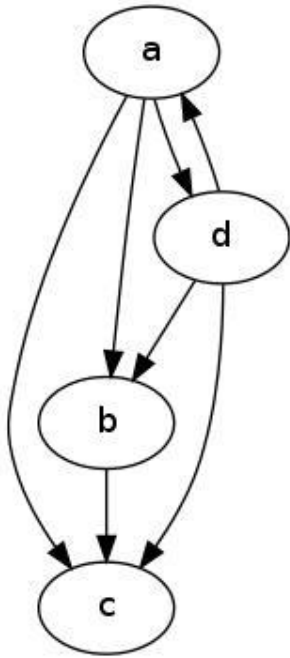


CSCI 3005 – Programming Assignment 3 – April 2022



In a social network, relationships between users can be modeled using a graph. For instance, if a user x “follows” a user y , the x **follows** y relation can be modeled as a directed graph which includes an edge from vertex x to vertex y .

Your task is to use the breadth-first search algorithm studied in class and similar techniques for analyzing the characteristics of a graph in a social networking application. In developing your solution, you are to use the `DiGraph.java` class provided by the instructor. Data for your program will be obtained from text files in which each line contains the names of two users corresponding to a **follows** relationship in the network.

Your solution is to be implemented as a class named `SocApp` containing the following public methods:

`SocApp(String filename)`: a constructor that reads in the data from a text file which contains a series of lines. Each line has the names of two users separated by a single space. The input file `spies.txt` corresponds to a graph such as the one in the figure on the left (with users represented by their initial).

`String mostPopular()`: user followed by the largest number of other users.

`String topFollower()`: user who follows the largest number of other users.

`boolean isInfluencer(String user)`: user is considered an influencer if at least 40% of all users follow u .

`double reciprocity()`: reciprocity is a measure of the symmetry in a graph, computed as the ratio of edges in the graph which are reciprocated over the total number of edges in the graph. The reciprocity of the sample graph above is 0.29 (two of the seven edges are symmetric).

`int distance(String user1, String user2)`: returns the smallest number of directed edges in the path from $user1$ to $user2$. If no such path exists, the method should return `Integer.MAX_VALUE` (the closest we can get to infinity in Java).

`String path(String user1, String user2)`: returns a `String` consisting of vertices in the shortest path from $user1$ to $user2$, using the format: `[user1|vertex] ... [user2]`. The path should be `[NONE]` if no path exists.

`Set<String> reachable(String user)`: a collection of users reachable from $user$.

The `SocAppTest.java` program and a sample text file have been provided for partial testing. Submit your `SocApp.java` (and any other `.java` files developed as part of your solution) to Mimir for testing. Your submission should include neither `DiGraph.java` nor any data files nor any `.class` files).