CS311:	**Homework 5, 2008**	Due in class, 11:15 am, Thursday, Oct. 16, 2008

Note that my treatment in class of DFS was almost exactly the same as the text's. The main difference is that what they call pre[v], I call $d[v]$, i.e., discover time, and what they call post[v], I call $f[v]$, i.e., finish time. Please make sure that you understand the diagram on page 89 which summarizes the relations of the four kinds of edges to the discover and finish times. Compare that table to the following table which can be used in DFSvisit(u) to calculate in constant time on looking at the edge (u, v) whether it is a tree, back, cross, or forward edge.

	edge type of (u, v)	color$[v]$	$f[v]$ vs. $d[u]$	nesting	relationship
1.	tree edge	white		$[u\ [v\quad v]\ u]$	v is a child of u
2.	back edge	gray		$[v\ [u\quad u]\ v]$	v is an ancester of u
3.	forward edge	black	$f[v] > d[u]$	$[u\ [v\quad v]\ u]$	v is a descendent of u
4.	cross edge	black	$f[v] < d[u]$	$[v\quad v]\ [u\quad u]$	none of the above

1. [25 pts.] Do problem 3.3, page 95 of the text: sample use of DFS to compute topological sort.
2. [25 pts.] Question 3.5, page 96: give a linear-time algorithm whose input is a graph, G, given via its adjacency lists, and whose output are the adjacency lists of G^R, the reversed graph.
3. [25 pts.] Question 3.14, p. 98: the algorithm is already sketched on the bottom of page 90 and top of page 91. Your job is to refine it with an appropriate simple data structure and carefully analyze it to make sure that it is linear time.
4. [25 pts.] Question 3.16, p. 98: minimum number of semesters to meet prerequisite structure assuming you can take and pass as many courses at once as needed. [Hint: use question 3.]