CMPSC 497B/CSE 597B HW1. DUE 9/26

Instructions: Why is this homework important? You will only get access to AWS after successfully completing this homework. AWS will be used in subsequent homework. You cannot share code with other students or view code written by other students. Violation = Academic integrity violation. Your submission will be a zip file containing three files: Controller.java, WordFreq.java, Reverser.java

Problem 1. *In this homework, we will extend the WordFreq code from Lab 2 so that the output is a list of words and frequencies sorted in decreasing order by word frequency. Make sure it runs on inputs like g1, g2 (from the lab). The output should have a format like:*

```
0.45    hello
0.23    world
0.22    hadoop
0.09999999    is
0.00000001    the
```

To do this, your code will probably need to set up 2 mapreduce jobs - one to get the word frequency, and another to rearrange the output in this format.

- *Your code should consist of 3 files:*
  (1) *Controller.java (with the main() function that sets up the mapreduce jobs to run one after the other).*
  (2) *WordFreq.java (which contains the WordFreq class that has the map, reduce, and any sort/partition code needed for computing word frequencies)*
  (3) *Reverser.java (which contains the Reverser class that has the map, reduce, etc. code that will be run after the word frequency computation to produce the final output)*

Hints:

- *Writables*
  - *There are different types of suitable Writables, such as org.apache.hadoop.io.FloatWritable*
  - *context.write(key, value) is used to output things to the hadoop framework. Both key and value have to be Writables (e.g., Text, IntWritable, FloatWritable, etc.).*
  - *Reuse your writables (i.e., declare them in the class, not the method, and use their set() function instead of creating new ones all the time).*
- *Directories*
  - *You will need to store intermediate results (like word frequencies computed after the first mapreduce job) in an intermediate directory (not the final output directory) and the subsequent mapreduce job (which runs Reverser) will read from this intermediate directory.*
- *Running/compiling*
  - *You may want to put them all in the package org.psuname.hw1 (where psuname is your psu name).*
  - *You can get all the files to compile with javac using *.java (rather than naming the files individually)*
  - *When you run the jar file, remember to give it the main class (org.psuname.hw1.Controller)*