Homework Assignment #7
Due: May 12, 4:00 p.m.

1. If T is a binary search tree, let $R(T)$ be the length of the path from the ro ot to the maximum element in the tree.

(a) Suppose some node in T has a left child. Show that there is a node x in T such that the tree T' that results from performing RIGHT-ROTATE(T, x) has $R(T') > R(T)$.

(b) Let T_1 and T_2 any be two binary search trees that contain the same set of n elements. Prove that there is a sequence of $O(n)$ RIGHT-ROTATES and LEFT-ROTATES that converts T_1 into T_2. Hint: start by converting one tree into a tree where no node has a left child.

2. Suppose we want to implement a dictionary whose elements have integer keys. We want to handle range-sum queries: RANGE-SUM(a, b) returns the sum of all keys in the dictionary that are greater than a and less than b. Describe how to augment a B-tree to answer range-sum queries efficiently. (If your algorithm takes $\Theta(n)$ time to answer a range-sum query on a dictionary with n elements, it is much too slow.) You should describe what changes are necessary for the B-tree insertion and deletion routines (if any), and those changes should not change the asymptotic running time for insertions and deletions. Use Θ notation to state the running time for your range-sum query as a function of n and t, the B-tree's minimum degree. Also state the number of nodes that must be accessed by your range-sum query.

Assume that key values are at most n and that one word of memory can store $\log n$ bits, so that adding two $(\log n)$-bit numbers can be done in a single step.