

# Data Structures and Algorithms

## Lab 4 (5% of final grade)

(100 points total)

All code must be your own and follow the guidelines in lecture notes and textbook. Use proper coding styles including comments and indentation. No code from Internet sources is allowed in this assignment. All code must be in C++ programming language.

1. Create a Binary Search Tree using 200 unique random words from the attached dictionary words\_alpha.  
As a first option, allow the user to input a word from the console and provide a response on whether the word provided by the user is contained in the Binary Search Tree or not.  
As a second option, allow the user to insert a new word in the Binary Search Tree.  
As a third option, allow the user to add a new word in the Binary Search Tree (if it does not already exist). There is no need to write this added word to the file.  
Submit screenshots of the results, and code files in a zip file.
2. Perform the same tasks as #1, except that now store the words in an AVL tree.
3. Perform the same tasks as #1, except that now store the words in a 2-3 tree.
4. Assume the input as the complete graph with 5 vertices ( $K_5$ ). Store  $K_5$  using each representation of Graphs as listed below. The output is a list of vertices of  $K_5$  in the order visited first by DFS traversal and then by BFS traversal, starting from the same vertex and assuming increasing order preference. Compare the order of vertices ordered by each of the traversals. Submit a writeup, screenshots of the results, and code files in a zip file.
  - a. edge-linked graph representation
  - b. adjacency list graph representation
  - c. adjacency matrix graph representation
5. Assume a graph represented by adjacency matrix and model player transfer in your favorite sport. Instead of 1 representing an edge in the adjacency matrix, add the number of transfers (e.g., if teams A and B had 3 transfers between each other, the cell corresponding to the AB edge in the adjacency matrix contains a 3 instead of a 1). Create and test the implementation with a realistic input. The output should answer the questions below. Submit a writeup of your design, screenshots of the results, and code files in a zip file.
  - a. the total number of transfers of a team inputted by the user

- b. the number of transfers between two teams inputted by the user