Grade: 53/60 = 89%

Graded by Lawrence Jerome, 8/2/2004.

A good job on this assignment, except for you confusion over breadth-first algorithm. I've made my comments in blue, and points subtracted in red.

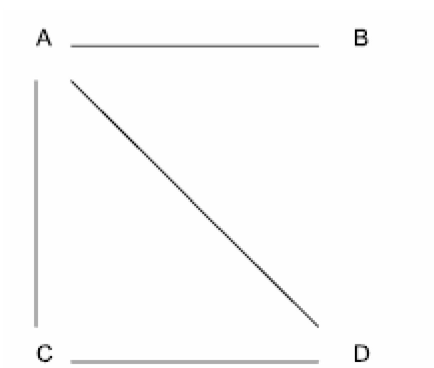
Name:	Janine Bouyssounouse
Student ID:	4782181
E-mail:	jcbouyss@yahoo.com
Course Number:	CS201
Course Code:	02A
Assignment Number:	2
Version of Word:	2002

This document includes symbols and pictures created with MathType, Paint, and Geometer's Sketchpad. If any symbols are unreadable, let me know to send a printed out version that is readable.

3.1 What is a graph?

A graph has vertices and edges. It has to have at least two vertices and at least one edge with its endpoints defined by the vertices. [Good definition.]

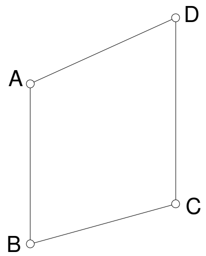
3.2 List the set of vertices and the set of edges for the graph below.

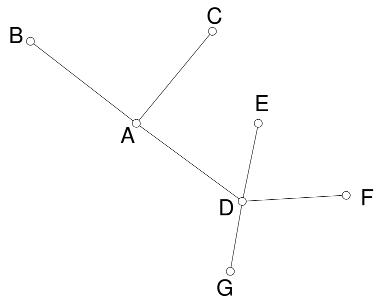


The vertices are: A, B, C, and D. The edges are: {A, B}, {A,D}, {A, C}, {C, D}. [Correct.]

3.3 What is a connected graph? Give two examples.

A connected graph is a multigraph with all vertices connected with at least one edge. [Correct. Good examples.]

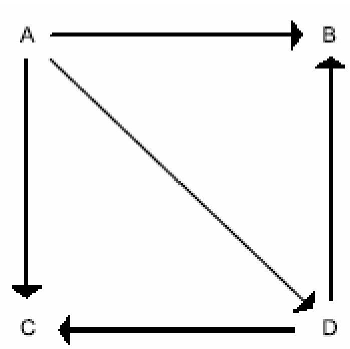




3.4 What is a weighted graph? What are its applications?

A weighted graph is a graph with numbers assigned to the edges of the graph. The number is the weight of that edge. These types of graphs can be used to represent freeway maps with the mileage as the weight of the edges to show the distance between cities, or they could be used for a map of airline mileages to fly between different destinations. It is then possible for a person to calculate the total distance between location A and C, going through B by adding the weight of the edges {A, B} and {B, C}. [Correct. Good explanation.]

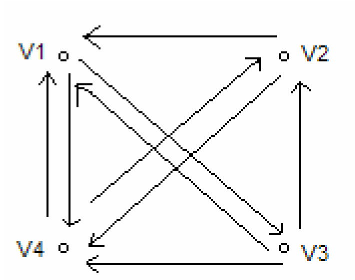
3.5 List the vertices and directed edges for the directed graph below:



The vertices are: A, B, C, and D. The directed edges are: (A, B), (A, D), (A, C), (D, C), and (D, B) [Correct.]

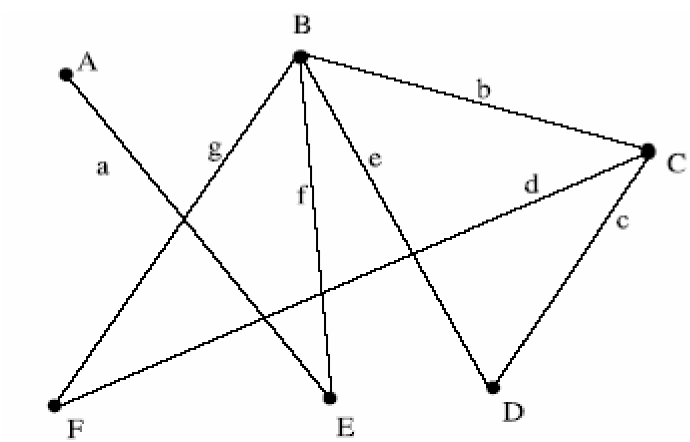
3.6 Construct the labeled directed graph for the adjacency matrix:

$$\begin{bmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix}$$



[Correct.]

3.7 Find an Euler path for the following graph:

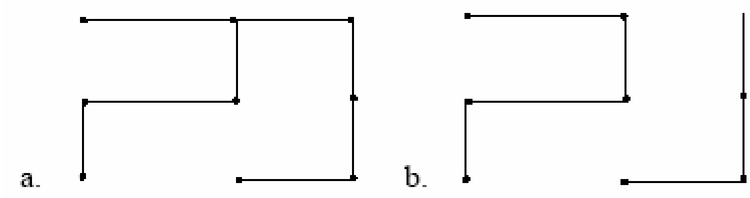


a, f, g, d, b, e, c [Correct. Good path.]

4.1 What is a tree? Is a PERT diagram always a tree? Explain.

A tree is a graph that is connected and has no cycles in it. PERT diagrams are trees because all of the events or vertices are connected and there are no repeat loops. Each event is signified by the completion of the even in its vertex. [Your definition of tree is correct. However, a PERT diagram cannot be a tree because a vertex can have two parents. Here's the formal proof: Theorem 4.1 states: "*U* and *V* are vertices in a tree. Then there is exactly one simple path from *U* to *V*." So there must be one simple path between any two vertices in a tree. Therefore, a PERT diagram is not always a tree. Though a PERT diagram can be drawn as a directed graph (e.g. see figures 1.1 and 1.2)—and consequently has no cycles—it still does not comply with Theorem 4.1. As the PERT diagram from figure 1.3 on page 5 of the text clearly shows, there are three simple paths from C to J (C-D-F-I-J, C-E-F-I-J, C-G-H-J); thus, a contradiction. Furthermore, Theorem 4.3 shows that "a tree with *n* vertices has exactly *n* – 1 edges." Figure 1.3 has 11 vertices and 12 edges. This also contradicts the theorem, proving that not all PERT diagrams are trees. Partial deduction = 1 point.]

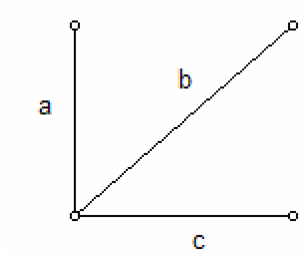
4.2 For each of the following graphs, determine if each is a tree and explain your answer.



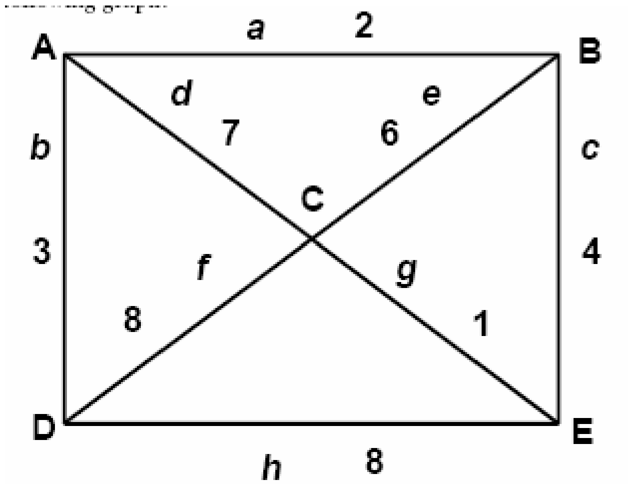
Graph a is a tree because it is connected and there are no circuits. Graph b is not a tree because it is not connected. [Correct. Both parts.]

4.3 What is a spanning tree? Give an example.

A spanning tree is a graph with all of the vertices connected and no vertices. There will be one less edge than there are vertices. [Correct. Good example.]



4.4 Explain the breadth-first search spanning tree algorithm, and then apply it to the following graph:



The breadth-first search spanning tree algorithm looks at the minimal paths to connect the tree. This would be the minimal spanning tree. If you always take the path with the smallest weight, then you are assured of getting the minimal weight of the tree or the minimal spanning tree. For this example the edges b, a, c, and g connect all of the vertices with the least weight possible. In this scenario, the highest weighted edges h, f, e, and d are completely avoided. [You must be confusing breadth-first with Prim's algorithm. Breadth first starts with one vertex, then adds all the edges leading from that vertex (called the first level). You move from level to level, always adding all the edges from each vertex. Deduction = 4 points.]

4.5 Use Prim's algorithm to find the minimal spanning tree for the graph in the previous problem.

I accidentally already did this for 4.4. The minimal spanning tree for this figure is b, a, c, and g. [Correct.]

4.6 How do you know if a graph is a binary tree?

In a binary tree, each vertex of a rooted tree has either zero, one or two children. No vertex has more than two children. [Correct.]

4.7 Explain the preorder traversal algorithm.

In the preorder traversal algorithm, first start at the root of the tree, then follow the left path, once the left path has been visited, then follow the right path. This process is followed at each vertex with children (subtree), so that the left and right paths are examined for each subtree. [Correct.]

4.8 Explain the binary search tree search algorithm.

In the binary search tree search algorithm, a variable is picked to be set to the root, then a while loop is entered. The loop continues until either the value being searched for is found or the end of the tree and all the subtrees is reached. If the value being searched for is found, the original arbitrary variable is set to the value being searched for. This allows the correct end message to be produced that states that the value was found if the searched for value is equal to the arbitrary variable, otherwise, the searched for value was not found. [Good explanation.]