

# 210CT Week 6 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

1. Implement the TREE\_SORT algorithm in a language of your choice, based on the template provided on Moodle, but make sure that the INORDER function is implemented iteratively, rather than recursively.

### ADVANCED TASK

1. Build a Binary Search Tree (BST) to hold English language words in its nodes. Use a paragraph of any text in order to extract words and to determine their frequencies. Input: You should read the words and frequencies from a file in a suitable format, such as .csv. The following tree operations should be implemented: a) Listing (word, frequency) pairs for each of the tree nodes. b) Printing the tree in pre-order. c) 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.

### 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.