

Plan

- discuss assignment 4
- preview assignment 6
- recap AVL-trees
- recap dynamic programming
- Peer-Grading (5.2)

Assignment 4 discussion

Common mistakes for 4.3

- mixing 0/1-indexing
- no specification for variable in I.H \Rightarrow for some integer k \times , for some integer k with $1 \leq k < n$ \checkmark
- mixing Eprog and AnD invariants
- using list as synonym for array

Common mistakes for 4.4

- the default log in AnD is \log_2 not \log_{10}
- not using $\lfloor \rfloor / \lceil \rceil$ in pseudocode (I just assumed java rounding ($\lfloor \rfloor$))
- using "=" for assignments and comparisons

\hookrightarrow good options:

assignment	Comparison
$a = 3$	$a == b$
$a := 3$	$a = b$
$a \leftarrow 3$	$a = b$

Java-Syntax

my personal choice

syntax used in most literature

- off-by-one errors for the binary-search which led to non-termination
- you can call algorithms from previous subtasks
 \hookrightarrow no point deduction if prev. subtask was incorrect

Assignment 6 preview

- recheck AVL-tree properties
- try to understand the DP-approach in general
- that's it for assignment 6 😊

AVL-trees

- max 2 children per node
- left child has key < parents key
- right child has key > parents key
- rebalance based on subtree(s) height

↳ difference in height of left and right subtree can't be more than one (AVL-condition)

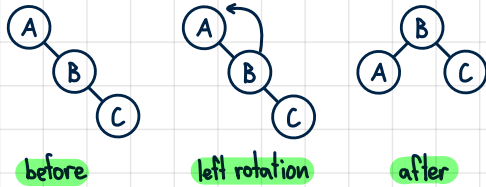
- on deletion: replace root with next-smallest leaf (if possible) else with next biggest leaf, then rebalance

also called self-balancing binary search tree

usually appears in exams

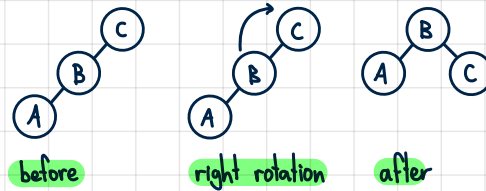
Left Rotation

- unbalanced due to right subtree of right subtree



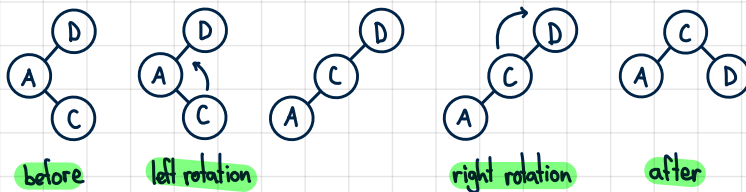
Right Rotation

- unbalanced due to left subtree of left subtree



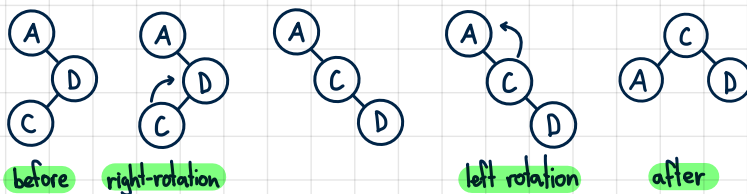
Left-Right Rotation

- combination of left and right rotation (in that order)
- unbalanced due to right subtree of left subtree

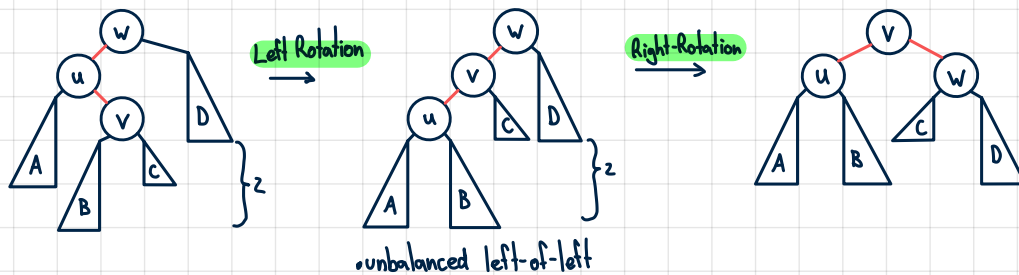
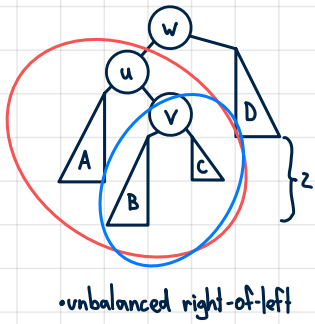
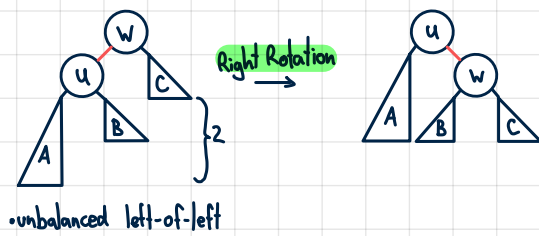


Right-Left Rotation

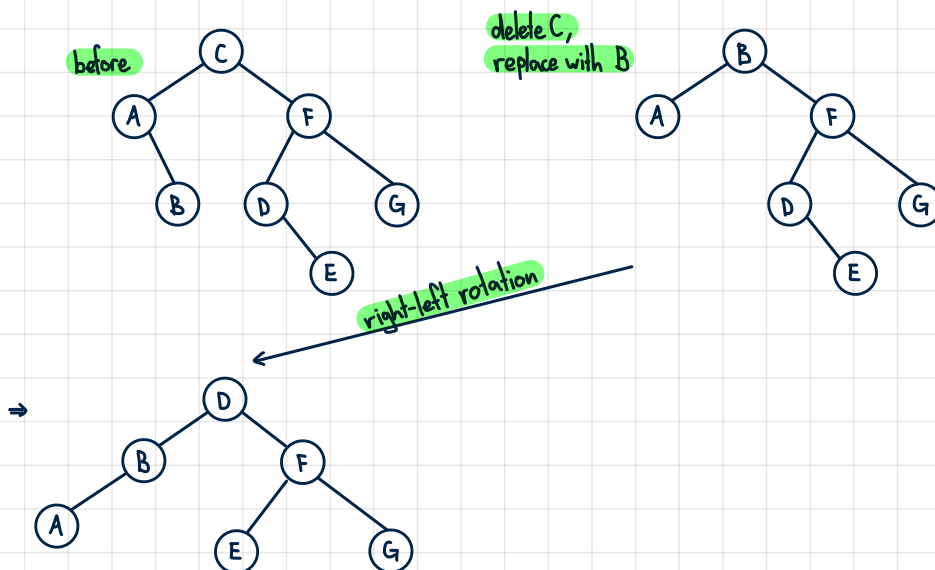
- combination of right and left rotation (in that order)
- unbalanced due to left subtree of right subtree



In the bigger picture: ~~not~~ rotated



Example of Deletion: we delete ③



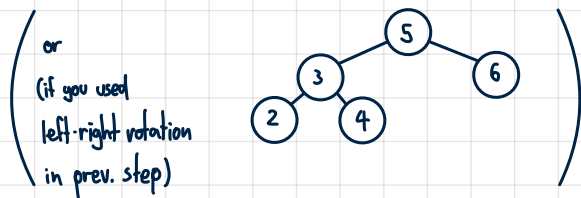
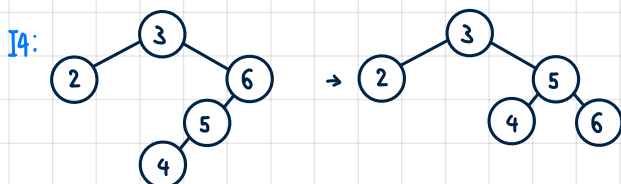
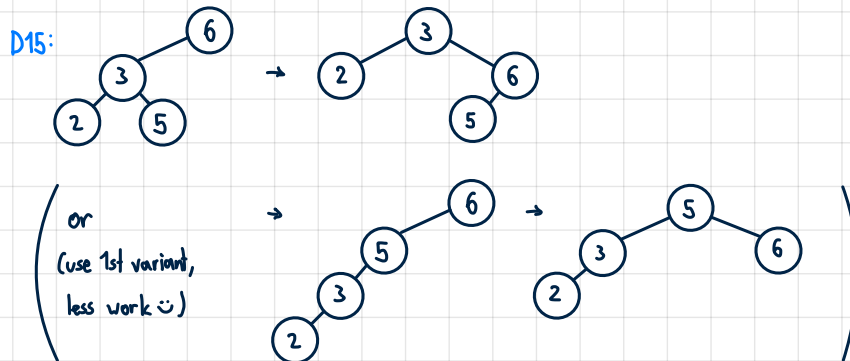
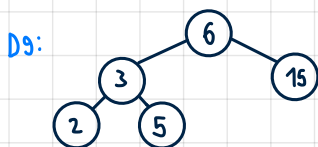
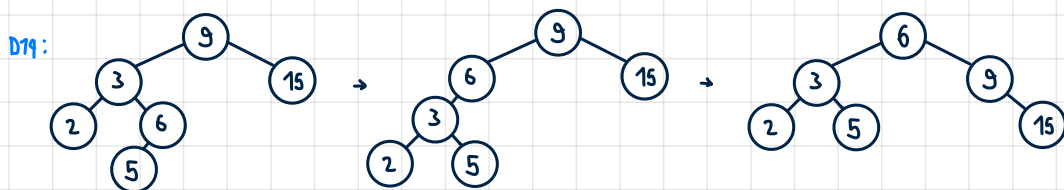
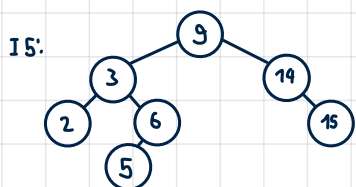
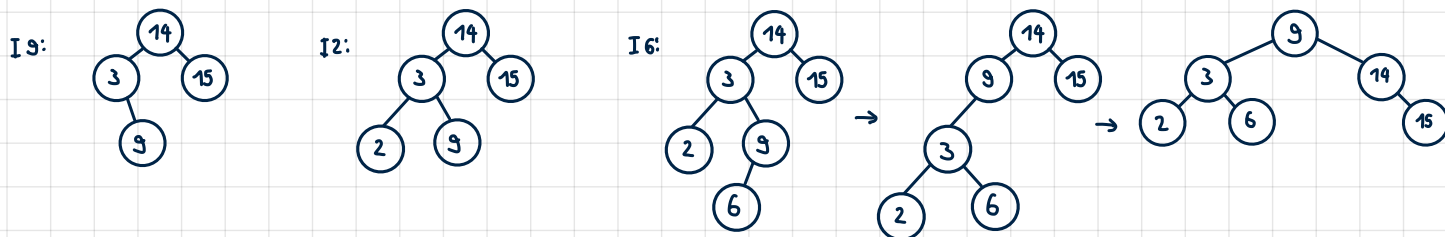
⇒ we now have an imbalance for ③, due to left subtree of right subtree of B
⇒ right-left rotation (for ③)

Exercises

FS22, T1 d.)

Draw the AVL-tree obtained when inserting 3, 14, 15, 9, 2, 6, 5 in this order into an empty tree (then delete 14, 9, 15 and insert 4, in that order)

not in exam question



⇒ Watch YouTube-video of Abdul Bari if you're still confused

Dynamic-Programming

idea: 1. define subproblem

2. solve subproblems recursive or iterative

are usually 2-/3-dimensional (can be 1-/k-dimensional, for $k \geq 4$, not discussed in course)

Discussion: Longest Palindromic-Subsequence (HS20 exam)

Given: array of characters $A[1 \dots n]$

Wanted: Length of longest palindromic subsequence in $A[1 \dots n]$

\Rightarrow subproblem: Length of longest palindromic subsequence in $A[i \dots j]$

Example: $A = \text{ETZHEEHU} \Rightarrow 4$

$A = \text{HBCH}$ oder $\text{HBCH} \Rightarrow 3$

Solution with DP-Table: 1-indexing, size is $n \cdot n$, $DP_{i,j}$ = Length of longest palindromic subsequence in $A[i \dots j] \Rightarrow$ solution in $DP_{1,n}$

j \ i	1	2	3	4	5	6	7	8
	E	T	Z	H	E	E	H	U
1	E	1	1	1	1	3 \rightarrow 3	4 \rightarrow 4	
2	T	0	1	1	1	2 \uparrow	4 $\uparrow \rightarrow$ 4	
3	Z	0	0	1	1	2 \uparrow	4 $\uparrow \rightarrow$ 4	
4	H	0	0	0	1 \rightarrow 1	2 \uparrow	4 $\uparrow \rightarrow$ 4	
5	E	0	0	0	0	1 \uparrow	2 \rightarrow 2 \rightarrow 2	
6	E	0	0	0	0	0	1 \rightarrow 1 \rightarrow 1	
7	H	0	0	0	0	0	0	1 \rightarrow 1
8	U	0	0	0	0	0	0	0

Base Case: $DP_{i,j} = \begin{cases} 0, & \text{falls } i > j \\ 1, & \text{falls } i = j \end{cases}$

Order of computation: bottom up, left to right (for $i=n \dots 1$
for $j=i+1 \dots n$)