Math 272 Discrete Mathematics
Lesson 6
Worksheet

What is a spanning tree?

A spanning tree for a connected simple graph G is a sub-graph of G that is a tree and contains all the vertices of G. Here is an example from the textbook.

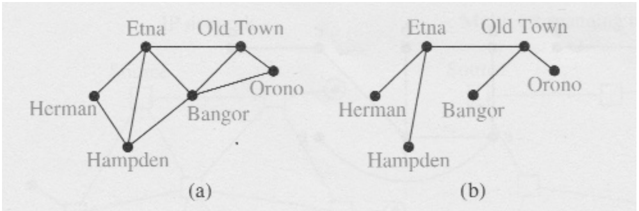


FIGURE 1 (a) A Road System and (b) a Set of Roads to Plow.

How do we find a spanning tree for a graph?

1 Removing edges from circuits

If G is already a tree, we are done: G is its own spanning tree.

If G is not a tree, it must have a circuit. Remove an edge from the circuit. (Keep the vertices.)

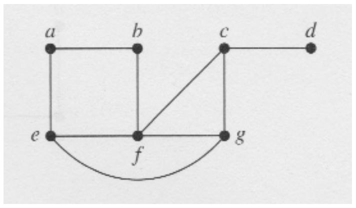
If the result is a tree, we are done.

If the result is not a tree, it must have a circuit, remove an edge.

Repeat until there are no more circuits.

Since G is finite, we must end up with a tree containing all the vertices.

Question 1. Use the above procedure on the following graph from the textbook to find a spanning tree.



How many edges are removed? If G has n vertices and e edges total, you need to remove $e-(n-1)$ edges to be sure the resulting tree has the required $n-1$ edges. Check that this is what happened when you created the spanning tree above.

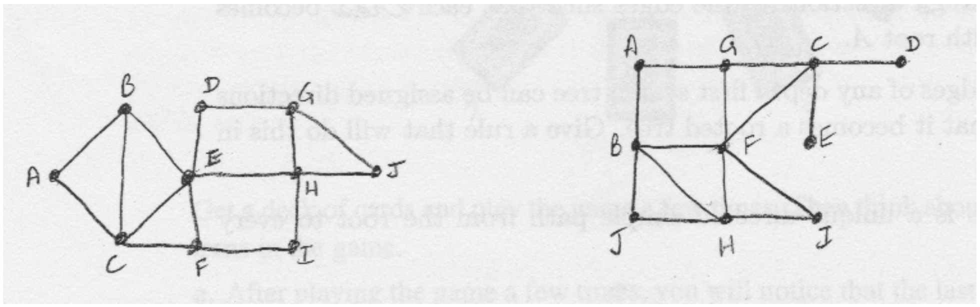
Breadth-first search for a spanning tree for a connected simple graph

Recall that the breadth-first search found the shortest path and distance from one vertex to all other vertices in a connected graph. Write down the simple sequence of steps for carrying out that algorithm, using the idea of circling and labeling vertices.

Question 2.

a. If we keep going until there are no vertices left in the graph, this procedure will create a spanning tree for the graph. Explain why this is so.

b. Apply this algorithm to the two graphs below, starting at vertex A. Break ties alphabetically.



Depth first search for a spanning tree for a connected simple graph

Procedure:

i. Choose a starting vertex and circle it.

ii. From the vertex where you are now, choose any adjacent un-circled vertex, if one exists, and circle it. Mark the edge. Move to the other end of that edge. (Use alphabetical order if there is more than one choice.)

iii. Repeat until you can go no further.

iv. From the vertex where you are now, back up along circled vertices until you come to the first vertex that is adjacent to an un-circled vertex, if there is one. From the circled vertex where you are after backing up, move to the un-circled vertex, circle it, and mark the edge.

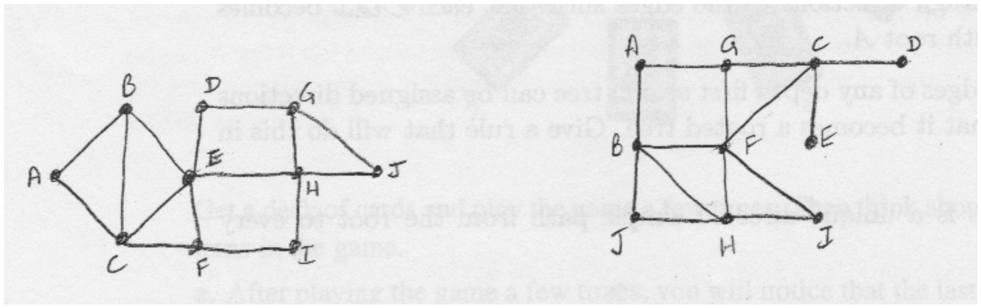
v. Return to step ii and repeat it until you can go no further.

vi. Repeat steps iv and v until you run out of vertices (which will be when you have reached all n vertices).

Question 3

a. If there are no vertices left in the graph, the marked edges form a spanning tree. Why?

b. Apply the depth first search algorithm to the same two graphs as above. Are the spanning trees the same or different from the ones you found using the Breadth-First Algorithm?



Shortest spanning tree in a weighted graph

MIU has six computers they want to link together without any junction box or server. They want to use the least amount of wire. Some pairs of computers cannot be linked to each other because of where they are in different offices or buildings. The matrix below shows which computers can be directly linked and how much wire is needed in yards.

	A	B	C	D	E	F
A	–	9	–	–	–	3
B	9	–	8	–	8	11
C	–	8	–	3	5	–
D	–	–	3	–	6	11
E	–	8	5	6	–	9
F	3	11	–	11	9	–

Question 4

a. What does the "5" in the C row mean? Why is the AB entry the same as the BA entry? Why isn't there a DD entry? Why isn't there a DB entry? Why should the matrix be symmetric?

b. Construct a graph containing the information in the matrix.

c. Find a shortest network in the graph. What is the total length of your shortest network? Mark the edges of this shortest network. What is the total length of your shortest network?

d. Explain why this shortest network must be a tree, and why it must be a spanning tree.

e. Describe the method you used to find the shortest network step by step.

Kruskal's Algorithm for finding a minimal spanning tree in a weighted graph

This algorithm is called a "best edge" algorithm.

(i) Mark the shortest edge in the graph. If there is more than one, choose alphabetically.

(ii) Mark the shortest unmarked edge that does not create a circuit with previously marked edges. Break ties alphabetically. The edge you mark does not have to be connected to any previously marked edges.

(iii) Repeat step (ii) until it is no longer possible to add an edge without creating a circuit.

(iv) If there are no vertices left that are not adjacent to any marked edges, the marked edges form a minimal spanning tree.

(iv) If there are still vertices left that are not adjacent to any marked edges, the graph is disconnected, and there is no spanning tree, although there may be a spanning forest.

Question 5.

Apply this algorithm to the computer network in the previous section, starting with vertex A. Is this spanning tree the same as the spanning tree you found before? If different, are their total lengths the same?

Prim's algorithm for finding a spanning tree in a simple connected weighted graph

Prim's algorithm follows a pattern similar to Dikstra's algorithm for finding a shortest path in a weighted graph.

(i) Choose a starting vertex and circle it.

(ii) Find all edges that have one vertex circled and one vertex not circled. Mark the shortest such edge that does not create a circuit. Break ties alphabetically.

(iii) Repeat step (ii) until you cannot proceed any further.

(iv) If there are no vertices left that are not adjacent to any darkened edges, the darkened edges form a minimal spanning tree.

If there are still vertices left that are not adjacent to any darkened edges, the graph is disconnected, and there is no spanning tree.

Question 6. Apply Prim's algorithm to the computer network above, starting with vertex A. Do you get the same minimal spanning tree as with Kruskal's algorithm and with your original algorithm?