CMPS 1600 Introduction to Computer Science II – Spring 14

2/21/14

5. Homework

Programming portion due **Thursday 2/27/14** at 11:55pm on Blackboard.
Written portion due the next day at the beginning of class.

Please zip the project directory for this homework, and use the following naming convention for the name of the project (and directory): `lastName_firstName_hw5`
In order to receive any credit for the programming portions, you are required to thoroughly comment your code.

1. **Hashing (10 points)**
   Consider the hash function $h(x) = x \bmod 10$, and hash tables of length 10.

   For each of the hash tables below, show the resulting hashtables after inserting the following sequence of numbers: $21, 53, 113, 89, 54, 69, 19$. Then discuss what happens when you add the number 29, and argue how many more numbers with hash code 9 you can add to this hash table before any problems occur, and describe nature of any problems.

   (a) (2 points) A chained hash table.
   (b) (2 point) Open addressing with linear probing, assuming the table will not be resized.
   (c) (2 point) Open addressing with quadratic probing, assuming the table will not be resized.
   (d) (2 point) Open addressing with linear probing, assuming the table will be resized according to the code provided in `OpenAddressingSimpleHashtable` in `hw5.zip`.
   (e) (2 point) Open addressing with quadratic probing, assuming the table will be resized according to the code provided in `OpenAddressingSimpleHashtable` in `hw5.zip`.

2. **Hashing Runtimes (10 points)**
   Consider the code provided in `hw5.zip`. It contains three text files `mardigras.txt`, `hodge.txt`, `searchTerms.txt`, as well as the `ArrayBag` class, and hashtable implementations `ChainedSimpleHashtable` and `OpenAddressingSimpleHashtable`. The main method in `Tester.java` contains code to read the text file `searchTerms.txt` and to store each word as an individual string into an `ArrayList`.

   Your task is to measure and compare the runtimes of the `add` and `find/contains` methods in `ChainedSimpleHashtable`, in `OpenAddressingSimpleHashtable` (both for linear probing and for quadratic probing), and in `ArrayBag`. To test the runtime of `add`, you should use `mardigras.txt` and `hodge.txt` as two separate test cases, add all the words from one file to the hashtable or bag, and measure the total runtime. To test the runtime of `find` and `contains`, you should measure the total runtime of iterating over all the words in `searchTerms` and attempting to find them in the hashtable or bag.

   Report your findings in a text document. Compare the runtimes. What conclusions can you draw? Do the runtimes change if you vary the size of the initial bag/hashtables?