

# 210CT Week 5 Coursework Tasks

## Dr. Diana Hintea

### LEARNING OUTCOMES

1. Understand the tree data structure, focusing on binary search trees, together with their related operations (insertion, deletion, searching, comparing trees, sorting).

### BASIC/INTERMEDIATE TASKS (MANDATORY COURSEWORK SUBMISSION)

1. Implement in the language of your choice a function that deletes a node in a Binary Search Tree. Integrate this function in the code provided to you on Moodle. The deletion operation should be performed based on the key (value) of the node. It is up to you how you print the binary search tree - you can just use a traversal method such as in-order or you can create a fancier, more visual representation.
2. Build a Binary Search Tree (BST) to hold English language words in its nodes. Use a paragraph (1 paragraph would suffice) of any text in order to extract words and to determine their frequencies. Input: You should read the paragraph from a suitable file format, such as .txt. The following tree operations should be implemented:
  - a) Printing the tree in pre-order.
  - b) Finding a word. Regardless whether found or not found, your program should output the path traversed in determining the answer, followed by yes if found or no, respectively.

Note: In order to deal with identical words, add another attribute to the Binary Search Tree node that would keep track of the frequency of that word.

### ADVANCED TASK (OPTIONAL)

1. Implement a system that maintains information for school classes. For each of the students in a class the following information is kept: a unique code, student name, birth date, home address, id of class he is currently in, date of enrolment, status (undergrad, graduate). For keeping track of the students, a computer program based on a binary search tree data structure is used. Write a program that implements the following operations:
  - Find a student by his/hers unique code, and support updating of the student info if found.
  - List students by class in lexicographic order of their names.
  - List all students in lexicographic order of their names.
  - List all graduated students.
  - List all undergrads by their class in lexicographic order of their names.
  - Delete a student given by its code.
  - Delete all graduates.

## READING

Sleator, D. and Tarjan, R. (1985). Self-Adjusting Binary Search Trees. *Journal of the Association for Computing Machinery*. Vol. 32, No. 3, July 1985, pp. 652-686.