MQF Object Oriented Programming I Fall 2019

Homework 3 Due 11/3/2019 before midnight

You may form a team of two people. Solo is allowed.

Specifications

In this assignment, you will develop a system written in C++, which will allow user to search data warehouse to retrieve documents based on user's input. More specifically, you will write a program that generates an "inverted index" of all the words in a list of text files.

Before your system display a menu to allow user to select menu item from a menu, your system needs to create **inverted index** of these text files so to facilitate and speed up the search work. This menu contains at least two menu items. The first item will ask user to enter a word. After user entered this word, your system will search the **inverted index** data structure to find out which document(s) contain this word. If there is none, simply output "There is no XXX in our system", where XXX is the word entered by the user. If one or more documents contain this word, your system will output all the corresponding document name and frequency count. For example, if user entered "finance" and in your system, Doc1.txt and Doc5.txt contain this word "finance" then your system will output

Doc1.txt , 120

Doc5.txt , 14

To simplify your system implementation you may treat lower case word and uppercase word the same. I.e., "Finance" and "finance", and "FinanCe" can be treated as the same word.

Also note, the output from the search are sorted based on frequency count. I.e., we rank the document based on how many times the search word appeared in these documents. More frequency count means higher rank in this project.

The last menu item is "Quit", if user choose this option, then your system output a farewell message and quit. Otherwise after serve one user's query, your system will continue to display the menu and wait for user to input an option from the menu.

NOTE, because file reading and processing are time consuming tasks, you must preprocess the files once and cannot search them again after you display the menu. Doing so will slow down your search task.

Requirements

Put all the file names on a text file.

Your program will read this file first and then retrieve each file inside this file and create inverted index.

Extra Credits

You can add some more menu items to score extra credits.

(1) Your system can do phrase search such as "*financial industry*". These two words "financial" and "industry" are adjacent to each other.
(2) Your system can do two word search such as "*future* and *option*". In this case, these two words don't have to be adjacent to each other.
(3) Your system can do wide card query. For example, if user type in "fin*" then it will match "fine", "finance", "financial", etc.

Testing

We will provide you with 12 text files for you to test. To simplify your work, all the files contains only simple English words and don't have formulas, or figures etc. To make it easier, your program can store these inverted index on data structures that you choose so that they will reside on main memory of your computer but not external storages (Searching engine companies store all these on external disk systems). Note you can use more files to test run your program.

Grading rubrics

(1)	Programming style	4
(2)	Create correct inverted index data structure	10
(3)	Appropriate testing	6
(4)	Correct results	20
(5)	Each extra implementation	5 (so max 15 possible points)

References

https://nlp.stanford.edu/IR-book/pdf/irbookonlinereading.pdf

https://en.wikipedia.org/wiki/Inverted_index