ShanghaiTech University

CS101 Algorithms and Data Structures

Fall 2020

Homework 7

Due date: 23:59, October 26, 2020

1. Please write your solutions in English.
2. Submit your solutions to gradescope.com.
3. Set your Full Name to your Chinese name and your STUDENT ID correctly in Account Settings.
4. If you want to submit a handwritten version, scan it clearly. Camscanner is recommended.
5. When submitting, match your solutions to the according problem numbers correctly.
6. No late submission will be accepted.
7. Violations to any of above may result in zero score.

1: (4*1') Multiple Choices

Each question has one or more correct answer(s). Select all the correct answer(s). For each question, you get 0 point if you select one or more wrong answers, but you get 0.5 point if you select a non-empty subset of the correct answers.

Note that you should write you answers of section 1 in the table below.

Question 1	Question 2	Question 3	Question 4
C	ACD	A	ACD

Question 1. *Which of the followings are true?*

(A) For a min-heap, in-order traversal gives the elements in ascending order.

(B) For a min-heap, pre-order traversal gives the elements in ascending order.

(C) For a BST, in-order traversal gives the elements in ascending order.

(D) For a BST, pre-order traversal gives the elements in ascending order.

Question 2. *Which of the following statements are true for an AVL-tree?*

(A) Inserting an item can unbalance non-consecutive 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) Removing an item in leaf nodes can cause at most one node imbalanced before the restructuring.

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

Question 3. *Consider an AVL tree whose height is h, which of the following are true?*

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

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

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

(D) None of the above.

Question 4. *Which of the following is TRUE?*

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

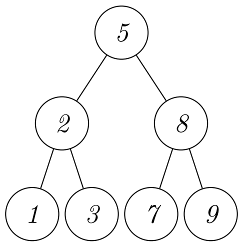
(B) The cost of searching an AVL tree is $O(\log n)$ *but that of a complete binary tree is* $O(n \log n)$

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

(C) The cost of searching an AVL tree is $O(nlogn)$ *but that of a binary search tree is* $O(n)$

2: (3'+8') BST and AVL Tree

Question 5. *Draw a valid BST of minimum height containing the keys 1, 2, 3, 5, 7, 8, 9.*



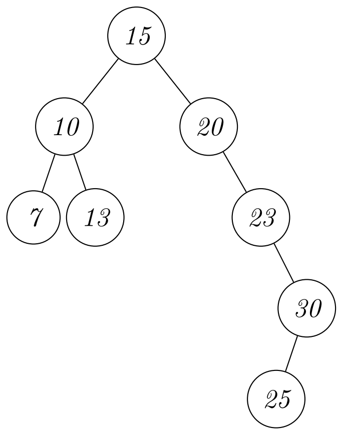
Question 6. *Show that for* $\forall N > 0$*, a complete BST containing N items (without duplicates) is unique. By unique we mean the BST is the only complete BST that contains exactly those N items. By complete we mean every level of the tree, except possibly the last, is completely filled, and all nodes are as far left as possible.*
Hint*: Try to prove by contradiction.*

To prove N can be any non-zero value:

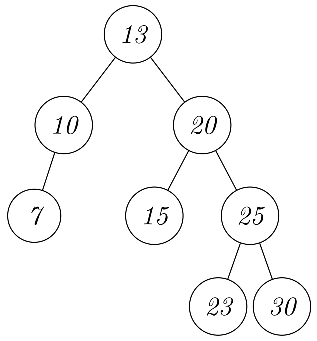
We know that there is at least one complete BST containing a specific set of N items, let's call it T. We also know that there is only one way to arrange the N nodes to form a complete BST (e.g. a linear chain of N nodes is not a complete BST). Now we show that the arrangement of values in T is unique. Suppose we take two different values x and y in T; without loss of generality, assume $x < y$ *(the same argument applies for* $y > x$*, and we know* $x \mathrel{!}= y$ *because all items are unique). If we try to swap the places of x and y, we would obtain a tree where y is to the absolute left of x, which would violate the property of BST's where all items to the absolute left of a node are less than or equal to the item in that node.*

Question 7. *BST and AVL Tree*

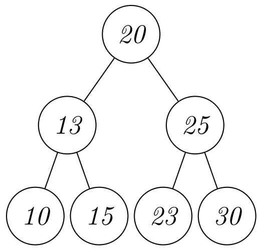
(1) Given an empty Binary Search Tree, insert the sequence of integers 15, 20, 23, 10, 13, 7, 30, 25 *from left to right into the BST. Draw the final BST.*



(2) Replace the BST in the question (1) with an AVL tree and insert the same sequence of integers into it. Draw the final AVL tree.

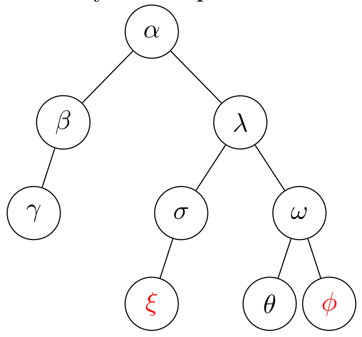


(3) For the final AVL tree in the question (2), delete 7*. Draw the AVL tree after deletion.*

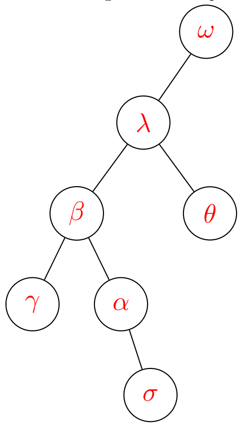


3: Magic BST

Consider the binary search tree below. Each symbol represents an object stored in the BST.



Question 8. *(3') Based on the ordering given by the tree above, fill in the BST below with valid symbols. Symbols must be unique. You may only use the 7 printed symbols (do not include any symbols from part b).*



Question 9. *(3') For each of the insertion operations below, use the information given to "insert" the element into the* ***TOP TREE WITH PRINTED SYMBOLS, NOT THE TREE WITH YOUR HANDWRITTEN SYMBOLS*** *by drawing the object (and any needed links) onto the tree. You can assume the objects are inserted in the order shown below. You should not change anything about the original tree; you should only add links and nodes for the new objects. If there is not enough information to determine where the object should be inserted into the tree, circle "not enough information". If there is enough information, circle "drawn in the tree above" and* ***draw in the tree AT THE TOP OF THE PAGE.***

insert(ϕ):	$\phi > \omega$	[✓ Draw In Tree Above]	Not enough Information
insert(χ):	$\chi > \gamma$	Draw In Tree Above	[✓ Not enough Information]
insert(ξ):	$\alpha < \xi < \sigma$	[✓ Draw In Tree Above]	Not enough Information
insert(μ):	$\lambda < \mu < \omega$	Draw In Tree Above	[✓ Not enough Information]