CMSC 12300: Computer Science with Applications III

The University of Chicago, Spring 2021

Quiz 1

Friday, April 30, 2021 – Saturday, May 1, 2021

This quiz has a **90-minute time limit**. This timer starts as soon as you view any material on the quiz beyond this page of instructions.

By submitting the quiz you are implicitly certifying that you have adhered to the standards of academic honesty. In particular, you are confirming that:

- You wrote your responses in one 90-minute sitting (or more time if by prior special arrangement).
- You did not discuss the quiz content with anyone else.
- You will refrain from communicating with anyone about the quiz content until after the quiz submission deadline, and refrain from any online posts until after it has been graded and returned.
- You did not refer to any materials during the quiz other than your own class notes, the class Ed Discussion site, your own prior assignments, recordings and slides of lectures from the class, and the class web site.

You may choose when you start the quiz, but must complete all work within 90 minutes from starting. You will need to commit your completed work to your git repository. There is no chisubmit step, but please make sure you have performed a `git add` for each of the files, a `git commit`, and a `git push`. This quiz is due Saturday, May 1 at 11:59pm Chicago time. Your submission will be collected from your git repository at that time.

This quiz consists of 3 pages, including this one. There are 5 problems. In total, all problems are worth 50 points.

For each problem, there is a file with a name like `problem1.py` in the `quiz1` directory of your repository. You must edit these files to contain your answers and commit the updated contents to your repository.

The individual tasks are explained on the following pages.

You may run and test your code, and debug it. Please keep in mind the time limit for the overall quiz, however.

You may not ask clarifying questions during the test; read the text of the test, including these instructions, closely and literally, and respond as best you can. It is unfair for certain students to interact with instructors while others don't. If you feel a question is unclear, state your assumptions.

For all problems, strive for the most efficient possible implementation. Use as few steps (e.g. as few reducers) as possible; provide a combiner that does meaningful summarization whenever possible; and avoid using large object attributes that would have the potential to consume large amounts of memory for a large data set whenever possible. Your code will be evaluated both for correctness and for optimality in these regards. That said, code that works, even if less efficiently, is always better than "efficient" code that does not work. Do not time your code to attempt to assess efficiency; these results may be misleading.

Introduction and Dataset Description

For this quiz, you will write multiple MapReduce tasks over the same dataset. Each task is considered a separate problem and will be graded individually. This section describes the dataset that will be used in all problems.

Our dataset consists of weather observations made at airports. The dataset includes daily high and low temperatures observed at all participating airports over some period of time. The format is comma-separated values (CSV).

Each row identifies:

- the airport code where the observation was made
- the date of the observation, in MM/DD/YYYY format
- the high temperature for the day
- the low temperature for the day

Here is a small excerpt that demonstrates the format of the dataset:

```
airport,date,high,low
MDW,04/29/2021,65,55
MDW,04/30/2021,70,50
MDW,05/01/2021,75,40
PIT,04/29/2021,65,60
PIT,04/30/2021,60,55
PIT,05/01/2021,63,60
```

Although this example is very small, assume that there may be an overwhelming number of airports, and time span, covered by the data set.

Problem 1 [*5 points*]

Write a task that determines each airport that has reported at least one observation, with each airport only appearing once in the final result.

Problem 2 [*10 points*]

Write a task to determine the average high in the month of April, 2021 for each airport. To be clear, this task should report per-airport averages, not a single average across the entire geography. If days are missing from the month for a given airport, only average across the days that are present.

Problem 3 [*10 points*]

Write a task to determine the number of unique airports that have reported at least one observation.

Problem 4 [*10 points*]

Write a task to identify the airports for which the high on 04/30/2021 was above (warmer than) the high on 04/29/2021.

Problem 5 [*15 points*]

Write a task to determine a “histogram” of the highs on 04/30/2021, across the entire geography. Temperatures should be grouped into ten degree ranges on the 10s (i.e. 60–70, 70-80, etc., with the lower bound inclusive and the upper bound exclusive), and your code should report the number of airports reporting a high in each of the corresponding ranges as an actual number. (In other words, this is the data you could then plot graphically to make a histogram, but you are not actually generating a plot.)