BLG 233E DATA STRUCTURES AND LABORATORY

EXPERIMENT 9 – TREES

IMPORTANT REMINDERS

1. It is not allowed to use USB sticks during the lab sessions.
2. You should unplug your ethernet cables during the lab sessions.
3. Any reference book or help material (C++) is allowed.

In this experiment, you will exercise basic properties of tree data structure. For this aim, you are required to write your own binary tree struct and implement the following functions.

a) **createTree():** Create a random integer array (size of N, values should also be between 1 and N) and insert each element of the array to your binary tree. You should not add an element to the depth (d) if depth (d-1) has an empty spot.

b) **removeTree():** Removes all of the tree nodes from the memory.

c) **printPreorder():** Prints the contents of the binary tree to the screen in preorder.

d) **printInorder():** Prints the contents of the binary tree to the screen in inorder.

e) **printPostorder():** Prints the contents of the binary tree to the screen in postorder.

f) **findMax():** Finds the maximum value in the binary tree and and returns it.

g) **findMin():** Finds the minimum value in the binary tree and returns it.

h) **findNumNode():** Finds the number of the nodes in the binary tree and returns this value.

i) **findNumLeaf():** Finds the number of the leaves in the binary tree and returns this value.

j) **calculateDepth():** Calculates the depth of the binary tree and returns this value.

k) **calculateSum():** Calculates the summation of the values in the binary tree and returns this value.

l) **calculateAverage():** Calculates the average of the values that are stored in the nodes of the binary tree and returns this value.