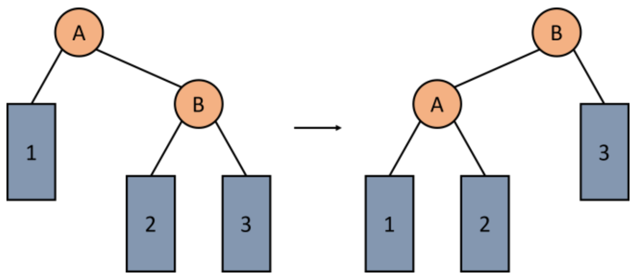
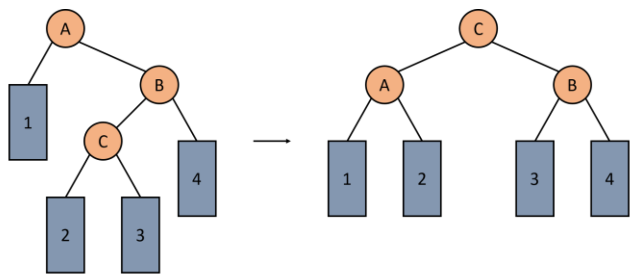
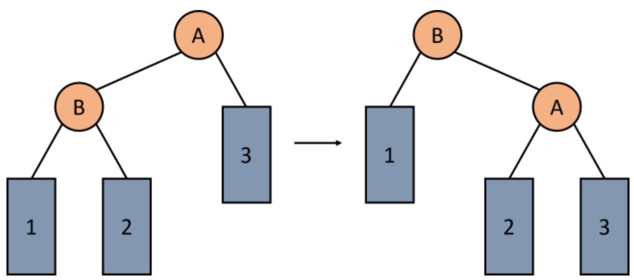
Лабораторная работа №2

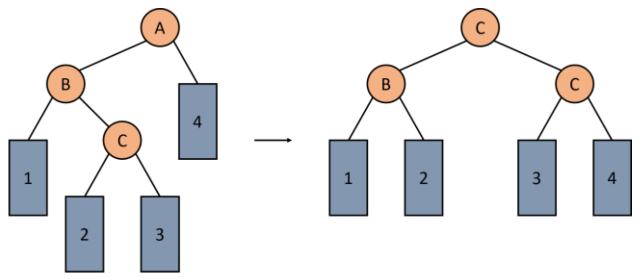
АВЛ-деревья

Основные понятия

Структура сбалансированного дерева была предложена Адельсоном-Вельским и Ландисом в 1962 году. Даётся следующее определение сбалансированности дерева.

Дерево является сбалансированным тогда и только тогда, когда для каждого узла высота его двух поддеревьев различается не более чем на 1. Для контроля сбалансированности узлов в структуру каждого узла введён критерий сбалансированности $balance = h_R - h_L$, где h_L и h_R – высота левого и правого поддеревьев узла. Узел сбалансирован, если значение $balance$ равно -1, 0 или $+1$. Операции вставки и удаления элементов могут привести к изменению высот поддеревьев в узлах, лежащих на пути к вставленному или удалённому элементу. После вставки или удаления элемента выполняется восходящая проверка и корректировка критериев сбалансированности всех узлов, лежащих на пути операции. Если в результате корректировки критерия значение $balance$ в узле становится равным -2 или $+2$, то выполняется балансировка разбалансированного узла. Балансировка заключается в выполнении поворота узла для выравнивания высот поддеревьев. В AVL-дереве используются четыре вида поворотов в зависимости о конфигурации поддеревьев узла с нарушенным критерием сбалансированности.

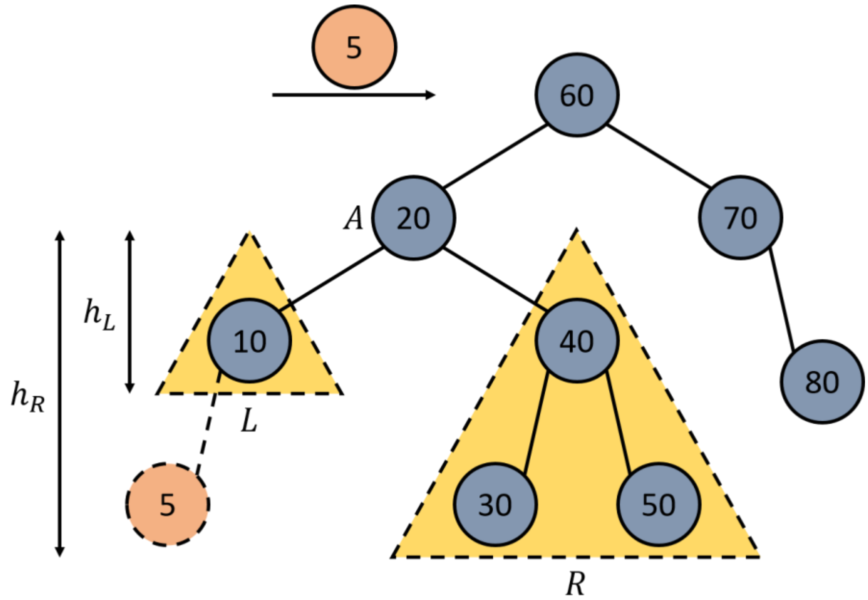
Малое левое вращение (L)		Если узел с нарушенным критерием перевешивает *вправо*, а его правый сын – *вправо*
Большое левое вращение (LR)		Если узел с нарушенным критерием перевешивает *влево*, а его правый сын – *вправо*
Малое правое вращение (R)		Если узел с нарушенным критерием перевешивает *влево*, а его правый сын – *влево*

Большое правое вращение (LR)		Если узел с нарушенным критерием перевешивает *вправо*, а его правый сын – *влево*

В каждом случае достаточно просто доказать то, что операция приводит к нужному результату и что полная высота уменьшается не более чем на 1 и не может увеличиться. Также можно заметить, что большое левое вращение — это композиция правого малого вращения и левого малого вращения. Из-за условия сбалансированности высота дерева $O\big(log(N)\big)$, где N – количество вершин, поэтому добавление элемента требует $O\big(log(N)\big)$ операций.

Включение нового элемента в AVL - дерево

Рассмотрим операцию включения в сбалансированное дерево нового узла. Пусть дан узел A с левым и правым поддеревьями L и R. Предположим, что новый узел включается в L, вызывая увеличение его высоты на 1.



Возможны три случая после включения:

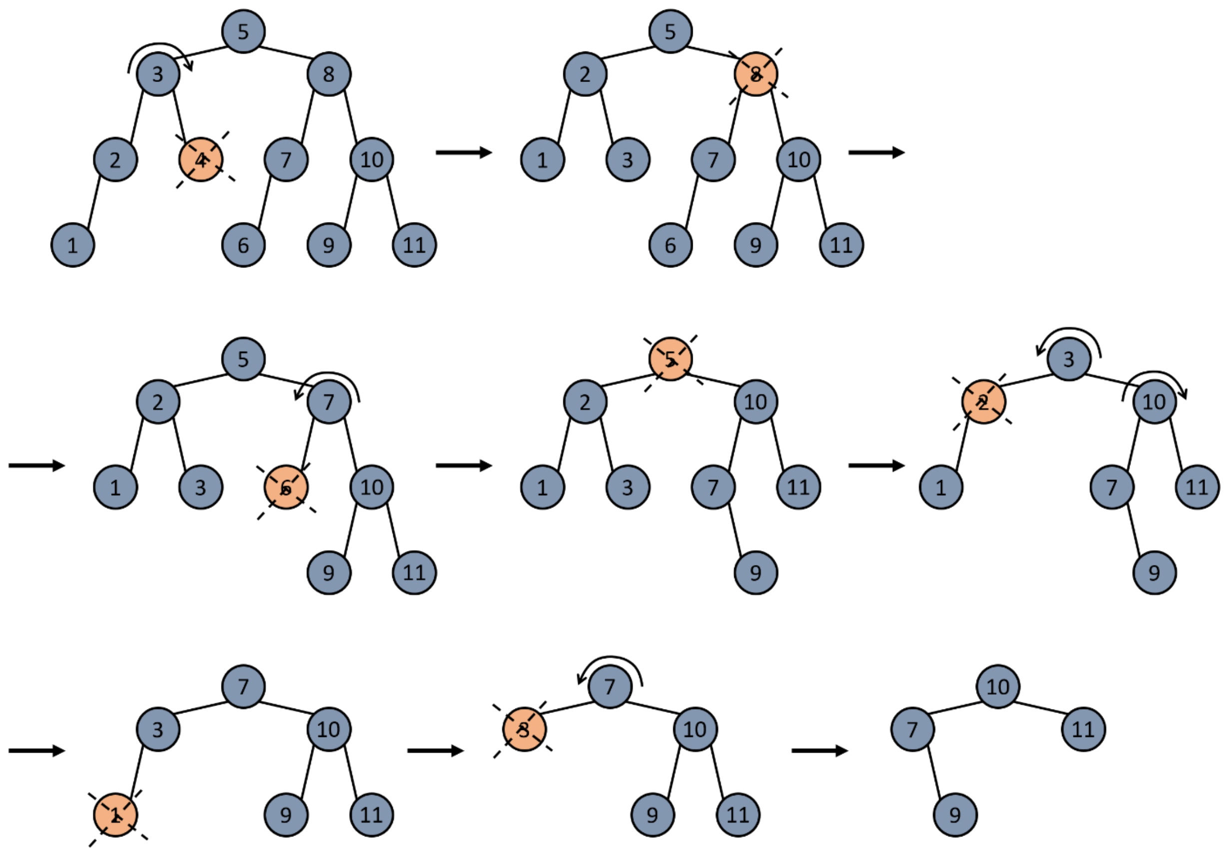
1. Если $h_L = h_R$ до включения, то L и R становятся неравной высоты, но критерий сбалансированности не нарушен.
2. Если $h_L < h_R$ до включения, то L и R становятся равной высоты и сбалансированность даже улучшается.
3. Если $h_L > h_R$ до включения, то критерий сбалансированности нарушается, и дерево нужно перестраивать.

Процесс включения узла в сбалансированное дерево состоит из трех этапов:

1. Поиск места включения узла в каком-либо листе.
2. Включение нового узла и определение нового показателя сбалансированности в месте включения.
3. Обратный проход по пути включения и проверка сбалансированности у каждого узла на этом пути.

Удаление элемента из AVL - дерева

При удалении узлов из AVL – дерева операция балансировки в основном остается такой же, что и при включении и заключается в однократном или двукратном повороте. При этом булевская переменная h, возвращаемая операцией удаления означает уменьшение высоты поддерева. При восходящем пересчете критерия сбалансированности выполняется балансировка тех узлов, у которых критерий стал равным $+2$ или -2. Балансировка выполняется с помощью тех же L- , R- , RL- и LR- поворотов.



Удаление элементов также имеет стоимость $O(\log_2 n)$. Но если включение может вызвать один поворот, удаление может потребовать повороты в каждом узле пути поиска при обратном ходе рекурсии. Но эмпирические проверки показали, что если при включении выполняется один поворот на каждые два включения, то при удалении один поворот приходится на 5 удалений.

Задание к лабораторной работе

В данной лабораторной работе необходимо:

1. Написать программу, которая демонстрирует работу с АВЛ-деревьями.
2. Реализовать функции:
   - создания дерева;
   - добавления элемента;
   - удаления элемента;
   - уничтожения дерева;
   - проверки дерева на пустоту;
   - вывода элементов двоичного дерева на экран одним из обходов.

   При необходимости должна выполняться балансировка дерева.
3. Выполнить индивидуальное задание в соответствии со своим вариантом.

Варианты заданий:

Вариант	**Задание**
1	Определить глубину дерева, представляемую как наибольшая длина пути от корня к листьям.
2	Определить ширину дерева, представляемую как: $1 + \text{Ширина левого поддерева} + \text{Ширина правого поддерева.}$
3	Определить самый левый лист дерева.
4	Определить самый правый лист дерева.
5	Определить глубину правого поддерева дерева.
6	Определить глубину левого поддерева дерева.
7	Определить ширину левого поддерева дерева.
8	Определить ширину правого поддерева дерева.
9	Определить лист, который находится дальше всего от корня.