CMPSC 497B/CSE 597B HW2. DUE 10/13

Instructions: Debug/develop on the VM. Afterwards use AWS

You cannot share code with other students or view code written by other students. Do not post code online. Violation = Academic integrity violation.

Your submission will be a zip file containing your code and the output generated by AWS. The controller logic should be in the file TopWords.java

In this homework, we will be processing tweets. Tweets are encoded as json objects, which you can examine from the file shorttweets.txt on the course webpage.

To extract tweets from json objects, download the zip file minimal-json.zip from the course webpage. Unzip it and copy/paste all 10 files into your vm in the same directory as your homework files. The file ReadTweets.java contains sample code for extracting tweets. To compile this code:

```
mkdir hw2classes
javac -d hw2classes *.java
## Don't forget the dot at the end:
jar -cvf tweets.jar -C hw2classes/ .
## The following will make the jar executable from the command line
## by specifying the main class to run
jar -ufe tweets.jar org.cse97b.json.ReadTweets
## execute the jar
java -jar tweets.jar
```

Problem 1. *In the file TopWords.java, write code to find the k most frequent words in tweets (sorted in decreasing order of frequency). Thus, if $k = 3$ and the most frequent words are "tweet" (occurring 1000004 times), "hadoop" (occurring 1000003 times), and "food" (occurring 1000001 times), then the output should be*

```
tweet   1000004
hadoop  1000003
food    1000001
```

Here are requirements for this homework:

(1) *The controller code is in the file TopWords.java*
(2) *The first command line argument is input directory, the second is output directory, and the third is the value k (communicating command line arguments to mappers is discussed in Lab 2)*
(3) *We only care about letters and no other symbols, so for each tweet extracted from a json object, split by non-letters: valueAsAString.split("[^A-Za-z]+")*
(4) *Convert all upper case letters to lower case (check the Java String operations)*
(5) *Once you have developed/debugged on the VM, run it on AWS using the input directory psucse97data/tweets/ using $k = 50$*
(6) *Zip your code and the output file from Amazon and submit it to ANGEL.*

You will be graded based on the following:

- *Correctness of your code.*
- *Quality of the code. This includes (but is not limited to):*
  - *How well your code uses public static final variables*
  - *If you use a comparator, how well your compare method satisfies the requirements of a comparator*
  - *Appropriate use of counters (e.g., for tracking exceptions)*
  - *Comments*