COP 5536 Spring 2020

Programming Project

Due Date: March 30th, 2020, 11:55 pm EST

1. General

Problem description

You are required to implement a system to find the n most popular hashtags that appear on social media such as Facebook or Twitter. For the scope of this project hashtags will be given from an input file. Basic idea for the implementation is to use a max priority structure to find out the most popular hashtags.

You must use the following structures for the implementation.

1. Max Fibonacci heap: use to keep track of the frequencies of hashtags.

2. Hash table: The key for the hash table is the hashtag, and the value is the pointer to the corresponding node in the Fibonacci heap.

You can assume there will be a large number of hashtags appearing in the stream and you need to perform increase key operation many times. Max Fibonacci heap is recommended because it has an amortized complexity of O(1) for the increase key operation. You should implement all Fibonacci heap functions discussed in class. For the hash table, you can use an implementation from any library that your programming languages supports (e.g., STL).

Programming Environment

You may implement this assignment in Java or C++. Your program must be compilable with the compilers and run on the following environment:
- Java or g++/gcc on the thunder.cise.ufl.edu server
You may access the server using Telnet or SSH client on thunder.cise.ufl.edu.

You must write a makefile document which creates an executable file named hashtagcounter.

2. Input and Output Requirements

Your program is required to take the input file name as an argument. The following is an example (at the command prompt '$') of a program that reads from a file named file_name.

```
$hashtagcounter file_name
```

Input Format

Hashtags appear one per line in the input file and start with # sign. After the hashtag an integer will appear and that is the count of the hashtag (There is a space between hashtag and the integer). You need to increment the hashtag frequency by that count. Queries will also appear in the input file and once a query appears you should append the answer to the query to the output file. A query appears as an integer number (n) without # sign in the beginning. The answer to the query n is n hashtags with the highest frequency. These should be written to the output file. An input line with the word “stop” (without hashtag symbol) causes the program to terminate. The following is an example of an input file.

```
#saturday 5
#sunday 3
#saturday 10
#monday 2
#reading 4
#playing_games 2
#saturday 6
#sunday 8
#friday 2
#tuesday 2
#saturday 12
#sunday 11
#monday 6
3
#saturday 12
#monday 2
#stop 3
#playing 4
#reading 15
#drawing 3
#friday 12
#school 6
#class 5
5
stop
```

Output Format

For each query n, you need to write the n most popular hashtags (i.e., highest frequency) to the output file in descending order of frequency (ties may be broken arbitrarily). The output for a query should be a comma separated list occupying a single line in the output “output_file.txt”. **There should be no space character after the commas.**

Following is the output file for the above input file.

```
saturday,sunday,monday
saturday,sunday,reading,friday,monday
```

You can have following assumptions

1. No uppercase letters in the input stream
2. No spaces in the hashtags. (only one word per one hashtag)
3. One query has only one integer.

4. Query integer, n<=20. i.e. you need to find at most 20 popular hashtags.
5. Input file may consist of more than 1 million hashtags.

3. Submission

The following contents are required for submission:

1.Makefile: Your make file must be directly under the zip folder and named "Makefile".
No nested directories.

2. Source Program: Provide comments.

3. REPORT:
· The report should be in PDF format.
· The report should contain your basic info: Name, UFID and UF Email account
· Present function prototypes showing the structure of your programs. Include the structure of your program.

To submit, please compress all your files together using a zip utility and submit to the canvas system. You should look for Assignment Project for the submission. Your submission should be named ***LastName_FirstName.zip.***

Please make sure the name you provided is the same as the name that appears on the Canvas system. Please do not submit directly to a TA. ***All email submission will be ignored without further notification. Please note that the due date is a hard deadline. No late submission will be allowed. Any submission after the deadline will not be accepted.***

4. Grading Policy

Grading will be based on the correctness and efficiency of algorithms. Below are some details of the grading policy.
Correct implementation and execution: 60%
Comments and readability: 15%
Report: 25%
Note: ***If you do not follow the input/output or submission requirements above, at least 25% of your score will be deducted, and then more will be deducted for additional mistakes.***
If your code does not compile or run in thunder.cise.ufl.edu server, at least 25% of your score will be deducted, and more will be deducted for additional mistakes. In addition, we may ask you to demonstrate your projects.

5. Miscellaneous

- Your implementation should be your own. **You have to work by yourself for this assignment** (discussion is allowed). Program code will be checked and you may have negative result if it has reasonable doubt.