CE 377K: Homework 2
Due Thursday, March 5

Problem 1. A common way of storing a network for use in computer algorithms is an *adjacency matrix*. If the network has n nodes, the adjacency matrix A is an $n \times n$ matrix, where the value in row i and column j is 1 if the link (i, j) is part of the network, and 0 otherwise.

(a) Write down the adjacency matrix of the network in Figure 1.

(b) Draw the network corresponding to the following adjacency matrix:

$$\begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

(c) Write a program (in the language of your choice) which processes the adjacency matrix from part (b) and answers the remaining parts of this question.

(d) How many links does the network have?

(e) How many links enter node 1? How many links leave node 1?

Problem 2. As mentioned in class, acyclic networks are much easier to work with than networks with cycles. This is mainly due to the existence of a *topological order*, that is, in an acyclic network it is always possible to re-number the nodes so that each link connects a lower-numbered node to a higher-numbered node. See the examples in Figure 1. In this problem the objective is to show that a network has a topological order if and only if it is acyclic.

(a) Show that if the network contains a cycle, then no topological order can possibly exist.

(b) Show that every acyclic network must have at least one node with no incoming links. (Hint: there is only a finite number of links.)

(c) Using the result from (b), show how a topological order can be constructed in any acyclic network. (Hint: can you find a way to label the nodes in increasing order?)

(d) What happens if you try to use the method from part (c) on a network with a cycle?

(e) Is the network represented by the adjacency matrix in Problem 1b acyclic? If so, find a topological order.

Problem 3. Use the basic search algorithm to determine whether the network represented by the following adjacency matrix is strongly connected:

$$\begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

.

Problem 4. Identify a minimum spanning tree on the network represented by the following "modified" adjacency matrix, where the entry in row i and column j is the cost of building a (bidirectional) link between nodes i and j, or ∞ if it is impossible to build such a link. (Only the upper part of the matrix is shown because the minimum spanning tree problem ignores direction: building a link from i to j is the same as building a link from j to i.)

$$\begin{bmatrix} \infty & 5 & \infty & 2 & 1 & 3 & \infty \\ & \infty & 6 & \infty & \infty & 5 & 4 \\ & & \infty & 7 & \infty & \infty & \infty \\ & & & \infty & 8 & \infty & \infty \\ & & & & \infty & 7 & \infty \\ & & & & & \infty & 6 \\ & & & & & & \infty \end{bmatrix}$$

Problem 5. Identify the shortest path between nodes 7 and 4 in the network represented by the "modified" adjacency matrix below, where the entry in row i and column j is the cost of traveling on link (i, j), or ∞ if no such link exists.

$$\begin{bmatrix} \infty & \infty & \infty & 5 & 3 & \infty & \infty \\ 4 & \infty & 7 & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & 6 & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & 2 & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty & 4 & \infty \\ 3 & 2 & \infty & \infty & \infty & \infty & \infty \\ \infty & 1 & \infty & \infty & \infty & 4 & \infty \end{bmatrix}$$

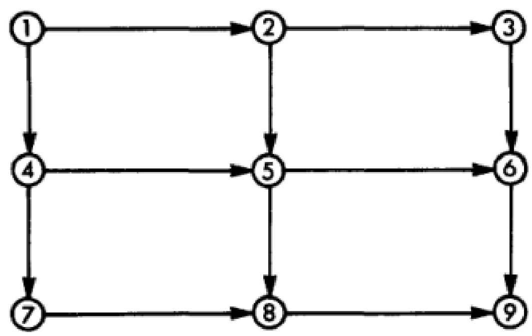

Figure 1: Example of nodes labeled in topological order.