CS146 Data Structures and Algorithms, Fall, 2018
Programming Assignment 1
Due at 11:59pm, October 15, 2018

Note:

This programming assignment is for individual work only. You are not allowed to use someone's codes or copy from internet, except for Web Crawler codes. Code plagiarism checker tools (Jplag, MOSS, etc.) will be used to check the similarity of codes. Please check on the University Policies in the syllabus.

Problem Statements:

Google's search engine is a powerful tool. Without search engines like Google, it would be practically impossible to find the information you need when you browse the Web. Like all search engines, Google uses sorting algorithm, indexing, searching, and priority queueing techniques to generate search results. Google has a large index of **keywords** where those words can be found and uses automated programs called **spiders** or **crawlers**. What sets Google apart is how it ranks search results, which in turn determines the priority order Google displays results on its search engine results page (SERP). Google uses a trademarked algorithm called **PageRank**, which assigns each Web page a relevancy score.

A Web page's PageRank depends on a few factors:

1. **The frequency and location of keywords within the Web page**: If the keyword only appears once within the body of a page, it will receive a low score for that keyword.
2. **How long the Web page has existed**: People create new Web pages every day, and not all of them stick around for long. Google places more value on pages with an established history.
3. **The number of other Web pages that link to the page in question**: Google looks at how many Web pages link to a particular site to determine its relevance.
4. **How much the webpage owner has paid to Google for advertisement purpose**: Website's owners pay a lump sum of money to Google to increase the priority of PageRank for advertisement of their services/products.

For this programming assignment, your job is to design and implement a simulator of Google Search Engine Results Page (SERP) using **Heap Sort** for the following two major features:

1. For each search keyword/term, display only the top 10 priority search result items/links based on the PageRank.
2. You have a billion google searches a day, design a data structure which lets you pull out the top 10 unique ones at the end of the day.

Programming Requirements:

1. The Google Search Engine Simulator MUST be written in Java and is required to use Max-Heap Priority Queue approach specified in lecture ppt slides and in textbook. (You MUST directly use/revise the pseudo codes provided in textbook to write your Java codes. Any other codes (e.g. copied from internet) will be treated as failing requirements and **will receive 0 point**.)
2. You can either download a Web Crawler software tool from Internet (links provided) or develop your own simple Web Crawler in Java (Link Provided) to collect research results (See the references.)
3. Based on the PageRank calculation, the priority queue with an integer score implemented in a Max-Heap Tree will store a collection of **at least 30 search results** of PageRank for the priority queue. (You may use either ArrayList or Vector in Java)
4. You can make your own way/assumption of assigning a score to each factor for PageRank calculation. The simulator must allow users to enter a score for each of the four factors.
5. When a PageRank factor of a keyword/search item changes it's value, let's say a website owner has paid more money than other websites in the results of search to Google, the display result of the specific priority order increased. In this case, you have to restructure the Max-Heap tree. So your simulator must provide the user interface for changing the score for each PageRank factor.
6. The Google Search Engine Simulator **MUST** contain the following functions:
   The following functions must be implemented.
   1) Max-Heapify()
   2) Build-Max-Heap()
   3) Heapsort()
   4) Max-Heap-Insert()
   5) Heap-Extract-Max
   6) Heap-Increase-Key
   7) Heap-Maximum
7. Each java file/class/subroutine/function call **MUST** contain a header comment at the beginning of it and each end of line in your codes. (Points will be taken off if your codes did not provide comments.)
8. The end of your implementation, compute the complexity of your codes.
9. You **MUST** write a MS Word or pdf report with the following items included in the report:
   a) Explain/Illustrate your Google Search Engine Simulator design and implementation details.

b) A list of classes/subroutines/function calls and explanations of each purpose.

c) Provide a lot of screen shots of each simulation process/procedure including inputs and outputs for each of function listed in item 6 associated with the two major features listed in the end of the Problem Statements. This is to verify your codes and to check to see if your codes match the functional requirements and run correctly by yourself. (You will receive 0 if no self-testing screenshots provided.)

d) The procedure (step by step) of how to unzip your files, install your application, and run/test your codes.

e) Problems encountered during the implementation.

f) Lessons Learned

Submission Requirements:

2. Zip all your files including your report, all *.java files, and a Java executable file (a jar file, so that I can run your codes) into one zipped file with file name: **PA1-Your_First_name-Last_Name.zip**, Any file with incorrect file name will not be accepted and 0 point will be given.
3. The due time to submit/upload your zip file to Canvas is before 11:59pm on Monday, 10/15/2018. After the due time, the submission link will be closed.
4. Grading is based on the creative design, full functioning, completeness and clarity of your codes and report.

Useful Reference Links:

1. Top 20 Web Crawler Tools to Scrape the Websites
2. Open Source Web Crawler for Java
3. How to make a simple web crawler in Java
4. **How to make a Web crawler using Java?**
5. Make Java executable Jar file
6. https://trends.google.com/trends/?geo=US