Lab 12 – Binary Search Trees

Download the bst.py and astack.py files from D2L and complete the functions as described below.

# Part A – Non-Recursive In-Order Traversal

Although using recursion is an easy way to implement an in-order traversal of a tree, it is possible to use iteration. To do this, you need a stack to keep track of where you were so that you can return to the place where you made you decision to go down one of the branches of the tree.

The algorithm is as follows:

1. Create an empty stack S.
2. Initialize current node as root
3. Push the current node to S and set current = current.left until current is None
4. If current is NULL and stack is not empty then:
   1. Pop the top item from stack.
   2. Print the popped item, set current = popped\_item.right
   3. Go to step 3.
5. If current is NULL and stack is empty then we are done.

Complete the debug\_in\_order function using iteration. You must use the Stack class provided from astack.py but do not modify the class.

# Part B – Node Deletion

Complete the remove() function. Remember that there are 3 cases to consider based on the number of children the found node has.