Name: Krishna Somani **Roll No: 2021058**

Brief overview of working of the assignment:

This assignment demonstrates the implementation of the Affiliation Graph Model (AGM) to detect communities within a dataset. The dataset comprises nodes (GitHub developers) and edges (mutual follower relationships). The primary goal is to use Spark's GraphFrames library to process this data and evaluate the detected communities using modularity as a metric.

The provided code performs the following steps:

1. **Setup Spark Session and GraphFrames**
   The code initializes a Spark session with the GraphFrames library for handling graph operations.

2. **Load and Prepare Data**
   The edges file is loaded and renamed to define the source (src) and destination (dst) columns. Similarly, the vertices file is loaded, keeping essential attributes like ID, name, and target labels.

3. **Create a GraphFrame**
   A graph is constructed using the edges and vertices dataframes. Each vertex is initially assigned a unique community ID matching its own ID.

4. **Community Detection with BigClam**

   - The bigclam_iteration function simulates one iteration of community detection. It uses a mapping of nodes to their communities and updates community assignments by analyzing the edges.
   - The update_community function merges community information between connected nodes.
   - The loop iterates 10 times, gradually refining community assignments.

5. **Modularity Calculation**

   - The compute_modularity function (illustrative in this code) calculates modularity, a measure of the quality of the community structure. It compares intra-community connections against random connections.

6. **Results**
   After all iterations, the final community assignments are displayed. Modularity score achieved is 0.1704983372.