**CS 2302 Data Structures**

**Fall 2019**

**Lab Report #4**

Due: 10/21/2019

Professor: Olac Fuentes

TA: Anindita Nath

**Introduction**

For this lab we were asked to work with Binary Search Trees (BST) and BTrees. We were asked to store word embeddings in the trees. For this lab it is key to remember how to access elements from both trees. The main objective of this lab is to get a deeper understanding of BST’s and BTrees and be able to access any information.

**Proposed Solution Design and Implementation**

**Operation #1:**

For this operation, I first saved all of the numbers of the word embedding so I could get an idea of how to ask the user to input the amount of numbers he wanted stored. I then asked the user to input the number of items to store in a node. I then read line per line and read the amount of items the user wanted to store in a node and ignored the rest.

**Operation #2:**

For this operation, I created a list in which I stored the word, and then the second item in the list was the list of numbers. I wanted to pass the word embeddings to insert and then display only the word and not the numbers. I compared the strings that were already inside the tree and determined whether the new string was greater or less than the root.

**Operation #3:**

For this operation, I approached this in the same way I approached **Operation #2** since I already had an idea of how to do it. I got it right the first time for the BST but the BTree became a problem. I had to find the word I was looking for and then access the numbers that came with it so I was able to make the calculations. For the BTree, I had a hard time comparing word embeddings since I did not notice that I was comparing a string to a word embedding. However, I solved it just as I solved the BST.

**Operation #4:**

For this operation, I opened the file that had all of the word embeddings and chose a word close to the end so it would show me meaningful results.

**Experimental Results**

**Operation #1:**

For this operation, I had to leave it until last because I was worrying about the rest of the lab. I just asked the user to input the maximum number to store in a node and I was able to get it the first time.

**Operation #2:**

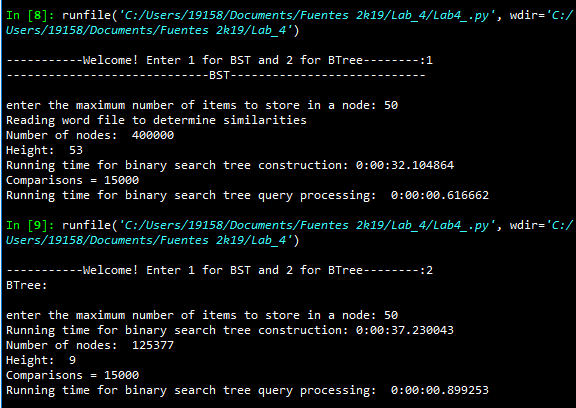
For this operation, I did not know what a word embedding was and I had to look up the definition. I wanted to pass all of the numbers and the word first so I could access the word fast. I then just passed the word and then the rest of the list so it would create the embedding. This did not take me long to do.

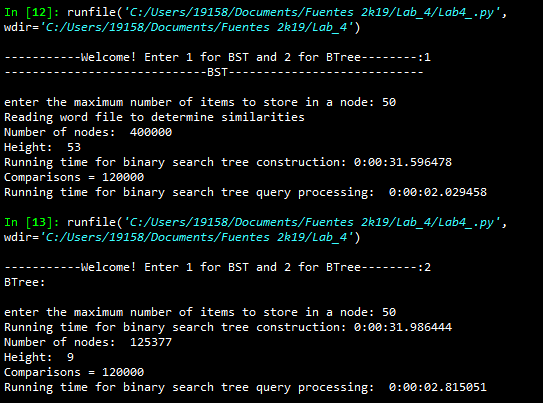
**Operation #3:**

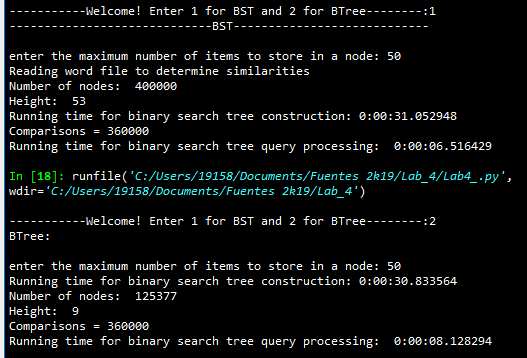
For this operation, as I said before, I got it the first time. I first made sure the word existed and then accessed the embedding of that word. I computed the calculation with np.dot(u,v) and np.linalg.norm(u).

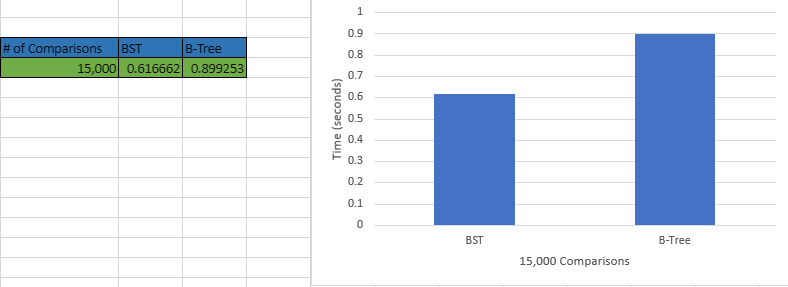
**Operation #4:**

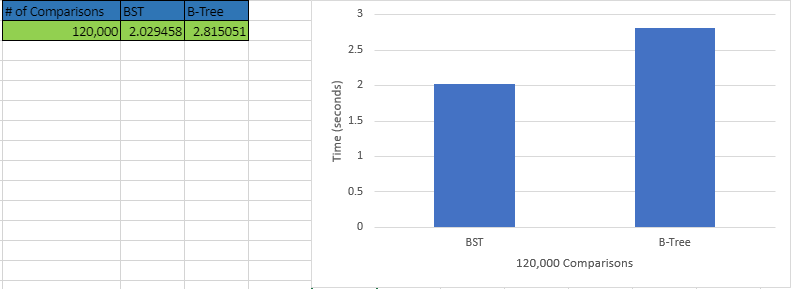
For this operation, I first made 100 comparisons, then 2,000 comparisons, and finally, 30,000 comparisons. Before making all these big comparisons, I went ahead and made the comparisons Dr. Fuentes made as an example on Lab 4 to make sure I was saving the embedding correctly. I realized that all of my previous labs took almost 0 seconds to compute a method and decided that I needed a big number of comparisons in order to actually show a difference between a B-Tree and a BST. Since the word

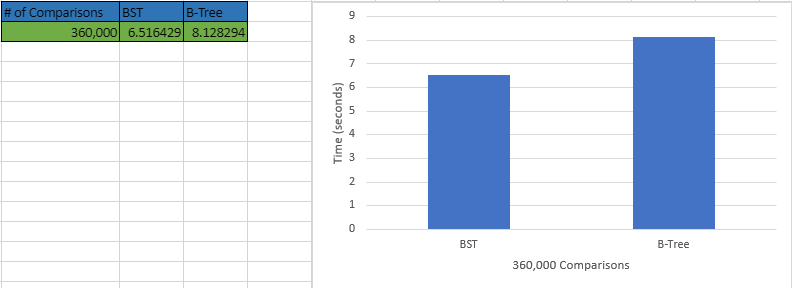
 # of comparisons for both BTree and BST = 15,000

 # of comparisons for both BTree and BST = 120,000

 # of comparisons for both BTree and BST = 360,000





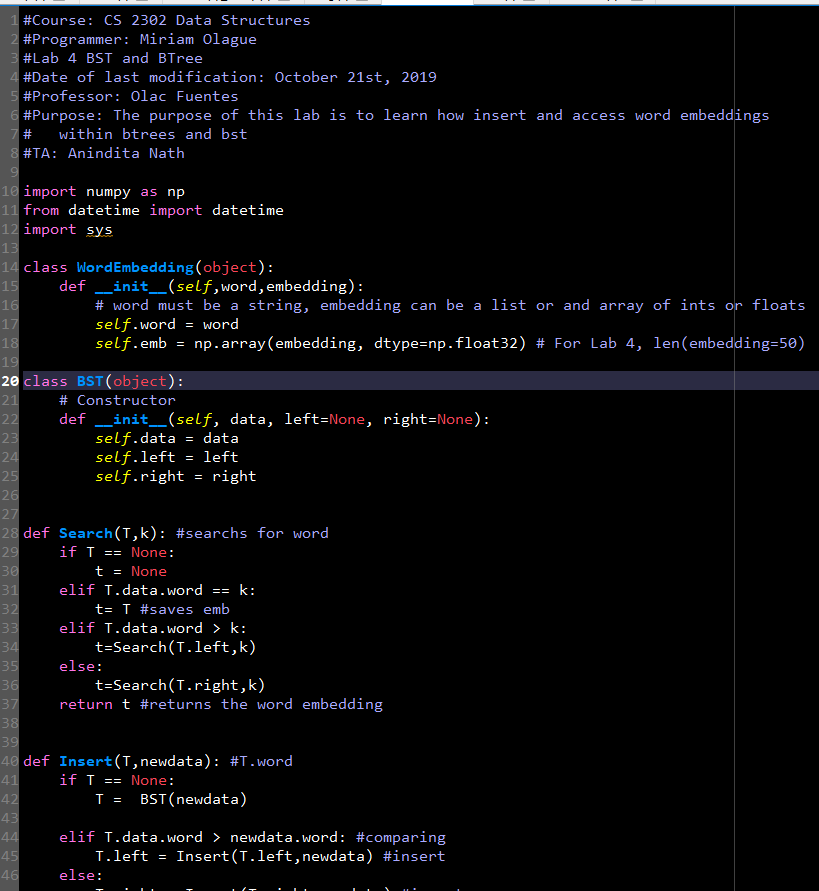


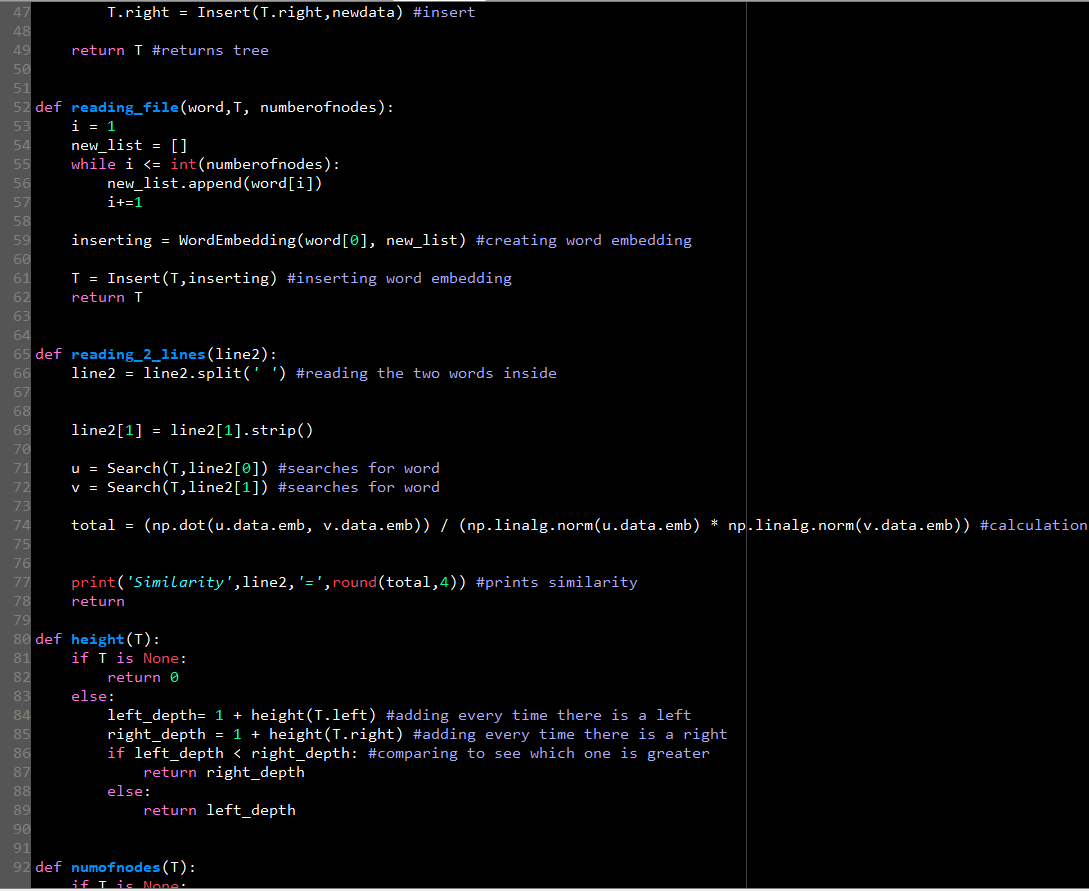
As the results show, the B-Tree took longer to compare the same words as in the BST. Even though they were the same exact words, B-Tree started to show a big time difference when making comparisons.

**Conclusion**

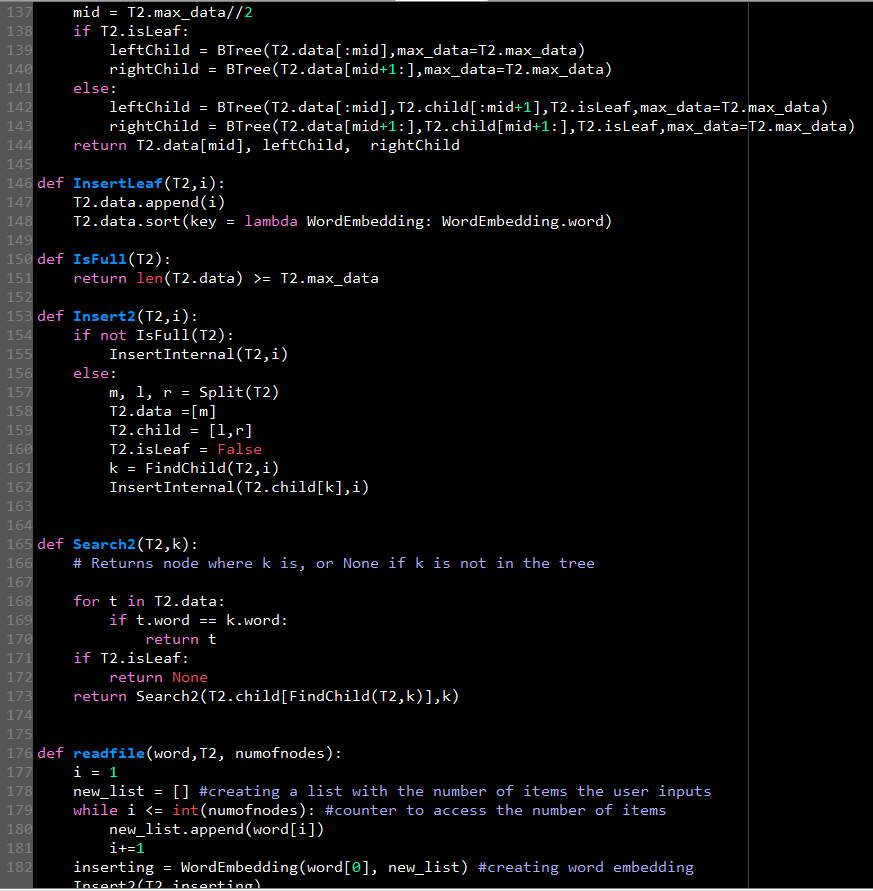
Since the word embeddings represent words, it makes it very efficient to access data of that word very quickly. Binary Search Trees (BST) were easy to understand when compared to B-Trees. I learned a lot about B-Trees and learned how to access data from a BST or a B-Tree. This lab helped me understand what the difference between B-Trees and BSTs.

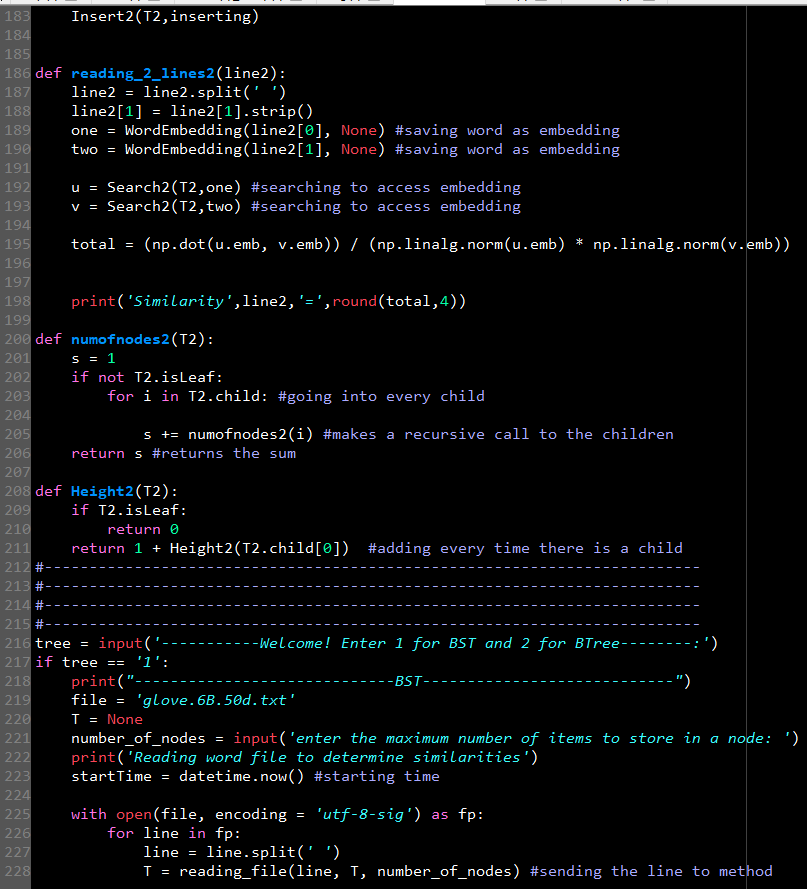
**Appendix**

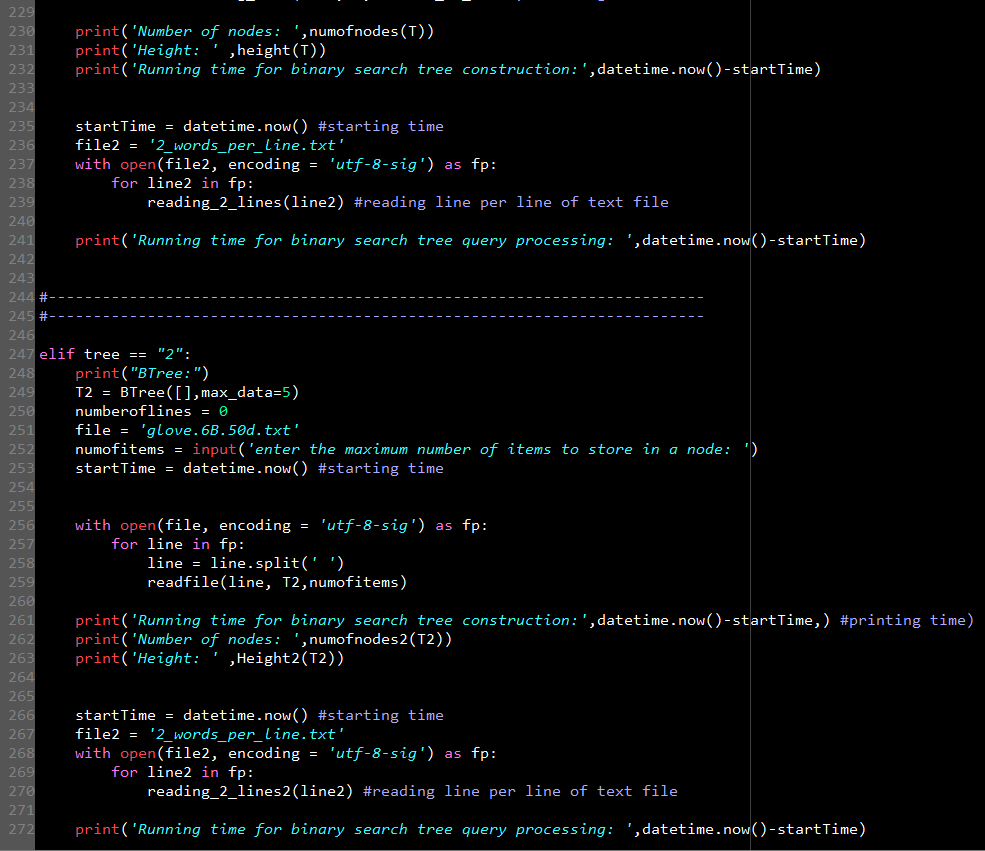
****

****

****

****

****

****

I certify that this project is entirely my own. 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.