Hash Table Assignment

Due: **Marks 25**

Create a generic class called HashTable that allows us to store whatever information we like. Your hash table must maintain a maximum load of 70% to ensure optimal performance. Your class must implement the following basic functionality and must do so efficiently.

(10 marks)

add(T) - Allows us to add an object of the generic type. This method must use the hashCode() of the object it is adding, handle collisions, and resize the array as needed. By default the array will resize when the load of the table is above 70% but the setMaxLoad () function can change this percent level.

remove(T) - When we remove an item simply set the position to null unless there was a collision in which case one of the collisions must be moved to the newly emptied position.

get(key) - Given the full hashCode of an object return the actual object or null if it's not there.

getLoad() - Return how full the hash table is as a number from 0-1, 0 being no elements in the array, and 1 being all spots in the array are full (please never return 1.)

setMaxLoad(percent)

- Allows us to set the load as a percentage. The parameter should be a number between 0.1 and 0.8. This is the number that add will look at to decide if it is time to resize.

setLoad(percent)

- Allows us to set the load as a percentage. The parameter should be a number between 0.1 and 0.8. If the number is greater than the maxLoad then do nothing, otherwise reset the hash table to force the load to be what was asked for. This may result in the array shrinking or growing.

toArray() - Return an ArrayList of objects with no nulls in it. The order of this list is unimportant

---

Practical applications

Use your Hash Table class to solve the following problems

1. At the last ANS I was defeated by both Mr. Ing and Mr. Delnea at scrabble. Next year I think I'll bring my laptop to help level the playing field. I want you to write a program that allows me to enter up to eight letters (seven plus one on the board) and quickly display all possible words I can make using all of the letters I entered. To help with this process I've included a file called dictionary.txt in the class folder. You should realize that an eight letter word has 40320 permutations and that my dictionary has 84219 words in it. If you did a linear search the problem that would be 3,395,710,080 comparisons and a binary search would be 645120 comparisons. I need much faster than that if I'm going to have a chance of winning next year.

   Save as HashAssign1.java **(5 marks)**

2. I'm a bit of a creeper. I wrote a computer program that scans blogs looking for certain emotional keywords then used an IP tracker to find exact locations of the bloggers. For each entry I rated the amount of Love/Hate, Happiness/Sadness and amount of Excitement/Boredom. For each of the three continuums my program gives a number between -100 and 100 for each blog entry. For example the quote "I Love physics tests they really tickle my senses, but it's too bad some people fail." my program rates as:

    LH: +80
    HS: -20
    EB: +40

I made a data file called "creeper.txt" that has a number of entries, each of the form

    X Y LH HS EB

where X Y is an x,y location on my map and LH, HS, EB are the emotion values. Write a program that reads in all of the entries in my file and stores them in your Hash Table in some custom objects. Then your program needs to have a visual interface where you display the map of Windsor and allow users to click on locations and you display the emotions in that area (within a circle of about 10 pixel radius from point clicked.)

Emotions are converted to colours in the following manner:

LH: -100 → 100		Red	0 → 255
HS: -100 → 100		Green	0 → 255
EB: -100 → 100		Blue	0 → 255

The colour values for each pixel should be an average of all the entries for that x,y location.

Save as HashAssign2.java **(10 marks)**

(note: If you think I'm too creepy look at "We Feel Fine" http://www.ted.com/talks/lang/eng/jonathan_harris_collects_stories.html)