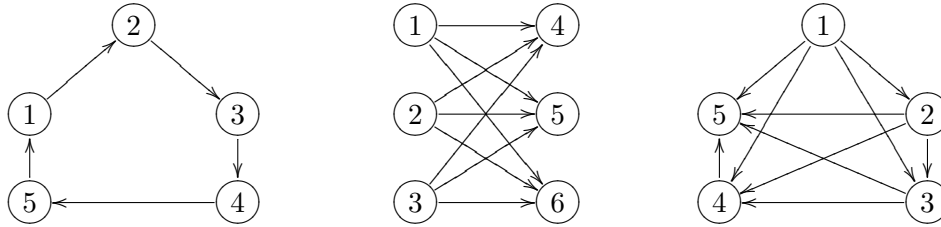


## Network Optimization (Fall 2008) – Brief Solutions to Homework 2

1.



2. Follows directly from the definition of a strongly connected (directed) graph. Since there must be a directed path to each node from every other node, there must be at least one arc incoming to each node.
3. Take a leaf node and put it in set  $N_1$ . Put all the nodes connected to this leaf node by an arc in set  $N_2$ . For each node in set  $N_2$ , put all nodes connected to that node by an arc in  $N_1$ . Continue this process until all nodes in the tree are classified either in  $N_1$  or in  $N_2$ . Since a tree contains no cycles (directed or not), there will be no arcs between two nodes both in  $N_1$  or both in  $N_2$ . Note that the result is true for the case of undirected trees as well.
4. We can store the forward and backward star representations simultaneously using an additional array called `trace`, which stores the number of each arc in the reverse star representation (assuming the arcs are numbered  $1, \dots, m$  in the forward star representation). The incoming arcs at node  $i$  are listed in the rows `trace(rpoint(i))` to `trace(rpoint(i+1)-1)` in the forward star representation matrix.

$$\text{data} = \begin{bmatrix} \# & \mathbf{T} & \mathbf{H} & \mathbf{COST} & \mathbf{UB} \\ 1 & 1 & 2 & 5 & \infty \\ 2 & 1 & 3 & 3 & \infty \\ 3 & 2 & 4 & -2 & 10 \\ 4 & 3 & 2 & -1 & 20 \\ 5 & 3 & 5 & 10 & \infty \\ 6 & 5 & 4 & 2 & \infty \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} 20 \\ -15 \\ 5 \\ -10 \\ 0 \end{bmatrix} \quad \text{point} = \begin{bmatrix} 1 \\ 3 \\ 4 \\ 6 \\ 6 \\ 7 \end{bmatrix} \quad \text{trace} = \begin{bmatrix} 1 \\ 4 \\ 2 \\ 3 \\ 6 \\ 5 \end{bmatrix} \quad \text{rpoint} = \begin{bmatrix} 1 \\ 1 \\ 3 \\ 4 \\ 6 \\ 7 \end{bmatrix}$$

5.  $G^T$  will be the “reverse” graph of  $G$ , with each arc  $(i, j)$  in  $G$  replaced by the corresponding reverse arc  $(j, i)$  in  $G^T$ .
6. Let  $\mathcal{H}$  be lower triangular. In this case, all  $(i, j) \in A$  will satisfy  $i > j$ . Assume  $i_1-i_2-\dots-i_r-i_i$  is a directed cycle in  $G$ . Then we should have  $i_1 > i_2 > \dots > i_r > i_1$ , a contradiction. Hence  $G$  must be acyclic.

For the “only if” part, we need to describe a method which reorders (or rennumbers) the nodes in the desired fashion, assuming  $G$  is acyclic to start with. We can use the idea of topological ordering, and one exists for every acyclic graph (but we have not discussed this topic yet). Alternatively, we can use induction on the number of nodes to prove the result. First, notice that there must be at least one node in  $G$  with zero indegree (if every node has positive indegree, then  $G$  must contain a cycle). Now, assume the result holds for all acyclic graphs with  $n$  nodes. Consider an acyclic graph with  $n + 1$  nodes. We can identify a node with zero indegree. The subgraph obtained by removing this node and all its outgoing arcs will be an acyclic graph with  $n$  nodes. By the induction assumption, we can reorder the nodes in the subgraph such that its  $\mathcal{H}$  is lower triangular. We then add the original node back (along with the arcs outgoing from it), and assign to it the number  $n + 1$ . We will be adding a row and a column to  $\mathcal{H}$ , preserving the lower triangular structure.