CS6905 (AGA) Winter 2023 – Assignment 9 (Major)
Due Thursday April 13, 2023, by 11pm.

The `GraphWtAL.java` file (on Desire2Learn) provides the `GraphWtAL` class, used to build and store an undirected graph with edge weights, as described for Assignment 1. It has been augmented by adding a method `edgeReset(int)` that sets all edge marks to the parameter value, and now with a method `delEdge(int, int)` that deletes an edge from the graph.

In this assignment, you need to implement the algorithm for calculating edge betweenness, as discussed in class, and use it to cluster data.

You need to write a `EBetween` class that extends `GraphWtAL`, adding the following:

- arrays to store the information needed by the calculation:
  - integer arrays for the minimum distance to each vertex and its number of parents
  - a floating point array to store the interim vertex betweenness calculations (for each search)
  - a public array of float, named `btwn`, to store the betweenness scores for each edge
  - Other variables and arrays as needed to manage the graph and betweenness calculations
- a constructor that calls the `GraphWtAL` constructor, allocates the arrays needed by `EBetween`, and initializes the values in `btwn[]` to 0
- `addEdge(int, int)`: adds an edge between the vertices given in the parameters, setting the edge weight to the correct index for the edge
- `betFind()`: calculates the betweenness of each edge. You should break this down into additional methods as needed. Results are to be stored in `btwn[]`.
- `cluster(float)` : as long as there is an edge with betweenness greater than or equal to the parameter value, will remove the edge of maximum betweenness from the graph, adjusting edge array(s) as needed, and recalculate edge betweenness. Once all remaining edges have betweenness less than the parameter, it traverses the graph to find the graph's connected components and marks each vertex with its component number (starting from 1).
- `toString()`: constructs a string of the overall data, composed of the graph (as built by the `toString` method of `GraphWtAL`), followed by a single line that gives the values in `btwn[]`, to 2 decimal places, separated by tabs. This method will be used by the driver to print out the graph and results.

Your class should also have other methods and classes as appropriate.

Ensure that your code works with the provided `DriverAGA9c.java` code, which will be used to test your submitted solution. Organize and comment your code appropriately.

Submit on D2L: your `EBetween.java` file, and the I/O from one test run of your solution that you choose to demonstrate your code. This test data should not be I/O provided by the instructor.

Note: the `DriverAGA9c.java` program does not ask for input in the form of the graph itself; it asks for pairwise similarity data and a similarity threshold from which the graph is then built.