CS 201: Data Structures
Homework 12

Layla Oesper
Fall 2017

This assignment is due by **10pm on Wednesday November 15** and is worth 18 points.
Note: You may not use late days on this assignment, even if you have some remaining.

# 1 Goals

The goal of this assignment is to get a better understanding of hash codes, and hash tables.

# 2 Setup and Instructions

This assignment is primarily a problem set, and should be completed individually. You are welcome to talk with others about the problems, but you must write up your assignment by yourself. If you talk with others about the problems, please note their names at the top of your homework submission.

You should not use any notes from discussions with others while writing up your assignment. If you need notes, it likely means you still don't quite understand how to do the problem. Go back over your notes, ask more questions, or try to solve a similar problem on your own. Then come back a day or so later and try to write up your solution without your notes.

As always, you will submit your work on Moodle. This means that you will need to create a single PDF of your solutions to the written exercises. You have a few options for this:

- Write it by hand and scan it into a PDF. This is likely to be the easiest thing to do for the drawings, and is perfectly acceptable. Just make sure it is legible. The library has a scanner.
- Write it in a word processing program and put in your drawings as scans or generated using some drawing program, then export or print it to PDF.
- Write it in LaTeX and use `\includegraphics` to put in your scanned drawings or or drawings generated from some other program and typeset it as a PDF.

Remember, the goal of your writeup is to communicate your understanding to us, so we can't accept writeups that we cannot read (whether that's due to handwriting, image quality, or file format issues).

# 3 Questions

Submit clear and concise answers to the following questions.

1. Suppose you are writing a `HashMap` class that uses open addressing to resolve collisions. You are going to create a `TableEntry` class that will be stored at each position in the array. What instance variables should you have in the `TableEntry` class and why? How will those instance variables allow you to remove entries from the table?
2. Suppose you are using quadratic probing to resolve collisions. Imagine you have an array of length 5 and the first spot you try to add a particular item is 0. If the array isn't full, could you ever fail to find a spot to add the item? *Hint: Try generating the sequencing of spots that quadratic probing will look at.*

3. Suppose you have keys that are Characters, a hashCode() function where $a = 1, b = 2, \dots, z = 26$, and you are using the standard mod operation as a compression function. Your hash table begins as an array of size 7 and you are using linear probing to handle collisions. Draw the array after you perform add on each of the following key-value pairs (key → value): (a → 4), (b → 2), (n → 14), (g → 9).

4. Now I call `hashMap.put('g',10)` on the map you created in question (3). What entries do I look through in the array? What will the array look like after I make this call?

5. After finishing question 4, what is the load factor $\lambda$ for the array?

6. Now imagine that you'r using chaining (with a linked list) to resolve collisions. Repeat question (3) with this new collision scheme, drawing the array after all of the key-value pairs have been added.

7. Let's say you're using chaining for collision resolution, and you have a spot in the array with a linked list containing 3 items. If you resize the array, would it be correct to compute the new spot for the first item in the list and then copy the linked list to that spot in the new array? Make sure you understand why or why not.

8. Suppose I have a `ZooAnimal` class that has instance variables for the animal's name, species, and year of birth. Propose a way to implement `hashCode()` for ZooAnimals such that two ZooAnimals that are equal (have the same values for all instance variables) will return the same hash code and ZooAnimals that are not equal will tend not to return the same hash code (i.e., just returning the same code for all ZooAnimals is not a correct answer). You may use any methods you want in your description.

9. Suppose you have a hash table of size 100 (yes, I know this isn't a prime number). You are using linear probing to handle collisions in the table and you find that the expected number of comparisons you need to make for finding an item in the table is 1.75. How many items are currently stored in the table?

## 4 Submission and Grading

Submit a PDF of your solutions named `[your_last_name]HW12.pdf` to Moodle. For example, I would submit `OesperHW12.pdf`.

This assignment is worth 18 points.

- Each questions is worth 2 points.
- You will only receive full credit for each question if you answer if correct as well as clearly and concisely articulated.

Start early, ask lots of questions, and have fun! Layla, the lab assistants, and the prefect are all here to help you succeed - don't hesitate to ask for help if you're struggling!