The purpose of this lab is to gain experience with binary search trees. Use the code provided in the class web page as starting point. The files provided contain the definition of TreeNode objects and several operations, including height, insertion, inorder traversal, extraction of a sorted array from a binary search tree, construction of a balanced binary search tree from an ordered array, among others.

Extend the code provided to support the following operations:

1. Determine if a tree is a binary search tree
2. Determine if a tree is balanced (that is, the difference between the minimum and maximum depth of any two leaves is at most one)
3. Return the minimum element in a BST
4. Return the maximum element in a BST
5. Delete an element from the BST
6. Remove all duplicate elements from a BST
7. Determine if two BSTs have the same elements
8. Determine if two BSTs are identical (that is, they have the same elements and shape)
9. Perform an inorder tree traversal using a stack instead of recursion
10. Print the contents of a tree ordered by depth, that is, first print the node at depth 0 (the root) then the nodes at depth 1, then the nodes at depth 2, and so on. Hint: use a queue

Write a main method that shows the functionality of your program and, as usual, write a report describing your program and experiments.