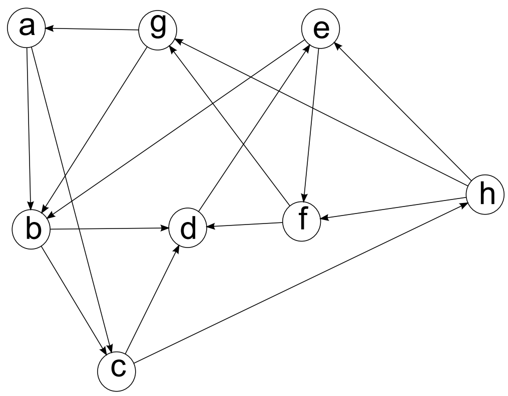
CS 473: Fundamental Algorithms, Fall 2014

Discussion 1

September 2nd 2014

1.1. DFS EXAMPLE.

Consider the following graph.



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 $\text{pre}(v) : \text{post}(v)$ time. Add in the remaining edges of the graph and label them as forward (F), backward (B), and cross (C) edges. Sort the vertices by their post visit order.

1.2. POST NUMBERING OF SCCS HAS TO BE DONE CAREFULLY.

Let G be a directed graph and consider a specific depth first search of G. Let S and S' be strong connected components in G. Suppose (S, S') is an edge in G^{SCC} we saw in lecture that $\text{post}(S) > \text{post}(S')$ (recall that $\text{post}(S)$ was defined as $\max_{u \in S} \text{post}(u)$). However it is not true that for every $u \in S$ and $u' \in S'$, $\text{post}(u) > \text{post}(u')$. Describe an example to show that this claim is not true.

1.3. PROVE/DISPROVE: SMALLEST POST NUMBER IMPLIES A VERTEX IS IN THE SINK.

Let G be a directed graph and G^{SCC} its strong connected component meta-graph (which is a DAG). Prove or disprove the following. For any **DFS** of G the vertex with smallest post-visit number is in a sink component of G^{SCC}.

1.4. NO NO NO! I WANT TWO CYCLES, NOT ONE.

Let $G = (V, E)$ be an undirected graph with n vertices ($|V| = n$) and m edges ($|E| = m$). Give an $O(n)$ time algorithm to check if G has at least *two* distinct cycles and output them if it does. Assume that the graph is represented using adjacency lists. Note that m can be much larger than n so the algorithm should not check all edges. Use **DFS**.