

CS 400/600 Data Structures and Software Design

Programming Assignment #4 Hash Table

Overview

The objective of this assignment will be to implement a closed-double hash, and submit a report that gives discussion of the result that you get based on the load factor of the hash table, alpha.

Programming

For the programming section, you will create a closed double hash to index positive integer values (employee IDs) between 1 and 999,999. You should create a hash of size $N = 32,768$ (32k) slots.

You will be provided a C++ abstract class called Hash, you should construct class ClosedHash as a subclass of class Hash, as follows:

```
class ClosedHash : public Hash { };
```

Your hash should use the EMPTY constant to mark empty hash buckets, and should use and fully support tombstones (using the TOMBSTONE constant) for deletions.

You will also be provided with an extensive main program for testing your hash implementation. The primary hash function $h_1(k)$ and the secondary hash function $h_2(k)$ are also given. The primary hash function gives the home location for a particular key. The secondary hash function determines the probe offset for a particular key.

Report

Along with your working source code (.cpp and .h files), you should submit a brief (< 5 pages) report describing how the closed hash function performed as a function of the load factor of the hash table, alpha. You should perform a variety of searches, and compare the number of nodes searched for each index. You should address questions such as:

- How does the hash perform in terms of search efficiency when the hash table is (1) nearly empty (the alpha is very small, like less than 0.1), (2) moderately full (alpha is around 0.5), (3) very full (alpha is larger than 0.8).
- Give some intuitive comments about the performance of your hash table regarding the load factor, alpha.

Your report should be supported with data in the form of tables, charts, graphs, etc.

Make sure to base your report on average results over many searches, and not just one search.

The preferred format for your report is a Microsoft Word file named proj4-report.doc, included in your project 4 directory and turned in with your source code. Other formats (like pdf format) may also be acceptable, as long as the report is clear.

Requirements

1. Your code should follow the Code Standards handed out during the first day of class. Your code will be graded according to its correctness, efficiency, organization, and readability.
2. Make sure that each file includes your name in the header comments.
3. Turn in all files needed to compile and execute your code (including any needed files from the previous lab) via webCT. If for some reason WebCT is unavailable, submit your source code by email to wlodarski.4 AT wright.edu. If you want, you can also cc to the instructor Meilin Liu, whose email address is meilin.liu AT wright.edu.

4. If there are any special instructions that I need to know about, be sure to include a file named Readme.txt in your project 3 directory detailing them.
5. The grader will test your programs under the schools UNIX environment, e.g., unixapps1.wright.edu. It is YOUR responsibility to make your programs workable and runnable by others under schools UNIX environment.
6. The programming assignment is individual. You must do the project by yourself. If you allow others to copy your programs or answers, you will get the same punishment as those who copy yours.

Graduate Students

CS 600 students should evaluate an additional hashing method (other than closed double-hashing), and include these results in the final report (for example, you might choose to investigate the performance of a closed hash using pseudo-random hashing. You can also implement a chained hash table if you like).