Assignment 5

Problem 1, (7 pts) Finding Twitter symmetric 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 spark program which takes a sample of Twitter’s social graph dataset and finds all pairs of tweeter users A ,B such that A follows B and B follows A.

Input Dataset (Twitter Social Graph):

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,

```
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 where each line contains users with symmetric “following” relation. Your output should only contain one the symmetric pairs (a,b) and (b,a) but not both. For example, for the above input, the output will be:

```
12        13       (or  13   12)
16        17       (or 16   17)
```

What you need to turn in :

- **Your .scala or .py spark program.**

Problem 2-Experimenting with Graphx package (8 points)

This problem is lab-oriented. I would like to expose you to Graphx package, which is a spark

library for processing large scale graph and supports common graph algorithms such as page rank, triangle count, clustering coefficient, etc.
For this problem, we want to find the **in-degree distribution** of the twitter social graph.

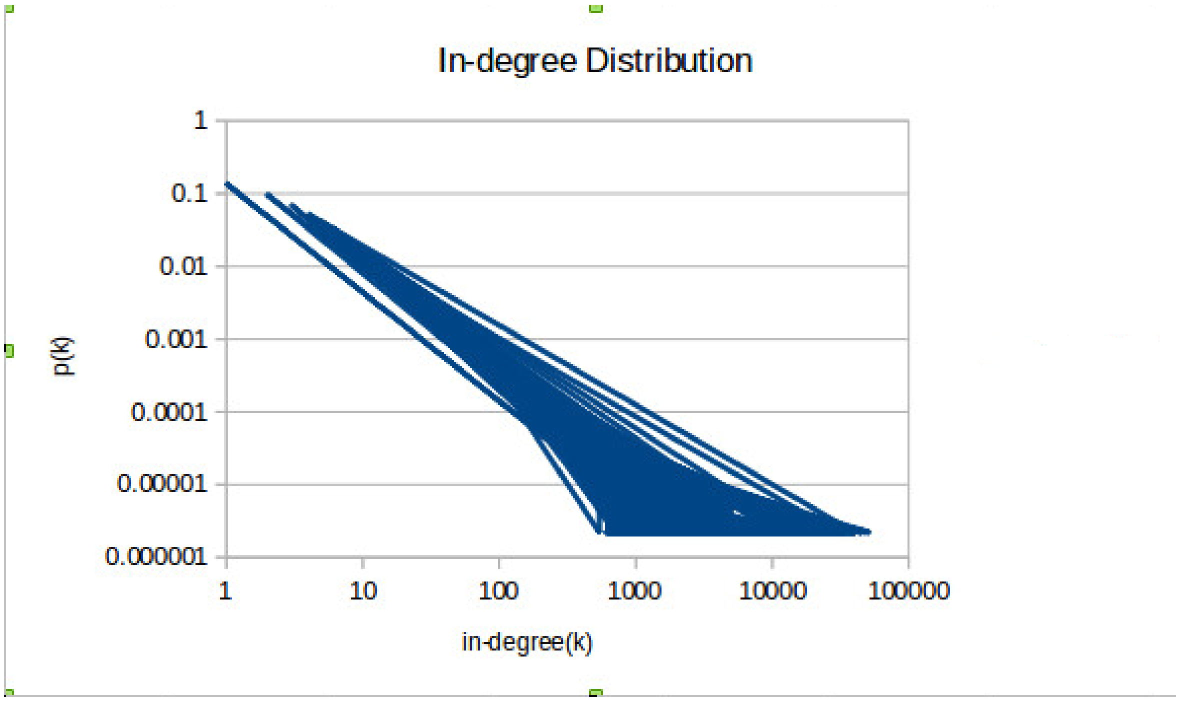
For a quick tutorial on graphx, please refer to this document from MapR (Please review this document before attempting to write a code for this problem)
https://www.mapr.com/blog/how-get-started-using-apache-spark-graphx-scala
For a more complete documentation on graphx package please refer to spark documentation:
http://spark.apache.org/docs/latest/graphx-programming-guide.html#the-property-graph

To create a graph from twitter social data, you need to follow the steps below:

1- Import org.apache.spark.graphx._
2- create a pair rdd in form of: (userid_1, user_id2) and call it "edges"
3- Use graph=Graph.fromEdgeTuples(edges,null) . This will create a graph rdd from "edges"
4- Using this graph rdd, you can call various functions such as vertices, edges, indegrees, outdegrees, pagerank, connected components, etc. (for a complete list, please visit spark graphx documentation).

1- Write a spark program to find in-degree Distribution of the twitter graph
The in-degree of a node (or vertex), is the number of incoming links to that node. We are interested to find the **in-degree distribution** of the twitter graph. The in-degree distribution p(k) of a network is the fraction of nodes in the network with degree k. That is
P(k)= $\frac{n_k}{n}$ where n_k is the number of vertices with in-degree k and n is the total number of nodes (vertices) in the graph.

You should write a spark program which produces a pair RDD of the form (k,p(k)) where k is an indegree and p(k) is the probability distribution for k in the twitter graph. For example an rdd element (1, 0.0.1) means that 1% of the nodes in the twitter graph have indegree 1. Once you produce such rdd, save it in a text file and draw a scatter-plot graph of the distribution. The x-axis in the graph should represent k (indegree) and the y-axis should represent p(k) (fraction of nodes with degree k). **Please use logarithmic scale for both x and y axis**. For example, the graph that I generated on a log-log scale is as follows:



Hints:

For this problem, you might find the following methods of graph rdd useful:

1- "graph.inDegrees" returns a pairRdd in the form: (vertex, indegree). In this case vertex would be a user_id and indegree would be the number of followers for this user_id.

2- "graph.numVertices" returns the total number of nodes in graph as a long integer.

What you need to turn in:

1- Your spark source code
2- The in-degree distribution graph.