Brandeis University
Department of Computer Science
Networked Information Systems - Fall 2016

Homework #2

Hand-out: October 02, 2016
Hand-in: October 11, 2016

1 Data Processing with MapReduce on Hadoop (100 points)

In this programming exercise, you are going to implement the "Facebook common friends" question from the previous exercise. You might find the tutorial materials useful in setting up and programming with Hadoop.

1.1 The Problem

Facebook has a feature which lists common friends with a person when you visit his or her profile page. For the sake of this exercise, we simply assume that common friends will **only be listed for people who are friends** (Let's assume Facebook introduced a new privacy rule :)). We would like to implement this feature using MapReduce. The friendship is a bi-directional relationship, if A is a friend of B then B is a friend of A too. Assume that you are given a file which consists of millions of lines in the following format, where every person in the social network is listed on a line together with his/her friends separated by a whitespace:

```
PersonA PersonB PersonC PersonD ...
...
```

1.2 Logistics

We are providing two input files and an output file for you to better understand the input/output formats and to run your code for testing & debugging. The small input file named **"friendship-20-persons.txt"** represents a small social network of 20 persons which you should use for testing and debugging. A correct implementation's output is given in the **"output-20-persons"** directory inside the logistics file. The input file named **"friendship-500-persons.txt"** represents a social network of 500 persons and your code will be validated for correctness of functionality with this data file. The logistics file **"HW02-logistics.zip"** containing all these mentioned files can be found on LATTE.

1.3 Output & Deliverables

The output of your code on the small input file should match the given sample output. We will run your code on the larger input file and compare its result to a correct implementation's output. Output file should contain on each line two persons' usernames concatenated with comma (,) in **lexicographical** order followed by space (amount of space unimportant) and finally followed by common friends between those two in **lexicographic** order joined by commas. If two persons do not have a common friend, instead of common friends there will be only white space. An example looks like following:

```
A_user,B_user C_user,D_user,E_user
B_user,F_user
...
```

Your source code must be named **CommonFriends.java**. You should return your source code and compiled Java classes as a deliverable tar/gzipped together and named as your last name (Ex: Papaemmanouil.tar.gz). Your code should not take more than **5 minutes** to execute on the larger input file.