CSE225 Data Structures, 2020(FALL)

Project #2 (Deadline: 24.01.2021, 24.00)

There will be no extension since the term ends on 24.01.2020 !

Ranking Documents for Information Retrieval with Priority Queues

In WWW, when you enter a “keyword” search engines try to rank the documents according to the similarities between your keyword and the documents lying on WWW. The most famous search engines are Google, Bing, Yahoo... Various machine learning and deep learning algorithms run behind these search engines. On the other hand, the common logic behind them is to rank the documents according to relevance.

In text classification studies, textual materials are represented with the frequencies of the words. Consider the following three documents:

Doc1: “*Text mining studies have gained importance in recent years because of the increasing number of electronic documents like news, social networks, research papers and digital libraries. There is no doubt that this enormous data continues to increase day by day with the contribution of lots of people.*”

Doc2: “*Automatically processing, organizing and handling this text materials are a central problem. The key aim of text mining is to allow users to get information from text materials. Text mining mainly deals with several important applications like information retrieval (IR), classification (i.e., supervised, unsupervised and semi supervised classification), document filtering, summarization, sentiment or opinion classification.*”

Doc3: “*Natural Language Processing (NLP), Machine Learning (ML) and Data Mining methods work together to detect patterns from the different types of the documents and classify them in an automatic manner.*”

When we want to rank documents according to some keyword there are some different methods in the literature. One of the simplest ways of achieving this task is to find a similarity score between the search keyword and the document by using common terms. For example, let’s calculate the similarity score between the search keyword (i.e., text) and three documents.

Similarity score between search keyword: “text” and Doc1: 1 since “text” occurs 1 times in Doc1.

Similarity score between search keyword: "text" and Doc2: 4 since "text" occurs 4 times in Doc2.

Similarity score between search keyword: "text" and Doc3: 0 since "text" occurs 0 times in Doc3.

The most relevant document with the given keyword is the one having the highest similarity score. Consequently, the ranking of the documents according to the relevance to the keyword is *Doc2, Doc1, Doc3*.

In this project, you are to use binomial heaps to implement a priority queue for the task of ranking documents. In other words, you need to rank documents according to given keyword with binomial heaps.

In the folder of files, there are 42 documents.

Steps:

a) Open the folder and read the files.
b) Ask the user for the keyword and pass it to your program as the keyword parameter
c) Decide your heap structure whether you choose max-heap or min-heap.
d) Build a priority queue with the keyword.
e) By using your priority queue data structures in (d), extract the relevant 5 documents with the keyword.
f) Discuss your results in (e) and advantages using priority queue for such a problem.

OUTPUT OF THE REPORT:

1.) (10 points) Screen shot that you are taking the keyword from the user.
2.) (5 points) Screen shot that you are taking the NUMBER OF RELEVANT DOCUMENTS.
3.) (25 points)Put your enqueue and dequeue implementation here (Just copy them from your source code) This part needs to be you own study. Taking the implementation from web will not be accepted.
4.) (50 points)Print your results in (e). The format will be:
Assume that the keyword is text and the NUMBER OF RELEVANT DOCUMENTS is 2.

The relevance order is: Doc2(4), Doc1(1).

Doc2(4): Automatically processing, organizing and handling this text materials are a central problem. The key aim of text mining is to allow users to get information from text materials. Text mining mainly deals with several important applications like information retrieval (IR), classification (i.e., supervised, unsupervised and semi supervised classification), document filtering, summarization, sentiment or opinion classification."

Doc1(1): Text mining studies have gained importance in recent years because of the increasing number of electronic documents like news, social networks, research papers and digital libraries. There is no doubt that this enormous data continues to increase day by day with the contribution of lots of people

5.) (10 points)*Your discussions about the advantages of* using priority queue for such a problem.

VERY IMPORTANT

The main goal of this project is to be familiar with priority queue**. So, use of arrays/linked lists instead of** priority queue **is not acceptable for this project.**

In this project, you are expected to develop an algorithm that is capable of finding a solution to the above problem and ***implement this algorithm in ANSI C that runs under either UNIX or Windows***.

CODE SUBMISSION:

You should use Google Classroom in order to submit your code:

Your any submission after the project submission due date, will not taken into consideration.

1. **Your c file**
   naming standart:
   name_surname.c

2. **Your report**
   naming standart:
   name_surname.doc/pdf

Answers to 1,2,3,4,5.
PLEASE OBEY THE REPORT OUTPUT FORMAT ABOVE.

DO NOT COMPRESS FILES, JUST SUBMIT TWO FILES SEPERATELY. SUBMIT THE REPORT FOR REPORT PART IN GOOGLE CLASSSROOM, SUBMIT C FILE IN C CODE PART IN GOOGLE CLASSROOM.

*****IF YOU COMPRESS YOUR FILES YOU WILL LOSS 20 POINTS.**
*****IF YOU WILL NOT FOLLOW NAMING STANDART YOU WILL LOSS 20 POINTS.**

GRADING:

When grading your project, first we will run your code and compare your results of the output with the results in your report. If you did not get the result by your algorithm, then do not write any answer for that part in your report. Because this is a programming homework.

Then, we will run your program by entering arbitrary words. We want to see if they are working correctly or not, and your project grade will be calculated based on number of your test cases, which are working correctly.

Individual effort is required and strictly expected in your projects. Your source code will be thoroughly checked both automatically by a cross check software tool and by your TA. Any attempt to any type of plagiarism will result in ALL involved students directly failing the class; their names appear in the black list of department and necessary disciplinary action taken by the instructor.

Good luck!!!