BST AVL
PRACTICE PROBLEMS

By: Zahra Bukhari

COUGARCS

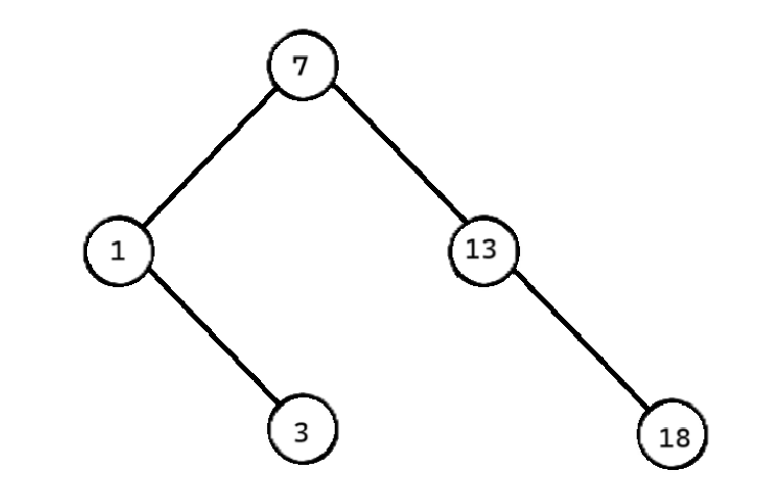
Time Complexities

1. What are the best/average/worst case time complexities for each of the following?
    - Inserting an element into a BST tree.
    - Deleting an element from a BST tree.
    - Finding an element in a BST tree.
    - Inserting an element into an AVL tree.
    - Deleting an element from an AVL tree.
    - Finding an element in an AVL Tree.

Tracing

1. To store an AVL in an array, the children of an index i are at the indices 2i+1 (left) and 2i+2 (right).
    - Insert the even numbers from **0 to 16** (inclusive) into an array where the root is at index 0. What is the final array that holds this AVL?
    - What is the pre-order traversal for this tree?
    - What is the in-order traversal for this tree?
    - What is the post-order traversal for this tree?
    - Check your Answers

2. Begin with this BST:

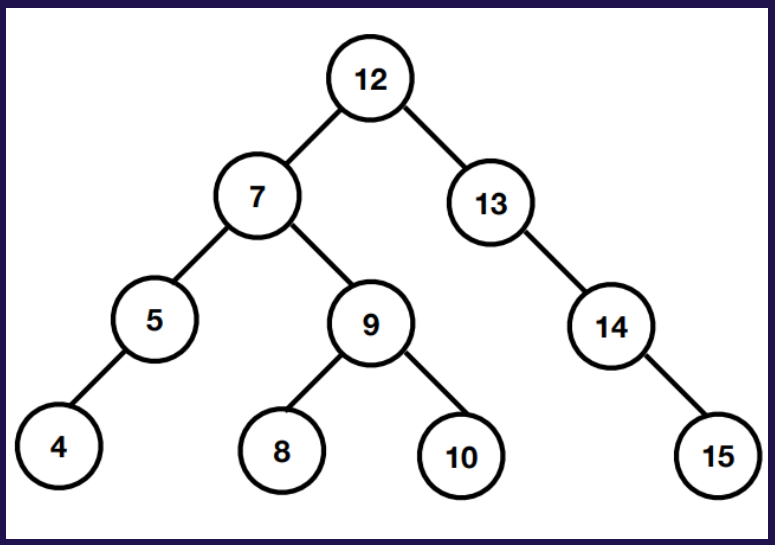


    - Insert: 0, 12, 21, 2, 9, 10
    - Write the Pre-Order for the new tree
    - Delete: 13, 12, 3, 7
    - Write the Post-Order for the new tree
    - Check your Answers

3. Describe, write code for, and show examples of the four rotations expected to maintain height balance when inserting an element into an AVL tree.

Programming

1. Write the function which produced the output below



   Output: 4, 5, 8, 10, 9, 7, 15, 14, 13, 12
2. Write a function minNode() that takes the Node* root of a BST and returns the smallest value in the tree.
3. Write a function countNodes() that takes the Node* root of a BST and returns the total number of nodes in a BST.
4. Write a function countFullNodes() that takes the Node* root of a BST and returns the number of nodes that have both left and right children.

GOODLUCK !

And remember, if you need help, you can message us at CougarCS Tutoring, or find us at PGH!

COUGARCS