A. Midterm Project – Social Network Edgelist I

This project is along the same lines as the homework assignments we have been doing, but it uses actual Twitter data collected between July 1st and 7th, 2012. The data is available from http://snap.stanford.edu/data/higgs-twitter.html and also from the course website. The file we will work with is called social_network.edgelist. It contains information about who is following whom. This information has been anonymized by replacing usernames with id numbers (from 0 to 456630). The format of the file is int <space> int, where the first int is the id of the user doing the following and the second int is the user being followed.

Your task is to create a TwitterUser class to represent the users in this dataset. The class should meet the following requirements:

- Store the user's id number and who that user has followed
- Implement Comparable to sort TwitterUser objects based on id number
- Implement Cloneable to make a deep copy of the object
- Write a recursive getNeighborhood method similar to the one you developed for the FacebookUser class in 4. Fourth Assignment. The method should take an id number and depth as arguments and return an ArrayList of the users that TwitterUser is following, and who those users are following, and so on, up to the requested depth
- Include any other methods to the TwitterUser class that are necessary to meet the project requirements

After you have developed the TwitterUser class, create a driver program to do the following:

- Read in the information in the data file and store it in a Collection of TwitterUser objects
- Unit test your getNeighborhood method
- Check that your clone method is creating a deep copy by cloning the first TwitterUserobject (id=0), setting the clone's "following" list to empty, and making sure the original object still has the contents of its following list (i.e. you want to make sure that changing an attribute of the clone does not affect the original)

Hints:

These data file is quite large, so you will need to be patient. It takes approximately 60 seconds to read in the data file using a Scanner object and an average laptop. Alternatively, you could use a BufferedReader to read it, which is more efficient than a Scanner object. The BufferedReader class is described in the Java documentation here:

http://docs.oracle.com/javase/7/docs/api/java/io/BufferedReader.html

When working on the getNeighborhood function, be careful not to add any id to the list that is already in there or a stack overflow may occur.

You will be graded according to the following requirements:

- The TwitterUser class contains the requested fields
- The TwitterUser class implements Comparable based on id
- The TwitterUser class implements Cloneable to make deep copies
- The getNeighborhood method works correctly
- The driver reads in the data file
- The driver creates TwitterUser objects that accurately represent the data in the file
- The driver program tests the getNeighborhood method
- The driver program tests that the clone method makes a deep copy
- The program compiles and runs
- The program is clearly written and follows standard coding conventions
- ***Note:*** *If your program does not compile, you will receive a score of 0 on the entire assignment*
- ***Note:*** *If you program compiles but does not run, you will receive a score of 0 on the entire assignment*
- ***Note:*** *If your Eclipse project is not exported and uploaded to the eLearn drop box correctly, you will receive a score of 0 on the entire assignment*
- ***Note:*** *If you do not submit code that solves the problem for this particular project, you will not receive any points for the program's compiling, the program's running, or following standard coding conventions.*