CIS 044:   
Introduction to Data Structures Using Java

Lab 10

**Instructor**

Dr. Maher Mneimneh  
Email:mahernm@gmail.com

**Guidelines**

**Please follow the guidelines below:**

**Submit the homework and lab solutions in the drop box. For a programming question, submit a .java file (for source code) and a .txt file for program output. If the problem involves other questions, submit a separate .txt file to answer the question.**

**P1 (25 points)**

Consider a method for a binary search tree that decides whether the tree is height balanced as defined in section 23.10. A tree is said to be height balanced or simply balanced if the subtrees of each node in the tree differ in height by no more than 1The header of the method could be as follows:

**public boolean isBalanced()**

Write this method for the class BinarySearchTree in **BinarySearchTree.java**. It should call a private recursive method of the same name. isBalanced() must run in O(h) time where h is the height of the tree.

**P2 (25 points)**

Write a method that checks that the underlying tree is a valid binary search tree. The solution should examine each node in the given binary search tree only once.

The header for the method is as follows:

**public boolean isBST()**

Add the method to the file **BinarySearchTree.java**

**P3 (25 points)**

Implement the following method in BinarySearchTree.java

**public T getPredecessor(T entry)**

which return the inorder predecessor of entry or entry if it’s, or **entry** if it’s the smallest item in the tree, or null if **entry** is not in the tree. The solution must run in O(h) where h is the height of the tree.

**P4 (25 points)**

Statisticians often are interested in the median value in a collection of data. In a collection, about the same number of values are greater than the median value as are less than the median value. When the data is sorted, the median value occurs at the midpoint of the collection. But when the data is not sorted, the median is not as easy to find.

A problem more general than finding the median is to find the k th smallest value in a collection of n values, where 0 < k < n. To find the median, k would be [n/2]—that is, the smallest integer greater than or equal to n/2. For example, the median value of 11 items is the 6th smallest one.

Design an algorithm that uses a maxheap to find the k th smallest value in a collection of n values. Implement your algorithm as a method

**public static Integer getKthSmallest(ArrayList<Integer> aList, int k)**

in the file **MaxHeap.java**