CSC 403 W18 Homework 5 part I

Objectives:

- Practice creating a hashCode function for a custom class
- Consider issues related to accurately determining algorithm runtimes.
- Appreciate the role of empirical testing in algorithm analysis
- Practice analyzing data.

Q1. What is the algorithmic complexity of creating/populating a hash table of size N?
Q2. What is the algorithmic complexity of inserting (put) the $N+1^{St}$ key into a hash table of size N?
Q3. What is the algorithmic complexity of searching (get) for a key in a hash table of size N?
Q4. Which Hash implementation performs better on the following scenario?
- a mixture of a large number (but near equal) number of puts and gets.

1. Create a class to use as your Key. This should have at least two fields. Suggestion: String and Integer. Your class must include: constructor, toString, hashCode and equals. Add other methods if you wish.

   Suggestion: use the framework on 462 to create your hashCode function

2. Devise and implement a testing program to empirically evaluate the performance of:
   - `testLinearProbingHashST algs34`
   - `testSeparateChainingHashST algs34`

   in the context of the above questions.

3. Create a report in which you:
   - Describe your testing approach and in particular how you attempted to 'isolate' the operations of interest.
   - Present your collected data (tables and graphs)
   - Present your analysis and/or conclusions.
   - Reflect on the possible limitations of your testing program/results

You may work with a partner on this assignment if you wish. In such case, each person will turn in a copy of the relevant documents with both partner's names. And each individual will upload a separate document describing how the partnership worked and an estimate of each person's % share of the work.

Turn in:

Report document
Source code

Extra for fun:
Examine the effects of changing the load factor balance points:

LinearProbing (e.g. 3/4 instead of ½)
SeparateChaining (e.g. 5 instead of 10)

Vers 1.0