Programming Assignment 1

Henry Z. Lo

Due: Tuesday, June 10 at midnight

1 Goals

This assignment will help you with the following:

1. Review hash tables.
2. Investigate various collision resolution methods.
3. Recall how to work with abstract classes.

2 Overview

Given an `AbstractHashSet` class, create the following classes:

1. `LinearHashSet`, a linear probing hash table.
2. `QuadraticHashSet`, a quadratic probing hash table.
3. `ChainingHashSet`, a chaining hash table.

You will also be given a `Main` class. The main method in this class will output data like this:

```
linear,0.00,0.00,0.00,...
quadratic,0.00,0.00,0.00,...
chaining,0.00,0.00,0.00,...
```

Each line here represents an experiment, in which values are added into the hash set, one by one. Each number represents the total number of collisions encountered after that add. First number is after adding one element, and so on. The numbers are floats, because they are averaged.

This comma-separated values (CSV) format can be put into any spreadsheet program. The idea is that if you plot them, you can see how well each collision resolution method resolves future collisions.

3 Instructions

1. Code `LinearHashSet`, `QuadraticHashSet`, and `ChainingHashSet`, which all subclass the `AbstractHashSet` given.
2. Implement the `size` and `add` methods.
3. You don't need to implement `get` or `remove`, but it should be easy.
4. Constructors should take in the number of slots in the hash set as an argument.
5. The collision variable should count the total number of collisions. Note:
   - Probing methods might collide multiple times per insert. Increment the collision variable every time this happens.
   - Use step size of 3 for the linear and quadratic implementations.
   - For `ChainingHashSet`, you can directly add an element to the end of the `LinkedList`. Consider the length of this `LinkedList` to be the number of collisions needed for this insert.
6. Classes must compile.
7. Run the main method, save the output into a file named "data.csv".
8. Put all deliverables in your cs310/pa1 folder. Deliverables include:
   - AbstractHashSet.java (do not modify)
   - Main.java (do not modify)
   - LinearHashSet.java
   - QuadraticHashSet.java
   - ChainingHashSet.java
   - data.csv
   - memo.txt
9. Answer the questions in section 4, put answers in memo.txt.

4 Questions

1. For each implementation, what is the average number of collisions for 1000 inserts? Which implementation resulted in the least collisions? What are its drawbacks?
2. The `collisions` variable in `AbstractHashSet` is cumulative. What would you change to have it represent the number of collisions for the previous insert, rather than for all previous inserts?
3. What could happen if your step size was not co-prime with the hash table size?

5 Tips

1. The only difference between `LinearHashSet` and `QuadraticHashSet` is one expression. You may want to have both subclass one parent class. This up to you.

2. `ChainingHashSet` is fundamentally different; its main data structure should be an `ArrayList` of `LinkedList` objects.

3. The number of collisions should constantly go up, because it is cumulative. You only need to find the places where it should be incremented.