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

Programming Project I
Today's Date: March 16^{th}, 2021
Due Date: April 19^{th}, 2021

PREAMBLE

In this project, you will design, implement and demonstrate a distributed graph processing algorithm to calculate a solution to the graph coloring problem.

PROBLEM DESCRIPTION:

The minimum number of colors required to properly color a graph is called a chromatic number of that graph. Finding the chromatic number of a graph is a well-known NP-Hard problem. Furthermore, it is not possible to approximate the chromatic number of a graph into a considerable bound.

If we relax the chromatic number we can come up with several polynomial time sequential algorithms for the graph coloring problem. Moreover, Maximal Independent Set algorithms can be used to solve the graph coloring problem in parallel.

IMPLEMENTATION ASPECTS:

You will use the `Apache Spark` framework in this project, to implement an iterative Pregel-like algorithm that computes a solution to the graph coloring problem and associates each node with a number representing the color that should be used.

Algorithm 1: `Local Maxima First()`

```
1  begin
2      Set Maxima = true;
3      if superstep==0 then
4          Set Vertex.Value = -1;
5          Send Vertex.Id to all neighbors;
6      else
7          if Vertex.Value==-1 then
8              foreach msg ∈ Messages do
9                  if Vertex.Id < msg then
10                     Set Maxima = false;
11             if Maxima==true then
12                 Set Vertex.Value = superstep;
13             else
14                 Send Vertex.Id to all neighbors;
15     VoteToHalt();
```

You have to create a Python script that uses the `Graphframes` library of Apache Spark. Algorithm **??** provides pseudocode for the calculation of a solution using the Local Maxima First algorithm. 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 a solution to the graph coloring with any input *undirected* graph. However, you must also provide a graph with your submission that can be used for verifying your approach. Moreover, you are expected to include in your report a solution to the graph coloring problem for the graph that you will provide.

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. A solution for the input graph.
3. Your Python script.

Finally, you will have to demonstrate your work.