ShanghaiTech University

CS101 Algorithms and Data Structures
Fall 2024

Homework 6

Due date: November 11, 2024, at 23:59

1. Please write your solutions in English.
2. Submit your solutions to Gradescope.
3. Set your FULL name to your Chinese name and your STUDENT ID correctly in Gradescope account settings.
4. If you want to submit a handwritten version, scan it clearly. `CamScanner` is recommended.
5. We recommend you to write in LaTeX.
6. When submitting, match your solutions to the problems correctly.
7. No late submission will be accepted.
8. Violations to any of the above may result in zero points.

1. (10 points) Multiple Choices

Each question has **one or more** correct answer(s). Select all the correct answer(s). For each question, you will get 0 points if you select one or more wrong answers, but you will get 1 point if you select a non-empty subset of the correct answers.

Write your answers in the following table.

(a)	(b)	(c)	(d)	(e)

(a) (2') Which of the following statement(s) is/are true for an AVL tree?

A. Inserting an item can unbalance non-consecutive (not directly connected) nodes on the path from the root to the inserted item before the restructuring.

B. Inserting an item can cause at most one node imbalanced before the restructuring.

C. Only at most one node-restructuring has to be performed after inserting an item.

D. Removing an item in leaf nodes can cause at most one node imbalanced before the restructuring.

(b) (2') Consider an AVL tree whose height is h, which of the following is/are true?

A. This tree contains $\Omega(\alpha^h)$ nodes, where $\alpha = \frac{\sqrt{5}-1}{2}$.

B. This tree contains $\Theta(2^h)$ nodes.

C. This tree contains $O(h)$ nodes in the worst case.

D. None of them above.

(c) (2') Which of the following about the comparison between AVL-tree and BST is/are true? Suppose n is the number of nodes in the tree.

A. The cost of searching an AVL tree is $O(\log n)$ but that of a complete binary search tree is $O(n \log n)$.

B. The cost of searching an AVL tree is $\Theta(\log n)$ but that of a binary search tree is $O(n)$.

C. The cost of searching a binary search tree with height h is $O(h)$ but that of an AVL tree is $O(\log n)$.

D. The corrections of both Insertion and Erasion cost $\Theta(\log n)$ time in worst cases in an AVL tree.

(d) (2') Which of the following statements is/are true for an AVL tree? Here one balance correction means a single rotation (in left-left or right-right cases) or a double rotation (in left-right or right-left cases).

A. In an AVL tree, during the insert operation there are at most two rotations needed.

B. Inserting an item causes at most one node imbalance before checking if any balance correction is needed.

C. At most one balance correction has to be performed after inserting an item.

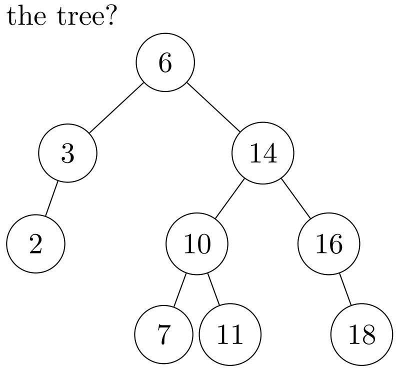
D. At most one balance correction has to be performed after removing an item.

Solution:

A. The AVL property is restored on every operation. Therefore, inserting another item will require at most two rotations to restore the balance.

C. Suppose x is the deepest imbalanced node. After the imbalance of x is corrected, the height of x's subtree always becomes the same as when the item is not inserted. Therefore other ancestors of x will not need to be corrected.

(e) (2') You are given an AVL tree as a blow. Suppose we promote the minimum element in the right sub-tree when erasing a node with 2 children. Which of the following operation sequences will cause 2 imbalances that must be corrected in total in order to rebalance the tree?



A. Erase 2, 6.

B. Insert 4, 5, 12. Erase 2.

C. Erase 6, 2, 3. Insert 20.

D. Insert 1, 0, 4, 13, 19.

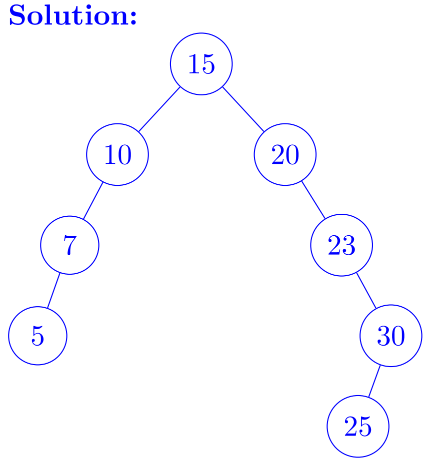
2. (14 points) BST and AVL Tree

In this question we are going to see what's the difference between general Binary Search tree and AVL tree.

Note: We uniformly stipulate that when erasing a non-leaf node x, we will fill its successor (the minimum value greater than x among all child nodes) to its original position. If x has no successor, we will fill its predecessor (the maximum value less than x among all child nodes) to its original position.

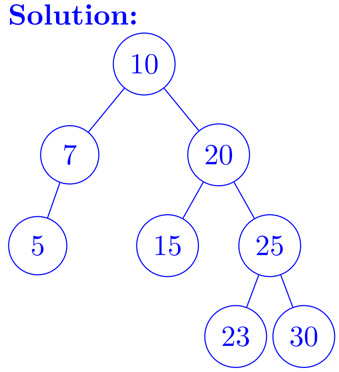
(a) (3') Given an empty Binary Search tree, insert the sequence of integers $15, 20, 23, 10, 7, 5, 30, 25$ from left to right into the tree. Draw the final BST.

Solution:



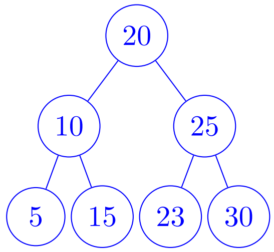
(b) (3') Given an empty AVL tree, insert the same sequence in part (a), draw the final AVL tree.

Solution:



(c) (3') For the final AVL tree in the question above, delete 7. Draw the AVL tree after deletion.

Solution:



(d) (5') For an AVL tree, define D = the number of descendants of the left child of the root - the number of descendants of the right child of the root. Then what is the maximum of D for an AVL tree with height n?

$D_{max} = k_1 \times 2^n + k_2 \times B^n + k_3 \times (-\frac{1}{B})^n$, please write down the value of B and k_i.

Solution: The maximum of D should be the maximum number of the nodes in the left subtree minus the minimum number of the nodes in the right tree.

$$D_{max} = L_M - R_m$$

Therefore, the height of the left subtree shoud be $n-1$ and the height of the right substree shuold be $n-2$. The number of nodes in the left subtree is maximized when the left subtree is a perfect tree. Then we have

$$L_M = 2^n - 1$$

The right subtree should also be an AVL tree. The minimum number of the nodes in the right subtree is the minimum number of the nodes in an AVL tree with height $n-2$. Let the minimal number of the nodes in an AVL tree with a given height h be denoted as $F(h)$. Each AVL tree can be separated into left subtree, right subtree, and root. Both the left subtree and the right subtree also have a minimal number of nodes. One of the heights of the subtree should be $h-1$ and the other should be $h-2$. Therefore, we have:

$$\begin{cases} F(h) = 1 + F(h-1) + F(h-2) \quad (*) \\ F(0) = 1 \\ F(1) = 2 \end{cases}$$

Then we are going to use some common tricks in solving general term in sequences. Notice that it has the potential to be actually homogeneous, which indicates that it could be rewritten into some form of $G(n) = F(n) + b$ and $G(n) = k_1 G(n-1) + k_2 G(n-2)$. Then $G(n) + b = G(n-1) + b + G(n-2) + b + 1$. We want to eliminate the constants, so $b = b + b + 1$. Hence,$b = -1, G(n) = F(n) + 1$ and the $(*)$ can consequently be represented as follows:

$$F(h) + 1 = [F(h-1) + 1] + [F(h-2) + 1]$$

Let $G(h) = F(h) + 1$

$$\begin{cases} G(h) = G(h-1) + G(h-2) \\ G(0) = 2 \\ G(1) = 3 \end{cases}$$

The chacracteristic funciton is

$$x^2 = x + 1$$
$$x_{1,2} = \frac{1 \pm \sqrt{5}}{2}$$

Then we have

$$\begin{aligned} G(h) &= c_1(x_1)^h + c_2(x_2)^h \\ G(0) &= c_1 + c_2 = 2 \\ G(1) &= \frac{1+\sqrt{5}}{2}c_1 + \frac{1-\sqrt{5}}{2}c_2 = 3 \\ c_{1,2} &= \frac{5 \pm 2\sqrt{5}}{5} \end{aligned}$$

Therefore

$$\begin{aligned} G(h) &= \frac{5+2\sqrt{5}}{5} \cdot \left(\frac{1+\sqrt{5}}{2}\right)^h + \frac{5-2\sqrt{5}}{5} \cdot \left(\frac{1-\sqrt{5}}{2}\right)^h \\ F(h) &= \frac{5+2\sqrt{5}}{5} \cdot \left(\frac{1+\sqrt{5}}{2}\right)^h + \frac{5-2\sqrt{5}}{5} \cdot \left(\frac{1-\sqrt{5}}{2}\right)^h - 1 \end{aligned}$$

Then we can calculate R_m

$$R_m = F(n-2) = \frac{5+2\sqrt{5}}{5} \cdot \left(\frac{1+\sqrt{5}}{2}\right)^{(n-2)} + \frac{5-2\sqrt{5}}{5} \cdot \left(\frac{1-\sqrt{5}}{2}\right)^{(n-2)} - 1 = G(n-2) - 1$$

Hence, by definition, in the best case we have maximum descendants when it is a balanced BST. In the worst case, we have $F(n-2) = G(n-2) - 1$ descendants.

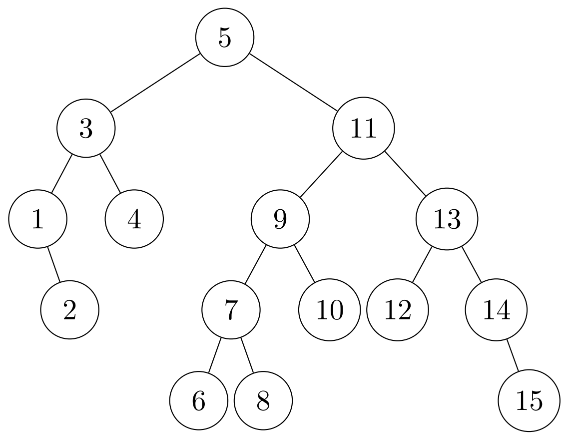
$$\begin{aligned} D_{max} &= L_M - R_m \\ &= (2^n - 1) - (G(n-2) - 1) \\ &= 2^n - \frac{5+2\sqrt{5}}{5} \cdot \left(\frac{1+\sqrt{5}}{2}\right)^{(n-2)} - \frac{5-2\sqrt{5}}{5} \cdot \left(\frac{1-\sqrt{5}}{2}\right)^{(n-2)} \end{aligned}$$

Therefore,$k_1 = 1, B = \frac{1+\sqrt{5}}{2}, k_2 = -\frac{5+\sqrt{5}}{10}, k_3 = -\frac{5-\sqrt{5}}{10}$

3. (8 points) AVL tree operations

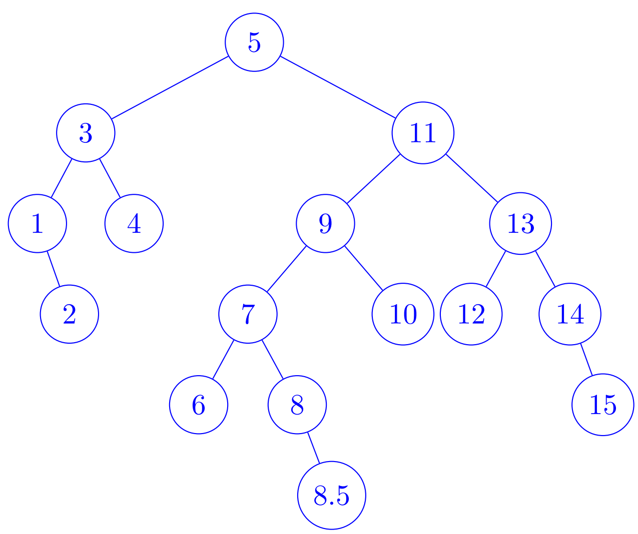
Here is an AVL tree. Denote it as T.

Note Again: We uniformly stipulate that when erasing a non-leaf node x, we will fill its successor (the minimum value greater than x among all child nodes) to its original position. If x has no successor, we will fill its predecessor (the maximum value less than x among all child nodes) to its original position.



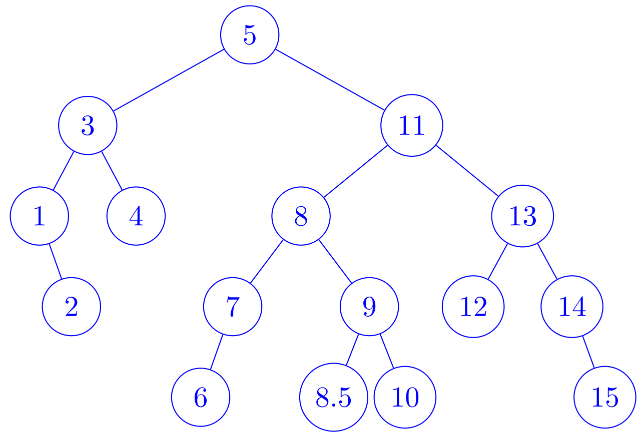
(a) (2') Insert 8.5 into T. Draw the AVL tree before checking if any balance correction is needed.

Solution:



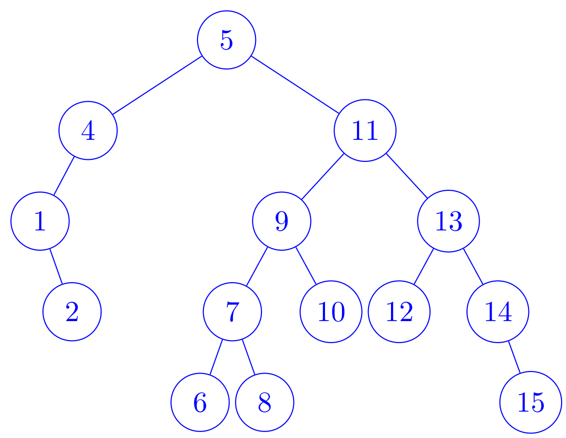
(b) (2') Insert 8.5 into T. Draw the AVL tree after balance corrections.

Solution:



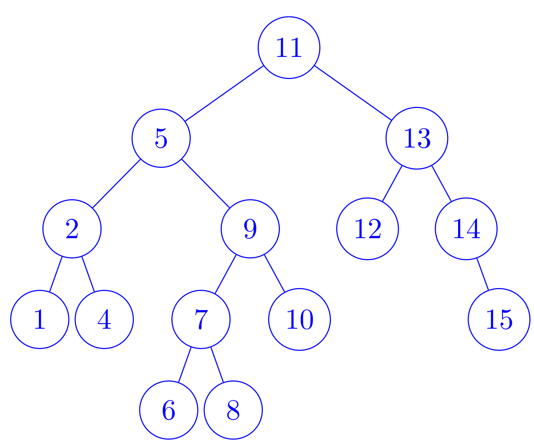
(c) (2') Remove 3 from T (**NOT from the previous answer!**). Draw the AVL tree after replacing and before checking if any balance correction is needed.

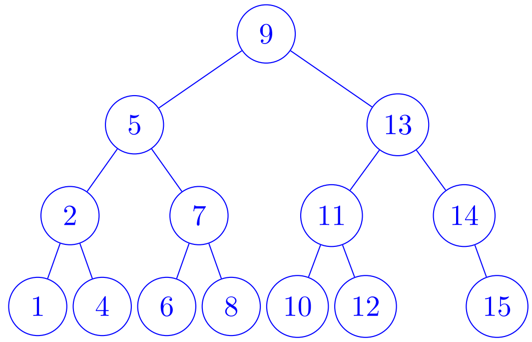
Solution:



(d) (2') Remove 3 from T. Draw the AVL tree after balance corrections.

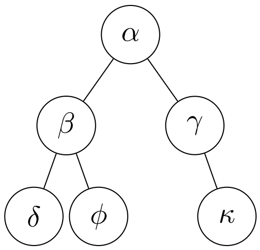
Solution:




9
5
13
2
7
11
14
1
4
6
8
10
12
15

4. (11 points) Alphabet Cognitive Battle

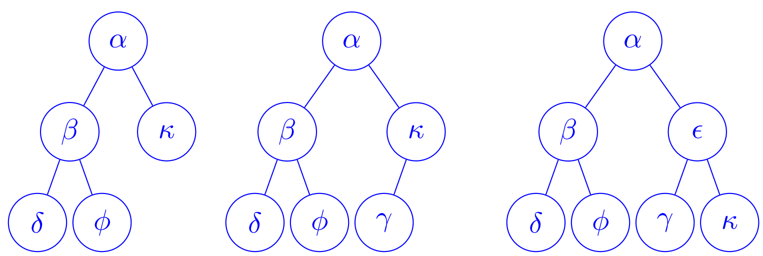
Consider the AVL tree below. Each symbol represents a unique object stored in the AVL tree.



The following 4 questions are **independent** of each other, i.e. for each question, your answer should be built on the original AVL Tree above, instead of the AVL Tree from the answer of the last question.

(a) (3') If we delete object γ, then insert object γ, then insert new object ϵ ($\gamma < \epsilon < \kappa$), please draw the AVL tree after each insert/delete operation.

Solution:



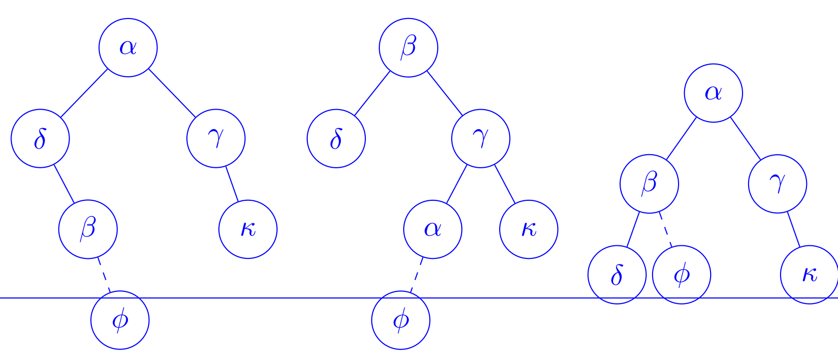
(b) (3') If we want to insert a new object ϵ (not equal to the 6 objects)but we don't want to change the tree's current structure to maintain balance. What are the ranges of the object we can insert? (Note: the range is denoted by the objects, for example, $\alpha < \epsilon < \beta, \epsilon < \delta$).

Solution:

$$\epsilon < \delta, \delta < \epsilon < \beta, \beta < \epsilon < \phi, \phi < \epsilon < \alpha, \alpha < \epsilon < \gamma$$

(c) (2') If we know the last object we insert is ϕ. What is the tree before inserting it? Symbols must be unique. You may only use the 6 printed symbols. If there are more than 3 correct answers, giving only 3 correct answers would lead to full credit.

Solution:



(d) (3') If we don't know the last object we insert, but we know we change the tree's structure to maintain balance. What is the object we may insert? Please write down the possible objects. Symbols must be unique. You may only use the 6 printed symbols. If there are more than 3 correct answers, giving only 3 correct answers would lead to full credit. Giving any wrong answer will lead to zero credit.

Solution:

$$\beta, \delta, \kappa, \phi$$