1. (5) When an adjacency-matrix representation is used, most graph algorithms require time $\Omega(V^2)$, but there are some exceptions. Show that determining whether a directed graph $G$ contains a universal sink vertex with in-degree $|V| - 1$ and out-degree 0 can be determined in time $O(V)$, given an adjacency matrix for $G$.

2. (3) We covered two routines for graph traversal - DFS($G$) and BFS($G,s$) - where $G$ is a graph and $s$ is any node in $G$. These two procedures will create a DFS tree and a BFS tree respectively. Show that if $G = (V,E)$ is a connected, undirected graph then the height of DFS($G$) tree is always larger than or equal to the height of any of the BFS trees created by BFS($G,s$).

3. (9) For the following problem, use the directed unweighted graph given by the following adjacency list. Be sure to consider the edges in the given order.

   A: C E B  
   B: E D  
   C: E  
   D: C F E  
   E: F  
   F:  

   (a) For the source vertex $s = A$ what is the order in which the vertices are visited by BFS (breadth first search)? Also, show the breadth-first search tree that you obtain.
   (b) What is the order in which the vertices are visited by DFS (depth first search)? You should assume that the top-level DFS procedure visits the vertices in alphabetical order. Set up a global counter which gets incremented every-time when the vertex if first explored or is finished being explored. For each vertex give the discovery and finishing time.
   (c) Suppose that this graph is a precedence graph. Using your work above either give a valid order in which to perform the tasks (call them task A, task B, . . ., task F) or prove that there is no valid order.

4. (5pt) Chapter 3.2 (discuss the solution using one the graph traversal algorithms BFS or DFS). Assume that the graph is connected. Starting from arbitrary node $s$, obtain BFS tree $T$. If every edge go G that appears in the tree, then $G = T$, so $G$ contains no cycles. Otherwise, there is some edge $e = (u,w)$ that belongs to $G$ but not $T$ connecting the nodes within the same level or to some of the ancestors of $u$. If such edge exists there is a cycle. The cycle can be printed by tracing all the ancestors of that node along the path in the BFS tree. To do that you may need to associate an additional field node.parent with each node. This can be done in $O(m + n)$ time.