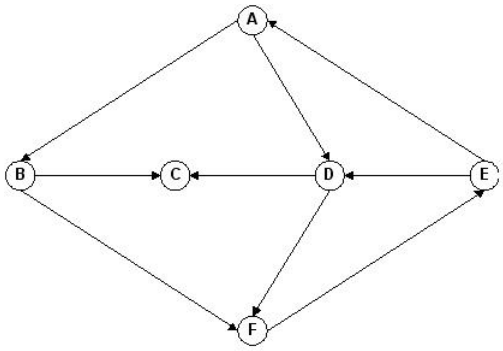
Head-Banging Session 2
Fall 2008 CS 473: Algorithms

Problem 1. **[Decorating the DFS Tree]**



Draw the DFS tree rooted at D for the above graph. Use alphabetic ordering to break ties. Label the vertices of the tree with their $pre(v) : post(v)$ time. Add in the remaining edges of the graph and label them as forward (F), backward (B), and cross (C) edges.

Problem 2. **[Can't Find a Bridge]**
Given a connected undirected graph $G = (V, E)$, an edge $e = (u, v)$ is called a bridge, or a cut-edge, if removing e disconnects the graph into two pieces, one containing u and the other containing v. A graph where the degree at each vertex is k is called a k-regular graph.
Prove or disprove the following: Every 3-regular graph has no cut-edge.

Problem 3. **[Largest Label]**
You are given directed graph $G = (V, E)$ where each of the nodes are labeled with an integer. For node u, $k(u)$ is the integer label. Give a linear time algorithm to compute for each node u the largest label of a node v that can reach u.

1. Solve problem assuming G is a DAG.
2. Solve the problem by extending to general graphs via the strong component meta-graph.

Problem 4. **[Properties of DAGs]**
Recall that given a directed graph G, its reverse G^{rev} is the directed graph where each edge (u, v) is replaced by its reverse (v, u). Recall also that the equivalence classes of $R = \{(u, v) \mid u \text{ is connected to } v\}$ are the strongly connected components of a directed graph G. Prove the following statements about DAGs:

1. G is a DAG if and only if G^{rev} is a DAG.
2. Every DAG G has at least one source and at least one sink.
3. G is a DAG if and only if each node is in its own strong component.
4. For any graph G, the graph of SCCs of G^{rev} is the same as the reversal of G^{SCC}