CMPSC/DS 410: PROGRAMMING MIDTERM

Instructions:

(1) **What you can access**
   (a) the cluster
   (b) the canvas pages for this class (but no external sites, even if linked from canvas)
   (c) your github classroom repository (e.g., on github.com or cluster)
   (d) tophat
   (e) gradescope
   (f) zoom for recording your work.
   (g) nothing else: no email, no google, no how-to sites, no facebook, no netflix, twitter, no other zoom sessions, etc. No chatgpt, no catgpt, no other large language models, no small language models, no medium language models. Improper internet access will result in an academic integrity violation.

(2) **Zoom**
   (a) Share full screen (not a window).
   (b) Record (only end recording after you have submitted to gradescope).

(3) Get the exam using updatestarter.

(4) **Submiting your exam:** gradescope

(5) **if terminal is not working, use Putty to connect to the cluster**

Question 1. This question uses the orders dataset in **/datasets/orders**. We want to study the correlation between words in an item description and its unit price. In this study, the cost of an item is shared equally among word appearances. For example, for an item with description "dog sized treat" and price 0.60, the word "dog" gets the share 0.20, "sized" gets the share 0.20 and "treat" gets the share 0.20. For an item with description "dog sized dog treat" with price 1.00, the words "sized" and "treat" get 0.25 each but "dog" gets the share 0.50 because it appears twice.

Write the mapreduce program that computes the total share for each word, and the total number of appearances of each word. However words that appear 200 or more times should be excluded from the output. Thus, if the output file has:

```
"cat"       [5.7654, 4]
```

it means that "cat" appeared 4 times with a total share of 5.7654.

- Fill in midterm/wordprice.py
- Fill in the run script
- Fill in the test data files and expected output.
- **The reducer should round total share to 4 decimal digits.**
- Your code is required to use a combiner.
- A word is a sequence of visible characters.

To document your code:

- For each component (mapper, combiner, reducer) add comments that:
  - Explain precisely what it expects as input (what do the keys and value or valuelists mean).
  - Provide examples of what an expected input should look like.
  - Explain what outputs the component should be producing, provide examples of the outputs it should be producing from those example inputs.
  - Do not make the explanations very long.
- Documentation is important if you want partial credit (I cannot read minds, but I can read documentation).