Problem

Build a binary search tree using Java and it must demonstrate the following:

1. Find your binary search tree's maximum depth
2. Calculate size of the tree with recursive and iterative methods
3. Print traversal of tree with pre-order, post-order and in-order methods

Solution

The program used the following classes to perform the requested operations:

1. **BSTNode:** Linked List node class that is used in the BinarySearchTree class
2. **BSTInterface:** Binary Tree Interface which will be implemented in BinarySearchTree class. This class extends CollectionInterface and Iterator class form java library
3. **BinarySearchTree:** This class implements the BSTInterface the class that defines our Binary Search Tree and has all required methods
4. **CollectionInterface:** Collection interface that will be extended by the BSTInterface. Since our BST uses a linked list that is a collection.
5. **LinnkedQueue:** The ADT LinnkedQueue class is used in the getIterator method that allows us to view our tree in the order we desire (PreOrder, InOrder, or PostOrder)
6. **LinkedStack:** This classes will be used in finding out the size of our tree iteratively
7. **LLNode:** This class is used in both LinnkedQueue and LinkedStack
8. **Order:** an enum used to support our Iterator and getIterator methods

Finally, **The BinarySearchTreeClient** class has the main method that uses instantiate a BinarySearchTree, adds Integer elements to it, finds out the min, max, depth, and the different variations of order to view the tree contents.