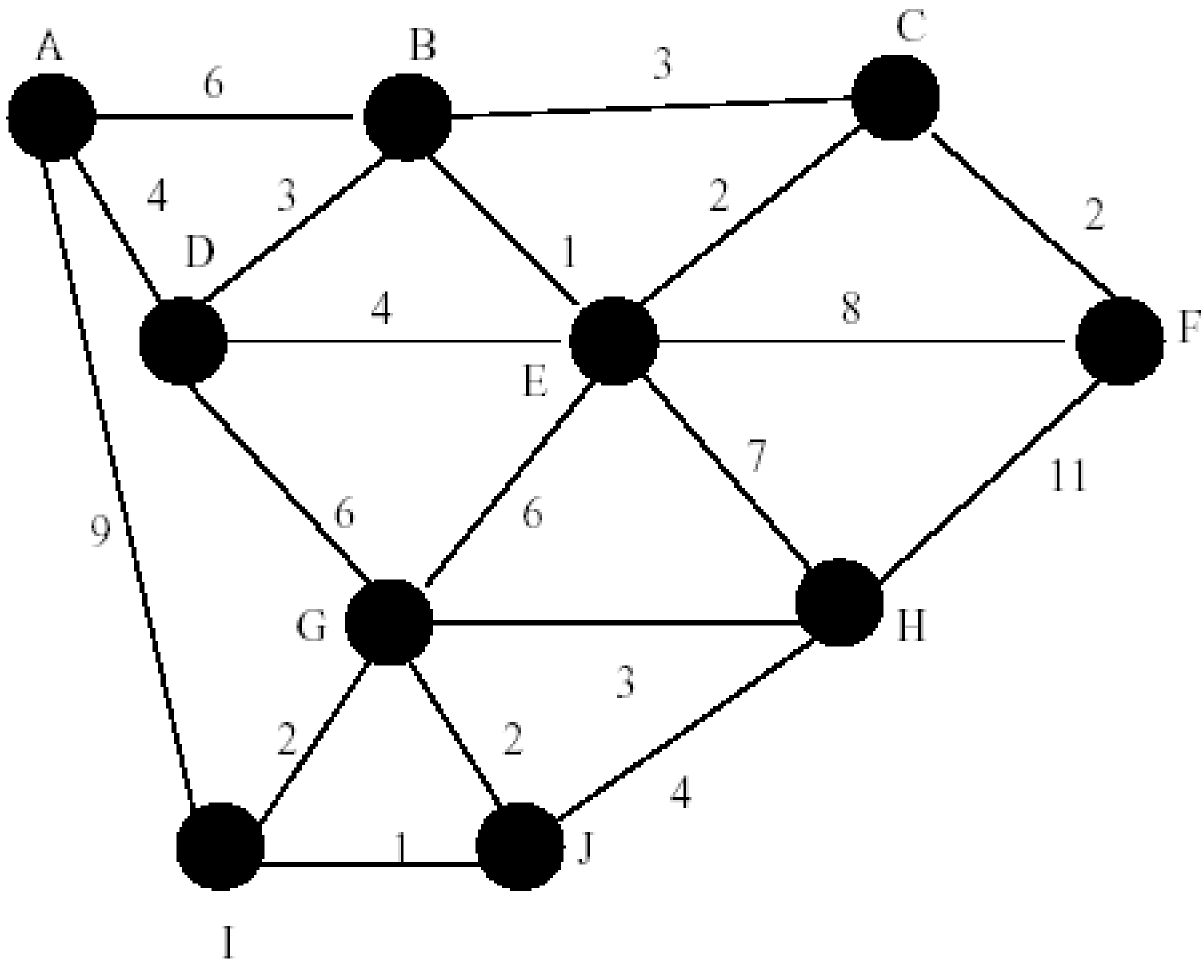
<u>CS 332 Homework Assignment 7</u>

Spring, 2005

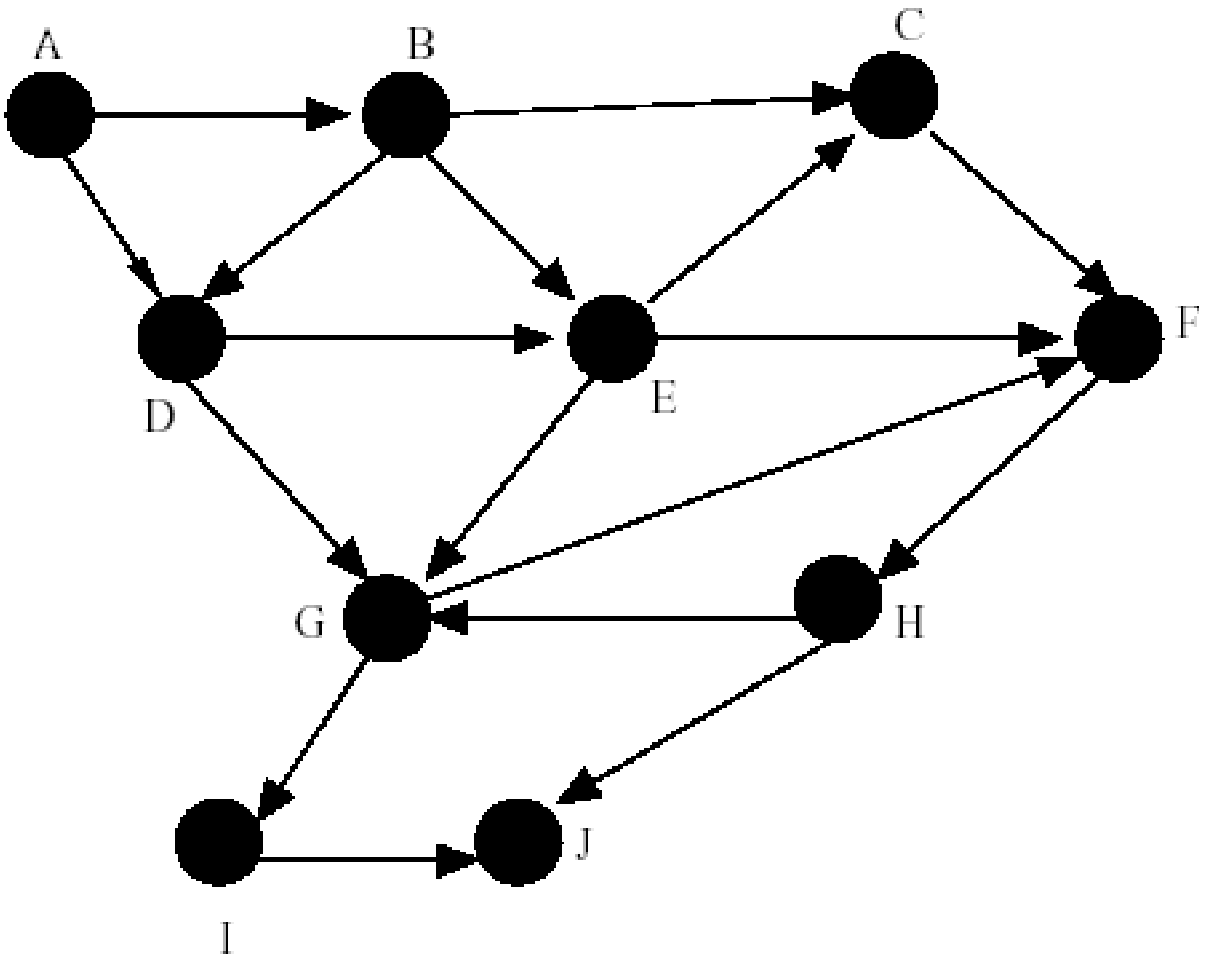
This assignment is not a programming assignment. You can either write up your solutions to these problems using pen and paper or you can produce an electronic document using programs like MS Word, LaTeX, or Maple (for example, this document was created using Maple). Whatever you turn in should be neat, readable, and well organized. This assignment is due on Friday, May 6.

Problem 1. Consider the following graph *G*.



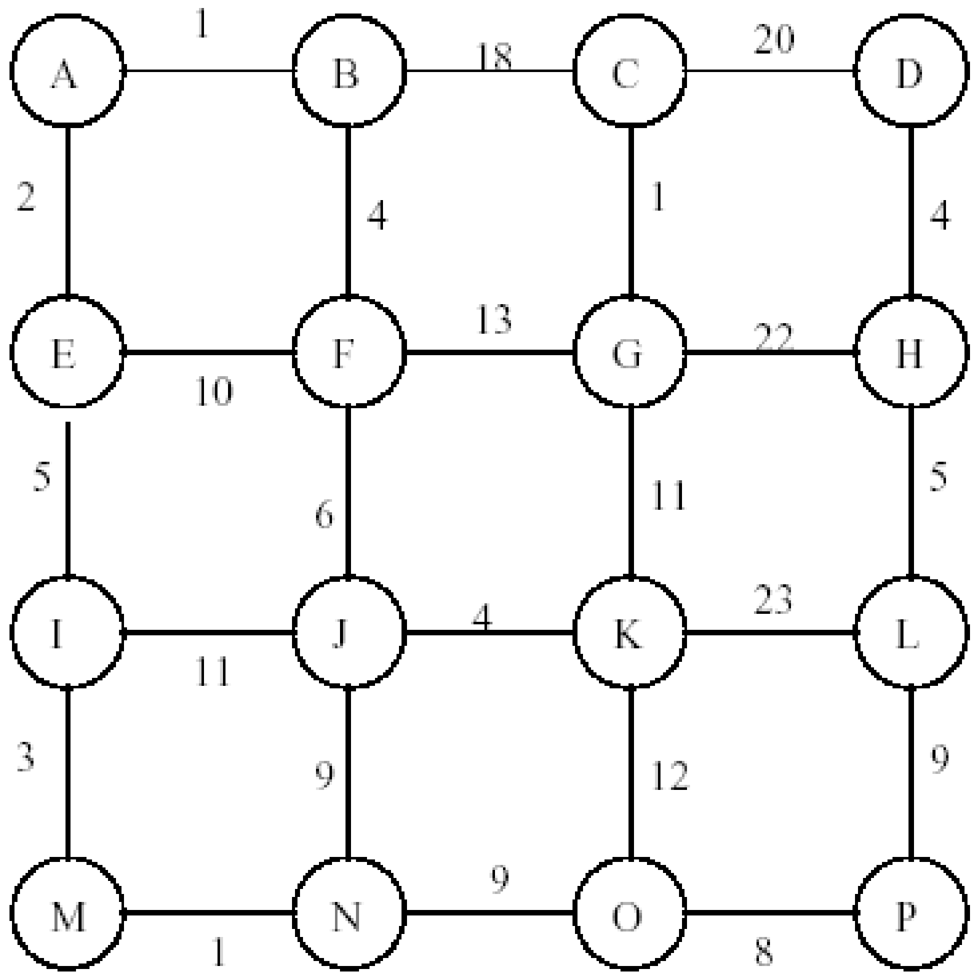
- Part (a) Report the order of the vertices encountered on a breadth first search of *G* starting from vertex *A*. Iterate through adjacent vertices using alphabetical order. Draw the BFS spanning tree.
- Part (b) Report the order of the vertices encountered on a depth first search of *G* starting from vertex *A* . Iterate through adjacent vertices using alphabetical order. Draw the DFS spanning tree.
- Part (c) Find the minimum spanning tree of *G*. Compare the weight of the minimum spanning tree with the weights of the BFS and DFS spanning trees from Parts (a) and (b).

Problem 2. Consider the following graph *G* (notice that this graph is directed).



- Part (a) Report the order of the vertices encountered on a (directed) BFS of *G* starting from vertex *A*. Iterate through adjacent (in the directed sense) vertices using alphabetical order. Draw the BFS spanning tree.
- Part (b) Report the order of the vertices encountered on a (directed) DFS of *G* starting from vertex *A*. Iterate through adjacent (in the directed sense) vertices using alphabetical order. Draw the DFS spanning tree.
- Part (c) The vertices of graph *G* have a certain hierarchical order to them (besides alphabetical order) that has something to do with the directed nature of the graph. Study the graph carefully and find this hierarchical order and explain it.

Problem 3. Consider the following graph *G*.

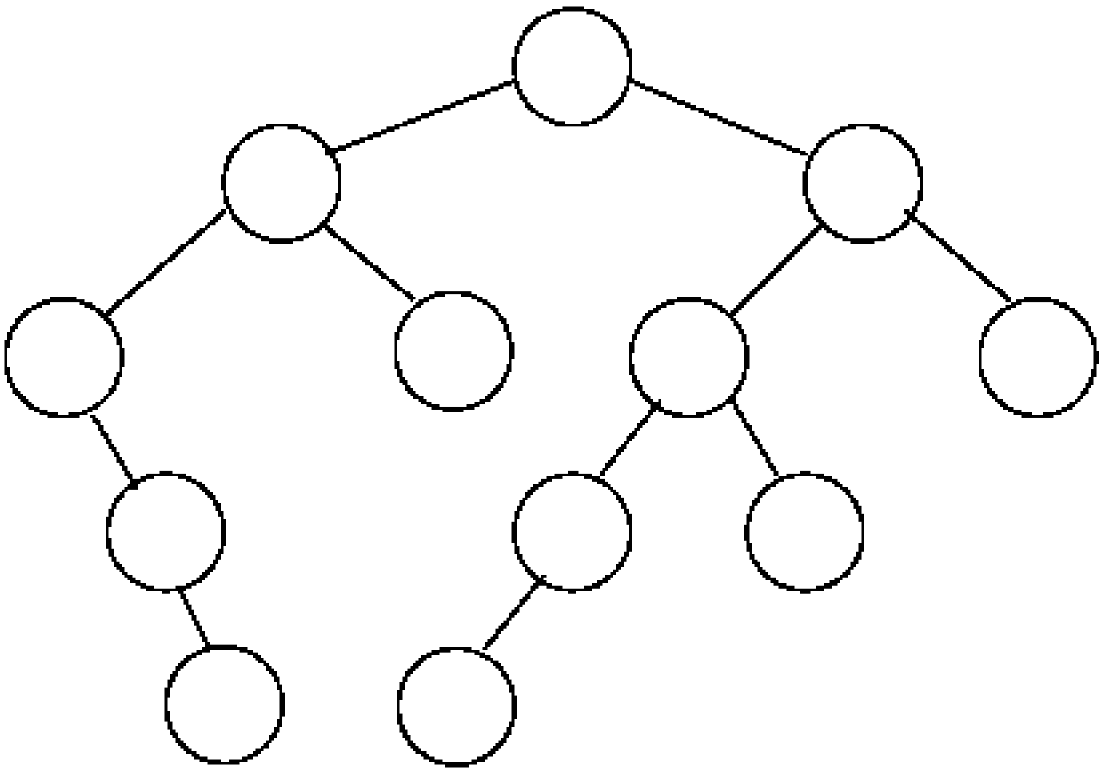


- Part (a) Find the minimum spanning tree of *G* using Prim's algorithm starting from vertex *A*. Show how the algorithm grew the minimum spanning tree by labeling the edges in the minimum spanning tree with the order in which they were discovered.
- Part (b) Find the shortest path tree of *G* using Dijkstra's algorithm starting from vertex *A*. Show how the algorithm grew the shortest path tree by labeling the edges in the shortest path tree with the order in which they were discovered

Problem 4. Given the pre-order and post-order traversals of a binary tree, is it possible to reconstruct the tree? If so, sketch an algorithm to do it. If not, give a counterexample.

Problem 5. Insert keys into the binary tree structure given so that a postorder traversal will give:

8	3	5	11	14	4	2	1	9	7	6	0



Problem 6. Suppose you have a tree with the adjacency list equal to:

0: 1 3 4 7
1: 0
2: 6
3: 0
4: 0 5 6
5: 4
6: 2 4
7: 0

Draw a graph of this tree. Show how this tree could be represented in an array of size 8. Explain what you are doing.

0	1	2	3	4	5	6	7