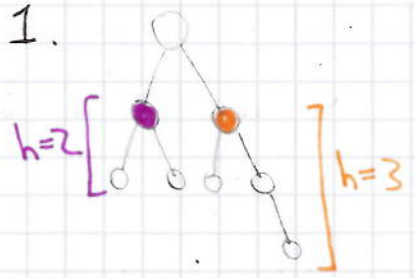
AVL Trees

Description

- An AVL tree is a binary search tree with the additional property of being self-balancing, that is, it fulfills the height balance property.
- Height balance property: For every internal node of the tree, the heights of its children can differ by at most 1.
- Any subtree of an AVL tree is an AVL tree itself.



⇒ fine!

Complexities

- height is always roughly $\log n$; n = number of nodes
- Space: $O(n)$
- Search: $O(\log n)$
- Insert: $O(\log n)$
- Delete: $O(\log n)$

Balancing

- An AVL tree can become unbalanced in four different ways:
  1. Left-left-case
  2. Right-right-case
  3. Left-right-case
  4. Right-left-case

> The process of rebalancing the tree (see next page) is called trinode restructuring.

- See next page on how to rebalance the tree in each case.
- We use two functions for rebalancing: left_rotate(k) and right_rotate(k) (same node)