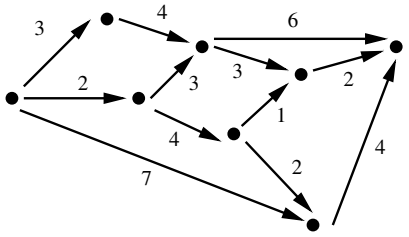
Starred exercises are optional.

1. You have to take over a project of another programmer which comprises the source-files A, $\ldots$, M. These source-files have the following dependencies: A imports G and L; B imports E; C, D, E and I all import L; G, H and J import F; K imports C; L imports H; M imports B, D and I; and F depends on no other source-files. Every source-file that is not imported by others, may be an independent program. To compile a program, the source-files must be processed in an order such that a file is only processed after the files it imports, i.e. in so-called topological order.
   a) Construct and draw the dag having the source-files as nodes and the dependencies as edges.
   b) Find out, how many independent programs the project maximally comprises.
   c) Find a topological sorting of $A, \ldots, M$ respecting the dependencies.

2. Compute the shortest path from the source (node without incoming edges on the left) to the sink (node without outgoing edges on the right) in the following dag, using the shortest path algorithm based on topological sorting



   showing (at least) 2 intermediate stages. Next, adapt the algorithm, and find the longest such path.

3. Let G be the undirected weighted multigraph having vertices $V = \{0, 1, \ldots, 9\}$ and edges $E = \{0, 1, \ldots, 13\}$, with end-points r and weights b given by:

e	$r(e)$	$b(e)$
0	$\{0, 1\}$	3
1	$\{0, 3\}$	1
2	$\{0, 6\}$	5
3	$\{0, 9\}$	1
4	$\{1, 6\}$	1
5	$\{1, 9\}$	2
6	$\{2, 5\}$	1
7	$\{2, 8\}$	3

e	$r(e)$	$b(e)$
8	$\{3, 4\}$	4
9	$\{3, 6\}$	3
10	$\{4, 6\}$	4
11	$\{5, 8\}$	2
12	$\{5, 8\}$	3
13	$\{7, 7\}$	1

   Compute a minimal spanning tree of G using Kruskal's algorithm. Is it unique?

4*) Complete the proof of the lemma on page 9 of the slides of the lecture of week 6, stating the correctness of topological sorting by repeatedly removing minimal nodes, after updating the distances of the nodes reachable from it.