CMSC 476/676 Information Retrieval
Programming Assignment 4 - Due Monday April 22, 2019
(timestamped on or before 11:59pm)
Do this assignment individually

The objectives of this assignment are to build a command line retrieval engine on top of the inverted files created in Assignment 3. The retrieval engine should take in queries which are lists of words, run them through the same preprocessing as the collection (i.e., downcase, optionally remove stopwords) and then match the query words against the inverted file to come up with document weights (the sum of the term weights in the document). Display the ten top-ranking document identifiers or filenames to the user.

You may use any algorithm you choose to locate the postings, but your report should discuss your choices (i.e., do binary search on index file on disk versus load entire inverted index with poistings into memory and search there). If you manage your inverted index between memory and disk, there will be a 3% bonus.

You must use something other than a simple array to tabulate the document-query similarity scores. A hashtable, heap, something with skiplists, a TDM that uses compression somehow, or whatever. You may design your own query syntax, however you only need to handle queries that are lists of words.

Entries in the inverted index shall be non-negative.

The query must run in a single command from the linux command line, i.e., not be an interactive menu-based system.
For example "retrieve dog cat bird" would be an example query.

You are welcome to add query term weights, e.g., retrieve Wt 0.3 dog 0.4 cat 0.3 bird. Weights may be negative. This implies, for example, that documents on cats will score higher than documents on dogs or birds, other things being equal. **If you add query term weights, there is a 5% bonus.**

Hand In
Sample queries and the top ten document identifiers). Test your system with the following queries:

- diet
- international affairs
- Zimbabwe
- computer network
- hydrotherapy
- identity theft
- Other queries showing multiple terms, effect of weights (if implemented)

Also, hand in listings of your code and a project report which outlines your algorithms, your data structures, and any files created. Discuss the complexity (roughly) of your algorithms. Include output with the queries listed above.

As usual, submit the code and your report in a ZIP file.
Email to our TA.

A sample session would look like:

```
$ cd /appropriate/directory
$ retrieve dog cat mouse
013.html 0.8
404.html 0.3
332.html 0.25
...
the last of the top ten files
```

If no files with non-zero scores exist, print an appropriate message. Don't bother printing files with scores of zero.

As usual, prepare a report and submit a zipfile to our TA.