## Computing Year 3: 2020-2021

## DSA II (DS and Algorithms)- Take Home Assignment

**INSTRUCTIONS**

* Answer all of the following questions and submit on a **single pdf document** by the due date. This submission document must have your student number in its name. The front page of this document must contain your name and your student number.
* Upload your submission to:  
  <https://instituteoftechnol663-my.sharepoint.com/:f:/g/personal/byrnea_itcarlow_ie/El4oU-Zx3edGnd3MWsEJqvEB-wJxO-_Qxyx9a-e6G_hS0w>
* **Due Date: Tuesday January 12th.**
* You must demonstrate how you arrive at any answers you get. Correct answers without any explanations/demonstrations will get no credit.
* This assignment has a 10% credit weighting for the whole DSA II module
* Submissions after January 12th will not be considered and a score of zero will be allocated.
* **You must do this assignment yourself**. Any group of assignment answers that I deem to be too similar for coincidence will be sent to the External Examiner for arbitration. If it is adjudged that plagiarism has occurred then **all parties involved will get a score of zero** on this assignment.

**Question 1: (10%)**

Create a Huffman tree and the Huffman codes for the letters of your full name.

**Question 2: (50%)**

1. Create a Binary Search Tree (BST) from the following list of names: use your first name and the first names of 8 of your friends. List the names you have used.
2. Write the algorithm to output this list (from the BST) in alphabetical order.

Demonstrate how your algorithm works on your BST.

1. Write the algorithm that returns the number of leaf nodes in the tree.

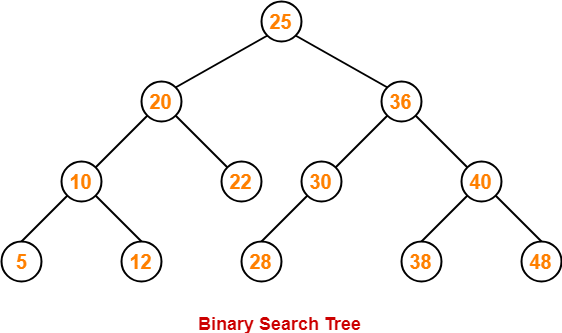
Demonstrate how your algorithm works on your BST.

1. Write the algorithm that outputs the contents of the leaf node with the highest value.

Demonstrate how your algorithm works on your BST.

1. Write the algorithm that searches for a given name in the tree. It will return whether the name has been found or not. It will also return the search path used.

For example, given a tree with the following structure:



Searching for 28 (from root) , would result in right, left, left, 28 Found

Searching for 25 (from root) , would result in left, right, Not Found  
  
Demonstrate how your algorithm works on your BST.

1. List the Big O value for each of your algorithms.

**Question 3: (40%)**

Design an app to store your friends' telephone numbers. Use a linked list to store the name/number information.

Explain the data structure(s) and six methods in your app and show how they will work together by writing the pseudocode of each method.

The app should be capable of adding a friend's telephone number to the list, searching for a friend's number, and printing out the entire list to the frontend.