

Programming Assignment V COP3502H

Due: 4/23/07 11:59pm

The CEO of the Computer Sciences Corporation, Dr. Link Queue has just hired Dr. Johnny Hash, the new vice president of product development. The first thing Dr. Hash does whenever he starts a new job is to have the company design and implement a hash table. The problem is Dr. Hash has memory loss and does not remember which hashing functions work best. Therefore, in this assignment, you will implement a hash table and several different hash functions, examine their performance and produce a report on your findings.

Requirements

You will develop a hash table that will be used to insert, delete, and search for keys. In this case, the keys will be social security numbers (a nine digit number). You will be given a file with 1000 random social security numbers to use for testing. For collision detection, your hash table should use chaining. You should be able to use your linked list code for the chains. The hash table will simply be an array where each element is a pointer to a linked list (or NULL if it is empty). The size of your hash table should be on the order of a 100 or so. You will also need to implement the 5 hash functions we discussed in class,

Hashing by

1. Division
2. Multiplication
3. Folding
4. Middle squaring
5. Truncation

For each of these hash functions, you must decide what the constants should be.

You should provide a menu driven interface that lets users try the different hash functions and perform other operations. These operations should include

1. Initialize Hash Table
2. Choose Hash Function
3. Load SSN Numbers from file
4. Insert a SSN number
5. Delete a SSN Number
6. Search for a SSN Number
7. Print out Hash Table

For operation 7, you should print out a hash table by simply printing out the number of elements in each slot. This will let you determine how many collisions there are in each slot. Make sure you handle error conditions.

You will also write a report on the performance of the hash functions. For each function, you must state what values you chose for the constants and why as well as report on the number of collisions from the data in the testing file. You must also answer the question, “Which hashing function performed best? worst? Why?” The report only needs to be one or two pages in length.

Extra Credit

Come up with a hash function of your own design and compare its performance with the others you have implemented.

Documentation

You must document your routines. Each function should be commented with what the function does, its input parameters, and what it returns. Comments are very important and will account for 20% of your grade.

Testing

You must test your program and ensure that it works. You should provide output files that show you can insert and delete, and search for keys in the hash table.

Deliverables

You must submit your report, all source files, and your test files to Sean and Dr. LaViola by the due date. Also, include a README file explaining what problems you encountered. Note the code must be able to be compiled and executed on the Olympus unix system.

Grading

Grading will be based on the following:

- 60% correct functionality
- 20% documentation
- 20% report