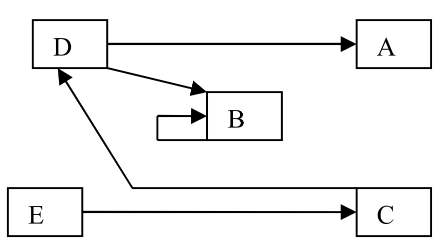
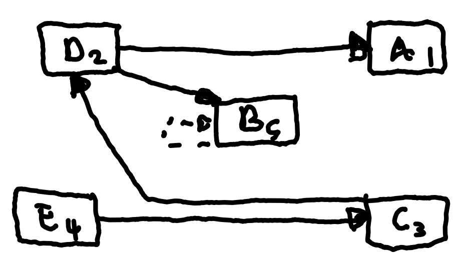
Homework #17

Name ______________________ Sec ____

Questions:	Answers:
1. Consider graphs with n nodes and m edges which are stored as adjacency lists. Think of various algorithms to solve the following graph problems. For each, give pseudo code and the worst-case big-Oh running time for the best algorithm you considered. a) For a given a node x, find all nodes in a directed graph that are a distance of two from node x. b) Find all edges with weight 2 in an undirected, weighted graph.	
2. For the graph G below, create the depth first search forest that starts with node A. Whenever there is a choice of which node to visit next, choose the node that comes alphabetically first. Use solid arrows for tree edges; add all other edges as dashed edges. Label all edges as tree, forward, backward, or cross edges. Add postorder numbers as subscripts to each node name. 	1. AD 2. ADB 3. ADBC 4. ADBCE 

3. The appearance of a backward edge in a depth-first-search forest of a directed graph tells us that the graph is cyclic. Graph G in Problem #1 is cyclic. Identify the backward edge in G by giving its tail node x and head node y ($x \rightarrow y$). How do we recognize backward edges by testing postorder numbers? Explain, for the backward edge in G, how the edge $x \rightarrow y$ satisfies this test.	
4. A topological sort of a strict partial ordering R is a total ordering such that x precedes y if xRy. a) Do a DFS on graph G (from #2), choosing nodes alphabetically first, to generate a postorder of the vertices, then reverse the postorder. Your answer should be the reversed postorder. Note that the reverse of the postorder is ***not*** always the same as the preorder. b) Give the total ordering produced if we do a DFS choosing nodes alphabetically last. c) Let R = {(a, b), (a, c), (b, d), (b, e), (c, e), (d, f)} and let the strict partial ordering be R+. List all topological sorts of R+. (You need not use any particular algorithm)	

5. a) Give all depth-first-search forests for G in Problem #2 starting with node E. b) Expressed in terms of reachability from node E, why do all of the “forests” only have one tree?	
6. Let F be a depth-first-search forest for an undirected graph G. (Recall that when we create a depth-first-search forest for an undirected graph, we replace each edge with two directed edges, one in each direction, and then run the ordinary algorithm to create a depth-first-search forest.) Prove: Two nodes, x and y, are in the same tree of F if and only if x and y are in the same connected component of G.	

7. Create the adjacency matrix A for the graph below. Compute A^2, A^3, and A^4 and then the union of A, A^2, A^3, and A^4, yielding the reachability matrix for A.