Math 382

Homework 5

Due Friday, March 20

1. Say we have a set of n items and want to build a binary search tree. Knowing nothing about the data, we can only interact with it through comparisons. Show that any algorithm that builds a binary search tree must take at least $\Omega(n \log(n))$ time.

2. In class we gave pseudocode for finding the successor of a given node in a binary search tree. Write the pseudocode for finding the predecessor, the node with the highest value less than that of the given node.

3. Given two nodes x and y that we wish to delete from a binary search tree, we could first delete x and then y or vice versa. Will the order of deletions affect the resulting tree? Prove that it won't or give a counterexample where it will.

4. Consder a red-black tree with n nodes created by calling the insert function n times. Show that as long as $n > 1$, the tree will always have at least one red node.

5. Consider a red-black tree that begins empty. We then insert in order the keys 41, 38, 31, 12, 19, 8. We then delete each key in the order 8, 12, 19, 31, 38, 41. Draw the tree that exists at each step of this process. (Please be careful to give yourself plenty of room and draw these trees clearly.)