COP 5536 Fall 2018

Programming Project

Due Date: Nov 15th, 2018, 11:55 pm EST

1. Problem Statement

A new search engine "DuckDuckGo" is implementing a system to count the most popular keywords used in their search engine. They want to know what the n most popular keywords are at any given time. You are required to undertake that implementation. Keywords will be given from an input file together with their frequencies. You need to use a max priority structure to find the most popular keywords.

You must use the following data structures for the implementation.

- **Max Fibonacci heap**: to keep track of the frequencies of keywords
- **Hash table**: keywords should be used as keys for the hash table and value is the pointer to the corresponding node in the Fibonacci heap.

You will need to perform the increase key operation many times as keywords appear in the input keywords stream. You are only required to implement the Fibonacci heap operations that are required to accomplish this task.

2. Guidelines

- **Do not use complex data structures provided by programming languages**. You have to implement Max-Fibonacci heap data structure by your own using primitive data structures such as pointers. But if you want you can use a library implementation for the hash table.

- 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.

- You may implement this assignment in Java or C++. Your program must be compatible 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 keywordcounter.

3. Input and Output Requirements

Your program is required to take the input file name as an argument. Following is an example of a program that read from a file named file_name.

For C++

```
$./ keywordcounter file_name
```

For Java

```
java keywordcounter file_name
```

3.1. Input format

Keywords appear one per each line in the input file and starts with $ sign. In each line, an integer will appear after the keyword and that is the count (frequency) of the keyword (There is a space between keyword and the integer). You need to increment the keyword by that count. Queries will also appear in the input file and once a query (for most popular keywords) appears, you should append the output to the output file. Query will be an integer number (n) without $ sign in the beginning. You should write the top most n keyword to the output file. When "stop" (without $ sign) appears in the input stream, program should end. Following is an example of an input file.

```
$facebook 5
$youtube 3
$facebook 10
$amazon 2
$gmail 4
$weather 2
$facebook 6
$youtube 8
$ebay 2
$news 2
$facebook 12
$youtube 11
$amazon 6
3
$facebook 12
$amazon 2
$stop 3
$playing 4
$gmail 15
$drawing 3
$ebay 12
$netflix 6
$cnn 5
5
stop
```

3.2. Output format

Once a query appears, you need to write down the most popular keywords to the output file in descending order. Output for a query should be comma separated list without any new lines. Once the output for a query is finished you should put a new line to write the output for the next query. You should produce all the outputs in the output file named “output_file.txt”. Following is the output file for the above input file.

```
facebook,youtube,amazon
facebook,youtube,gmail,ebay,amazon
```

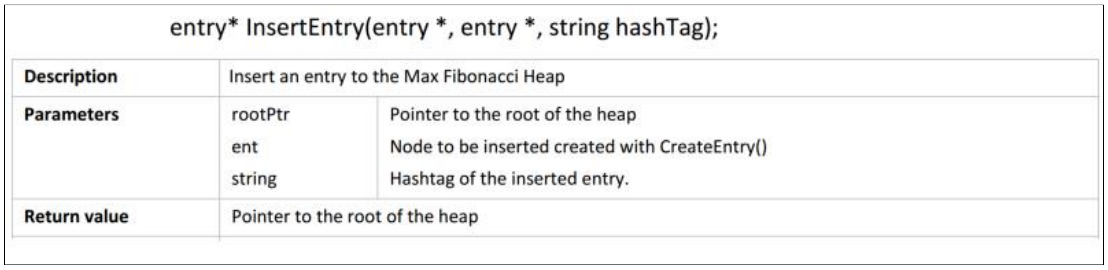
3.3. Assumptions

- Input keyword can be any arbitrary string including alphanumeric characters and special symbols.
- The count/frequency can only be a positive number greater than 0.
- No spaces in the keywords.
- For two keywords to be same, whole word should match. i.e. #youtube and #youtube_music are two different keywords.
- One query has only one integer.
- If there are more than one keyword with similar frequencies, you can print them in any order.
- Query integer $n \leq 20$. i.e. you need to find at most 20 popular keywords.
- Input file may consist of more than 1 million keywords.

4. Submission Instructions

The following contents are required for submission:

1. Makefile: Your make file must be directly under the zip folder. 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. A sample function prototype is shown below;



entry* InsertEntry(entry *, entry *, string hashTag);

Description	Insert an entry to the Max Fibonacci Heap	
Parameters	rootPtr	Pointer to the root of the heap
	ent	Node to be inserted created with CreateEntry()
	string	Hashtag of the inserted entry.
Return value	Pointer to the root of the heap	

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 same 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 day is a hard deadline.

5. Grading Policy

Grading will be based on the correctness and efficiency of algorithms. Below are some details of the grading policy.

- Code compiles: 10%
- Execution of given test case with accurate results: 20%
- Execution of two new test cases with accurate results: 40%
- Comments and readability: 10%
- Report: 20%

Note: If you do not follow the input/output or submission requirements above, 25% of your score will be deduced. In addition, we may ask you to demonstrate your projects.

- Late submission penalty: 20% deduction per day