COSC–254 Data Mining
Homework 02 – MapReduce/Hadoop & Itemsets
Due: Wednesday, February 13, 2019, 1.59pm

Exercise 1 Design a MapReduce algorithm to compute, given a file containing one integer per line, the count of the number of distinct integers. You need to:

- Formally define the functions map and reduce (for each round, if your algorithm takes multiple rounds). Remember to specify the input domains, output domains, and the actual function.
- Analyze the complexity of the algorithm in terms of communication cost and elapsed communication cost, as functions of (potentially a subset of) the following parameters: the input size, number of machines, and number of distinct integers in the input.
- Can you use the reduce functions in a combiner? Prove it formally either way. If you can use them, do the communication costs change and how?

Exercise 2 Exercise 2.3.5 from MMDS, page 41. You can assume that all the numbers will be non-negative. Please call the class `JoinLess`, and call your JAR file `jl.jar`. It must be possible to run your work as

```
$ hadoop jar jl.jar JoinLess PATH_TO_R PATH_TO_S PATH_TO_OUT
```

where `PATH_TO_R` is the path to the HDFS directory containing the first relation R, `PATH_TO_S` is the path to the HDFS directory containing the second relation S, and `PATH_TO_OUT` is the path to the HDFS output directory. You may want to use the org.apache.hadoop.mapred.lib.MultipleInputs class to handle multiple inputs. An example of using it is in this blog post.

The relations are plaintext files containing one row per line. Each row is a pair of non-negative integers separated by a white space, such as

```
2 5
4 23
42 43
```

where the first number corresponds to attribute A for relation R and to attribute C for relation S, and the second number corresponds to attribute B for relation R and to attribute D for relation S. The output should be plaintext with one row per line, with each row composed of a white space followed by four non-negative integers separated by a white space, as in

```
 2 5 42 43
```

(there a leading whitespace before the '2'), where the order of the attributes is A, B, C, D . Example input files `R.txt`, `S.txt`, and expected output `output.txt` are available.

Exercise 3 Exercises 2 and 7 from Sect. 4.9 of DMT, page 132.

How to submit Submit your work at `https://www.cs.amherst.edu/submit` or via `cssubmit` from romulus/remus, as a *single* archive file with name `username.ext` where `username` is your user name and `ext` is one of `.zip`, `.tar.bz2`, or `.tar.gz`.

The archive must contain a *single* directory with name `username`. This directory must contain a subdirectory with name `X` for each Exercise `X`. All files (source code or otherwise) for each exercise must be in the directory for that exercise. Directories containing source code should contain a `README.txt` file explaining how to run the code in that directory. You can find an example archive at `http://bit.ly/DM19sub`.

Please post to the Moodle forum if you have problems with the submission.