CSC 120 **ICA-23**

Work with your neighbor. (This will be graded for participation only.)

1. What does the Binary Search Tree look like if we add the values below in order from left to right?

```
18, 10, 40, 44, 7, 26
```

ANS:

```
          18
         /  \
      10      40
     /       /  \
   7       26    44
```

2. What does the Binary Search Tree look like if we add the values in a different order? Draw the tree that results from adding the values below from left to right:

```
40, 26, 18, 44, 7, 10
```

ANS:

```
          40
         /  \
      26      44
     /
   18
   /
  7
   \
    10
```

3. Using the same values, what order of insertion would result in the tallest tree possible? (There may be more than one order.)

ANS:

If the sequence of values is in sorted order (either ascending or descending), you will get the tallest tree possible when inserting them into a binary search tree (that starts empty).

4. Construct the BST by inserting the values below from left to right:

```
7, -2, 10, 0, 13, 14, 3
```

ANS:

```
          7
        /    \
     -2        10
       \         \
        0          13
         \           \
           3           14
```

5. Modify the code below for searching a BST T to define insert(T, v).

```
def search(T, v):
   if T == None:
      return False
   if v == T._value:
      return True
   if v  < T._value:
      return search(T._left, v)
   else:
      return search(T._right, v)
```

```
def Node:
   def __init__(self, value):
     self._value = value
     self._left = None
     self._right = None
```

ANS:

```
def insert(tree, value):
    if tree is None:
        return Node(value)

    if value < tree._value:
        tree._left = insert(tree._left, value)
    elif value > tree._value:
        tree._right = insert(tree._right, value)

    return tree
```