CSc 220: Algorithms
Homework 5
Due in Class on Thursday October 15

Return the homework written on sheet(s) of paper with your name and CSc220 written at the top of each sheet. Please staple multiple sheets together. Remember that collaboration is allowed, but that you must write the solution on your own. Also you must acknowledge all collaborators and all sources (other than the textbook) in your solutions. Each problem is worth 10 points.

Problem 1: We saw in class that given a binary search tree with $n$ elements, we can output the elements in sorted order using a $O(n)$ in-order tree walk. Use this fact and the $\Omega(n \log n)$ lower bound on the number of comparisons needed to sort $n$ elements, to prove a lower bound on the number of comparisons needed to build a binary search tree from a set of $n$ elements.

Problem 2: Give an algorithm that traverses the tree and stores on each node the height of that node in the tree. Show how to maintain this information when you insert or delete nodes. Argue the correctness and analyze the complexity of your algorithm.

Problem 3: Consider a binary search tree augmented with the following information. At each node $x$ we also store $m(x)$: the number of nodes in the subtree rooted at $x$ (including $x$). We require that for every node $x$ in the tree $|m(L(x)) - m(R(x))| \leq 1$.

- Prove that $h = O(\log n)$ where $h$ is the height of a tree with $n$ nodes satisfying this property [4pts]
- Show how to maintain this property when you do insertions or deletions, via rotations [6pts]