**CMPSC 412 – Lab-5** (25 points)

**Tree - Basics**

**Due date: 3/1/2022**

**Lab Exercises:**

**Exercise-1: (10 points)**

Develop a Binarytree.py which can perform the following functions:

* Insert node to a tree
* Perform In-order traversal
* Perform Pre-order traversal
* Perform Post-order traversal
* Find a node
* Minimum value in the tree
* Maximum value in the tree

**Exercise-2: (7.5 points)**

Write a function to remove a node from a tree data structure? This function should consider all the three cases: case-1: remove a leaf node, case-2: remove a node with one child and case-3: remove a node with two children.

Perform the time complexity for this function. Briefly explain?

The time complexity in the worst case for the removal function will be log n. This is because in the case where the node to be removed has 1 or 0 children, the replacement is very straightforward. In the case that the removed node has 2 children, then we must determine the appropriate node to be replaced which will take at most log n time.

**Exercise-3: (7.5 points)**

Write a function which takes two trees and merges the two trees. The function should return the merged tree. Perform the time complexity for this function. Briefly explain?

The time complexity of this function will be O(N\*M) where M is the number of nodes in the second tree to be merged and N is the number of elements in our initial tree. This is because for each element in the second tree, we will add it to our initial tree which executes in, at worst, N time, when all elements are one side of the tree. Since each element M, takes N time to execute, the worst case will be O(N \* M)

 