04-630: Data Structures and Algorithms for Engineers

Assignment III: Text Analysis Using Binary Search Trees

Deadline: 11:59pm CAT, Friday, 8th March 2024

POINTS TOTAL: 100 PTS

Problem Definition

Your program will analyze a set of files. For each file, compile a list of all the distinct words by reading them from the file and inserting them in a binary search tree (BST). When all the words have been read from the file, traverse the BST and write the words to an output file, one word per line, in alphabetic order, together with the number of occurrences of each word in the file. Your program should be case-insensitive, i.e., it should treat "Tomato" and "tomato" as the same word.

For each file, compute the average and the maximum number of probes (i.e., the number of nodes examined) to find a word in the BST.

Although there may be many input files, there is just one output file.

When writing the words and their frequency of occurrence to the output file, begin each list with the name of the file from which they were taken. This should be followed by the maximum number of probes and the average number of probes, each on its own line. At the end of each list, write a line comprising of **20 dashes** (i.e., ----------).

For the purposes of this task, a word is defined to be a contiguous sequence of alphanumeric characters (a-z, A-Z or 0-9) and possibly including a hyphen.

If a word is followed by an apostrophe and an 's', e.g., 'let's, the 's' following the apostrophe should not be treated as a word.

Your BST only needs to support insertion since no deletions will occur.

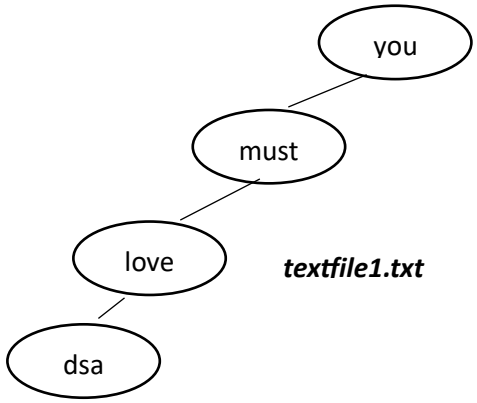
Input

The input file (input.txt) contains a list of file names with one filename per line. The filenames should include the relative path to that file from the directory where the executable code is stored, e.g. '../data/textfile.txt'. You can assume that there are no spaces in the filename or path. You can assume that all input provided is valid and contains no errors.

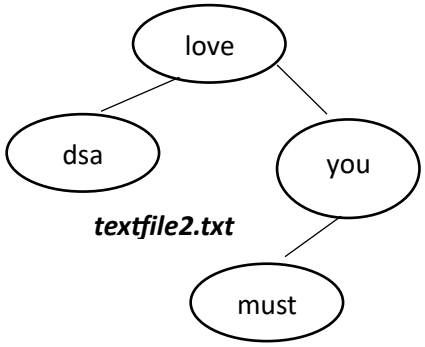
Sample Input

```
../data/textfile1.txt
../data/textfile2.txt
textfile1.txt
you must love dsa
textfile2.txt
love you love dsa must must love dsa must
```

Sketch of trees [Showing only words]



textfile1.txt



textfile2.txt

Output

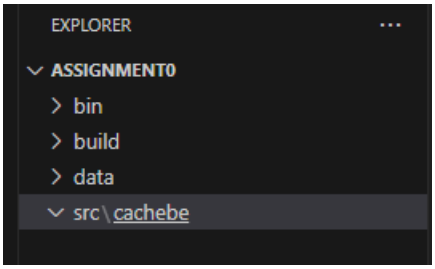
The program should write to the output file ("output.txt") your Andrew ID, followed by (for each file): the maximum number of probes on a separate line, the average number of probes on a separate line, followed by the list of distinct words in alphabetic order, one word per line with its frequency of occurrence and the level at which it is found in the tree, and, finally, the separator (----------) on a separate line.

Sample Output

```
cachebe
../data/textfile1.txt
Maximum number of probes: 4
Average number of probes: 2.5
dsa       1 (3)
love      1 (2)
must      1 (1)
you       1 (0)
--------------------
../data/textfile2.txt
Maximum number of probes: 3
Average number of probes: 2
dsa       2 (1)
love      3 (0)
must      3 (2)
you       1 (1)
--------------------
```

Submission Instructions

For your work on an assignment, say **assignment0**, you should have the following directory structure in your computer. Assume **cachebe** is your Andrew ID:



For submission, submit the following in a zip file named *yourAndrewID.zip:*

1. The source code: TextAnalysisBST.h, TextAnalysisImplBST and TextAnalysisAppBST.cpp. TextAnalysisAppBST.cpp must include the ***main*** function.
2. The test input file: input.txt
3. All the text files listed in input.txt
4. The test output file: output.txt
5. The cmake file: CMakeLists.txt

Do not include any other files. Submit only the source code files, the CMake file, and the input & output files. Do not include subdirectories.
The source code should contain adequate internal documentation in the form of comments.
The source code should contain adequate internal documentation in the form of comments.
Internal documentation should include the following.

- Author of the program. Date of submission.
- Functionality of the program.
- Format of input and output to the program.
- Solution strategy: a description (e.g. using pseudo-code) of the algorithms used to compute the three values: level, maximum, and average number of probes.
- A summary of the way the code was tested, e.g., a description of the different test cases.
- Complexity analysis of the essence of the algorithm in Big O notation with adequate justification.

Place this documentation at the beginning of the **application** file (the file with the **main** function).

Note: Submission will be made both on Canvas and Gradescope. Gradescope will be used only for purposes of checking similarity.

Marking

Marks will be awarded *if and only if acceptable internal documentation is included*. What constitutes acceptability is up to the examiner, but any reasonable attempt will be considered acceptable. Simple one-line descriptions will not be considered acceptable.

Marks will be awarded for the performance on a blind test data set made up of ten (10) text files (containing the words), as follows:

a) Maximum number of probes. 1 mark per text file. **[1x10=10 pts]**
b) Average number of probes. 1 mark per text file. **[1x10=10 pts]**
c) Frequency of occurrence for word. 2 marks per text file. **[2x10=20 pts]**
d) Level of word in tree. 2 marks per text file. **[2x10=20 pts]**
e) Alphabetic listing of output. 2 marks per text file. **[2x10=20 pts]**
f) Quality internal documentation. Notice that you will be awarded **0/100** for the entire assignment if there is no acceptable internal documentation. **[20 pts distributed below]**
    i. Functionality of the program. **[2 pts]**
    ii. Format of input and output to the program. **[2 pts]**
    iii. Solution strategy: a description (e.g. using pseudo-code) of the algorithms used to compute the three values: level, maximum, and average number of probes. **[4 pts]**
    iv. A summary of the way the code was tested, e.g., a description of the different test cases. **[6 pts]**
    v. Complexity analysis of the essence of the algorithm in Big O notation with adequate justification. **[6 pts]**

Use of Standard Template Library (STL)

The use of data structures from STL or similar advanced data structure libraries is prohibited.

Use of Generative AI

The use of generative AI e.g., ChatGPT is prohibited for this assignment. Therefore, any use of generative AI even to generate internal documentation will be considered unauthorized use and will lead to instant penalties as per Academic Integrity Violation (AIV) guidelines.