

# COMP 2540 Data Structures and Algorithms

University of Windsor

School of Computer Science

## Assignment 5: Due August 2<sup>nd</sup>, 2020, 11:59 pm

---

This assignment is to understand binary trees, binary search trees and its operations. Please Read and understand the definition for binary trees and binary search trees from lectures before you start to work on the assignment.

### Tasks:

You have to write a program for the following tree operations. Implement your programs on an array.

1. Consider a complete binary tree (check the definition from lecture). Write a program to build the following sequence in the given order on a complete binary tree.

1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15

2. A traversal visits the nodes of a tree in a systematic manner. Write a program for the following order of visit that we discussed in the class and test your code on the above binary tree in question 1 and print them.

- a. Preorder traversal
- b. Postorder traversal
- c. Inorder traversal

3. Assume a BST considers the keys (integer) and no values. Here keys are stored in each node of the tree. Consider the following sequence in the given order. Write a program to create the binary search trees on the following sequences.

i. 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15

ii. 8, 12, 14, 10, 4, 6, 13, 11, 2, 5, 9, 1, 7, 3, 15

- a) Print the above trees using Inorder traversal method that you implemented in question 2.

- b) Implement a search method for a BST. Check whether keys 15 and 25 are existing in the above BST or not. (your method can return Boolean value).
- c) Implement a delete method for a BST.
  - I. Delete key 13 and print Inorder traversal of the BST after removal.
  - II. Delete key 8 and print Inorder traversal of the BST after removal.
- d) Analyze and compare the performance (the worst and best-case) of above operations based on your implementation. (Submit a pdf document for this part)

**NOTE:**

- 1. You should test the given test cases in your program and create a make file (i.e. Script file). (submit your make file with your source code.)
- 2. You **CAN NOT** use inbuilt functions for **Tree ADT operations**. If you do so, you will get **ZERO**. Other required inbuilt functions are allowed to use.

**Requirements:**

- I. **UNDOCUMENTED OR IMPROPERLY DOCUMENTED** code will automatically lose 50% marks.
- II. Readability: The code is exceptionally well organized and very easy to follow.
- III. **PLAGIARIZED** work will not be graded and receive a mark of **ZERO** and reported according to the Senate bylaws.

**Submission:**

- I. Your assignment must be **RECEIVED** by the due date and time. Late assignment submissions are **NOT** accepted will receive a mark of **ZERO**.
- II. Before the submission, you must contact your GA/TA through your lab MS team channel for the feedback.
- III. Submit your work through Blackboard: Source code

**Rubric:**

- 5 - The program works and meets all the above requirement.
- 4 - The program works, fair readability, most of the code could be reused and fair documentation.
- 3 - The program works, fair readability, most of the code could be reused and NO documentation.
- 2- The program produces some results but does not display them correctly, poor readability and NO documentation.
- 1- The program produces compilation error, poor readability, and NO documentation
- 0 - Did not complete any task.

**PLUS**

- \*\*\* 1 - Contact the GA/TA to answer the questions. (attendance + students' involvement)