
АВЛ-деревья (AVL trees). Фактор балансировки АВЛ-дерева. Добавление элемента в АВЛ-дерево. Повороты поддеревьев. Алгоритм ленивого удаления узлов. Доказательство утверждения о высоте АВЛ-дерева.

АВЛ-деревья/Фактор балансировки

АВЛ-дерево (*AVL tree*) – сбалансированное по высоте двоичное дерево поиска, в котором у любой вершины высота левого и правого поддеревьев различаются не более, чем на 1

- Основная идея:
 - Если вставка или удаление элемента приводит к нарушению сбалансированности дерева, то необходимо выполнить его балансировку
- **Коэффициент сбалансированности узла** (*фактор балансировки*) – это разность высот его левого и правого поддеревьев
- В АВЛ-дереве коэффициент сбалансированности любого узла принимает значения из множества $\{-1, 0, 1\}$
- **Высота узла** (*height*) – это длина наибольшего пути от него до дочернего узла, являющегося листом

Вычислительная сложность для всех операций во всех случаях: $O(\log(n))$

Структура узла АВЛ-дерева:

```
struct avltree {  
    int key;  
    char *value;  
    int height;  
    struct avltree *left;  
    struct avltree *right;  
};
```

Добавление элемента в AVL-дерево

1. Находим лист для вставки нового элемента по такому же принципу как и в бинарном дереве поиска
2. Если при вставке или удалении узла фактор балансировки для какого-либо узла становится 2 или -2 это означает, что дерево нарушило свой баланс. Для восстановления баланса производится операция вращения (поворота), которая меняет расположение узлов, чтобы вернуть фактор балансировки в допустимый диапазон

Повороты AVL-дерева

Правый поворот AVL-дерева (R-rotation)

- В левое поддереву узла 3 добавили элемент 1
- Дерево с корнем в узле 3 не сбалансировано ($\text{balance} = h(\text{left}) - h(\text{right}) = 1 - (-1) = 2$ (нарушен баланс дерева))
- $h(\text{left}) = 1$; $h(\text{right}) = -1$
- Необходимо увеличить высоту правого поддерева

Поворачиваем ребро, связывающее корень и его левый дочерний узел, вправо



15

Левый поворот AVL-дерева

- В правое поддереву узла 1 добавили элемент 3
- Дерево с корнем в узле 1 не сбалансировано ($\text{balance} = h(\text{left}) - h(\text{right}) = -1 - 1 = -2$ (нарушен баланс дерева))
- $h(\text{left}) = -1$; $h(\text{right}) = 1$
- Необходимо увеличить высоту левого поддерева

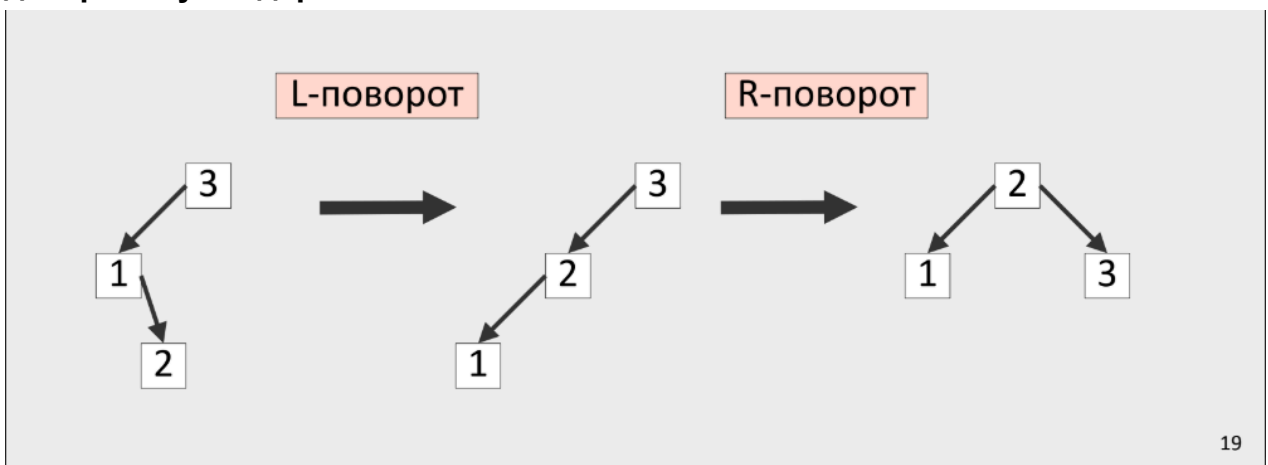
- Поворачиваем ребро, связывающее корень и его правый дочерний узел, влево



17

Двойной лево-правый поворот АВЛ-дерева (LR-rotation)

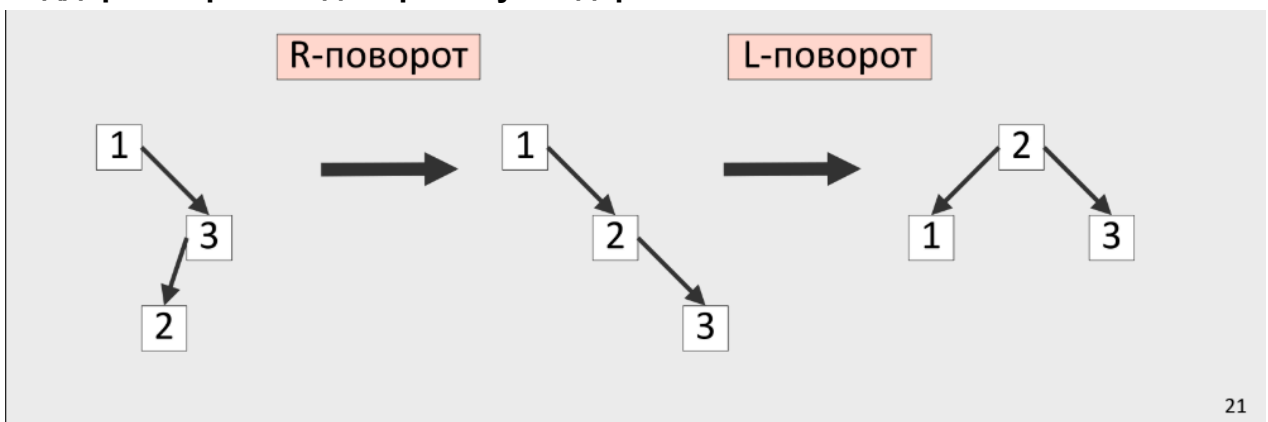
- LR-поворот выполняется после добавления узла в **правое поддерево левого дочернего узла дерева**



19

Двойной право-левый поворот АВЛ-дерева (RL-rotation)

- RL-поворот выполняется после добавления узла в **левое поддерево правого дочернего узла дерева**



21

Алгоритм ленивого удаления узлов

- Удаление элемента выполняется аналогично добавлению
- После удаления может нарушиться баланс нескольких родительских вершин

- После удаления вершины может потребоваться порядка $O(\log(n))$ поворотов поддеревьев

Алгоритм ленивого удаления узлов:

- С каждым узлом AVL-дерева ассоциирован флаг *deleted*
- При удалении узла находим его в дереве и устанавливаем флаг *deleted* = 1 (реализация за время $O(\log(n))$)
- При вставке нового узла с таким же ключом, как и у удаленного элемента, устанавливаем у последнего флаг *deleted* = 0 (в поле данных копируем новое значение)
- При достижении порогового значения количества узлов с флагом *deleted* = 1 создаем новое AVL-дерево, содержащее все не удалённые узлы (*deleted* = 0)
- Поиск не удалённых элементов и их вставка в новое AVL-дерево реализуется за время $O(n \log(n))$

Доказательство утверждения о высоте AVL-дерева

Теорема: AVL-дерево с n ключами имеет высоту $h = O(\log(n))$.

Лемма: Пусть m_h — минимальное число вершин в AVL-дерева высоты h , тогда $m_h = F_{h+2} - 1$, где F_h — h -ое число Фибоначчи.

Доказательство:

Если m_h — минимальное число вершин в AVL-дерева высоты h . Тогда как легко видеть, $m_{h+2} = m_{h+1} + m_h + 1$. Равенство $m_h = F_{h+2} - 1$ докажем по индукции.

База индукции: $m_1 = F_3 - 1$ — верно, $m_1 = 1, F_3 = 2$.

Допустим $m_h = F_{h+2} - 1$ — верно.

Тогда $m_{h+1} = m_h + m_{h-1} + 1 = F_{h+2} - 1 + F_{h+1} - 1 + 1 = F_{h+3} - 1$.

Таким образом, равенство $m_h = F_{h+2} - 1$ — доказано.

$F_h = \Omega(\varphi^h), \varphi = \frac{\sqrt{5}+1}{2}$ То есть $n \geq \varphi^h$

Логарифмируя по основанию φ , получаем

$$\log_{\varphi} n \geq h$$

Таким образом, получаем, что высота AVL-дерева из n вершин — $O(\log(n))$.