

Homework 2: Data structures

By Vladimir Hardy

GitHub link with the most up-to-date code and full requirements <https://github.com/Spexon/LinkedLists>

Requirements:

Write a program that reads a text file and outputs a list of all words in alphabetical order. Each word should be displayed with a list of line numbers on which it occurs in the file. Use an AVL binary search tree to store the words seen.

Requirements:

1. The program must work for arbitrary large files sizes.
2. You must test the program using the file I will provide.
3. Each node is to have a linked list to store all of the line numbers. Use the list you created in assignment 1.
4. Make the tree a class with the necessary functions.

Example:

Input

```
Hello Bill,  
Did you hear about Aunt Betty or  
about Aunt White? I cannot believe that  
...
```

Output

```
about      2, 3  
Aunt       2, 3  
believe   3  
Betty      2  
Bill       1  
cannot     3  
Did        2  
hear       2  
Hello      1  
I          2  
or         2  
that       3  
White      3  
you        2  
...
```

Description:

I approached this problem by first doing plenty of AVL tree graphing on paper to practice my understanding behind the logic. Then I created a new class for this algorithm and followed some of the code provided in class when explaining the AVL tree algorithm. This was when I encountered my first problem, and to this day I still do not fully understand why it occurs. When I would test my program, I would occasionally get errors of access denial, and wouldn't be able to run my code until I did a fresh reboot (perhaps because my pointers were pointing somewhere they shouldn't be). This issue greatly halted my progression on the assignment, and as a result I was unable to complete the majority of it. I would use the debugger and discovered that there were some segmentation fault errors occurring in my code. Here is the output of my partly sorted list (the sorting algorithm couldn't finish because the pointers would never know when they reached the end of the linked list). The AVL tree would partially work, however I couldn't find the time to implement it into the code, I used a whiteboard to follow along with my algorithm and it did not seem to work the same way as it was presented in class.

```
Bill,  
Did  
Hey  
hear  
about  
Aunt  
Betty  
or  
about  
Aunt  
White?  
I  
cannot  
believe  
that  
you  
Signal: SIGSEGV (Segmentation fault)
```