**IMPLEMENTATION OF ADVANCED DATA STRUCTURES AND ALGORITHMS**

**SHORT PROJECT – 2 REPORT**

**Group:** G19

**Team Members:**

Jayakarthigayan Sridharan

SathyaNarayanan Srinivasan

Sagarikha Srinivasan

Sahithireddy Andem

**Question 1:**

Topological ordering of a DAG.

Implement two algorithms for ordering the nodes of a DAG topologically.

Both algorithms should return null if the given graph is not a DAG.

**Compiling and Running Instructions:**

Compile: javac TopologicalOrder.java

Run: java TopologicalOrder 6 71 2 11 3 12 4 13 5 14 3 14 6 15 6 1

Output: g19\_topological\_1.txt and

g19\_topological\_2.txt

The output file is automatically generated.

**Question 2:**

Diameter of a tree.

In this problem, you are given an unrooted tree as input. Since the tree may not be a binary tree, we will represent it with an adjacency list (i.e., it is a graph that happens to be a tree).

**Compiling and Running Instructions:**

Compile: javac Diameter\_Tree.java

Run: java Diameter\_Tree

<no. of vertices> <no. of edges>

(loop through number of edges: ie; press enter after giving the tail and head edge connection)

<vertex\_tail> <vertex head>

Output:

The diameter of the givent tree, if not, it gives -1 as the output. It takes the first tail to be the source/ arbitrary node.

**Question 3:**

Strongly connected components of a directed graph. Implement the algorithm for finding strongly connected components of a directed graph

**Compiling and Running Instructions:**

Compile: javac Strongly\_Connected.java

Run: java Strongly\_Connected

<no. of vertices> <no. of edges>

(loop through number of edges: ie; press enter after giving the tail and head edge connection)

<vertex\_tail> <vertex head>

Output:

The total number of strongly connected components in the graph.

**Question 4:**

Finding an odd-length cycle in a non-bipartite graph. Given a graph, find an odd-length cycle and return it. If the graph is bipartite, return null.

**Compiling and Running Instructions:**

Compile: javac OddCycle.java

Run: java OddCycle 6 71 2 11 3 12 4 13 5 14 3 14 6 15 6 1

Output: odd\_cycle\_g19.txt

The output file is automatically generated.

**Question 5:**

Is a given graph Eulerian?

Write a function that outputs one of the messages that applies to the given graph.

**Compiling and Running Instructions:**

Compile: javac EulerianGraph.java

Run: java EulerianGraph