CS435 Homework Assignment 7

Due date: ____________________

Turn in all code on both paper and by email (to dbahr@regis.edu with “CS435 Homework” in the subject line).

Please include comments in all of your code. Describe what your code does, and describe anything that is tricky or unclear. More comments are better than fewer comments.

Problem #1: Given keys {4371, 1323, 6173, 4199, 4344, 9679, 1989} and a hash function hash(x) = x mod 10, show the resulting (with pictures):
   (a) Open addressing hash table with linear probing. Use an array of size 13.
   (b) Open addressing hash table with quadratic probing. Use an array of size 13.
   (c) Separate chaining hash table. Use an array of size 10.

Problem #2: You want to put 1000 elements in a hash table and you want the search to examine just 1.2 or fewer elements on average. How big does the table’s array need to be for the separate chained method? Show your calculation.

Problem #3: Consider a hash table for the Department of Agriculture that maps the cost of a bushel of grain in dollars and cents (e.g., $2.99) to the name of the grain (e.g., corn). The cost, without the dollar sign, is the hash table key. Without using ASCII conversions, write code for a simple hash function that converts the key to an array index. Assume that the cost can vary from 0.01 to some arbitrarily huge number; but assume that the hash table array will have a reasonably small size (say somewhere between 50 and 100 elements). For full credit you must include comments in your code that completely describe how your hash function works.

   Note: I want a complete Java method. It should take the key as a parameter and return an array index. You should attempt to minimize collisions.

Problem #4: What is the growth rate of your function in Problem #3? You may assume that any pre-existing Java methods have O(1) growth rate.
Problem #5: Write a Java method that calculates the load factor for any given array. For full credit you must include comments in your code that completely describe how your load factor method works.

Note: I want a complete Java method. It should take an array as a parameter and return the load factor.