M222: Analysis of Social Networks
University of Athens
Deprt. of Informatics & Telecommunications

Programming Project I
Today's Date: March 23th, 2020
Due Date: April 27th, 2020

PREAMBLE

In this project, you will design, implement and demonstrate a distributed graph processing algorithm to calculate a measure of tie strength in social networks.

PROBLEM DESCRIPTION:

A recent research effort[1] shows that using the social network of 1.3 million individuals on Facebook, it's possible to predict who e were partners and whether they were going to break up in the next two months. This was of course achieved without using their relationship status, as the researchers employed only the network structure – the friendship relationships between individuals.

The concept behind this research effort is social dispersion. Basically, just looking at the number of friends two individuals have in common–called embeddedness– does not provide enough information and is a low predictor metric. Dispersion, on the other hand, measures how much these common friends are not well connected. In other words, high dispersion between two people means they have friends in common but only a few of those friends are friends with each other. According to the data, couples with long-lasting relationships tend to present high dispersion. Intuitively, the results suggest that strong romantic relationships are those in which people participate in different social groups, which they share with their partners but which remain separate. Looking at one individual and selecting from her social network individuals with whom she has high dispersion generates a list of possible partners for that individual; couples without this particular social structure are more likely to split in the near future.

IMPLEMENTATION ASPECTS:

You will use the `Apache Spark` framework in this project, to implement an iterative Pregel-like algorithm that computes a simple version of dispersion and calculates for each node:

1. the neighbor that exhibits the largest dispersion with the node,
2. the respective dispersion value.

You have to create a Python script that uses the `Graphframes` library of Apache Spark. Algorithm 1 provides pseudocode for the calculation of dispersion. You have to write an iterative version of this algorithm using `graphframes.lib.AggregateMessages`. You can find a simple example (`example.py`) using the library here:
`https://github.com/panagiotisl/spark-graphframes-aggregate-messages-examples`.

Your algorithm should be able to compute dispersion with any input *undirected* graph. However, you must also provide a graph with your submission that can be used for verifying your approach.

[1] Romantic Partnerships and the Dispersion of Social Ties: A Network Analysis of Relationship Status on Facebook. Lars Backstrom, Jon Kleinberg `https://arxiv.org/abs/1310.6753`

Algorithm 1: `dispersion()`

```
begin
    foreach $u \in G$ do
        send list of neighbors to all neighbors;
    foreach $u \in G$ do
        compute the common neighbors with each of $u$'s neighbors;
        for neigbor $v$ of $u$ do
            $dispersion_{u(v)} = 0$;
            foreach pair $(s, t)$ of common neighbors between $u$ and $v$ do
                if $s$ and $t$ are not connected with and edge and do not share neighbors other than $u$ and $v$ then
                    $dispersion_{u(v)} += 1$;
```

Moreover, you are expected to include in your report calculations of the dispersion of a particular node in your graph with its neighbors.

COOPERATION:

You do not have the option of forming a team in this project. However, discussions on most aspects of the project in the classroom's forum are encouraged.

REPORTING:

The final *typed* project report (brief report) must consist of:

1. The adjacency list of a small undirected graph (10-15 nodes) that can be used as input.
2. Dispersion calculations for one of the graph's nodes.
3. Your Python script.
4. Results of your execution.

Finally, you will have to demonstrate your work.