Unit 7 Day 7

Section 5.4
Trees And Their Properties
& Section 5.5
Spanning Trees

WARM UP Day 7!

a. Construct a tree diagram showing all possible circuits that begin at *S,* visit each vertex of the graph exactly once, and end at *S.*

b. Find the total weight of each route.

c. Identify the shortest circuit.

d. Use the nearest-neighbor algorithm to find the shortest circuit.

e. Does the nearest-neighbor algorithm produce the optimal solution?

WARM UP Day 7 ANSWERS!

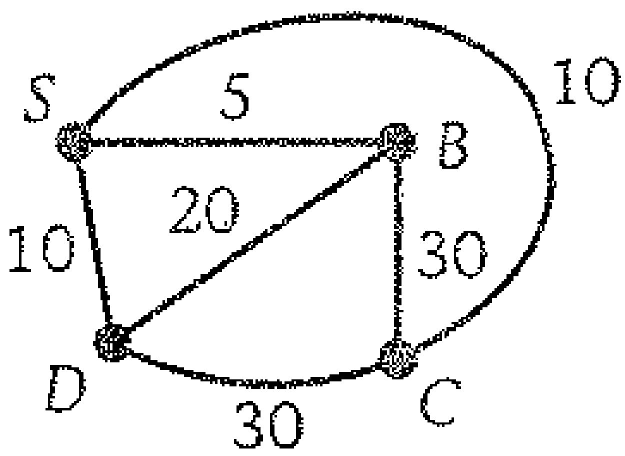
a. Construct a tree diagram showing all possible circuits that begin at *S*, visit each vertex of the graph exactly once, and end at *S.*

b. Find the total weight of each route.

c. Identify the shortest circuit.

d. Use the nearest-neighbor algorithm to find the shortest circuit.

e. Does the nearest-neighbor algorithm produce the optimal solution?



b. SBCDS = 75 SCDBS = 65

SBDCS = 65 SDBCS = 70

SCBDS = 70 SDCBS = 75

c. SBDCS or SCDBS = 65

d. SBDCS = 65

e. Yes, the nearest-neighbor algorithm produces the optimal solution.

Homework Answers

p. 226 # 5-7

5. a. 0.36288 sec., 87178.2912 sec. or 24 hours

   b. 0.00000036288 sec., 0.08718 sec.

6. Answers will vary.
   Examples: A mailman's delivery route, automated mail delivery in an office, bank courier

7. SACBS, 18.75 millimeters

Homework Answers

p. 232 # 5-8

5. a. Albany, CEH, Ladue 2 + 2 + 1 + 4 = 9

   b. Albany, BD, Fenton, GK, Ladue = 15 . This is a different solution than part a because you must travel through Fenton.

6. Answers will vary. Students should explain the best method to finding the shortest path without having to use Dijkstra's Algorithm.

7. ABCEDF = 20

8. Shortest path: DCAB least charge 1 +2 + 7 = 10

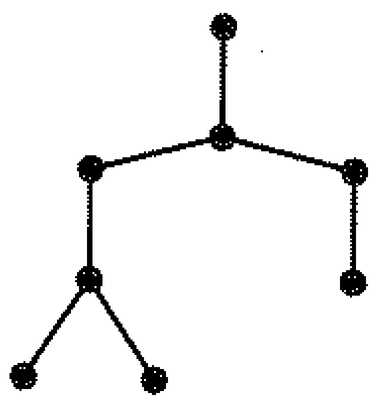
Note Section 5.4 Trees And Their Properties

What is a tree?

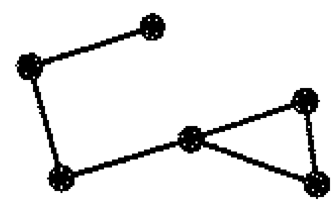
- ***Recall:*** **A <u>cycle</u> (circuit) in a graph is any path that begins and ends at the same vertex and no other vertex is repeated.**

- **<u>TREE</u>- connected graph with NO cycles.**

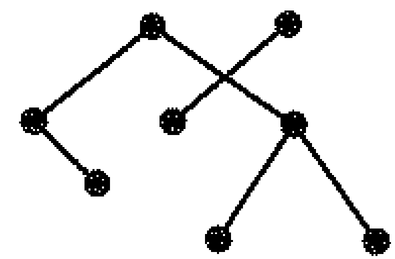
Which are trees?



a.

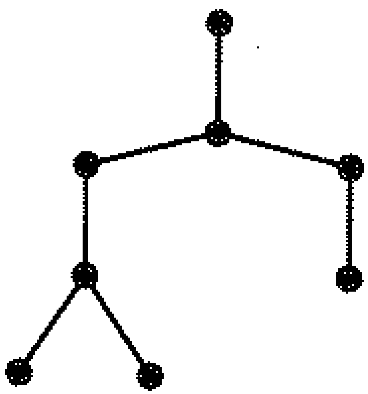


b.



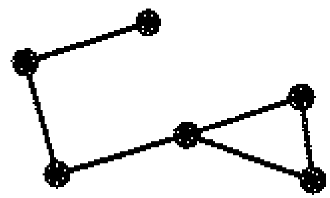
c.

Which are trees? ANSWERS



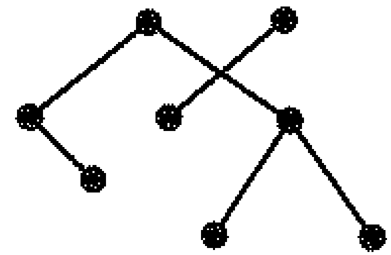
a.

Yes!



b.

No! It contains a cycle.

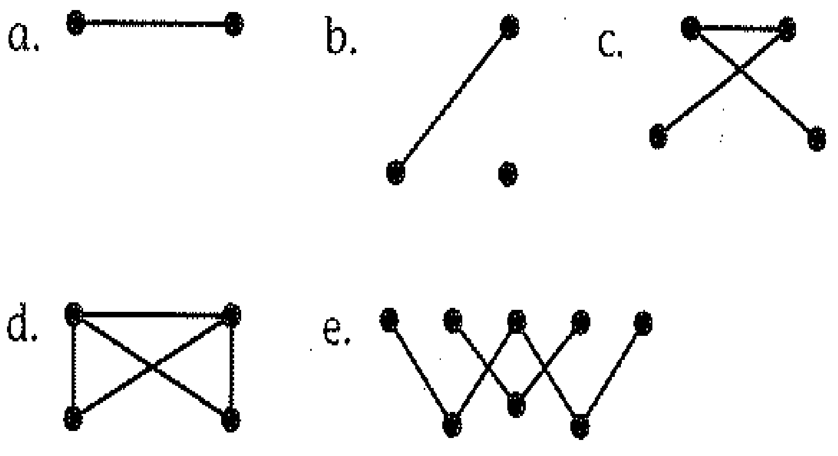


c.

No! It is not connected.

Look at Homework #2

Determine whether the following graphs are trees.



a. ______________________________

b. ______________________________

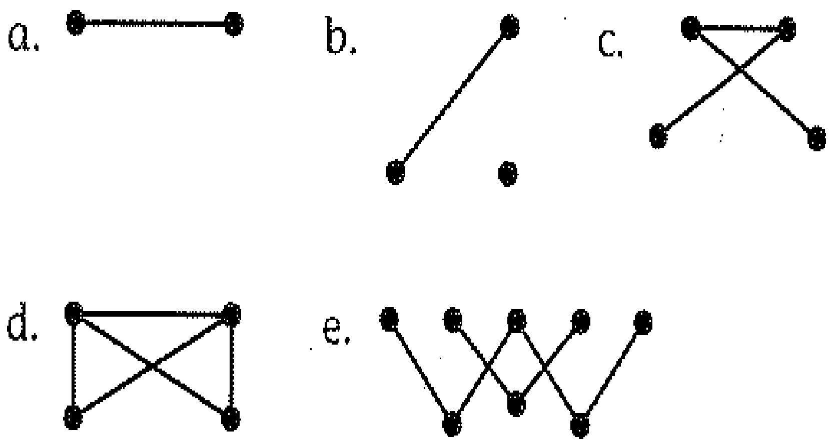
c. ______________________________

d. ______________________________

e. ______________________________

Look at Homework #2 -> ANSWERS

Determine whether the following graphs are trees.



a. Yes

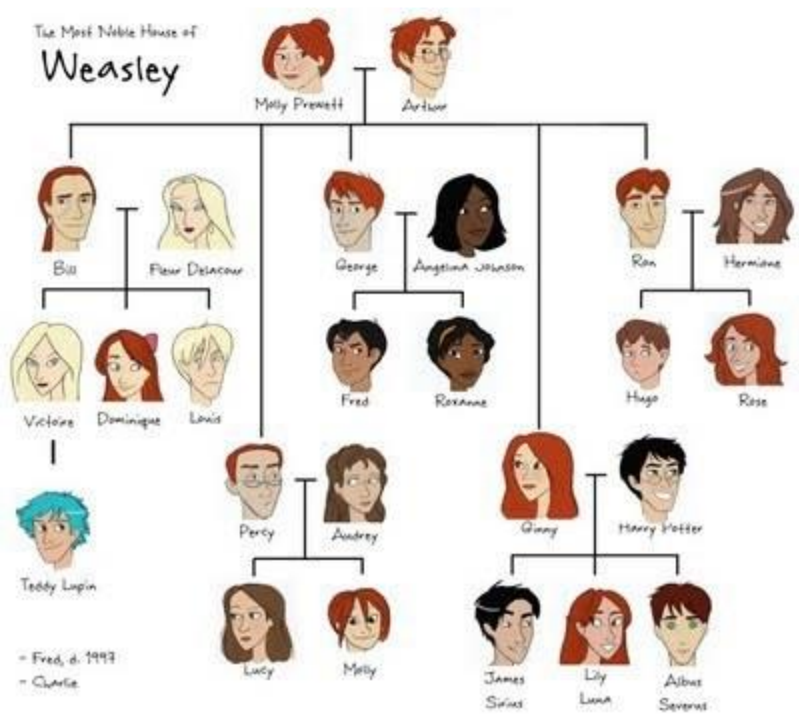
b. No, not connected

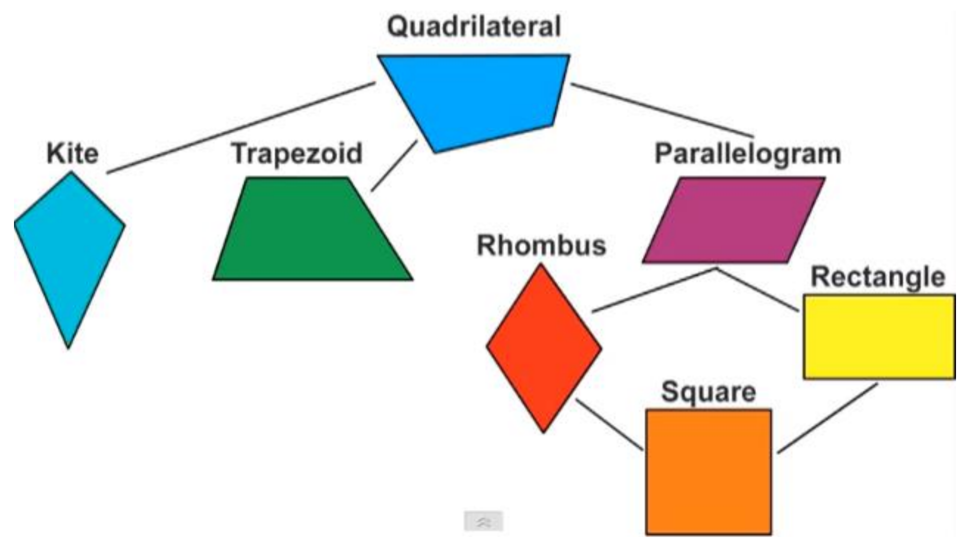
c. Yes

d. No, contains cycles

e. No, not connected

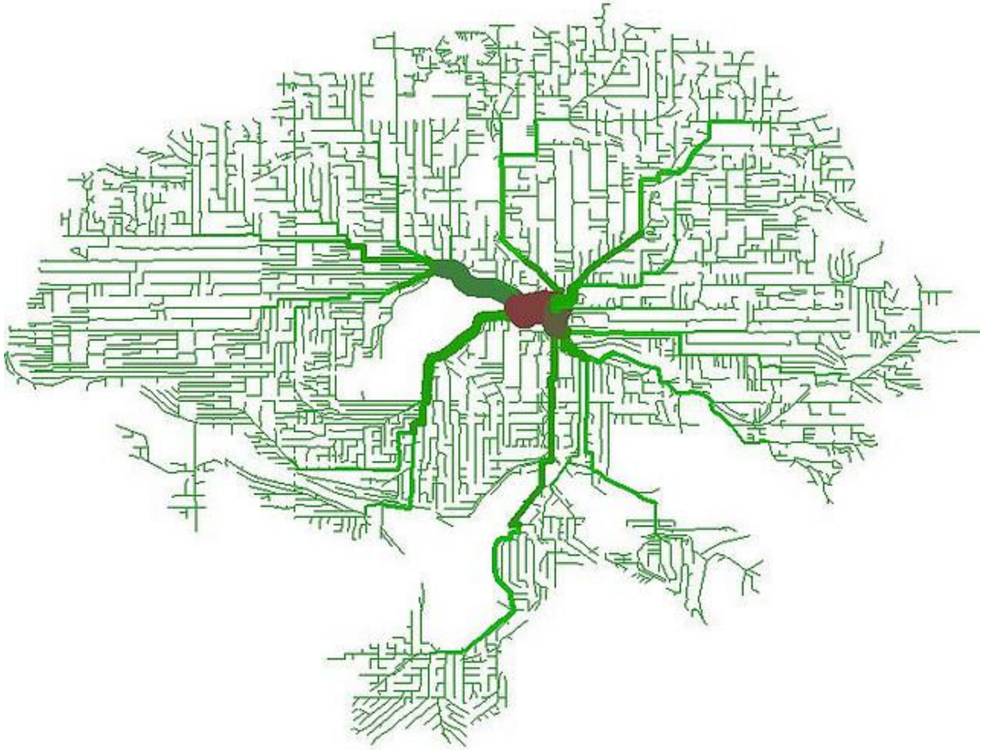
- You have used trees many times in this class already.
  (TSP, counting techniques, probability)
- Trees have many applications in the real world.





What are some other applications for trees?

Notes Section 5.5
Spanning Trees



Spanning Trees

Many of our lessons have dealt with <u>OPTIMIZATION</u> (minimizing cost or time or distance).

In this section we are dealing with two situations.

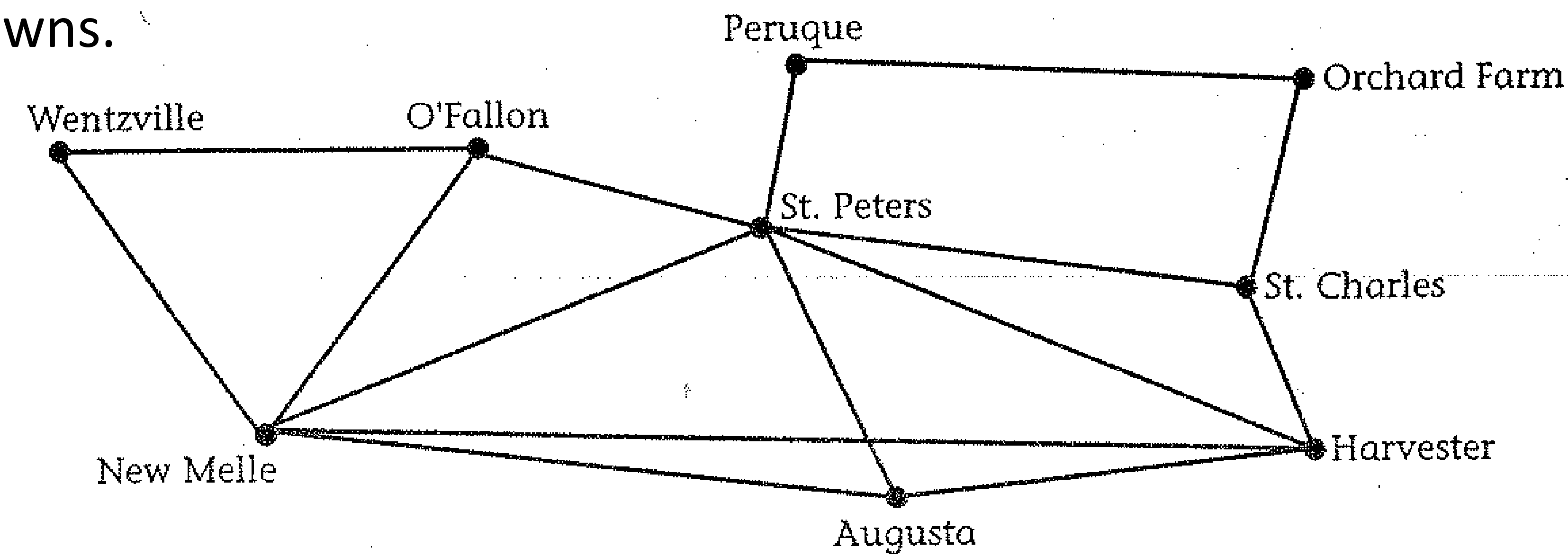
1) **Connecting vertices with the least number of edges.**
2) **Connecting vertices with edges that have the least total weight.**

A <u>SPANNING TREE</u> is a subgraph that will contain <u>every vertex</u> of the graph.

1) **It will contain the <u>minimum number of edges</u> needed to connect the vertices.**
2) **It will contain <u>no cycles</u>.**

Spanning Trees

In making earthquake preparedness plans, the St. Charles County needs a design for repairing the county roads in case of an emergency. The figure below is a map of the towns in the county and the existing major roads between them. Devise a plan that repairs the least number of roads but keeps a route open between each pair of towns.

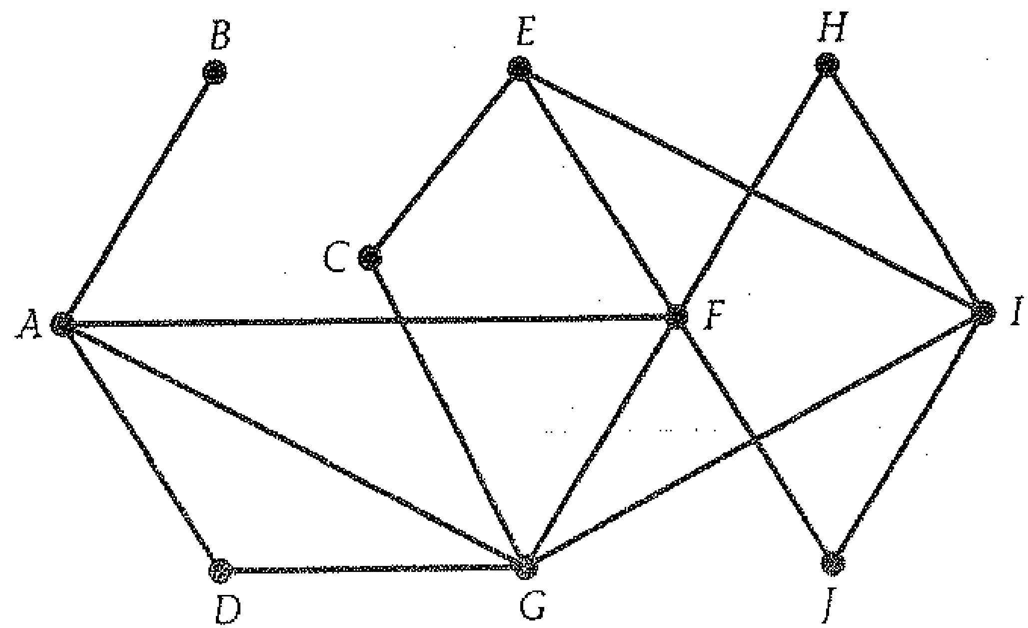


Examine your graph. If it connects each of the towns (vertices) and has no cycles, you've found a spanning tree.

How many edges are in your spanning tree? **8**

Spanning Trees

We need a systematic way of doing this.

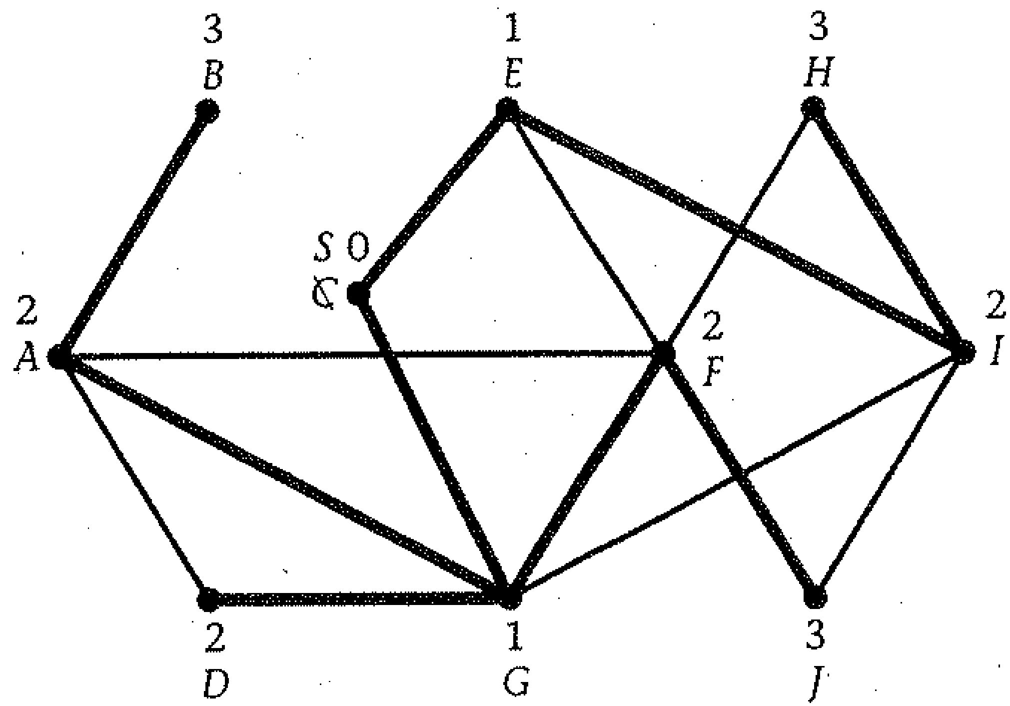


Breadth-First Search Algorithm for Finding Spanning Trees

1. Pick a starting vertex, S, and label it with a 0.
2. Find all vertices that are adjacent to S and label them with a 1.
3. For each vertex labeled with a 1, find an edge that connects it with the vertex labeled 0. Darken those edges.
4. Look for unlabeled vertices adjacent to those with the label 1 and label them 2. For each vertex labeled 2, find an edge that connects it with a vertex labeled 1. Darken that edge. If more than one edge exists, choose one arbitrarily.
5. Continue this process until there are no more unlabeled vertices adjacent to labeled ones. If not all vertices of the graph are labeled, then a spanning tree for the graph does not exist. If all vertices are labeled, the vertices and darkened edges are a spanning tree of the graph.

Spanning Trees

We need a systematic way of doing this.

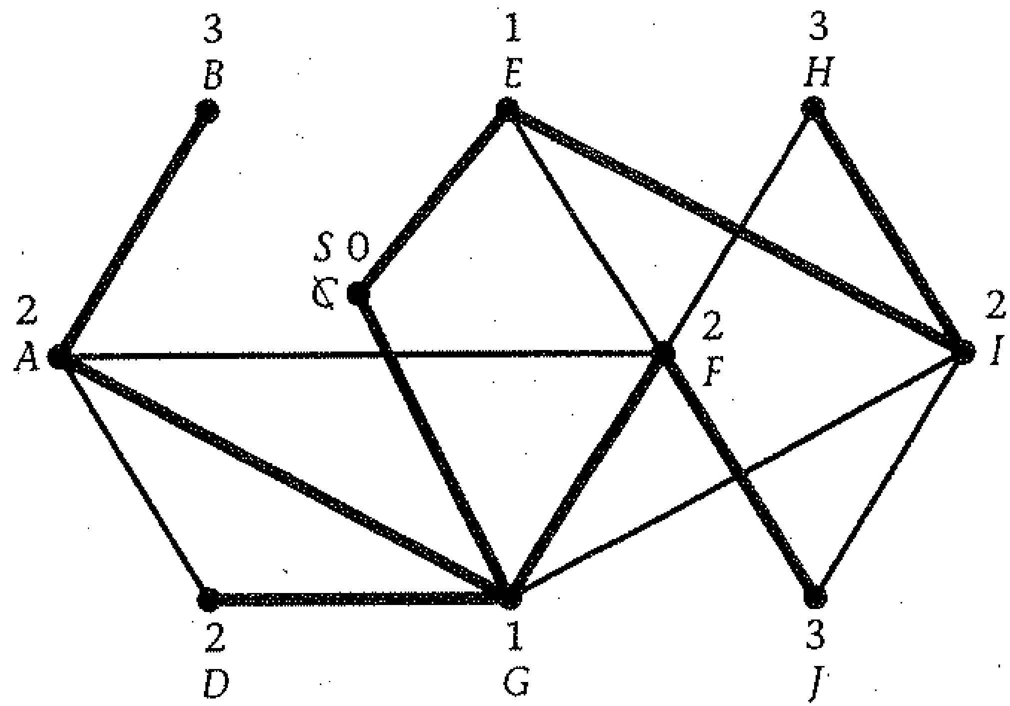


Breadth-First Search Algorithm for Finding Spanning Trees

1. Pick a starting vertex, S, and label it with a 0.
2. Find all vertices that are adjacent to S and label them with a 1.
3. For each vertex labeled with a 1, find an edge that connects it with the vertex labeled 0. Darken those edges.
4. Look for unlabeled vertices adjacent to those with the label 1 and label them 2. For each vertex labeled 2, find an edge that connects it with a vertex labeled 1. Darken that edge. If more than one edge exists, choose one arbitrarily.
5. Continue this process until there are no more unlabeled vertices adjacent to labeled ones. If not all vertices of the graph are labeled, then a spanning tree for the graph does not exist. If all vertices are labeled, the vertices and darkened edges are a spanning tree of the graph.

Spanning Trees ANSWERS

We need a systematic way of doing this.



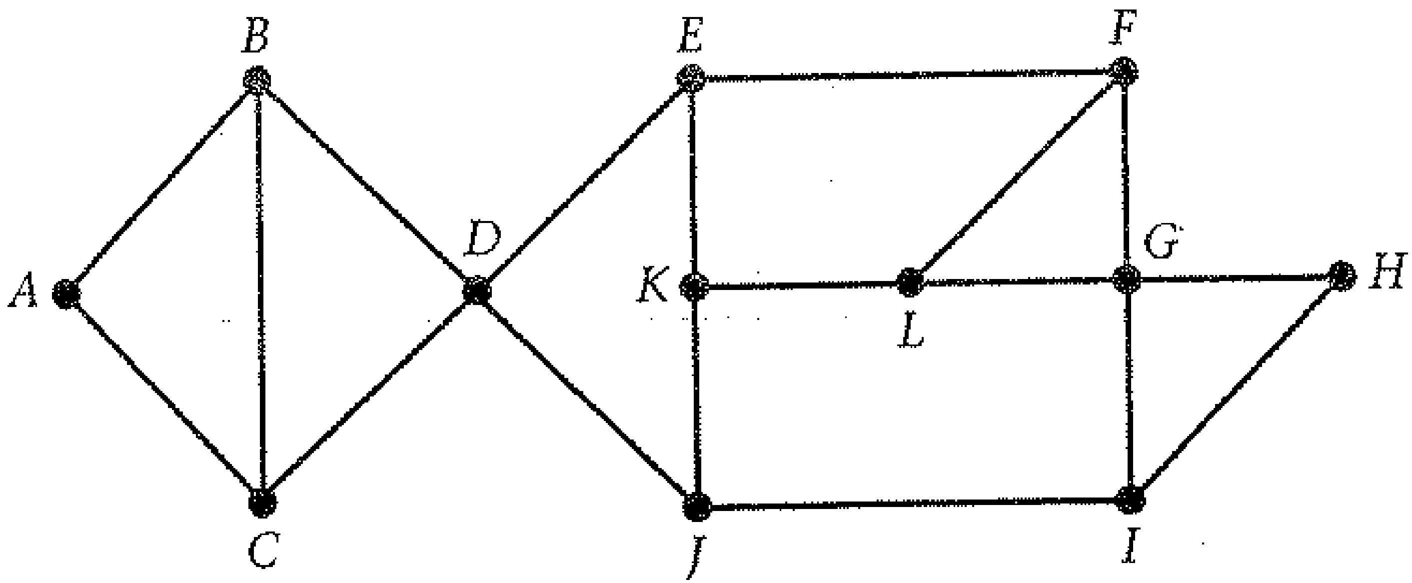
Breadth-First Search Algorithm for Finding Spanning Trees

1. Pick a starting vertex, S, and label it with a 0.
2. Find all vertices that are adjacent to S and label them with a 1.
3. For each vertex labeled with a 1, find an edge that connects it with the vertex labeled 0. Darken those edges.
4. Look for unlabeled vertices adjacent to those with the label 1 and label them 2. For each vertex labeled 2, find an edge that connects it with a vertex labeled 1. Darken that edge. If more than one edge exists, choose one arbitrarily.
5. Continue this process until there are no more unlabeled vertices adjacent to labeled ones. If not all vertices of the graph are labeled, then a spanning tree for the graph does not exist. If all vertices are labeled, the vertices and darkened edges are a spanning tree of the graph.

Spanning Trees

Attempt the Breadth-First Algorithm with this graph.

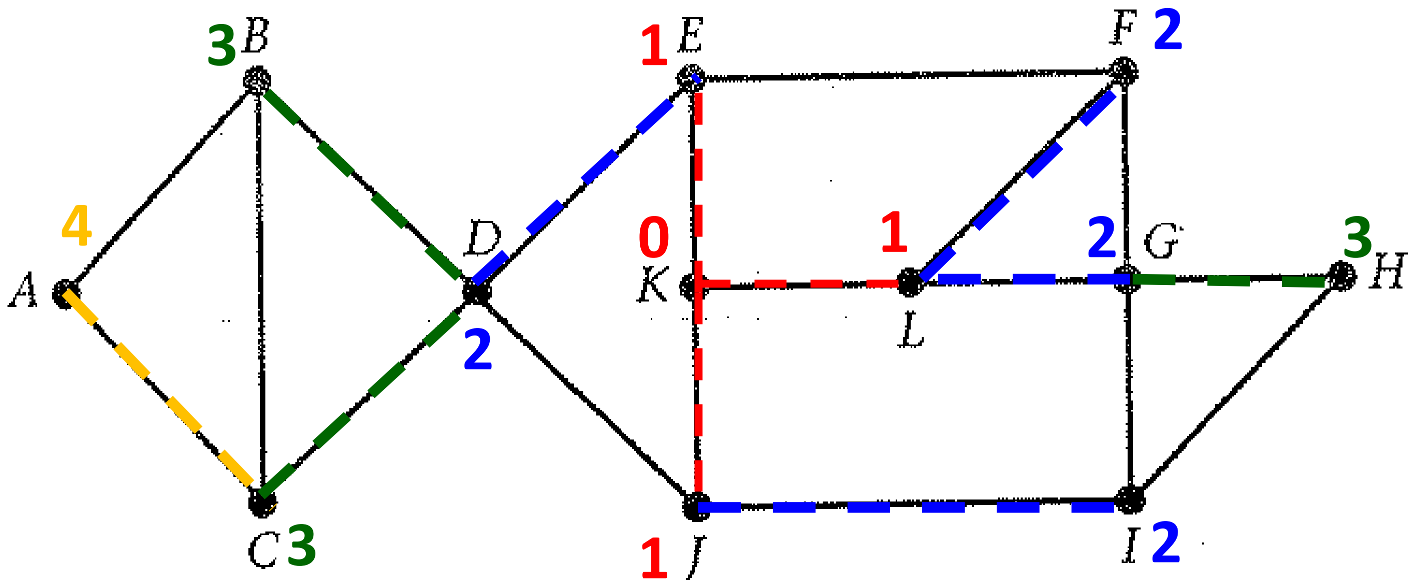
Let's start at K, so we are all on the same page ☺



Spanning Trees ANSWERS

Attempt the Breadth-First Algorithm with this graph.

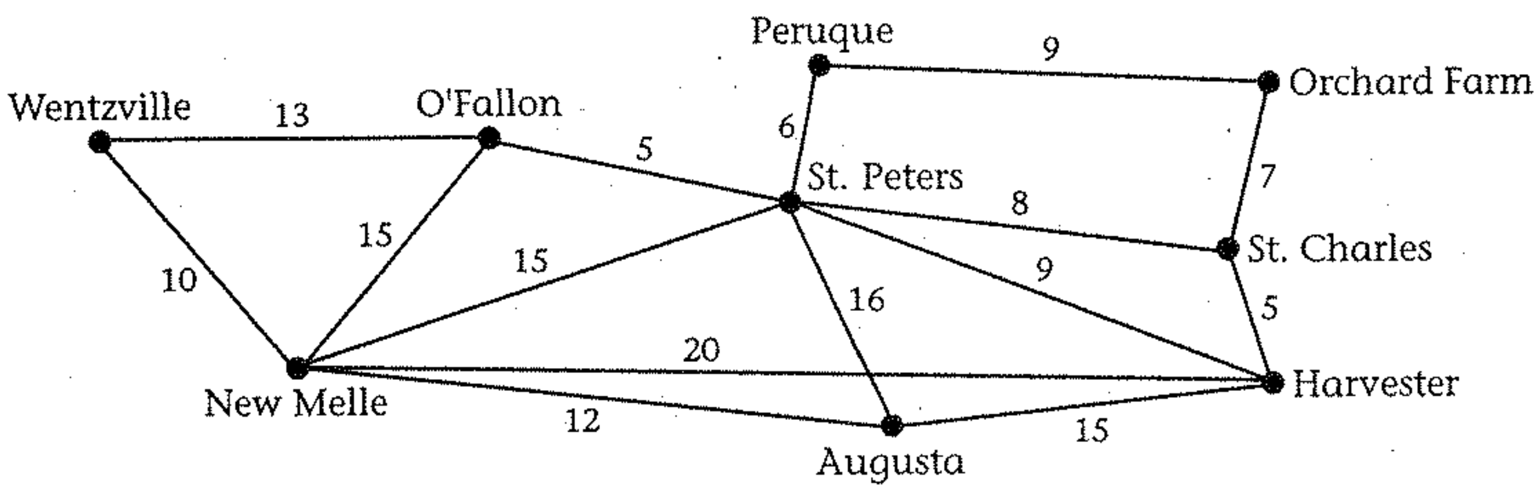
Let's start at K, so we are all on the same page ☺



Minimum Spanning Trees

Many applications are best modeled with weighted graphs. When this is the case, it is often not sufficient to find just any spanning tree, but to find one with minimal or maximal weight.

*Let's return to the earthquake preparedness situation and reconsider the problem when distances between towns are added to the graph.



Now, we want to find more than a Spanning Tree. We want to find an OPTIMAL Spanning Tree, the one with the minimum distance.

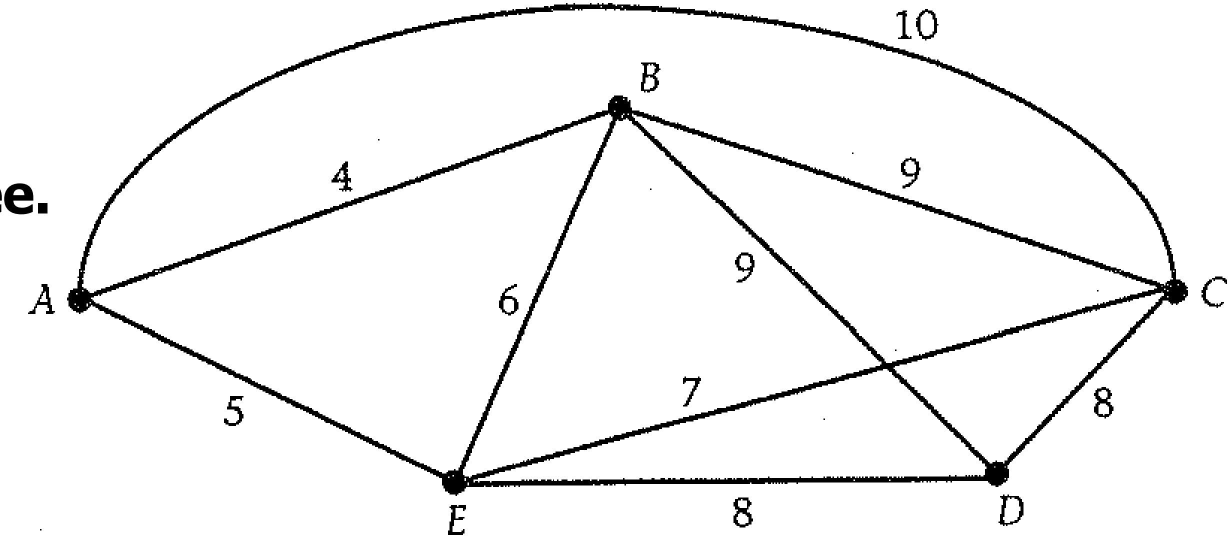
Were any of the graphs we made earlier optimal?

A Spanning Tree with minimal weight is called a MINIMUM SPANNING TREE.

Minimum Spanning Trees

We need a systematic way of finding the Minimum Spanning Tree.

Examine Kruskal's algorithm.



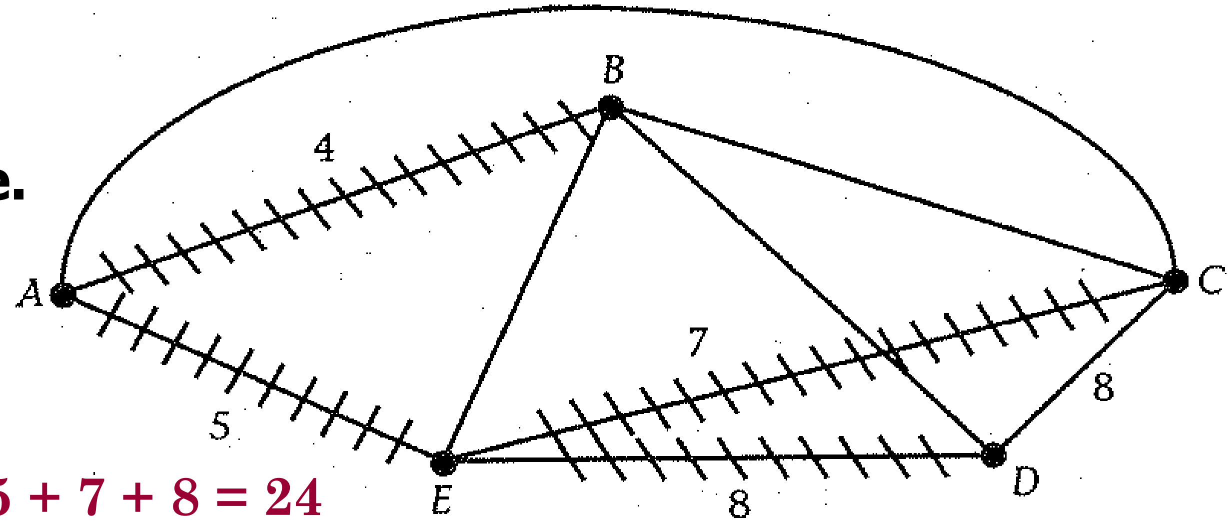
Kruskal's Minimum Spanning Tree Algorithm

1. Examine the graph. If it is not connected, there will be no minimum spanning tree.
2. List the edges in order from shortest to longest. Ties are broken arbitrarily.
3. Darken the first edge on the list.
4. Select the next edge on the list. If it does not form a cycle with the darkened edges, darken it.
5. For a graph with n vertices, continue step 4 until $n - 1$ edges of the graph have been darkened. The vertices and the darkened edges are a minimum spanning tree for the graph.

Minimum Spanning Trees ANSWERS

We need a systematic way of finding the Minimum Spanning Tree.

Examine Kruskal's algorithm.



Minimal weight is $4 + 5 + 7 + 8 = 24$

Kruskal's Minimum Spanning Tree Algorithm

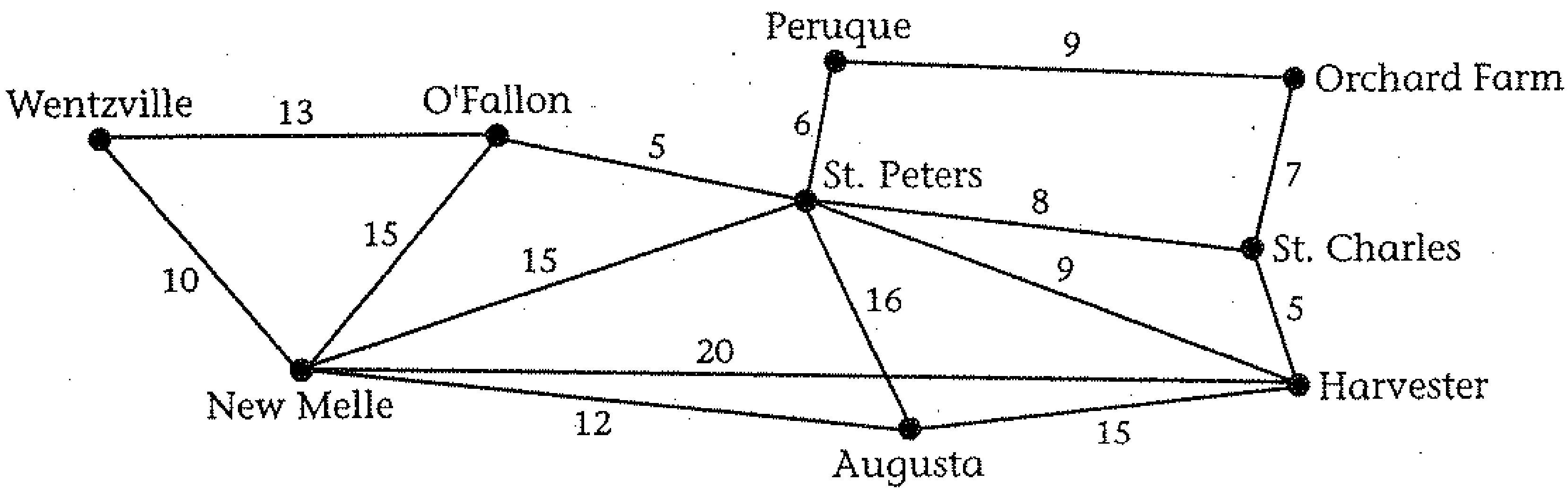
1. Examine the graph. If it is not connected, there will be no minimum spanning tree.
2. List the edges in order from shortest to longest. Ties are broken arbitrarily.
3. Darken the first edge on the list.
4. Select the next edge on the list. If it does not form a cycle with the darkened edges, darken it.
5. For a graph with n vertices, continue step 4 until $n - 1$ edges of the graph have been darkened. The vertices and the darkened edges are a minimum spanning tree for the graph.

List of Edges from Shortest to Longest

Edge	Length
AB	4
AE	5
BE	6
EC	7
CD	8
ED	8
BD	9
BC	9
AC	10

Minimum Spanning Trees

Now try Kruskal's algorithm with the St. Charles County example.

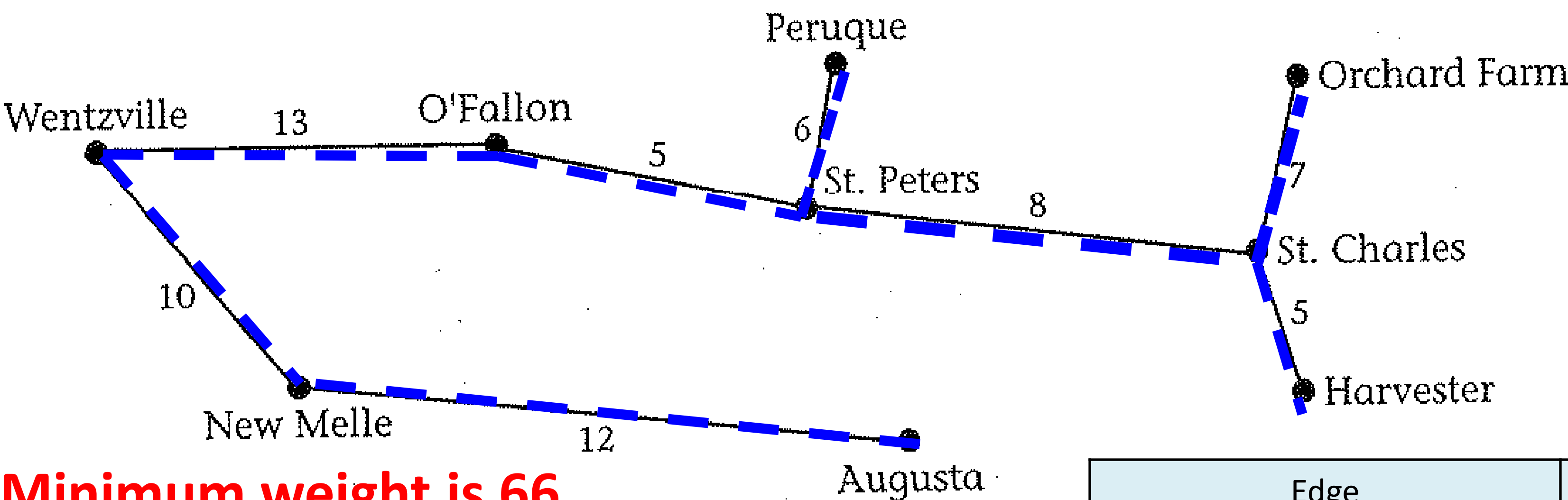


Kruskal's Minimum Spanning Tree Algorithm

1. Examine the graph. If it is not connected, there will be no minimum spanning tree.
2. List the edges in order from shortest to longest. Ties are broken arbitrarily.
3. Darken the first edge on the list.
4. Select the next edge on the list. If it does not form a cycle with the darkened edges, darken it.
5. For a graph with n vertices, continue step 4 until $n - 1$ edges of the graph have been darkened. The vertices and the darkened edges are a minimum spanning tree for the graph.

Minimum Spanning Trees

Now try Kruskal's algorithm with the St. Charles County example.



Minimum weight is 66.

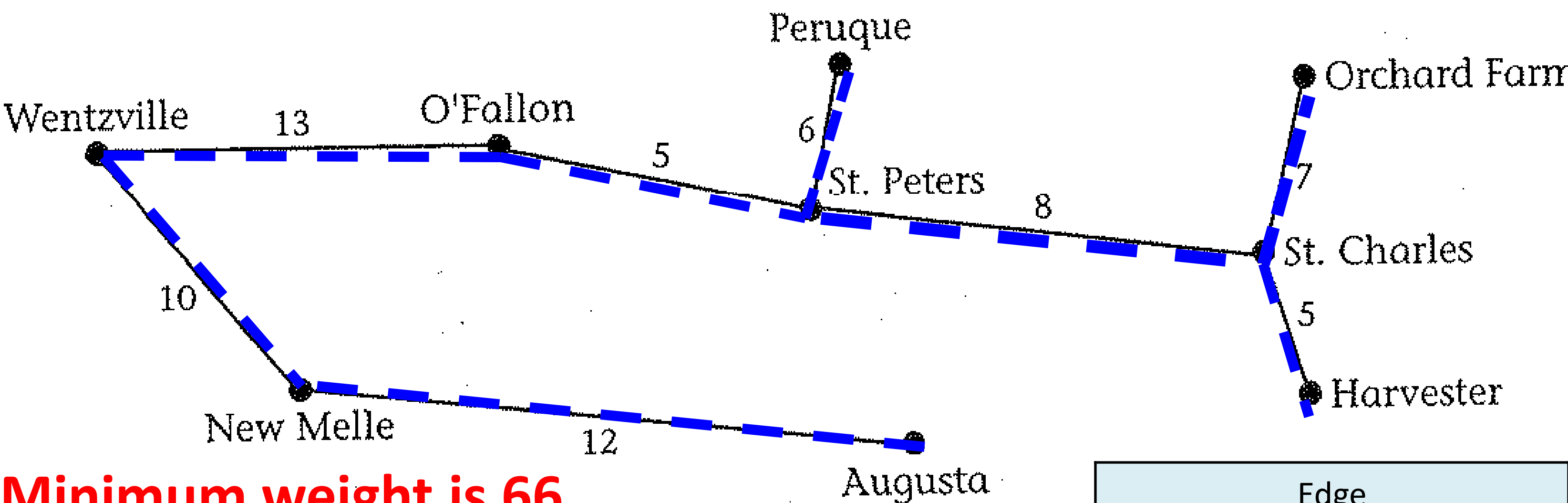
Kruskal's Minimum Spanning Tree Algorithm

1. Examine the graph. If it is not connected, there will be no minimum spanning tree.
2. List the edges in order from shortest to longest. Ties are broken arbitrarily.
3. Darken the first edge on the list.
4. Select the next edge on the list. If it does not form a cycle with the darkened edges, darken it.
5. For a graph with n vertices, continue step 4 until $n - 1$ edges of the graph have been darkened. The vertices and the darkened edges are a minimum spanning tree for the graph.

Edge	Distance
O'Fallen- St. Peters	5
Harvester- St. Charles	5
Peruque- St. Peters	6
St. Charles- Orchard Farm	7
St. Peters- St. Charles	8
St. Peters- Harvester	9
Peruque- Orchard Farm	9
Wentzville- New Melle	10
New Melle- Augusta	12
Wentzville- O'Fallen	13
New Melle- O'Fallen	15
New Melle- St. Peters	15
Augusta- Harvester	15
St. Peters- Augusta	16
New Melle- Harvester	20

Minimum Spanning Trees ANSWERS

Now try Kruskal's algorithm with the St. Charles County example.



Minimum weight is 66.

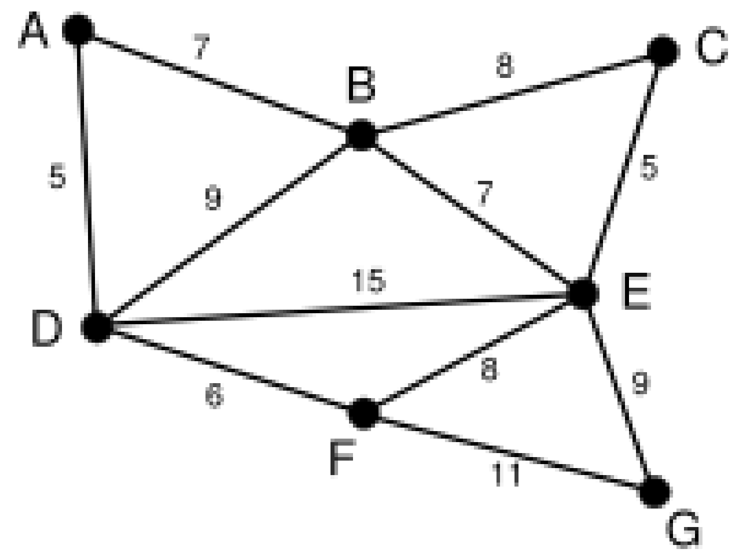
Kruskal's Minimum Spanning Tree Algorithm

1. Examine the graph. If it is not connected, there will be no minimum spanning tree.
2. List the edges in order from shortest to longest. Ties are broken arbitrarily.
3. Darken the first edge on the list.
4. Select the next edge on the list. If it does not form a cycle with the darkened edges, darken it.
5. For a graph with n vertices, continue step 4 until $n - 1$ edges of the graph have been darkened. The vertices and the darkened edges are a minimum spanning tree for the graph.

Edge	Distance
O'Fallen- St. Peters	5
Harvester- St. Charles	5
Peruque- St. Peters	6
St. Charles- Orchard Farm	7
St. Peters- St. Charles	8
St. Peters- Harvester	9
Peruque- Orchard Farm	9
Wentzville- New Melle	10
New Melle- Augusta	12
Wentzville- O'Fallen	13
New Melle- O'Fallen	15
New Melle- St. Peters	15
Augusta- Harvester	15
St. Peters- Augusta	16
New Melle- Harvester	20

Traveling salesperson problems, shortest route problems, and minimum spanning tree problems are often confused because each type of problem can be solved by finding a subgraph that may include all of the vertices.

Edge	Distance
AD	5
CE	5
DF	6
AB	7
BE	7
FE	8
BC	8
BD	9
EG	9
FG	11
DE	15

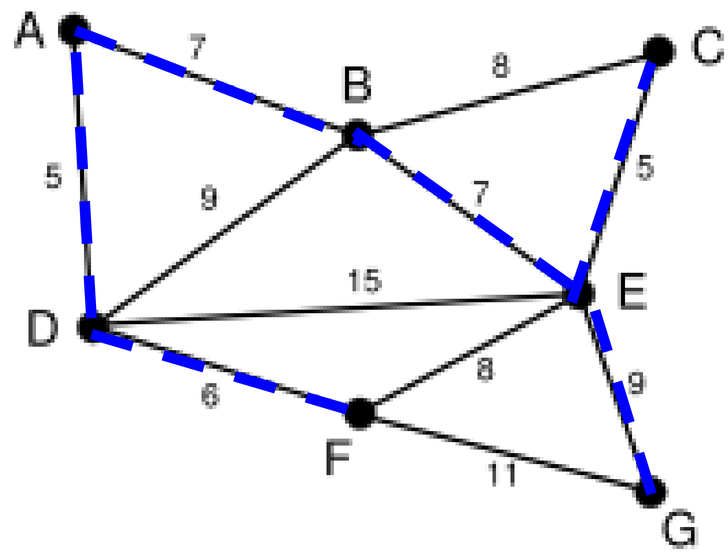


Using the graph to solve the following:

a. Explain how you would use a TSP Tree to determine the best path.
b. Find the shortest route from A to G.
c. Find the minimum spanning tree and its weight.
d. Compare and contrast what each type of problem asks and when each type is used.

Traveling salesperson problems, shortest route problems, and minimum spanning tree problems are often confused because each type of problem can be solved by finding a subgraph that may include all of the vertices.

Edge	Distance
AD	5
CE	5
DF	6
AB	7
BE	7
FE	8
BC	8
BD	9
EG	9
FG	11
DE	15



Using the graph to solve the following:

a. Explain how you would use a TSP Tree to determine the best path.
b. Find the shortest route from A to G. ADFG = 22
c. Find the minimum spanning tree and its weight. Min Weight = 39
d. Compare and contrast what each type of problem asks and when each type is used.

Day 7 Homework

Packet p. 15-18