EECS 293
Software Craftsmanship
2020 Spring Semester

Programming Assignment 6

Due at the beginning of your discussion session on

February 25-28, 2020

Reading

In addition to the following topics, the quiz syllabus includes any material covered in the lectures:

- In canvas, the Pseudo-Code Cheat Sheet and the Quick Reference on Routine Names,
- In Code Complete, Section 9.3 ("Design the Routine" only) and 19.6.

Grading Guidelines

In Programming Assignment 6, high-complexity and code repetition do not necessarily trigger an automatic C.

Starting with Programming Assignment 7, an automatic C (or less) will be triggered by improperly named routines.

Assignment

In this assignment, you will design a data structure to find the most assiduous clients to a Web server.

Gigatron has created a Web site that, much to everybody's surprise, has evolved into a major success. The Web site is hit by millions of views on normally days, and many billions on peak days. At this point, Gigatron should consider a major upgrade, yet the Gigatron team is unconvinced that the upgrade is needed. Gigatron suspects that most hits originate from automatic bots and search engines. To confirm or refute their suspicion, Gigatron needs a data structure to

collect the IP address of the most frequent clients. The data structure should support the following operations:

- NEW: create a new instance of the data structure in which all IP addresses start with no hits
- INCREMENT(x): increment the number of hits originating from IP address x
- COUNT(x): returns a count of the hits from IP address x

Gigatron had originally implemented this data structure as a hash table. However, the implementation was impractical because an excessive number of IP addresses have to be tracked, and the hash table ran out of memory. In other words, Gigatron needs a data structure that uses much less memory than the number of IP addresses. On the other hand, the data structure can be simplified by making COUNT(x) into an approximate estimate that is roughly informative of the client behavior, since even an approximate count would enable Gigatron to classify clients according to their general activity level and to select a few for further deeper analysis. You have been tasked with creating the pseudo-code and implementation of the Gigatron data structure.

Submit the pseudo-code of your algorithm. You should state the classes to which each method belongs and explain the abstraction that your classes capture. You should submit a justification for the algorithm's correctness, an analysis of its running time, and a few examples of the algorithm operation. It is your job to specify the representation and format of the algorithm's arguments, such as the IP addresses, and your submission will be evaluated in part on the adherence of argument format to good craftsmanship principles. Your algorithm should be written as a flexible package that can be reused in other projects, and thus should avoid using any explicit form of input and output.

No implementation is required: you will implement your pseudo-code in the next programming assignments. After this assignment, you will not be allowed to make major changes to the pseudo-code.

Canvas Resources

A module (Pseudocode Examples) contains a cheat sheet and five examples of pseudo-code from various textbook.

Discussion Guidelines

The class discussion will focus on the pseudo-code. The pseudo-code must be of sufficiently good quality that you can easily generate code from it in the next programming assignment. You may also be required to walk through your pseudo-code on some examples. It is better to present correct pseudo-code for a clearly specified subset of requirements rather than incorrect pseudo-code, or to address an unclearly stated subset of requirements.

Submission

Submit an electronic copy of your pseudo-code and any other ancillary documents on Canvas.