**CS 2302 Data Structures**

**Spring 2019**

**Lab Report #4**

Due: October 21th, 2019

Professor: Olac Fuentes

TA: Anindita Nath

Author: Kevin Ramirez

**Introduction**

For lab#4 we had to implement both a binary search tree, and a b-tree, these would read and save a great amount of words and arrange them in an ordered manner. These words would have numbers embedded within; these would be used to compare the similarities between all the words by using cosine distance. The objective was to demonstrate how AI systems utilize these data bases, to distinguish and understand what we are saying to them. There are two methods, “Btree”, and “Tree”, Tree is a binary search tree, but forgot to change the name back to BST. The rest of the other methods are used as tools, and just for debugging. And the other major method, is “readTXT”, and like the name explains, this read and inserts the data in the corresponding class, either, binary search tree, or the B-Tree.

**Proposed Solution Design and Implementation**

**Operation #1 class Btree:**

It’s a simple constructor, used for creating the necessary data structure, the code is an example that was given to us.

**Operation #2 class BST(Tree):**

It is also given to us, so the code part was very much simple, and the insertion of elements is simple, if the given node is smaller than the previous node, add to the left, otherwise, to the right.

**Operation #3 readTXT:**

This method will read the txt files and insert them in already made B-Trees. We will just use the open method and read the file one line at the time, then we insert every word and their number embedding in the corresponding tree.

**Experimental Results**

For the test cases in this lab I decided to use only all the words, and their embeddings from the “glove” txt file. And for the case of an empty set, we set an if statement that would prompt the user that the list is empty.

**Case #1 Class Btree: Full Txt**

****

**Time (no print): 18.5 sec**

**Height: 9**

**Case #2 Class BST: Full Txt**

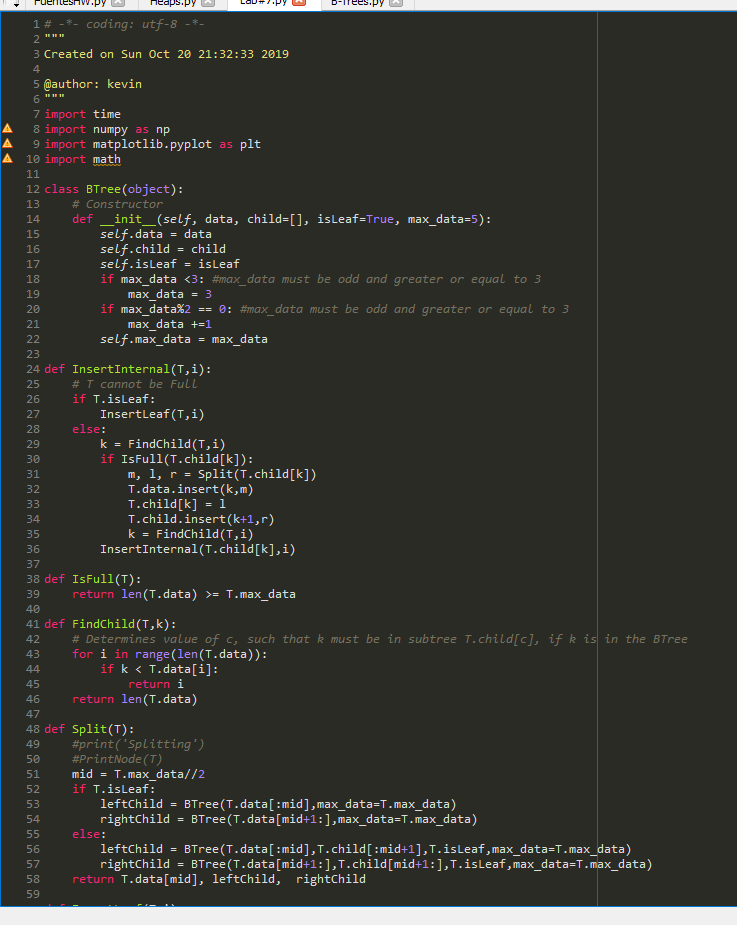
****

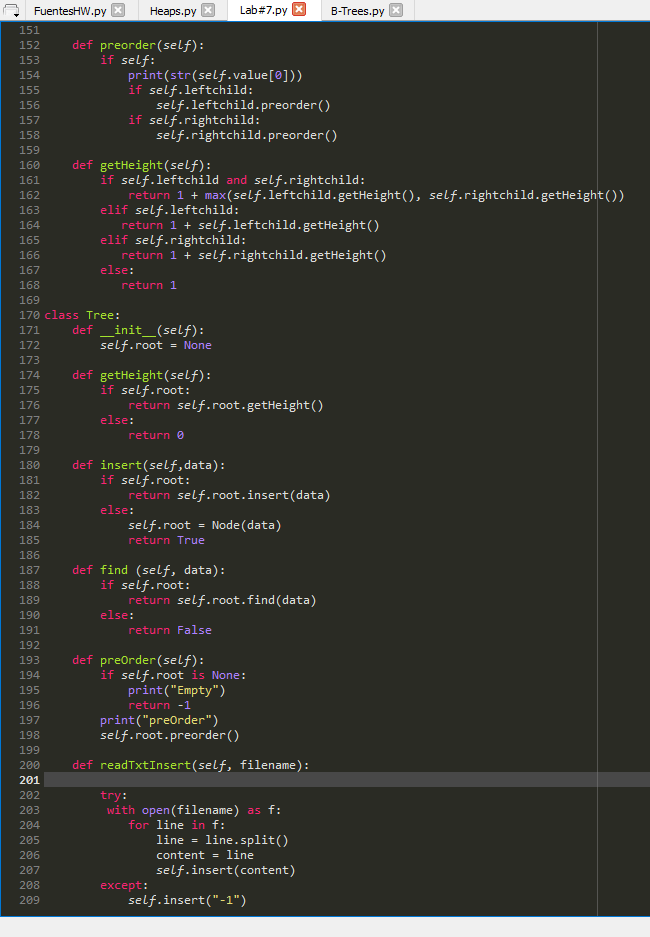
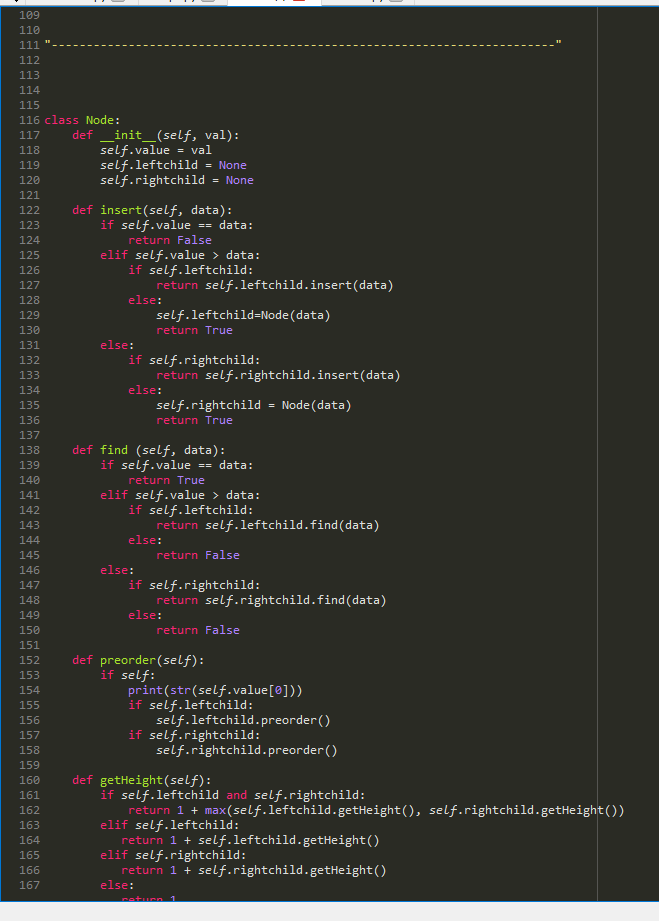
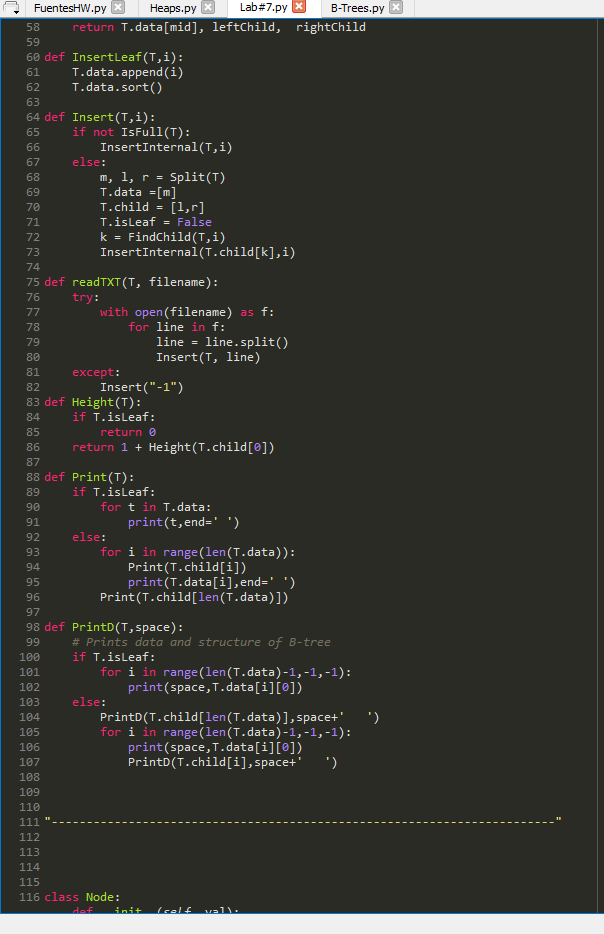
**Timer (no print): 19.40**

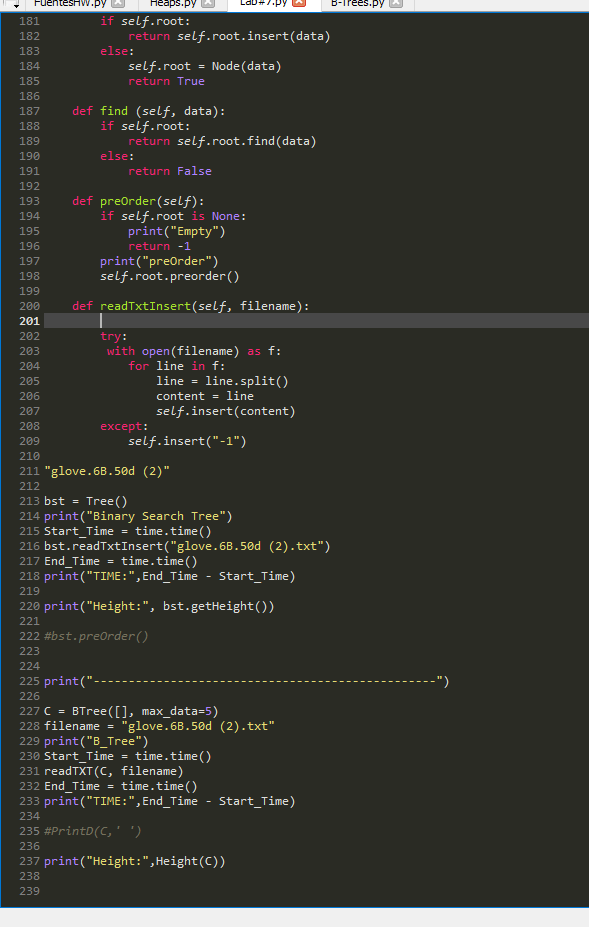
**Height: 53**

**Conclusion**

For the conclusion of this lab, I had no problems with the first two part for the lab but failed to properly implement the third part. I have no idea what was asked of me to do, and because of my over confidence I thought I could solve it before the due day, but yet again I proved myself wrong. For the rest of the lab it was simpler than I thought, and had no problems reading the txt files. And no problem implementing the data into both the classes of trees. Overall it was a good exercise to learn about B-trees and binary search trees.

**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