Broadcasting a message within a Social Network

Code Overview

Class name: [Graph.java]

Purpose and description of the class: public interface that represents a Graph with the necessary methods and class variables in order to reproduce the Graph’s behaviour.

Class name: [CapGraph.java]

Purpose and description of the class: This class is the implementation of the Graph Interface. Fort he purpouse of this part of the Project, the method that resolves the Dominant Set problema is: public Set<Node> findDominantSet()

Class name: [Node.java]

Purpose and description of the class: This class represents a Node in the Graph. This class overrides hashCode and equals method in order to find a Node in collections.

Class name: [Edge.java]

Purpose and description of the class: This class represents an Edge in the Graph. An Edge represents a relation between two nodes in the Graph. This class overrides hashCode and equals method in order to find an Edge in collections.

Class name: [GraphLoader.java]

Purpose and description of the class: This class is an utility class that calls the methods in CapGraph in order to load data in the Graph.

Class name: [TestSetCover.java]

Purpose and description of the class: This class if for the purpose of testing the behavior of the CapGraph and GraphLoader class to see if the Graph behavior is as expected. The framework I used for testing is JUnit.

Overall Design Justification

The network has been laid out as a classic graph using an adjacency list. Each individual in the graph is a vertex and an edge between vertices represents a relationship. In order to represent the Graph I use a map, each entry in the map is a vertex(key) and the value is a list with its neighbors which is the main data structure. This has worked fine for this problem.

The main method in order to resolve this part of the Project is findDominantSet(). A method that return a Set of nodes which are a dominant set in the Graph.