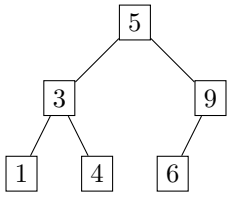
CS 161 **Pre-Lecture Exercise for Lecture 7**
Autumn 2017 **Date:** October 16, 2017

Pre-lecture exercises will not be collected for credit. However, you will get more out of each lecture if you do them, and they will be referenced during lecture. We recommend **writing out** your answers to pre-lecture exercises before class. Pre-lecture exercises usually should not take you more than 20 minutes.

In this pre-lecture exercise, we'll recall (or see for the first time) *binary search trees* (BSTs). Here is an example of a binary search tree:

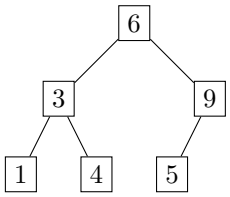


T is a binary search tree.

It is a binary tree, where each node has a *key*, that has the **BST property:**

> For any node n, $n.key$ is larger than all of the keys in the subtree under n's left child, and is smaller than all of the keys in the subtree under n's right child.

For example the following is **not** a BST (why not?):



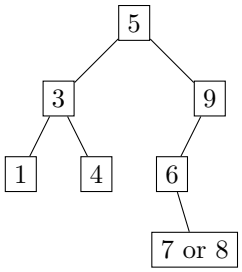
T' is a **NOT** a binary search tree.

If you have never seen binary search trees before, you might want to read up on then in CLRS ahead of lecture (Sections 12.1, 12.2, 12.3). Answer the following exercises, which refer to the BST T drawn above.

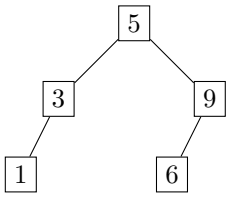
1. Suppose you wanted to insert a node with key 7 into T. Where should it go? How about a node with key 8? Try to think about how you'd write an algorithm to insert a node with any given key into T.

2. Suppose you wanted to delete the node with key 4 from T. What would happen? How about the node with key 3? Try to think about how you'd write an algorithm to delete a node with any given key from T.

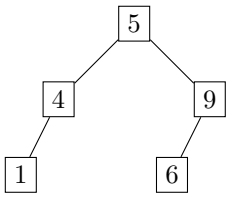
SOLUTIONS: Here's how we would insert/delete nodes as required above. (Note, there is *not* a unique way to do this!)



Inserting 7 or 8



Delete 4



Delete 3

To see how we would implement these operations in pseudocode, see Lecture 7 slides, Lecture notes, or CLRS.