CS 6350 - ASSIGNMENT 5

Please read the instructions below before starting the assignment.

- There are 3 parts in this assignment. You do not have to create separate folders, but you should mark clearly the parts.

- You should use a cover sheet, which can be downloaded at: http://www.utdallas.edu/~axn112530/cs6350/CS6350_CoverPage.docx

- You are allowed to work in pairs i.e. a group of two students is allowed. Please write the names of the group members on the cover page. Only one submission per team is required.

- The deadline for this assignment is Friday November 4 at 11:59 PM. No extensions are allowed.

- You have a total of 4 free late days for the entire semester. You can use at most 2 days for any one assignment. After that, there will be a penalty of 10% for each late day. The submission for this assignment will be closed 2 days after the due date.

- Please ask all questions on Piazza, and not through email to the instructor or TA.

ASSIGNMENT 5

In this assignment, you will use Spark GraphX to analyze the high energy physics collaboration network data that is available at:
https://snap.stanford.edu/data/ca-HepTh.html

The data consists of a large list of collaborators, i.e. scientists that have collaborated in the past. You will use this data to construct a GraphX graph and run some queries and algorithms on the graph.

Below are the steps that you will perform. Ideally, you should use Scala or Python under Spark to accomplish all of these tasks:

Step I:

Load the data into RDD using Spark. Define a parser so that you can identify and extract relevant fields.
Note that the dataset contains two-way relationships for each collaborator. That is, if X and Y have collaborated, there will be two lines in the file as:

X Y
Y X

Step II:

Define edge and vertex structure and create property graphs.

Step III:

Run the following queries using the GraphX API:

a. Find the nodes with the highest outdegree and find the count of the number of outgoing edges

b. Find the nodes with the highest indegree and find the count of the number of incoming edges

c. Calculate PageRank for each of the nodes and output the top 5 nodes with the larges PageRank values. You are free to define the threshold parameter.

d. Run the connected components algorithm on it and find the nodeids of the connected components.

e. Run the triangle counts algorithm on each of the vertices and output the top 5 vertices with the largest triangle count. In case of ties, you can randomly select the top 5 vertices.