# CIS*2520 Data Structures

Fall 2018

## Assignment 4

Write a C program to implement the AVL tree paradigm. Your AVL tree implementation should deal with nodes, each of which has a key and a count. Your program reads keys in file `A4_data_f18.txt`, and inserts them into the tree. A key is a character string like `flr830`, and the count is an integer which serves as the frequency count of the key. For example, if the same key has been inserted into the tree three times, there is only one node for the key and the count value of the node is 3. Your program should show a menu with the following options:

1. Initialization
2. Find
3. Insert
4. Remove
5. Check Height, Size, and Total Count
6. Find All (above a given frequency)
7. Exit

## Description:

- The user of your program enters 1 – 7 to select the options.
- When option 1 is selected, your program should open that file, read and insert keys into the AVL tree on the word by word basis. The data file is provided along with this assignment handout.
- If option 2 is selected, your program should ask for a key, then search the AVL tree that your program has created, and display the key and its frequency count if it finds the key, or display *No_such_key* if it could not find the key in the tree.
- For option 3, your program should ask for the key to be inserted and then search for the position where the key should be placed. Your insertion function shall increase the count if the key has already been in the AVL tree, or carry out the actual insertion if the key is a new one (set the count to be 1).
- For option 4, your program should ask for the key to be removed and then search for the matching key. If it could not find the given key, it should display *No_such_key*. If it finds the key, it removes the tree node only if the frequency count of the key is 1, or it decreases the count of the node by one otherwise.
- When option 5 is selected, your program should compute and display the height and the size of the AVL tree, which is the number of nodes, and the total count (sum of all the counts).

- When option 6 is selected, your program should ask for a positive integer, and then display all keys whose frequency counts are greater than or equal to the given integer.
- Option 7 terminates your program.

**Due time: 09:00am, Monday, Nov 26, 2018**.