Assignment 3 (Social Graph Analysis with MapReduce)

Problem 1, 10 pts) Finding Twitter Asymmetric Following Relations

The "following" relation in twitter is not necessarily a symmetric relation. User A may follow user B while user B may or may not follow user A.

For this assignment you are to write a MapReduce program which takes Twitter's social graph dataset and finds all pairs of tweeter users A ,B such that A follows B but B DOES NOT follow A. You must run your program on EMR.

Input Dataset (Twitter Social Graph):

Tweeter's social graph is large text file that contains over a billion social relations between tweeter users. The dataset is retrieved from this website : (http://an.kaist.ac.kr/traces/WWW2010.html) The size of the original dataset is about 25GB. However, for the purpose of this assignment and to save you some AWS credits, I have created smaller sample (about 4GB) of this data and posted in on my Amazon S3. In your MapReduce Step Configuration, load data from the following path . This will read the data directly from my s3 bucket and save you from having to downloading/uploading it to your own s3 .

s3://class-data-set/twitter_social_small.txt

Each record in the input file has two fields. The first field indicates a user id and the second field indicates who the user_id follows. The fields are separated by tab. Here is an example of a few records of this file

```
12  13
12  14
12  15
16  17
13  12
17  16
```

In this example, user 12 follows users 13 ,14, and 15. User 16 follows user 17. User 13 follows user 12 and user 17 follows user 16.

Output File

Your program should produce an output file (or multiple output files depending on the number of reducers) where each line contains users with asymmetric "following" relation. For example, for the above input, the output will be:

```
12      14
12      15
```

What you need to turn in :

1. **Your Mapper, Reducer, and Driver classes. Please name your classes as AsymmetricMapper, AsymmetricReducer, and AsymmetricDriver.**
2. **A URL to the first output part on your Amazon S3. Don't forget to make the output file public so I can access it. Just right-click on the output part generated on s3 and choose, make public. Then click on its properties and send me the public URL.**

Problem 2-Optional (+5 bonus pts) Finding common Followers in Twitter Dataset

For this assignment, you are to write a MapReduce program which gets a sample of twitter following graph and produces an output which shows all their common followers for each pair of users.
The input dataset can be downloaded from here: https://snap.stanford.edu/data/higgs-social_network.edgelist.gz and is in the following form:

<user_id1> <user_id2)

Where <user_id1> follows <user_id2>. For example,

1	3
1	5
1	4
2	4
3	4
2	3
4	5
5	3
6	4
7	4
6	2
7	5

This means that user_id 1 follows user_ids 3,5, and 4. User_id 2 follows user_id 4, user_id 3 follows user_id 4, and so on.

Your output should be in the following form:

<user_id_1> <user_id2> <A comma separated list of common followers>

For example, for the above sample input, your output must be as follows:

3	4	{1,2}
3	5	{1}
2	4	{6}
4	5	{1,7}

This means that users 3 and 4 have two common followers {users 1, 2} or in other words, users 1 and 2 follow both users 3 and 4. Users 3 and 5 have one common follower (user 1) and so on.

Hints:
You need a chain of two MapReduce jobs to solve this problem.

1- Your first MR job performs a self-join to find a pair of users with a common follower. That means your first MR job, joins the dataset by itself to produce the following output (You need to think about which column you should perform the join on):

<user_id1> <user_id2> <a common follower>

For example, for the above sample dataset, your first MR job should produce the following output (The order of rows does not matter)

3	4	1
3	4	2
3	5	1
2	4	6
4	5	1
4	5	7

2- The second MR job is simply a "**group by**" operation which takes the output of the first job and groups it by the first and second columns to produce the final output:

3	4	{1,2}
3	5	{1}
2	4	{6}
4	5	{1,7}

Note: You need to have only one driver class for your two jobs and chain the jobs using SequenceFileInputFormat and SequenceFileOutputFormat. Please refer to the drive of MatrixMultiplication example in module 4 to understand how to chain two MR jobs using SequenceFile Format.

When you write the output of the first job as SequenceFileOutputFormat, and read the input of the second job as SequenceFileInputFormat, the key and value that you emitted from the reducer of the first job, will be redirected to the key and value of the mapper of the second job. In other words, if your first job emits Text as key and Text as value from the reducer, the Mapper of the second job will receive Text as key and Text as value.

What you need to submit:
Mapper and Reducer for both of your MR jobs as well as your driver program. Please name your driver program as CommonFollowersDriver.java