CE 377K: Homework 2 Solutions

Problem 1.



- (d) See attached code
- (e) See attached code

Problem 2.

- (a) By contradiction, assume that a topological order does exist in a network with a cycle. Let the cycle of nodes be $[i_1, i_2, \dots, i_n, i_1]$. (Note that the first and last node is the same.) Since the topological order exists, $i_1 < i_2$, $i_2 < i_3$, and so on, up to $i_n < i_1$. Putting all of these inequalities together we have $i_1 < i_1$ which is a contradiction. Therefore no topological order can exist.
- (b) By contradiction, assume that there is some acyclic network where every node has at least one incoming link. This network contains arbitrarily long paths: start at any node, identify one of its incoming links. Go to that link's tail node, identify one of that node's incoming links, and add it to the start of the

path. Since every node has at least one incoming link this process can be repeated to generate a path as long as one wants. However, since there is only a finite number of nodes (say n), any path which passes through more than n nodes must repeat at least one of them. The segment of the path between this repeated node is a cycle, contradicting the assumption of an acyclic network. Therefore every acyclic network has at least one node with no incoming links.

- (c) Identify any node with no incoming links; number that node as 1. Now, identify any node whose only incoming links (if any) come from node 1; number that node as 2. Continue as follows: after numbering n nodes, identify any node whose only incoming links (if any) come from nodes numbered 1 to n, and number that node n + 1. This labeling is a topological order, since at any stage nodes are numbered so that any incoming nodes have lower node. Furthermore, this procedure can always be followed: it can always be started since there is at least one node with no incoming links. To find a node whose only incoming links come from nodes numbered 1 to n, delete these nodes from the original network along with any links with this node as tail. This process certainly cannot create any cycles, so the resulting network is still acyclic, and has some node with no incoming links. In the original network, this node has no incoming links numbered 1 to n.
- (d) The algorithm will be unable to find a network with no incoming links (when it reaches a node in the cycle) and cannot proceed.
- (e) Yes; see attached code

Problems 3–5. See attached code.