Written Homework #1

Data Structure 2021 Spring
Last update: Apr 22, 2021

Deadline: 11:59pm, April 29, 2021

Submission Instruction (i): You may write your answer using MS word or scan your hand-written answer, but your submission **must be a single "PDF" file**. You will get a 15% penalty if you don't follow this instruction. Once you have your PDF file, please submit your file to eTL.

Submission Instruction (ii): For all the questions, you should shortly justify your answer (please do not write a long answer!). You should not simply answer without any explanation.

1. (10 pts) Describe how to implement two stacks in one array. In particular, you are given two stacks A and B, and explain how following operations would be implemented using one array:
   - pushA()
   - pushB()
   - popA()
   - popB()
   - topA()
   - topB()

   You may write (short) pseudo code, or you may shortly describe the algorithm of each operation.

2. (10 pts) In order to store the height information for each node in a N-node AVL tree, how many bits should you allocate for each node? Answer using landau symbols, either big-oh or big-theta notation.

3. (10 pts) The commutative property is the mathematical property in which changing the order of operands does not change the evaluation result. Now, we look at the "delete" operation in a binary search tree if it's commutative. Specifically, suppose you delete item A and then item B from the binary search tree, BST. Would it be the same as deleting item B and then item A from that BST?
   (a) Answer "True" if you think this is true. Answer "False" if this is false.
   (b) If you answered "True" in (a), proof it. Otherwise, provide a counterexample.

4. (10 pts) Proof the following statement: The maximum number of nodes in a binary tree of height h is $2^{h+1} - 1$.

5. (20 pts) In order to sort n numbers, you may first construct a binary search tree having all these numbers (i.e., keep performing "n" insert operations to insert all the numbers one by one). After constructing the tree, you can perform in-order tree traversal, which would visit the numbers in a sorted way. Answer following two questions. Each answer should be represented with either big-theta or big-oh notations, and you should shortly justify your answer as well.
   (a) What would be the worst-case runtime of doing this binary search tree based sorting algorithm?
   (b) What would be the best-case runtime of doing this binary search tree based sorting algorithm?

6. (20 pts) Suppose you are given a binary search tree with n-nodes in which the average depth of a node is $\Theta(lg\ n)$. What would be the asymptotic upper bound on the height of this tree?
   Hint: You may imagine a binary search tree, looking similar to a complete binary tree but one leaf node is hanging down with a certain length.

7. (20 pts) Suppose you are given an empty AVL tree. Now you insert the following keys in order: $1, 2, \ldots, 2^k - 1$. Show that after these $2^k - 1$insertions, all the leaf nodes have the same depth.