Programming Project and Paper Assignment

Collaboration Policy:

Individual

Code Policy:

You must implement your own code. Plagiarism will not be tolerated.

Objectives:

To empirically test Open and Closed Hashing techniques to confirm average case time efficiency (i.e. constant time $O(1)$) for search operations. To write a research report on the experimental process and the time efficiency results.

Program Requirements

Compare load factor and search time efficiency of Open Hashing (Separate Chaining) and Closed Hashing (Open Addressing). The hashing algorithms and data structures are fully described in Section 7.3 in the supplemental course textbook (also found on OAKS in the content section).

Paper Requirements

- Develop hypothesis and experiment design
- Implement your own code that adequately tests your hypothesis
- Conduct experiments
- Record results
- Write paper

The submitted paper must be formatted in ACM (http://www.acm.org/sigs/publications/pubform.doc) style, and must include (at minimum) the following sections:

- Introduction
- Methods
- Results
- Conclusion
- References

Some additional advice on writing technical reports can be found at http://www.csee.umbc.edu/~sherman/Courses/documents/TR_how_to.html.

Code Specification

You shall create two new classes named `OpenHashing` and `ClosedHashing`.

You are responsible for fully defining these two classes (i.e. constructors, methods, instance variables, class variables, etc.) and fully implementing the search, insert, and delete operations needed to conduct the experiments. Note: you are also responsible for defining *h(K)* the hashing function. You may research existing ones, or come up with your own, your choice.

Where needed, these two classes must reuse other data structures developed in this course (e.g., `ArrayList`, `SinglyLinkedList`, `Stack`, `Queue`). You may not use data structures found in the Java API.

Experiment Specification:

Create a test class named `TestHashing` with a main method that is used to execute the experiments in the submitted paper.

The input to your experiments shall be words found in some random text file. You may use any text file used in the past, or publically available ones that can be found on textfile.com. To assess search efficiency you shall:

- Calculate the load factor (α) that is then used to determine the time efficiency (i,e, number of probes) for *successful searches* (S) and *unsuccessful searches* (U). Specifically, for Open Hashing S and U are defined in Eq. 7.4 on page 271 in the supplemental textbook, and for Closed Hashing S and U are defined in Eq. 7.5 on page 273 in the supplemental textbook.
- Calculate the estimated time (in seconds) required for successful searches and unsuccessful ones.

The two assessments outlined above shall use:

- Different word combinations, i.e. different text files, or one text file that is equally partitioned, and
- At least two different hashing functions.

An analysis of the experiment results must be provided in this section, and how they support, or do not support, average case time efficiency for search operations.

Submission:

Create a zip file that **only** includes: 1) completed paper as a PDF document, 2) all the source code (i.e. .java files) and text files required to compile and run all your experiments. The name of the zip file must be your last name. For example, *ritchie.zip* would be correct if the original co-developer of UNIX (Dennis Ritchie) submitted the assignment. Only assignments submitted in the correct format will be accepted (no exceptions). Please submit the zip file (via OAKS) to the Dropbox setup for this assignment by the due date. You may resubmit the zip file as many times as you like, Dropbox will only keep the newest submission.

If (for any reason) the assignment is not submitted in the proper format, it will not be graded, i.e. a zero will be given.

Late assignments will not be accepted – no exceptions (please do not email me your assignment after the due date, I will not accept it).

Please feel free to setup an appointment to discuss the homework problem. I'll be more than happy to listen to your approach and make suggestions. However, I will not tell you how to code the problem.

If you're unsure about any aspect of this assignment, you must ask! It is your responsibility to flush out questions you may have about the assignment prior to the due date. I can't help resolve an issue, or add clarity, if you don't ask.