

**CS6350**  
**Big data Management Analytics and Management**  
**Fall 2018**  
**Homework 1**  
**Submission Deadline: 1<sup>th</sup> October, 2018**

In this homework, you will be using hadoop/mapreduce to analyze social network data.

**Q1**

**Write a MapReduce program in Hadoop that implements a simple “Mutual/Common friend list of two friends”.** The key idea is that if two people are friend then they have a lot of mutual/common friends. This program will find the common/mutual friend list for them.

For example,

Alice's friends are Bob, Sam, Sara, Nancy  
Bob's friends are Alice, Sam, Clara, Nancy  
Sara's friends are Alice, Sam, Clara, Nancy

As Alice and Bob are friend and so, their mutual friend list is [Sam, Nancy]  
As Sara and Bob are not friend and so, their mutual friend list is empty. **(In this case you may exclude them from your output).**

Input:

[Input files](#)

1. [soc-LiveJournal1Adj.txt](#)

**The input contains the adjacency list and has multiple lines in the following format:**

<User><TAB><Friends>

2. [userdata.txt](#)

The userdata.txt contains dummy data which consist of

column1 : userid  
column2 : firstname  
column3 : lastname  
column4 : address  
column5: city  
column6 :state  
column7 : zipcode  
column8 :country  
column9 :username  
column10 : date of birth.

Here, <User> is a unique integer ID corresponding to a unique user and <Friends> is a comma-separated list of unique IDs corresponding to the friends of the user with the unique ID <User>. Note that the friendships are mutual (i.e., edges are undirected): if A is friend with B then B is also friend with A. The data provided is consistent with that rule as there is an explicit entry for each side of each edge. So when you make the pair, always consider (A, B) or (B, A) for user A and B but not both.

**Output: The output should contain one line per user in the following format:**

<User\_A>, <User\_B><TAB><Mutual/Common Friend List>

where <User\_A> & <User\_B> are unique IDs corresponding to a user A and B (A and B are friend). < Mutual/Common Friend List > is a comma-separated list of unique IDs corresponding to mutual friend list of User A and B.

Please find the output for the following pairs:

(0,1), (20, 28193), (1, 29826), (6222, 19272), (28041, 28056)

Submit the source code and the output via the elearning website.

## **Q2.**

Please answer this question by using dataset from Q1.

Find friend pairs whose number of common friends (number of mutual friend) is within the top-10 in all the pairs. Please output them in decreasing order.

Output Format:

<User\_A>, <User\_B><TAB><Mutual/Common Friend Number>

## **Q3.**

Please use in-memory join to answer this question.

Given any two Users (they are friend) as input, output the list of the names and the states of their mutual friends.

Note: use the userdata.txt to get the extra user information.

Output format:

UserA id, UserB id, list of [states] of their mutual Friends.

Sample Output:

26, 28 [Evangeline: Ohio, Charlotte: California]

## **Q4.**

Using reduce-side join and job chaining:

Step 1: Calculate the minimum age of the direct friends of each user.

Step 2: Sort the users by the calculated minimum age from step 1 in descending order.

Step 3. Output the top 10 users from step 2 with their address and the calculated minimum age.

Sample output.

User A, 1000 Anderson blvd, Dallas, TX, minimum age of direct friends.