MAPREDUCE LAB, DUE: SEPTEMBER 12, 11:59PM EST

1. Instructions

1.1. **What to look at.** The only references you may use for this lab are:

- Material on canvas

Do not use stackoverflow or chegg for two reasons: (1) it will result in an academic integrity violation and (2) you are likely to get a hilariously incorrect answer from them (don't be an anecdote).

1.2. **Getting and submitting the homework.** Run **updatestarter**, submit to gradescope. Make sure to only commit homework files, but no other files into your repository.

2. The Assignment

This is not a coding homework. **The correct answer for you will be different from the correct answer for your friends.** This homework has two questions that ask you to do similar things. Question 1 uses the files in **mrtrace/q1** and Question 2 uses the files in **mrtrace/q2**. The goal of the homework is to correctly trace the data flow through a hypothetical MapReduce program like we did in the slides (if you have problems, first consult the slides then the instructions in this assignment).

The starting points are:

- **mapper_in.txt** files (e.g., **mapper0_in.txt**, **mapper1_in.txt**). The number of such files is the number of mappers in our hypothetical mapreduce job (**mapper0_in.txt** is the input to Mapper 0, etc).
- **mapper.py** This file has a python function called mapper, whose inputs are **key** (when the input to a mapper are lines of text, the key is **None**) and **value** (which represents the contents of a line).
- **reducer.py** This file has a python function called reducer, whose inputs are the key and valuelist.
- **partition.py** This file has a python function **what_would_partitioner_do** that tells you what the partitioner in our hypothetical program does. Suppose you are wondering which reducer should get the key "omg!!" and that you have 4 reducers. Then you would call this function (knowing how to call a function inside a python file is a prerequisite for this class) like:

```
emailprefix = "abc123" # the stuff before @ in the email you use for gradescope
key = "omg!!"
numreducers = 4 # use the correct number of reducers. Don't just copy/paste 4
                # then ask us why it doesn't work
what_would_partitioner_do(emailprefix, key, numreducers) # returns which reducer
                                                         # gets this key

```

The use of your gradescope email is what makes the answers different for different people.

Fill in the following files (see formatting instructions at the end)

- **mapper_out.txt** files. There is one such file for each mapper. E.g., **mapper0_out.txt** should become the list of key value pairs produced by Mapper 0 **in the order they are produced**, etc. Each line should have a key value pair.
- **reducer_in.txt** files. There is one file for each reducer **so that is how you know how many reducers** there are in the hypothetical mapreduce job. **reducer0_in.txt** should show the input to reducer 0 (each line is a key valuelist pair), etc.
- **reducer_out.txt files.** These files should show the outputs of the reducers.

Formatting instructions for the files you fill in

- We will be using the same format that MRJob uses.

- You can run the check_format.py file to check whether you formatted your files correctly. Knowing how to run a python program from the command line is a prerequisite for this course.
- Each line should have a key, followed by a tab (not spaces) followed by a value or value list.
- The key, value, or value-list should be JSON-encoded. This means:
  - If the thing is a string, surround it with double quotes: `"Yo"`
  - If the thing is a number, just write it: **3** (this means 3 and `"3"` are different. One is a number and the other is a string).
  - If the thing is a list, use square brackets and separate list items with commas. The items inside the list should also be JSON formatted. Here are some examples:
    - **[1, 2, 3]**
    - **[1, "2", ["red", "green"]]**
- So a line with key 3 and value `"4"` should have formatting like `3	"4"` (note keys are not separated from values by commas)

Gradescope can check your files for correctness. The autograder will either crash, tell you that your files are correct, or identify the next mistake.