1. Introduction: demonstrate your understanding of the problem.

An AVL tree must be utilized, with other appropriate data structures, to create a frequency dictionary. A text file is inputted into a C++ program with the words stored into a struct that contains two variables, an int to represent the frequency of each word and a string which represents the key; each struct object is stored into a node in the AVL tree.

2. Algorithms: Choose two algorithms (functions) in your program to explain your idea of implementation using pseudocode or simplified source code (list the major operations only).

**AVL\_Retrieve2:** This function is a modified version of AVL\_Retrieve; AVL\_Retrieve originally being a boolean function that returns true or false if the key is found in the node. I modified this function so that the return type is the found node (if the key is found).

If there is no tree, function cant properly run, return NULL.

Node equals (retrieve function, key, tree).

If the key was found in a node of the tree, return pointer to node.

Else not found, return null.

**Print:** Function used to print the content of the struct wordNodes objects.

Pass the object as a parameter

Cout its key

Cout its frequency

3. Data Structures: List all the major data structures you have used for your implementation and briefly explain why they need to be used.

Struct: Used to store key(string/words) and frequency of each word stored in the nodes of the AVL tree. Needed to be used as each node must have variables to properly create a frequency dictionary.