Informatik II

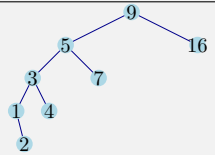
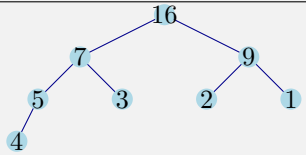
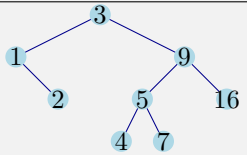
Übung 6

FS 2020

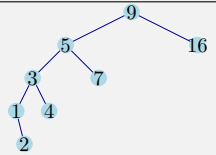
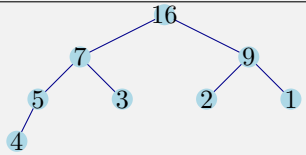
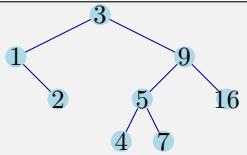
Program Today

1. Recap Binary Trees
2. Repetition Lectures
   - AVL Condition
   - AVL Insert
3. In-Class-Exercises

Comparison of binary Trees

	Search trees	**Heaps** Min- / Max- Heap	**Balanced trees** AVL, red-black tree
in Java:		PriorityQueue	TreeSet
			
Insertion	$\mathcal{O}(h(T))$	$\mathcal{O}(\log n)$	$\mathcal{O}(\log n)$
Search	$\mathcal{O}(h(T))$	$\mathcal{O}(n)$ (!!)	$\mathcal{O}(\log n)$
Deletion	$\mathcal{O}(h(T))$	Search + $\mathcal{O}(\log n)$	$\mathcal{O}(\log n)$

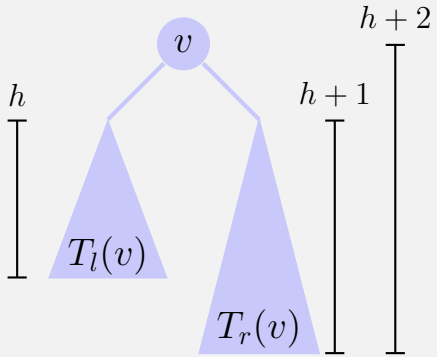
Comparison of binary Trees

	Search trees	**Heaps** Min- / Max- Heap	**Balanced trees** AVL, red-black tree
in Java:		PriorityQueue	TreeSet
			
Insertion	$\mathcal{O}(h(T))$	$\mathcal{O}(\log n)$	$\mathcal{O}(\log n)$
Search	$\mathcal{O}(h(T))$	$\mathcal{O}(n)$ (!!)	$\mathcal{O}(\log n)$
Deletion	$\mathcal{O}(h(T))$	Search + $\mathcal{O}(\log n)$	$\mathcal{O}(\log n)$

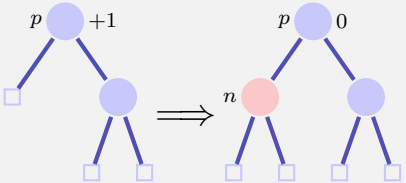
Recall: $\mathcal{O}(\log n) \leq \mathcal{O}(h(T)) \leq \mathcal{O}(n)$

AVL Condition

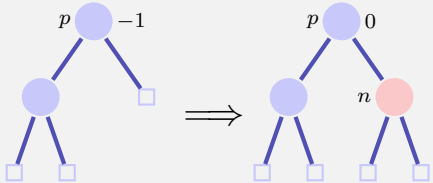
AVL Condition: for eacn node v of a tree $\mathrm{bal}(v) \in \{-1, 0, 1\}$



Balance at Insertion Point



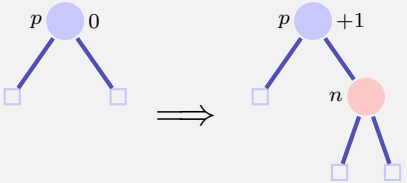
case 1: $\mathrm{bal}(p) = +1$



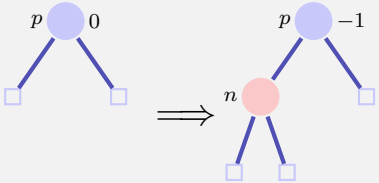
case 2: $\mathrm{bal}(p) = -1$

Finished in both cases because the subtree height did not change

Balance at Insertion Point



case 3.1: $\mathrm{bal}(p) = 0$ right



case 3.2: $\mathrm{bal}(p) = 0$, left

Not finished in both case. Call of `upin(p)`

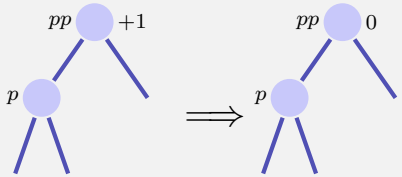
upin(p) - invariant

When `upin(p)` is called it holds that

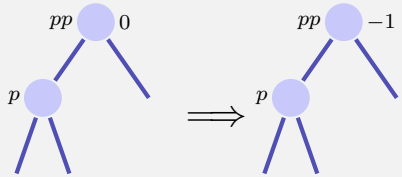
- the subtree from p is grown and
- $\mathrm{bal}(p) \in \{-1, +1\}$

upin(p)

Assumption: p is left son of pp[1]



case 1: $\mathrm{bal}(pp) = +1$, done.



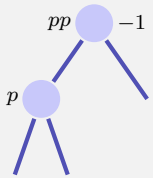
case 2: $\mathrm{bal}(pp) = 0$, `upin(pp)`

In both cases the AVL-Condition holds for the subtree from pp

[1] If p is a right son: symmetric cases with exchange of $+1$ and -1

upin(p)

Assumption: p is left son of pp



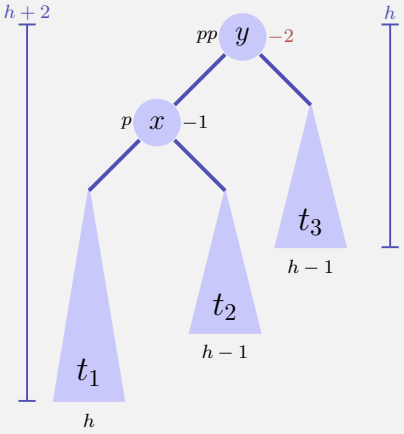
case 3: $\mathrm{bal}(pp) = -1$,

This case is problematic: adding n to the subtree from pp has violated the AVL-condition. Re-balance!

Two cases $\mathrm{bal}(p) = -1$, $\mathrm{bal}(p) = +1$

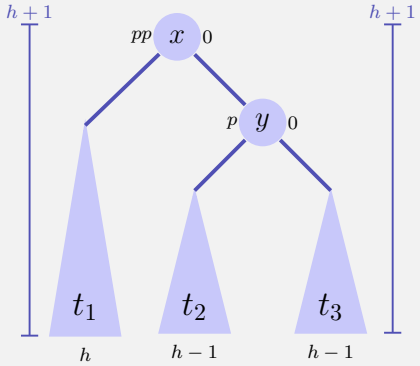
Rotations

case 1.1 $\mathrm{bal}(p) = -1$. [2]



$\Longrightarrow$

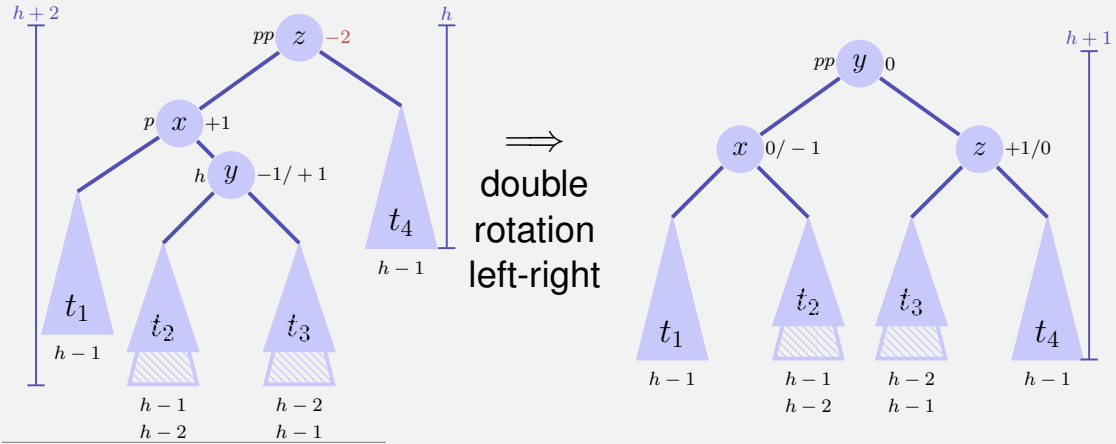
rotation right



[2] p right son: $\Rightarrow \mathrm{bal}(pp) = \mathrm{bal}(p) = +1$, left rotation

Rotations

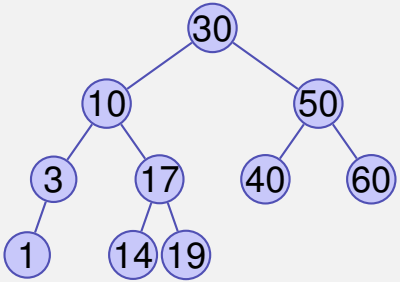
case 1.1 $\mathrm{bal}(p) = -1$. [3]



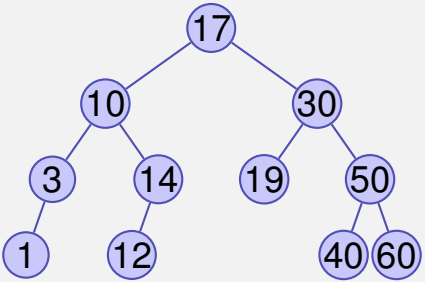
[3] p right son $\Rightarrow \mathrm{bal}(pp) = +1$, $\mathrm{bal}(p) = -1$, double rotation right left

Quiz

In the following AVL tree, insert key 12 and rebalance (as shown in class). What does the AVL tree look like after the operation that has been shown in class?



Solution



3. In-Class-Exercises

1. Recursion in Trees

Exercise:
Implement a recursive function to compute the height and weight of (a node of) of a binary search treee

[Code Expert, Code Examples 6]

2. Augment a Tree

Exercise:
Augment the nodes `n` of a binary search tree with their heights `n.height`. Make sure the height stays consistent when nodes are inserted.

[Code Expert, Code Examples 6]

Questions / Suggestions?