CSC212

Tutorial 9

Binary Search Trees

Problem 1:

Write the recursive method *maxKey* member of the class *BST* that returns the maximum key
value in the tree. Assume that the tree is not empty.

Method: *public int maxKey()*

Problem 2:

Write the recursive method *sumKeys* member of the class *BST* that returns the sum of all the keys.

Method: *public int sumKeys()*

Problem 3:

Write the method *range* member of the class **BST** that returns the range of the binary search tree.
The rage is defined as the difference between the maximum key and the minimum key. Assume
that the tree is not empty.

Method: *public int range()*

Problem 4:

Write an efficient method *inRange*, 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** ≤ **k** ≤ **k2**,
false otherwise. Try to minimize the number of visited nodes.

Method: *public boolean inRange(int k)*

Problem 5:

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