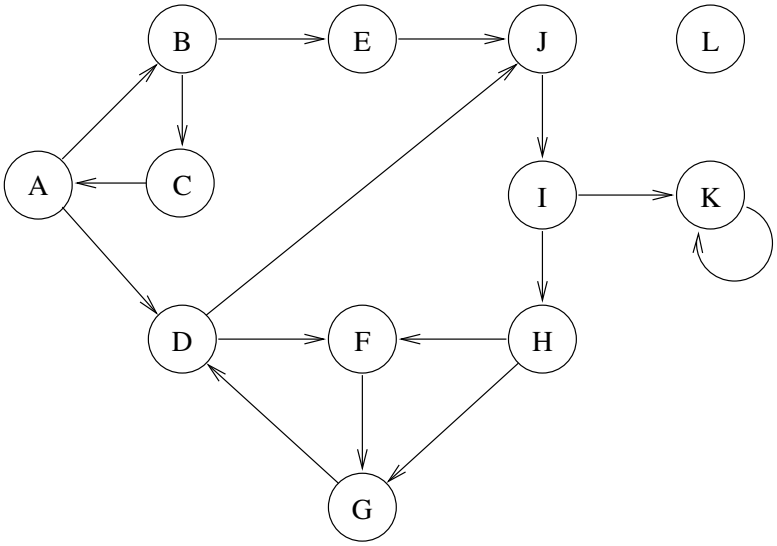
Data Structures 2018
Exercise 11 (Week 47)

- **Notice that based on university's new regulations on degrees, you can have a degree fail from the course which is put to the register.**

  If a student does not participate in the course and does not cancel his/her enrollment, or if he/she discontinues the course, he/she will be assigned a fail grade for the course in question.

- **Students who participate in exercise group must be in place before the exercise group begins (12.15/14.15/16.15). Students who come late do not get the exercise points.**
- **Check the numbers of exercises made before you come to the exercise group. By this means we can save a lot of time when filling the exercise point list.**
- **Notice that pseudocode does not mean the same this as Java code. Pseudocode is not a programming language dependent presentation for an algorithm.**

1. Write the class declarations for all the graph representations described on the course (adjancency matrix, edge list, adjancency list) in Java, including a couple of operations (without implementations). You don't have to write the declarations for utility data structures, e.g. List. The declarations do not have to compile.
2. Represent the following graph with



a) the adjancency matrix representation,

b) the edge list representation, and

c) the adjancency list representation.

Form the names of the edges from the names of the nodes, e.g. the edge from A to B can be named AB.

3. Using the graph of the previous assignment.

   a) Is the graph connected?

   b) Is the graph a tree, a forest or neither ?

   c) Give two different directed paths between nodes C and K.

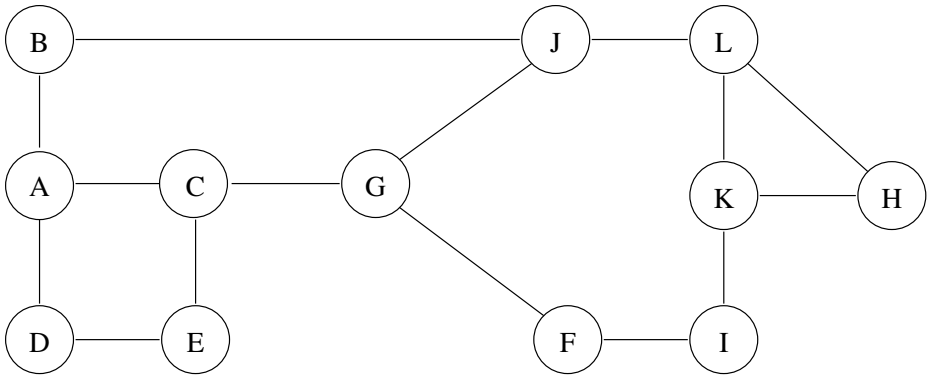
4. Looking at the graph of problem 2.

   a) Form a subgraph of the graph with at least 2 nodes and 2 edges.

   b) Find the connected components of the graph, when we consider the edges undirected.

   c) Find all the directed cycles of graph.

5. Perform for the graph



   a) depth-first search beginning from the node A.

   b) breadth-first search beginning from the node A.

   When you have make a choice between nodes, pick the alternative first in the alphabet.

6. Looking at the graph of problem 5.
   a) Form some subgraph of at least 3 nodes, which is a tree.
   b) Form some subgraph of at least 6 nodes and 3 edges, which is a forest.
   c) Form a subgraph of the graph, which doesn't have cycles, but has as many edges as possible.
7. Form an algorithm based on depth-first search, which finds if there are cycles in a graph. Hint: Mark the nodes visited, unfinished or unvisited during the execution and use this to stop unending recursion. Describe the algorithm in pseudocode.