

CSDS233 Spring Session 6

SI Leader: Jakob Danninger

2/16/2023

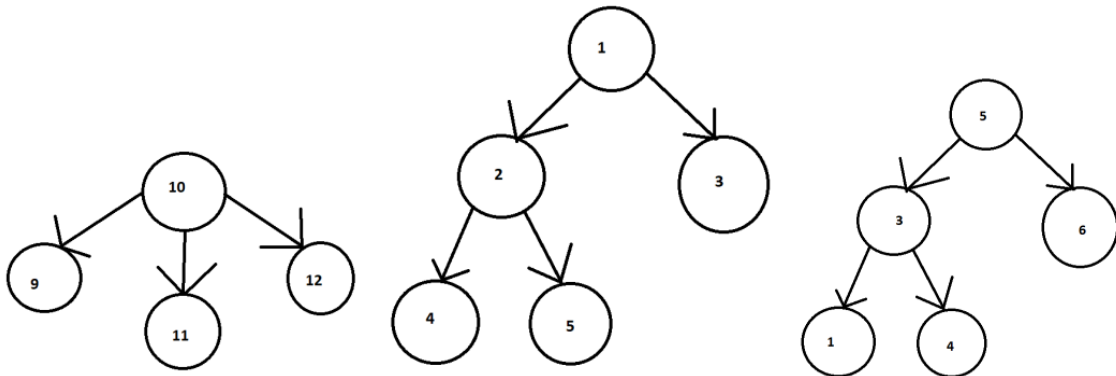
Disclosure: This is a supplement to class, not a replacement. This should not be your only study activity for exams, it should aid you in studying. I do not have the actual exam so questions here will differ from those on the exam.

Session Objectives:

- 1) Understand what a binary tree is and how to implement recursive searching
- 2) Understand how to add and delete a node (this part is tough)

Worksheet

- 1) Classify the following tree as either: binary search tree, binary tree, neither. Explain why?



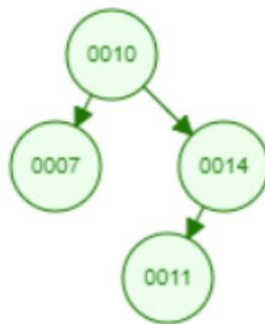
- 2) Draw the final binary search tree after we add the following nodes in order: 13, 9, 8, 15, 14, to the tree below



3) Is this (the tree from question 2 after you added everything) a balanced tree? (ask me what a balanced tree is if you don't remember or what balance even is)

4) Draw a BST by adding the following numbers in the following order: 15, 16, 17, 12, 11, 13, 12.5

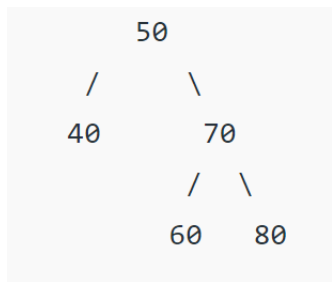
5) Draw the tree after delete node 11



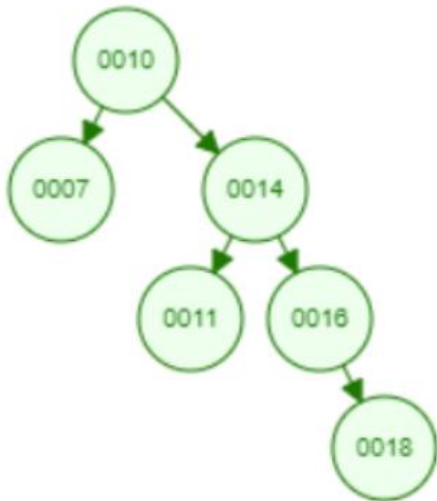
6) Delete node 16



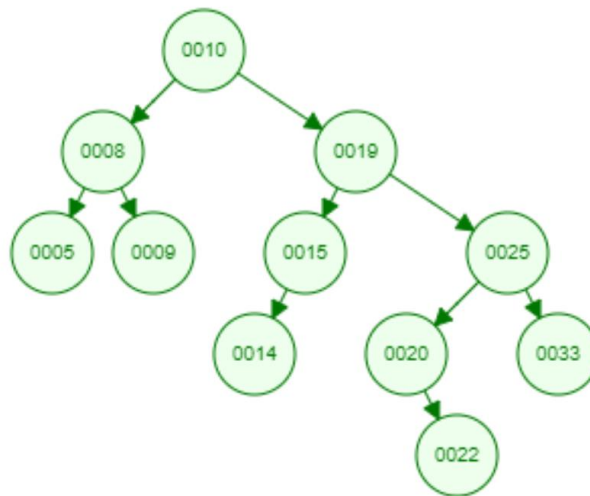
7) Delete node 50



8) Delete 10



9) Delete 19 (good luck)



Coding: <https://github.com/jdanninger/CSDS233-Supplemental-Instruction> – code is here

The code is under the session 6 folder along with the coding answer, you should copy and Paste the code into your development environment of choice or clone the code using git.

I want you to code and add method that adds a value into the correct location in the Binary search tree. I wrote test cases in the main.

This is a hard task and frankly you probably don't have enough time, but as long as you can understand the answer, I am confident it's a massive help.

Next session is next week Tuesday and will cover the intro to balance, rotations, and AVL trees

Further Practice (totally optionally and time consuming, but cool):

<https://leetcode.com/problems/delete-node-in-a-bst/>

Leetcode is a great platform to prepare for technical interview, I highly recommend for when you start getting CS Internship/Job interviews as prep, though some of their questions are legitimately hard, so be ready for that, I get stumped on how to make the most efficient iteration all the time.