Eidgenössische
Technische Hochschule
Zürich

Ecole polytechnique fédérale de Zurich
Politecnico federale di Zurigo
Federal Institute of Technology at Zurich

Department of Computer Science
Markus Püschel, David Steurer
Gleb Novikov, Tommaso d'Orsi, Ulysse Schaller, Rajai Nasser

22. November 2021

Algorithms & Data Structures — Exercise sheet 9 — HS 21

Exercise Class (Room & TA): ____________________

Submitted by: ____________________

Peer Feedback by: ____________________

Points: _____

Submission: On Monday, 29 November 2021, hand in your solution to your TA *before* the exercise class starts. Exercises that are marked by * are challenge exercises. They do not count towards bonus points.

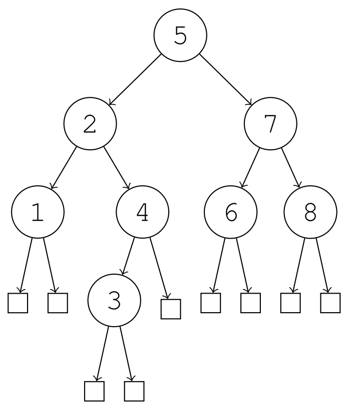
Exercise 9.1 *Search Trees* **(2 points)**.

a) Draw the resulting tree when the keys 2, 7, 8, 4, 5, 6, 3, 1 in this order are inserted into an initially empty binary (natural) search tree.

b) Delete key 1 in the above tree, and afterwards delete key 7 in the resulting tree.

c) Draw the resulting tree when the above keys are inserted into an initially empty AVL tree. Give also the intermediate states before and after each rotation that is performed during the process.

d) Consider the following AVL tree:



Delete key 1 in this tree, and afterwards delete key 7 in the resulting tree. Give also the intermediate states before and after each rotation is performed during the process.

Exercise 9.2 *Exponential bounds for a sequence defined inductively.*

Consider the sequence $(a_n)_{n\in\mathbb{N}}$ defined by

$$\begin{aligned} a_0 &= 1, \\ a_1 &= 1, \\ a_2 &= 2, \\ a_i &= a_{i-1} + 2a_{i-2} + a_{i-3} \quad \forall i \geq 3. \end{aligned}$$

The goal of this exercise is to find exponential lower and upper bounds for a_n.

a) Find a constant $C > 1$ such that $a_n \leq \mathcal{O}(C^n)$ and prove your statement.

b) Find a constant $c > 1$ such that $a_n \geq \Omega(c^n)$ and prove your statement.

Exercise 9.3 *Online supermarket.*

Assume that you work in a large online supermarket that offers different types of goods. At every moment you have to know the number of goods of each type that the supermarket currently offers. Let us denote the number of goods of type t by S_t. At any moment S_t can either be decreased (if someone has bought some goods of type t) or increased (if some goods of type t have been delivered from the manufacturer). Also your boss can ask you how many goods of type t does the supermarket currently offer. So you can receive three types of queries: to decrease S_t by $0 < x \leq S_t$, to increase S_t by $x > 0$ or to return S_t.

Assume that at each moment number of different types of goods that the supermarket offers at that moment is bounded by $n > 0$, but the number of types of goods that the supermarket can potentially offer might be much larger than n. Consider the following example: $n = 3$, at 14:00 the supermarket can offer 5 balls, 1 doll and 4 phones and at 14:15 it can offer 6 balls, 3 chairs and 12 pencils.

Provide an algorithm that can handle each query in time $\mathcal{O}(\log n)$. You may assume that initially all S_t are zero.

Exercise 9.4 *Augmented Binary Search Tree* **(1 point)**.

Consider a variation of a binary search tree, where each node has an additional member variable called SIZE. The purpose of the variable SIZE is to indicate the size of the subtree rooted at this node. An example of an augmented binary search tree (with integer data) can be seen below (Fig. 1).

a) What is the relation between the size of a node and the sizes of its children?

b) Describe in pseudo-code an algorithm VerifySizes(ROOT) that returns TRUE if all the sizes in the tree are correct, and returns FALSE otherwise. For example, it should return TRUE given the tree in Fig. 1, but FALSE given the tree in Fig. 2.

   What is the running time of your algorithm? Justify your answer.

c) Suppose we have an augmented AVL tree (i.e., as above, each node has a SIZE member variable). Describe in pseudo-code an algorithm Select(ROOT, k) which, given an augmented AVL tree and an integer k, returns the k-th smallest element in the tree in $O(\log n)$ time.

   Example: Given the tree in Fig. 1, for $k = 3$, Select returns 8; for $k = 5$, it returns 10; for $k = 1$, it returns 3; etc.

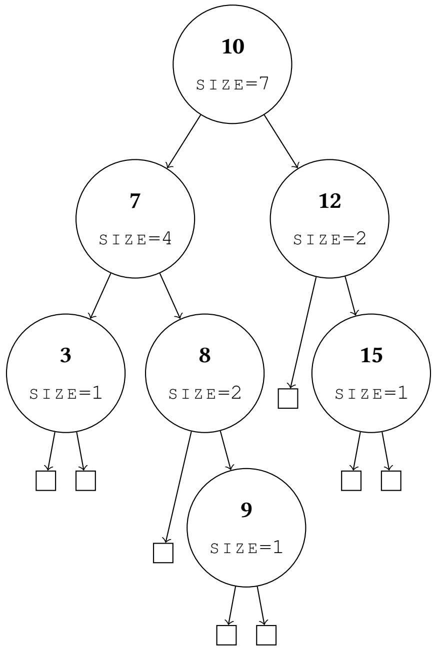


Figure 1: Augmented binary search tree

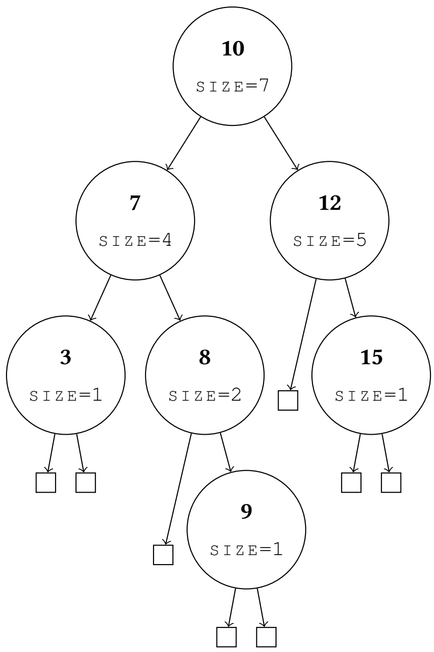


Figure 2: Augmented binary search tree with buggy size: incorrect size for node with data "12"

d)* To maintain the correct sizes for each node, we have to modify the AVL tree operations, insert and remove. For this problem, we will consider only the modifications to the AVL-INSERT method (i.e., you are not responsible for AVL-REMOVE). Recall that AVL-INSERT first uses regular INSERT for binary search trees, and then balances the tree if necessary via rotations.

- How should we update INSERT to maintain correct sizes for nodes?

During the balancing phase, AVL-INSERT performs rotations. Describe what updates need to be made to the sizes of the nodes. (It is sufficient to describe the updates for left rotations, as right rotations can be treated analogously.)

Exercise 9.5* *Maximum Depth Difference of two Leaves.*

Consider an AVL tree of height h. What is the maximum possible difference of the depths of two leaves? Describe which structure such trees need to have, and draw examples of corresponding trees for every $h \in \{2, 3, 4\}$. Derive a recursive formula (depending on h), solve it and use induction to prove the correctness of your solution. Provide a detailed explanation of your considerations.

Hint: *For the proof the principle of complete induction can be used. Let $\mathcal{A}(n)$ be a statement for a number $n \in \mathbb{N}$. If, for every $n \in \mathbb{N}$, the validity of all statements $\mathcal{A}(m)$ for $m \in \{1, \ldots, n-1\}$ implies the validity of $\mathcal{A}(n)$, then $\mathcal{A}(n)$ is true for every $n \in \mathbb{N}$. Thus, complete induction allows multiple base cases and inductive hypotheses.*