CME 2101 Project-Based Learning III
Project-1

A Text-Based Search Engine

As you know, most of the information you need is provided by internet search engines. One of the most preferred search engines is still Google. The fundamental factors behind the success of Google are quickness and correctness. How does Google retrieve quick and correct results for user queries? In this project, you will investigate the answers of this question. You will develop a simple text-based search engine. Your program only operates on text files that are kept in a single directory. We will provide necessary text documents into our document database. You can use these documents for your tests.

This project will be implemented in four phases respectively:

1. File Operations
2. Indexing
3. Query Executing
4. Ranking.

Four phases of this project are:

1. Text documents include punctuation marks and stop words. Since stop words are not important, they should be eliminated. Your program should parse the entire document to extract the individual words. In this phase, your program should be able to print out only the words of a particular document as the output.
2. In indexing phase, search engines process documents and map the unstructured data to a well-known mathematical model. You need to build an index for all documents in a specified directory. Numbers representing occurrence counts of words in a document (i.e., frequency of a word in a document) should be printed out as the output of this phase. Note that the data structure chosen for indexing documents will directly affect the performance (i.e., running time) of the search query.
3. In query executing phase, you are expected to use the index structure you have created in the previous phase. User enters the query (e.g., *java tool*) to search in a document collection. By using the index structure above, you can reach documents without searching whole documents again. In this phase, your program should list the documents satisfying the user query.
4. Another problem is the ranking of results generated in the query executing phase. Since your program might return hundreds of results (i.e., documents) in a large file system. Returning the most relevant documents in the first places can save user time (like Google generally does). So, you should write a relevance measure to prioritize the most relevant documents into the top of the result list. After calculating distances of the resulting documents to the given query, you should sort documents according to these distance values, i.e., the smaller distance gets a higher rank in the resulting list.

Good Luck.

Notes:
Your project outcome should include a report as well as a working computer program. In your report, you should explain indexing method, data structures, algorithms and relevance measure used by your programs. The programs will be checked for the plagiarism; similar codes will get zero points.

CME 2101 Project-Based Learning III

Project-1

Table 1. Weekly Schedule

Week	Date	Type	Description
1	27.09.2016 Tuesday	-	No Lecture
	29.09.2016 Thursday	Lecture	Project Introduction
2	04.10.2016 Tuesday	Discussion	Project discussion and Class Design
	06.10.2016 Thursday	Discussion	File Operations, Parsing, Eliminating Stop Words
3	11.10.2016 Tuesday	Milestone	**Milestone #1 Class Design, File Operations, Parsing, Eliminating Stop Words**
	13.10.2016 Thursday	Discussion	Data structures and Indexing
4	18.10.2016 Tuesday	Discussion	Data structures and Indexing
	20.10.2016 Thursday	Milestone	**Milestone #2 Data Structures and Indexing**
5	25.10.2016 Tuesday	Discussion	Executing the query and Ranking Results
	27.10.2016 Thursday	Discussion	Performance Comparison
6	01.11.2016 Tuesday	Discussion	Testing, reporting, fine tuning, etc.
	03.11.2016 Thursday	Presentation	**Project Presentation**

CME 2101 Project-Based Learning III
Project-1

Table 2. Grading Policy

Item	Expectations & Requirements	Percentage	
Milestone 1	• Class Design (30%) • Reading and Parsing File (40%) • Eliminating Stop Words (30%)	**%10**	**Group (%75) * Contribution Factor**
Milestone 2	• Data structure and Indexing (%100)	**%10**	
Functionality	• Reading and Parsing File (5%) • Eliminating Stop words (5%) • Data structures and Indexing (30%) • Executing the query (Showing the relevant documents) (25%) • Ranking the results from the most relevant one to least relevant (25%) • Performance Comparison (10%)	**%45**	
Project Report		**%10**	
Active participation and contribution to sessions		**%5**	**Individual (%25)**
Attendance		**%10**	
Presentation		**%10**	

Attendance notice: If you don't attend more than 3 meetings, your overall project score will be graded with zero. Non-attendance of 1, 2 or 3 meetings will be punished by reduced marks in your final score for the course.