CSC 212 Tutorial #9
Binary Search Trees

Problem 1

Write the method *getRange* member of the class *BST* that returns the range of the binary search tree. The range is defined as the difference between the maximum key and the minimum key. Assume that the tree is not empty. The method signature is: `public int getRange()`.

Problem 2

Write an efficient method *isInRange* member of the class *BST* that takes as input a key k and returns true if the binary search tree contains at least two keys *k1* and *k2* such that $k1 \leq k \leq k2$, false otherwise. Try to minimize the number of visited nodes. The method signature is `public boolean isInRange(int k)`.

Problem 3

Write the member method *countNodesIn* member of the class *BST* that returns the number of nodes in the subtree rooted at the node with key k. Assume that k exists. You are not allowed to call any of the *BST* methods. The method signature is `public int countNodesIn(int k)`.

Problem 4

1. Insert the following keys into an empty binary search tree: 37, 23, 18, 65, 25, 62, 20, 59, 63, 90, 18.
2. Remove the following keys from the final tree in part 1: 18, 90, 37.
3. If we wish to print the keys in increasing order, which traversal method should we use?