CS 2302 Data Structures

Fall 2019

Lab Report #4

Due: October 21, 2019

Professor: Olac Fuentes

TA: Anindita Nath

**Introduction**

In this lab the task was to implement both binary trees and binary search trees in a way that would acice the same goal. This goal was to read a file of words and their similarity vectors and search through the trees for specific words and retrieve those vectors to produce a similarity value, as well as compare their running times

**Proposed Solution Design and Implementation**

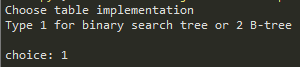
For the lab i wanted to keep the word and its vector together for the duration of its run time. To this end, I want to read a file and immediately convert it to an object and anytime I need to reference a specific word i will retrieve an object containing both a word plus its vector. Every method that required the word only to function receives the created object to ensure that the word never loses its paired set of vectors.

**Experimental Results**

**Part 1:**

This was very simple to implement. The task was to implement a user prompt to determine which tree to use for the similarity trial. For this all i did was add a user input to choose between 1 and 2. Then , using if statements I would then redirect them to the right tree.

TEST:



**Part 2:**

This was also a fairly simple task. I created a readfile method that would run through the text line by line and simultaneously convert each line to an object and save the objects in an unsorted list. I would then convert this list to a tree(either tree depending on the users choice) using their respective insert functions and due to how the trees are built they would turn into a sorted tree.

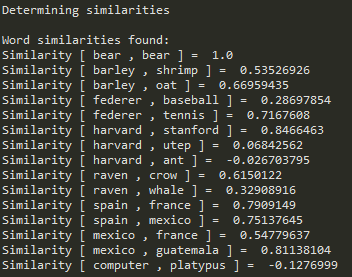
TEST:



**Part 3:**

This part was slightly confusing because I had never been exposed to the fundamentals of word similarities. After browsing through public python libraries I was able to create a method that would take in the vectors of a pair of words and produce a similarity rating from -1 to 1 depending on how similar they were.

TEST:



**Part 4:**

This part was not an issue at all. This was a repeat of a requirement from lab 1. I would record 2 times and subtract the start time from the end time to determine my run time.

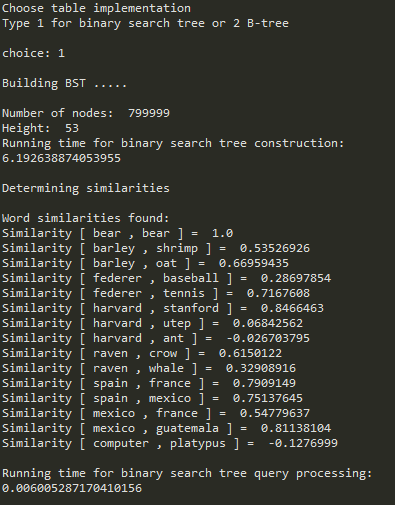
TEST:



**Overall Lab Results:**

After I finish this lab I was required to do a few reflections on the methods used. Such as what would happen if on the Btree method the max number of objects per node was increased. This would result in longer run times due to more objects needing to be checked as you iterate through the tree. Another reflection was to think about what the actual embedding of each word was. In this case it was a 50 dimensional representation of the word in vector form as determined by its association and relative position to other words in the 50 dimensional graph.All in all it ran well and unfortunately I was unable to finnish the B Tree portion of the lab I did fully complete the BST portion and this is the final run output.

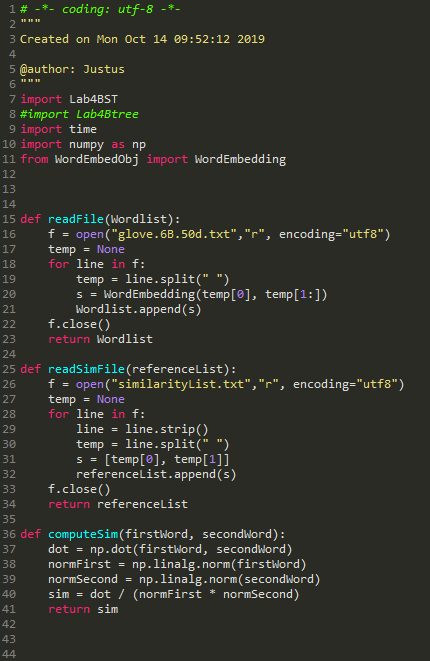
OUTPUT OF BST:

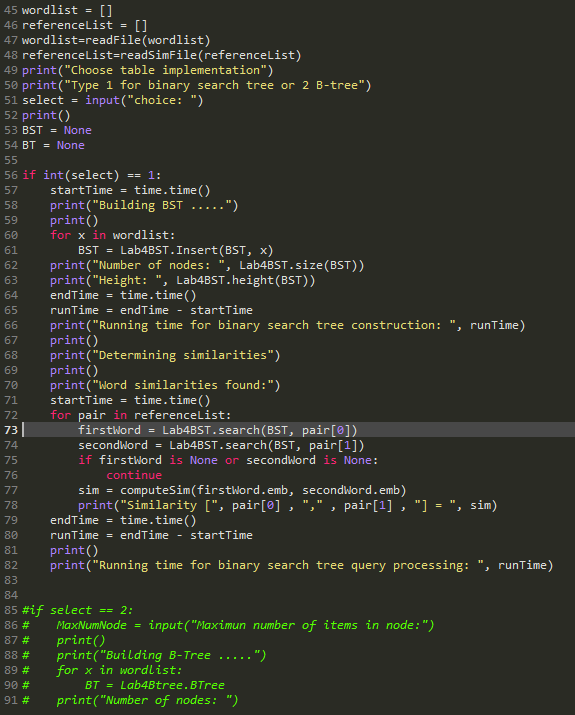
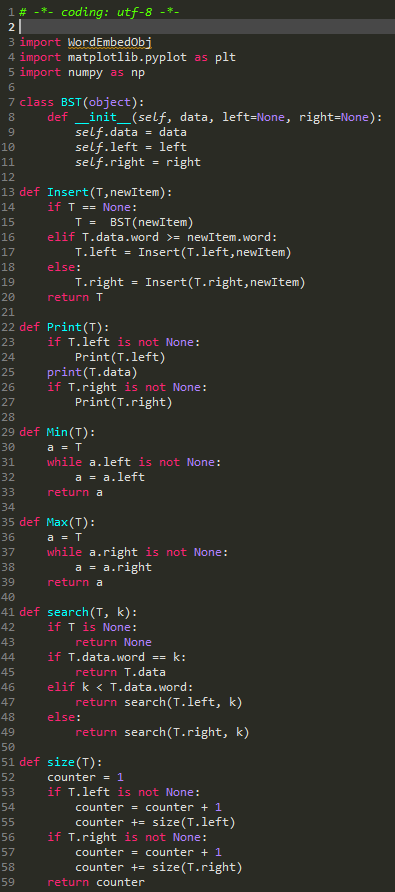


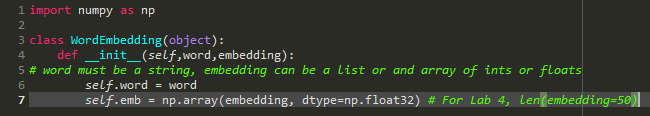
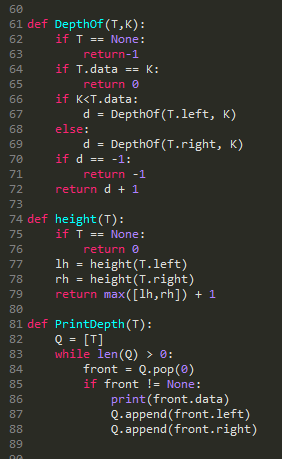
**Conclusion**

This lab was hard for me to complete due to a time crunch but was fun to work on as this was a glimpse into the practical uses of trees. I do plan on working to fully finish this lab.

**Appendix**

****



*I certify that this project is entirely my own work. I wrote, debugged, and tested the code being presented, performed the experiments, and wrote the report. I also certify that I did not share my code or report or provided inappropriate assistance to any student in the class*

Justus Frausto