CS6905 (AGA) Winter 2023 – Assignment 5 (Minor)
Due Thursday March 2, 2023, by 5pm.

This assignment is for you to build some of the pieces of the Ford-Fulkerson algorithm, specifically the construction of the residual graph and the modification of your reachability method to find an $s - t$ path.

The updated `DGraphWtAL.java` file (on Desire2Learn) provides the `DGraphWtAL` class, as discussed and used on previous assignments, now updated to allow for public access to the data, methods, and subclass. The `addEdge` method has also been modified to take a fourth parameter, the mark for the edge.

You need to:

1. write a `DGraphReach` class that extends `DGraphWtAL`, to add the following method:

   - `reach(int, int)`: takes a pair of vertex indices (x and y) as its parameters, and returns an integer array that contains, in order, the vertices on a path from x to y in the graph. (This path should start with x and end with y.) If no such path exists, it should return *null*.

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

2. write a `FlowGraph` class that uses both `DGraphWtAL` and `DGraphReach`, to store a flow graph and its residual graph. It should have data as follows:

   - `network` field – of class `DGraphWtAL`. Used to store the flow graph, where the capacity will be stored in an edge's `weight` field and its current flow in the edge's `mark` field.
   - `residual` field – of class `DGraphReach`. Used to store the residual graph. Residual capacities of the edges will be stored in the `weight` field of each edge.
   - `source` – an integer storing the index of the source node of the flow graph.
   - `target` – an integer storing the index of the target node of the flow graph.

   You need to write the constructor for this class, which will take a properly constructed flow graph (of class `DGraphWtAL`) and the indices of the source and target nodes as parameters, store these parameters in the appropriate fields, and build the residual graph.

Ensure that your code works with the provided `DriverAGA5.java` code, which will be used to test your submitted solution. Organize and comment your code appropriately.

Submit on D2L: your `DGraphReach.java` and `FlowGraph.java` files, and the I/O from one test run of your solution. Please submit each file as a separate attachment.