CS2302 - Data Structures

Fall 2020

Exercise - Binary Search Trees

1. Write the function *in_order_stack(t)* that receives a reference to a binary search tree t and performs the *in order* traversal of the tree using a stack instead of recursion.

2. Write the function *tree2List(t)* that receives a reference to a binary search tree t and returns a sorted list containing the elements in the tree. This should be done in time $O(n)$. Remember that sorting takes time $O(n \log n)$

3. Write the function *list2Tree(L)* that receives a sorted list L and builds and returns a balanced (as much as possible) binary search tree containing the elements of L. This should be done in time $O(n)$.

4. A binary tree is full if all its nodes have either 0 or two children). Write the function *is_full(t)* that receives a reference to a binary search tree t and determines if the tree is full.

5. A binary tree is perfect if all its nodes have either 0 or two children and all its leaves have the same depth. Write the function *is_perfect(t)* that receives a binary search tree t and determines if the tree is perfect. Hint: find the height and the number of nodes in the tree.