CS6905 (AGA) Winter 2023 – Assignment 4 (Major)
Due Wednesday February 15, 2023, by 5pm.

The `Voting.java` file (on Desire2Learn) provides the `Voting` class, which is used to store a matrix of ranked results. In this matrix, a column contains the preferences of a voter, while a row contains the preference ranks of an item. So, the value in $Grid[i][j]$ gives the position in voter j's preference list where the item i is located. All rows, columns, and ranks are counting from 0.

Its constructor sets up the $Grid[][]$ matrix, which is populated by repeated calls to the `addRank()` method.

You need to write a `Prefs` class that extends `Voting`, to add the following method:

`prefList()`: takes no parameter, and executes the following process:

1. Construct a directed graph G, where each vertex represents an item being ranked. Directed edges between each pair of vertices are based on majority rule: where at least half of the voters give x a lower rank number than y, there is a directed edge $\langle x, y\rangle$, to indicate that at least half of the voters prefer x over y. (For a tie, there would be edges in both directions.)

2. Find the strongly connected components of this graph, using your `Kosoraju` class that you completed for assignment 3.

3. Build another directed graph G', where each vertex represents a strongly connected component of the original graph G. A directed edge from x to y in G' represents that there is at least one directed edge from the vertices of SCC x to the vertices of SCC y.

   Also build, for each SCC, a list of its vertices in G.

4. Find the topological order of G', using your `DGraphTopo` class that you completed for assignment 2.

5. Output the SCCs in topological order, as follows:

   - one row per SCC, output in topological order of the SCCs
   - each row lists the items that are in that SCC

   So the first row will list the items that are tied for first according to majority preference, the next row will list the items that are tied for the next position, and so on until all the SCCs have been listed.

Your class will also need a constructor to pass on the size parameters to the `Voting` constructor, and should also have private methods and classes as appropriate. Your class should not have any additional variables declared globally for the class.

Note: The `DGraphWtAL.java` code has been updated to provide a `testEdge()` method, which you should use when building the graph of SCCs. Testing an edge is not very efficient in adjacency lists, but that's acceptable for this assignment.

Ensure that your code works with the provided `DriverAGA4.java` code, which will be used to test your submitted solution. Organize and comment your code appropriately.

The driver takes as input the ranked list from each voter, which it then puts into the $Grid[][]$ matrix.

Submit on D2L: your `Prefs.java` file, your `DGraphTopo.java` and `Kosoraju.java` files, and the I/O from one test run of your solution.