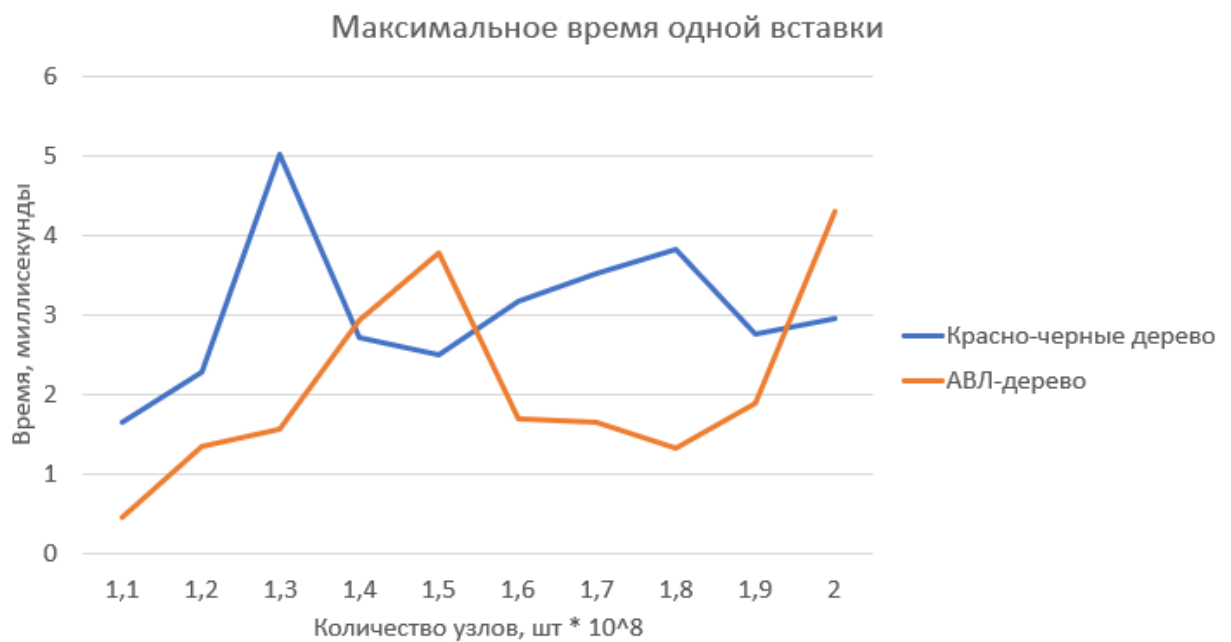


Рисунок 2.2 – График максимального времени одной вставки из  $10^7$



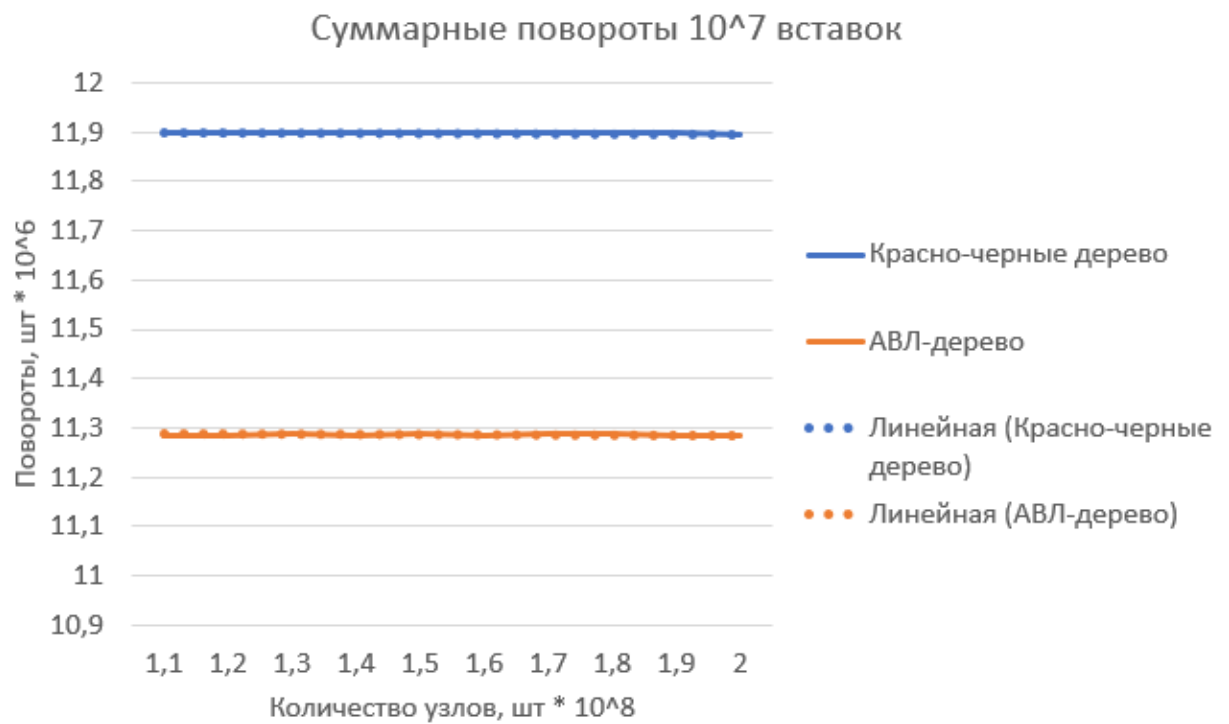


Рисунок 2.3 – График суммарных поворотов при выполнении  $10^7$  вставок



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

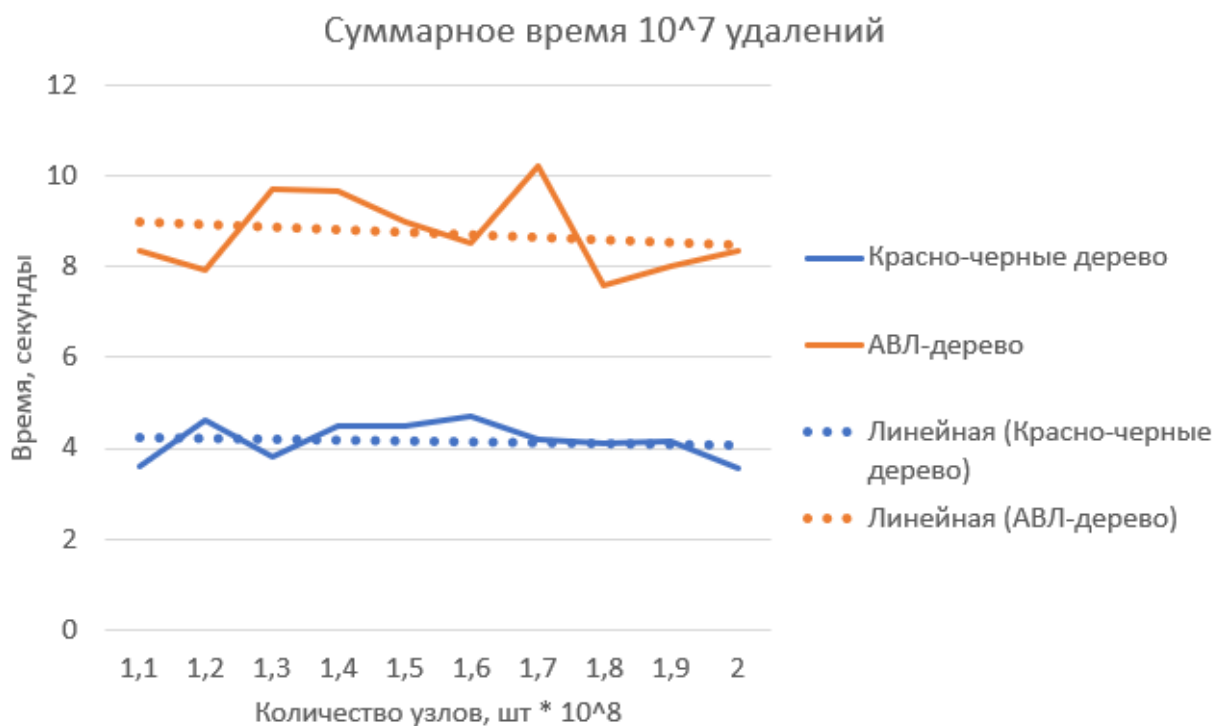


Рисунок 2.5 – График суммарного времени при выполнении  $10^7$  удалений

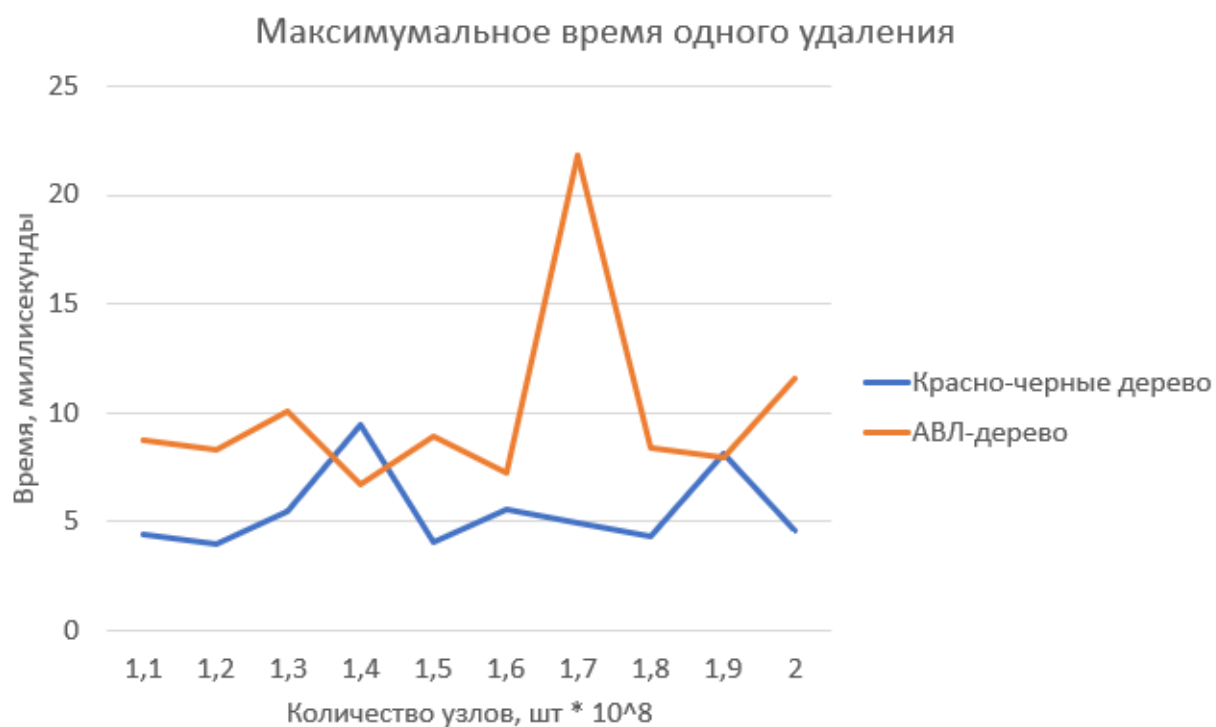


Рисунок 2.6 – График максимального времени одного удаления из  $10^7$

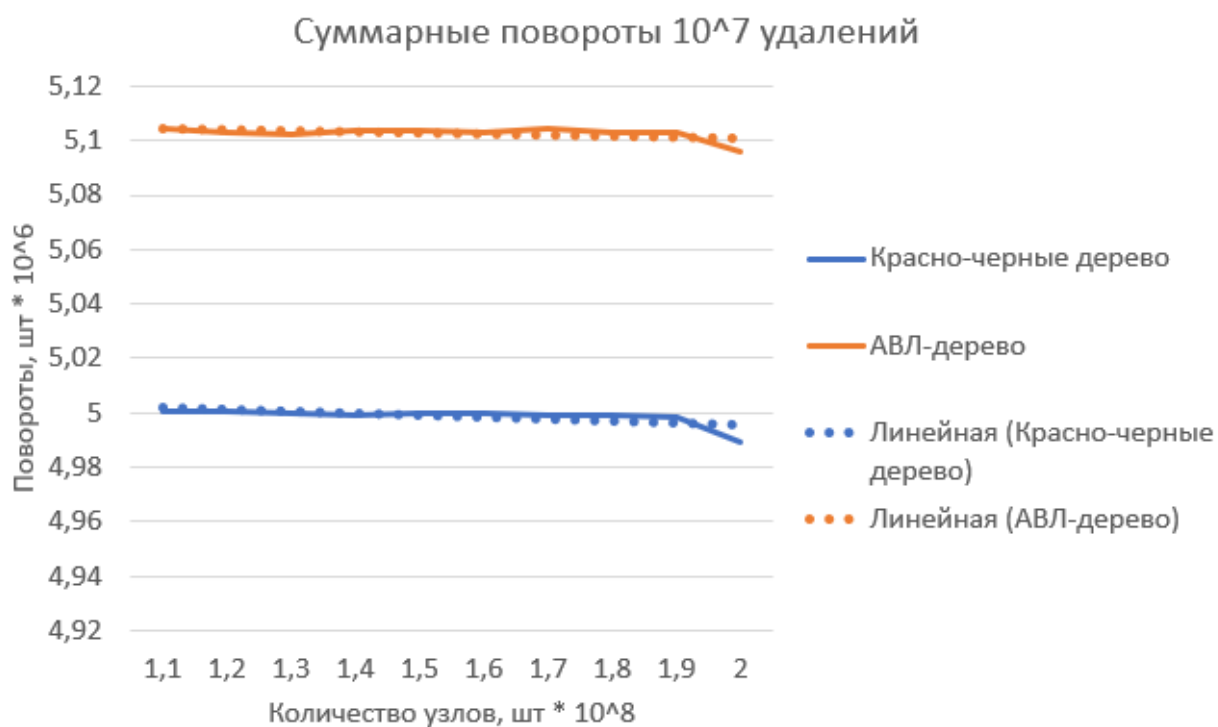


Рисунок 2.7 – График суммарных поворотов при выполнении  $10^7$  удалений

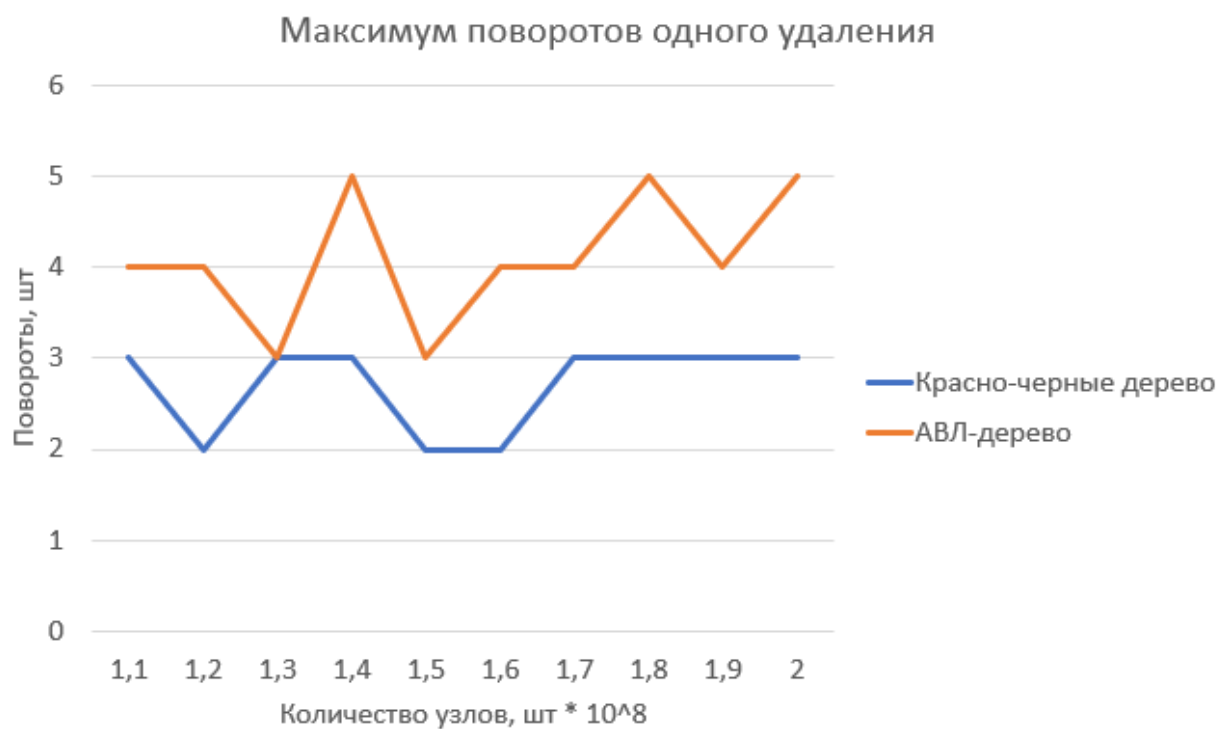


Рисунок 2.8 – График максимума поворотов одного удаления из  $10^7$

### 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.