

CSCI 104L Lecture 21: Binary Search Trees

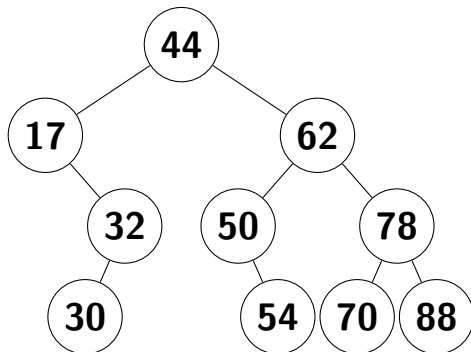
Question 1. How would you insert a new value into a (non-balanced) binary search tree?

Question 2. For the integers 1 through 7, is there an order you can insert them into an (initially empty) binary search tree such that any search will be efficient?

Question 3. For the same set, is there an order that will provide a “bad” binary search tree?

Deleting from a Binary Search Tree

Consider the following binary search tree:



Question 4. Starting from the above tree, what does the tree look like if we remove 30?

Question 5. What if we had deleted 17 *instead*?

Question 6. What if we had deleted 62 *instead*?

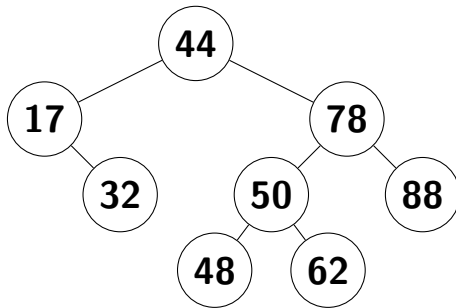
Question 7. What if we had deleted 44 *instead*?

AVL Trees

We say that a tree is an **AVL Tree** if the following two conditions both hold:

- The **binary search tree** property holds for all nodes.
- For every node v of T , the heights of the children of v differ by at most 1. This is referred to as the **height-balance property**.

Note that this means that any subtree of an AVL tree is itself an AVL tree.



Question 8. Is the above tree an AVL Tree? Why or why not?

Question 9. What is the minimum number of nodes an AVL tree can have if it has height h ?

Question 10. How long does `find(v)` take in an AVL tree?