Homework 2

Due Friday, July 27

1. The following directed graph is given in adjacency list format

Node	Incoming	Outgoing
1	{}	{2,7}
2	{1,6}	{3,7}
3	{2,8}	{4,7}
4	{3,5}	{8}
5	{9}	{4,9}
6	{}	{2,7}
7	{1,2,3,6}	{8}
8	{4,7,9}	{3}
9	{5}	{5,8}

   (a) Perform a DFS on G starting at node 1.

   (b) Label each edge of G as a Tree edge, forward edge, back edge, and cross edge according to this DFS tree.

   (c) List the strongly connected components of the graph.

   (d) Draw the meta-graph that results when each strongly connected component is collapsed to a node.

   (e) Give a topological ordering on this meta-graph.

2. An **Euler Tour** of a connected, directed graph $G = (V, E)$ is a cycle that traverses each edge of G exactly once, although it may visit a vertex more than once.
   Show that G has an Euler Tour if and only if in-degree(v)=out-degree(v) for each vertex $v \in V$.
   (in-degree(v) = number of incoming edges of v, out-degree(v) = number of outgoing edges of v)

3. Prove or give a counter example for the following statement. Every directed graph is a DAG of its strongly connected components.

4. Let $G = (V, E)$ be a directed graph. Give an algorithm that determines whether G contains a vertex v s.t. every other vertex is reachable from v, and outputs one such vertex if it exists. Your algorithm should run in $O(|V| + |E|)$ time. (Hint: Use a meta-graph)
   You may use any algorithm covered in class in your algorithm. So, a step of your algorithm could be run BFS. For this problem, just describe how your algorithm works, no proof of correctness is needed.

5. Exercise 9, on p.110 of Kleinberg and Tardos