CSC-361: Problem Set for Binary Search Trees

1. Indicate true / false for each statement. For these statement, you may assume the *visit* operation prints the key stored at the node.

   (a) It is possible for a levelorder traversal of a binary search tree to result in an ordered list of keys.

   (b) It is possible for a levelorder traversal of a binary search tree to result in a reverse-ordered list of keys.

   (c) A binary search tree can store arbitrary objects as keys.

   (d) The inorder predececssor of the minimum node in a binary search tree does not exist.

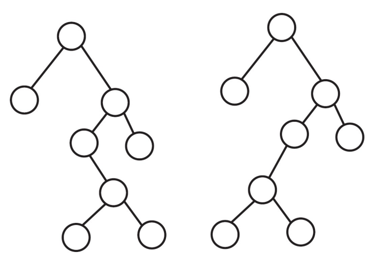
   (e) For every node except the minimum node in a binary search tree, the inorder predececssor of a node n corresponds to the maximum of the left subtree of n.

   (f) The search operation for a binary search tree consisting of $n > 0$ nodes is $O(n)$.

   (g) In the best case, the search operation in a binary search tree consisting of $n > 0$ nodes is $\Theta(1)$.

2. Show that the maximum number of nodes in a binary tree of height h is $2^{h+1} - 1$.

3. Suppose T_1 and T_2 are binary search trees with respective roots r_1 and r_2. Given a node u, let `LEFT`(u) and `RIGHT`(u) refer to the respective left and right children of u. We say that T_1 and T_2 are isomorphic if they have the exact same structure (in other words, the drawings of T_1 and T_2 are the same). For example, the two binary search trees below are not isomorphic.



   Give pseudocode for `isIsomorphic`(T_1, T_2) which determines if two binary search trees are isomorphic. Assume `ROOT`(T) returns the root of a tree T.