Section 8.0: Introduction to Graph Theory

This chapter provides an introduction into **graph theory—**the study of graphs. Graph theory is a large area of research for mathematicians and computer scientists. However, only two mathematical techniques involving graphs are presented in this chapter.

First, a technique called Kruskal's algorithm is used to find paths between houses after a tornado destroyed a town's road system. The roads need to be rebuilt so that emergency workers and volunteers have a way to travel from one house to another. However, the lengths of these rebuilt roads need to be minimized due to time and money constraints. Kruskal's algorithm is used to find the minimum total lengths of the rebuilt roads.

Second, a mathematical technique called Dijkstra's algorithm is used to determine the best route for a company that transports urgently needed medical supplies. With the high cost of gasoline and sometimes short notice to transport the supplies, the route used to transport these medical supplies is an important consideration. Dijkstra's algorithm is used to find the shortest routes to transport medical supplies from a supplies company to medical offices. Next, this same medical supplies company wants to build a new warehouse next to one of its customers. The possible locations for the new warehouse are found by performing Dijkstra's algorithm repeatedly.

Third, Dijkstra's algorithm is used again, this time to explore how long it would take for a rumor to spread. Then, the algorithm is used repeatedly to find which person should start the rumor in order for it to spread most quickly. Although the medical supplies context and the rumor context are very different, both problems can be solved by utilizing Dijkstra's algorithm to find the shortest path.

Before Kruskal's algorithm and Dijkstra's algorithm are presented, graph theory is introduced through the context of social networks.

8.0.1 Six Degrees of Separation

The phrase **six degrees of separation** is common in our country. The idea is that, through social networks, any person in the United States can be connected to any other person by an average of six people.

A Social Experiment

The theory of *six degrees of separation* has been around for several generations, but in 1967, Stanley Milgram performed an experiment to study this theory. In this experiment, Milgram sent out several hundred letters to people in Nebraska and Kansas. These letters contained instructions for the recipients to follow. The eventual goal was for the letter to reach a specific person in Boston, Massachusetts. Therefore, the letter instructed the recipients to either (1) send this letter to the person in Boston if they were familiar with them (i.e. if they were on a first-name basis with them), or (2) send it to someone who they were on a first-name basis with and who may have a better chance of knowing the person in Boston. If the recipient did not know the person in Boston, they may choose to send it to someone who lives closer to Boston than they do, or they may decide to send the letter to someone who has a large group of friends and acquaintances so that there was a better chance that the letter eventually arrived at the person in Boston. For the people who chose to participate, the average number of people it took to get to the person in Boston was 5.5. Hence, this experimented supported the idea of *six degrees of separation*.

Although this theory is sociological, it has leaked into the world of pop culture. For example, in the early 1990s, many people began to play the game **six degrees of Kevin Bacon**. In this game, people choose a random actor/actress and try to find a connection through movies to the popular actor Kevin Bacon. Kevin Bacon is known for his roles in Mystic River, Apollo 13, A Few Good Men, and Footloose.

To better understand the six degrees of Kevin Bacon game, consider the following example. If Angelina Jolie is chosen as the beginning actor/actress, she was in *Mr. & Mrs. Smith* (2005) with Brad Pitt, and Brad Pitt was in *Sleepers* (1996) with Kevin Bacon. Thus, Angelina Jolie is connected to Kevin Bacon through one person, so her **Bacon number** is 2. The Bacon number refers to the minimum number of degrees of separation (e.g., Kevin Bacon's Bacon number is 0 and Brad Pitt's Bacon number is 1).

This is not the only connection from Angelina Jolie to Kevin Bacon. Figure 8.0.1 shows a graph with a few of the connections between Angelina Jolie and Kevin Bacon. Some paths are obviously longer than others. A **path**, in this context, refers to a connection between two people through movies. The goal, when playing this game, is to find the path with the minimum number of connections. There could be several connections from Angelina Jolie to Kevin Bacon, but the shortest connections are of length 2.

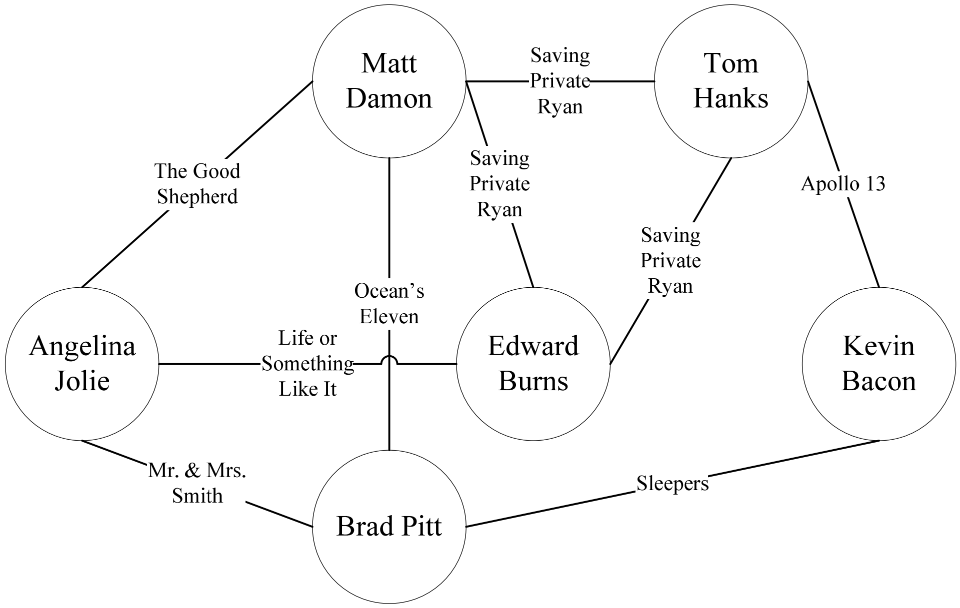


Figure 8.0.1: An example of a graph for the Kevin Bacon game

Q1. Choose an actor/actress and connect him/her to Kevin Bacon through movies.
  a. Create a graph similar to the one in Figure 8.0.1 showing more than one way to connect this actor/actress to Kevin Bacon.
  b. Determine the Bacon number (i.e., the minimum number of people it takes to get from this actor/actress to Kevin Bacon).

For the rock fans out there, there is also *six degrees of Dave Grohl*, where many rock music forums discuss the connections between various musicians and Dave Grohl, the drummer from Nirvana and the lead vocalist from Foo Fighters.

Also, social networking sites such as Friendster, MySpace, and Facebook use the idea of degrees of separation. For example, a graph similar to the one in Figure 8.0.1 can be created for Facebook friends. A fictitious example is given in Figure 8.0.2, where the connections between people represent Facebook friendships. In this example, Joey is not friends with Jonathan. However, they are connected through friends. Possible paths from Joey to Jonathan include:

- Joey-Jordan-Jonathan
- Joey-Donnie-Jonathan
- Joey-Danny-Donnie-Jonathan

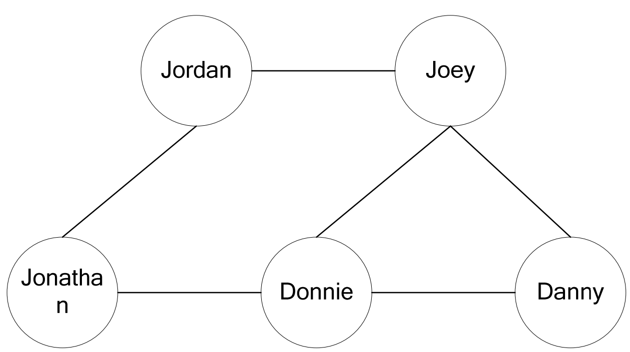


Figure 8.0.2: An example of a graph connecting Facebook friends

Figures 8.0.1 and 8.0.2 are examples of graphs. However, the kinds of graphs seen in this chapter are not the same as what you are used to seeing in Algebra (e.g., number lines or Cartesian planes). A **graph** is a broad term meaning a set of points, called **nodes**, connected by segments, called **arcs** (note: nodes are sometimes called *vertices* and arcs are sometimes called *edges*, but in this chapter, only the terms *nodes* and *arcs* are used). Figure 8.0.3 shows an example of a graph with 6 nodes and 10 arcs. Note that arcs can cross, like arc *AD* and arc *BC* in the figure. Crossing arcs do not form another node.

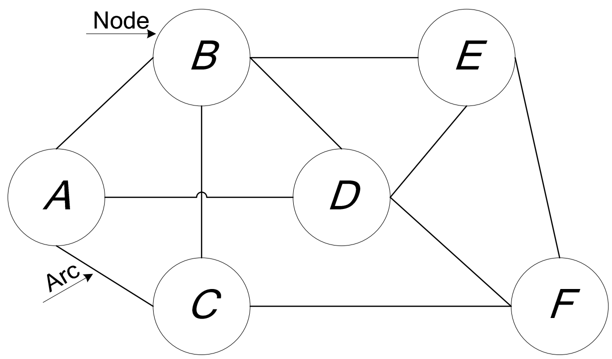


Figure 8.0.3: Example of a graph with 6 nodes and 10 arcs

Graphs, such as the one shown in Figure 8.0.3, are explored throughout this chapter. In the next section, a graph is used to determine which roads should be reconstructed after a tornado came through a town.

Section 8.1: Road Reconstruction

A tornado passed through a small Midwestern town called Nashatuck, destroying the entire road system in the town. Some roads need to be reconstructed so that it is possible to get from each house to every other house. The path from one house to another may be indirect, but there needs to be some sort of path between each house for emergency workers and volunteers to travel.

In the graph shown in Figure 8.1.1, the nodes represent houses and the arcs represent the roads between the houses before the tornado. The numbers on the arcs correspond to the distance, in miles, between each house. For example, house *A* is two miles from house *B*.

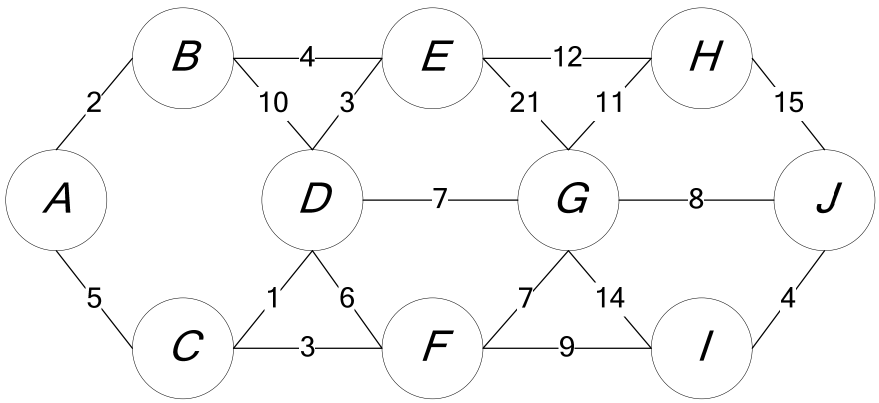


Figure 8.1.1: A graph representing the road system in Nashatuck

Note that the graph is not scaled to represent the distances as lengths. For example, the distance between house *C* and house *D* is 1 mile, and the distance between house *D* and house *F* is 6 miles. However, in Figure 8.1.1, arc *CD* and arc *DF* have the same length. This is common practice in graph problems. However, the distances and not the lengths are used to work through the problem.

To reduce the cost of reconstruction, the Nashatuck government officials want to minimize the lengths of the roads they rebuild. They need to determine which roads should be reconstructed. To do so, they create a minimum spanning tree.

8.1.1 Minimum Spanning Trees

Minimum spanning trees are used to determine which roads should be reconstructed. To understand minimum spanning trees, some definitions need to be given.

First, a **tree** is a graph in which there are no circuits. A **circuit** is a path that starts and ends in the same node. Figures 8.1.2 and 8.1.3 show examples and non-examples of trees, respectively. The graphs shown in Figure 8.1.3 are not trees because they contain circuits, shown with dotted lines.

Next, a **spanning tree** is a tree (so there are no circuits) that connects all of the nodes. Thus, a path exists from each node to every other node, but it may not be a direct path. Examples of spanning trees are given in Figure 8.1.4.

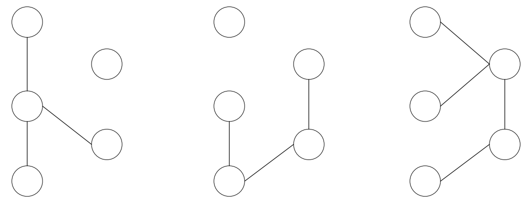


Figure 8.1.2: Examples of trees

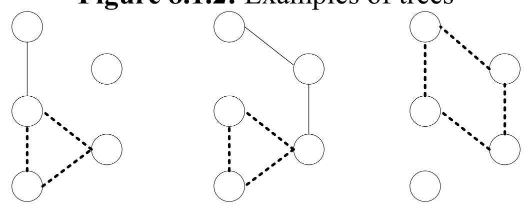


Figure 8.1.3: Non-examples of trees

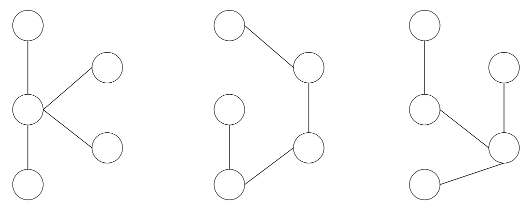


Figure 8.1.4: Examples of spanning trees

If a tree is **weighted** (i.e., the arcs are given a "weight" such as distance or cost), then a minimum spanning tree can be found. A **minimum spanning tree** is a spanning tree of the least total weight. Therefore, in a minimum spanning tree, every node is connected directly or indirectly to every other node in the graph, and the total weight as small as possible.

Minimum spanning trees can be used to determine which roads need to be rebuilt after being destroyed by the tornado. The weights in this problem refer to the number of miles between houses (shown on the arcs in Figure 8.1.1). Some, but not all, roads in the town need to be reconstructed to allow emergency workers and volunteers to have access to all of the houses. They need to determine which roads to rebuild so that the cost of reconstruction is minimized. To minimize this cost, the total number of miles of the rebuilt roads should be as small as possible. Therefore, it makes sense that minimum spanning trees should be used to solve this problem.

Q1. In the definition of a minimum spanning tree, circuits are not allowed. Why would circuits not be needed when rebuilding roads after a tornado?

Q2. Using Figure 8.1.1, find a spanning tree that connects all of the houses. Find the total weight of your spanning tree (i.e., add up the number of miles on the arcs in your spanning tree).

Q3. Compare your spanning tree and the total weight of your spanning tree with a neighbor.

Q4. What is the minimum total weight of the spanning trees for all students in the class? Is this the minimum possible total weight? How do you know?

To solve this problem, we need to implement an algorithm. An **algorithm** is a set of steps followed to solve a problem. In many cases, these steps are repeated until there is a reason to end. For example, if someone was asked how to walk to a wall, he/she might say, "Take a step and repeat until you reach the wall." That is a very simple algorithm.

For this problem, Kruskal's algorithm is used. Kruskal's algorithm is an example of a **greedy algorithm** because at each step of the problem, the best choice is made for that particular step. When this is done, the result is a minimum spanning tree that connects each of the houses in the network.

8.1.2 Kruskal's Algorithm

Step 1: Find the arc of the least weight and mark it.
Note: the arc of the least weight does not necessarily correspond with a particular node. For example, the algorithm does not necessarily begin with node *A*.

In this example, the arc of the least weight is *CD*, which is of weight 1, meaning that house *C* is 1 mile from house *D*. Since this road is relatively short, it makes sense that the government officials would make sure to rebuild this road. The graph in Figure 8.1.5 shows arc *CD* marked.

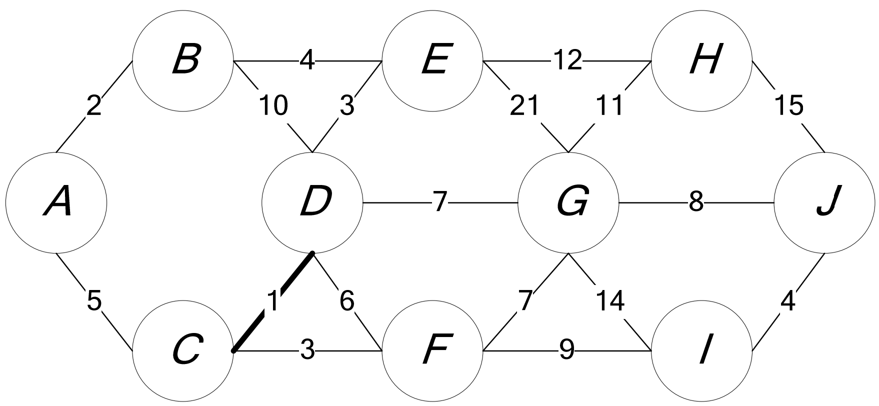


Figure 8.1.5: Step 1 of Kruskal's algorithm, where the arc of the least weight is arc *CD*

Step 2: Find the arc of the next least weight and mark it.
In this example, the arc of the next least weight is arc *AB*. Since house *A* is only 2 miles from house *B*, this road should also be reconstructed. Arc *AB* is marked in Figure 8.1.6.

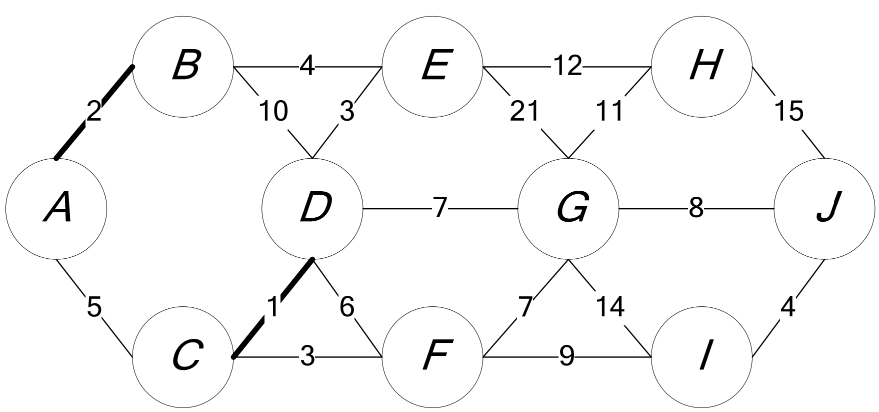


Figure 8.1.6: Step 2 of Kruskal's algorithm, where the arc of next least weight is arc *AB*

Step 3: Repeat Step 2 until each node is connected. If the arc with the next least weight creates a circuit, skip it and go to the arc with the next least weight after that.

Remember that a circuit is a path that starts and ends at the same node. If a circuit is created, then the graph is no longer a tree.

For the road reconstruction example, there are two arcs of with the next least weight; *DE* and *CF* are both of length 3. Both of these arcs can be marked, as shown in Figure 8.1.7.

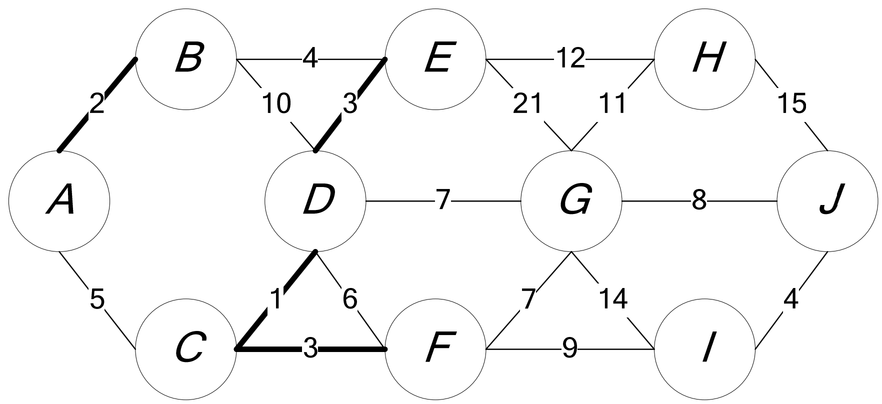


Figure 8.1.7: Step 3 of Kruskal's algorithm, where arcs *DE* and *CF* are both marked

Again, two arcs share the next least weight: *BE* and *IJ* are both of length 4. That is, the length of the road between houses *B* and *E* is four miles, and the length of the road between houses *I* and *J* is also four miles. Both of these arcs are marked in Figure 8.1.8.

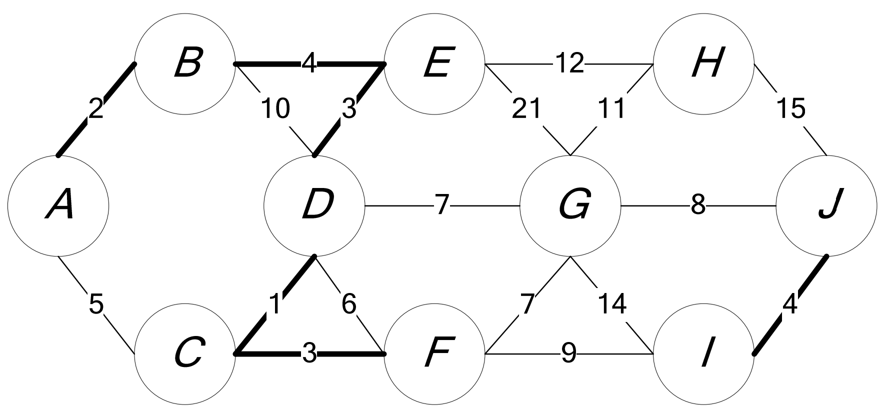


Figure 8.1.8: Continuing Step 3 of Kruskal's algorithm, where arcs *BE* and *IJ* are both marked

At this point, houses *A*, *B*, *C*, *D*, *E*, and *F* are connected to one another as are houses *I* and *J*. However, the emergency workers are still unable to access all of the houses in the town. For example, if an emergency worker is at house *A*, they would have no way to get to houses *G*, *H*, *I*, or *J*. Therefore, the algorithm is continued until all houses can be accessed.

The arc of next least weight is arc *AC*, with a weight of 5. However, if *AC* is marked, there would be a circuit, as shown in Figure 8.1.9. Since houses *A* and *C* are already connected to one another via the path *A-B-E-D-C*, it would be unnecessary to rebuild the road connecting house *A* to house *C*.

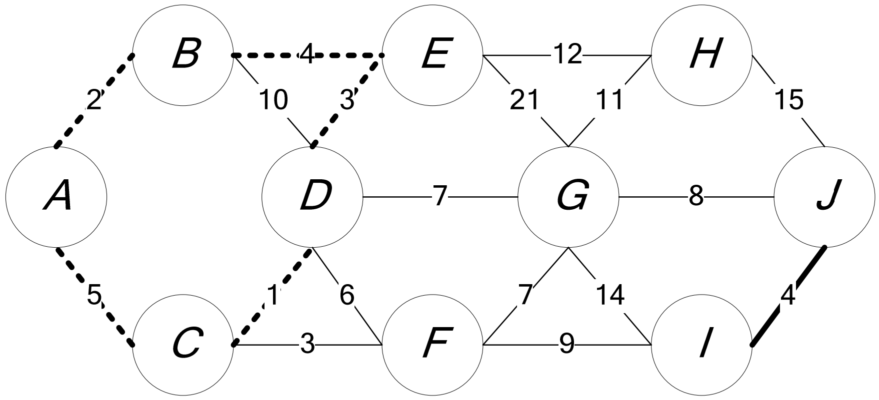


Figure 8.1.9: Choosing arc *AC* creates a circuit

Therefore, the government officials skip this arc and go on to the arc of next least weight, which is *DF*, of weight 6. Notice again that a circuit is created if this arc is marked, as shown in Figure 8.1.10.

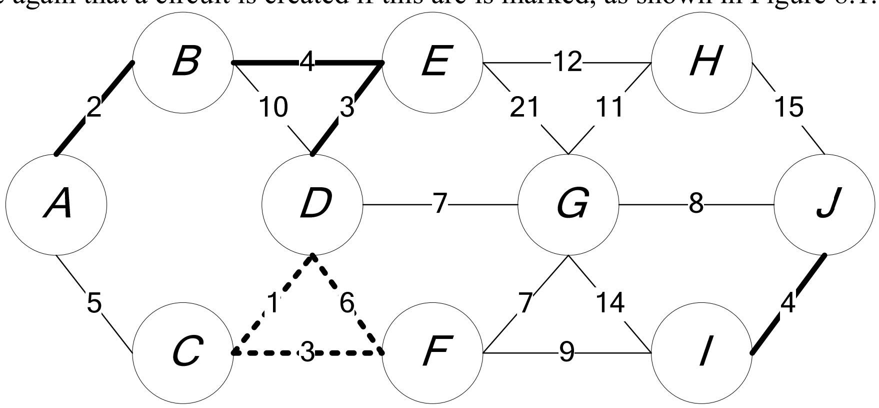


Figure 8.1.10: Choosing arc *DF* creates a circuit

Thus, the government officials skip arc *DF* and move on to the next least weight. Both arcs *DG* and *FG* have a weight of 7. However, if *both* arcs are marked (as was done earlier), a circuit is created, as shown in Figure 8.1.11.

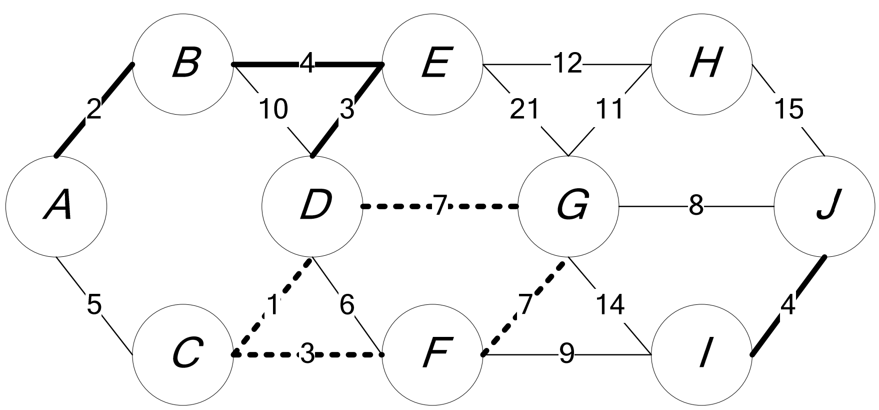


Figure 8.1.11: Choosing both arcs *DG* and *FG* creates a circuit

Therefore, either arc *DG* or arc *FG* must be marked, but not both. It is not clear which one to choose because they are of the same weight. Therefore, the government officials choose *FG* for now. Later, they go back, choose *DG*, and see which arc was the better choice. Figure 8.1.12 shows the graph with arc *FG* marked.

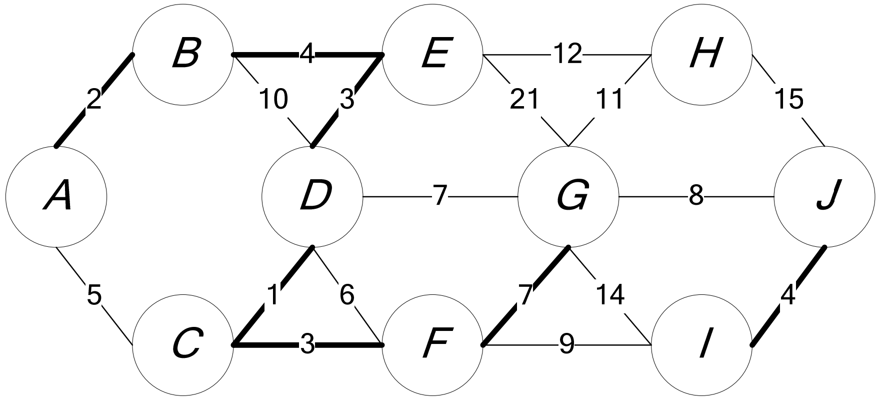


Figure 8.1.12: Continuing with Kruskal's algorithm after choosing arc *FG*

Looking at the graph in Figure 8.1.12, only two more roads need to be reconstructed so that each house can be accessed. Continuing with Kruskal's algorithm, the last two arcs chosen are *GJ* and *GH*, respectively. Figure 8.1.13 shows the complete minimum spanning tree.

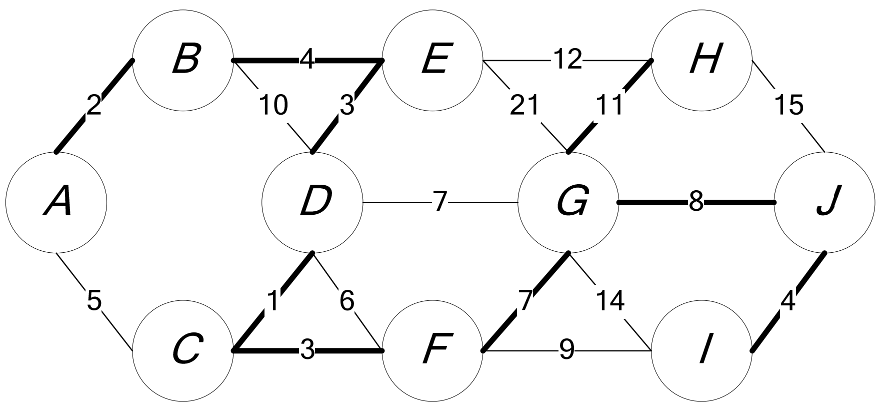


Figure 8.1.13: Completed minimum spanning tree

Q5. How do you know when to stop the algorithm?

Q6. Find the total weight of this minimum spanning tree. What does this mean in terms of the problem?

The steps of Kruskal's' algorithm are given below.

Kruskal's Algorithm

Step 1: Find the arc of the least weight and mark it.

Step 2: Find the arc of the next least weight and mark it.

Step 3: Repeat Step 2 until each node is connected. If the arc with the next least weight creates a circuit, skip it and go to the arc with the next least weight after that.

Recall that the government officials also wanted to find the minimum spanning tree using arc *DG* rather than arc *FG*, as shown in Figure 8.1.14.

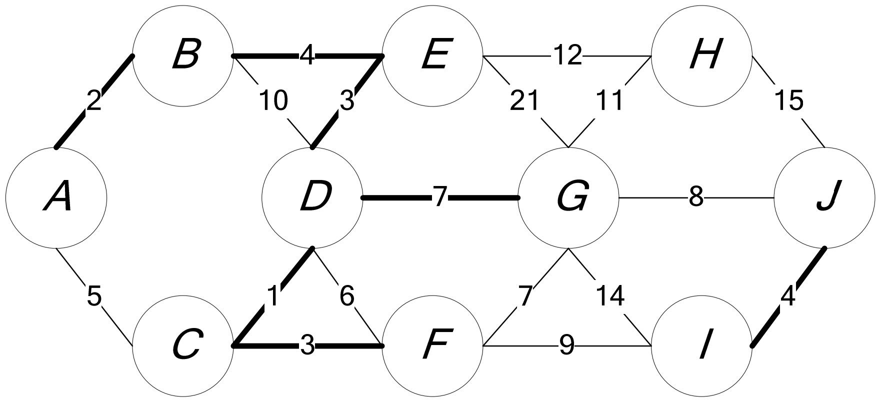


Figure 8.1.14: Continuing with Kruskal's algorithm after choosing arc *DG*, instead of arc *FG*

Q7. Find the minimum spanning tree using arc *DG* instead of arc *FG*.

Q8. Find the total weight of this minimum spanning tree.

Hopefully, you found that the total weight did not change depending on whether arc *FG* or arc *DG* was chosen. This will always be the case. In other words, if you have to choose between two arcs of the same length when creating a minimum spanning tree, which arc you choose will not make a difference in the total weight.

Q9. Using the minimum spanning tree you found in Q7, determine the path an emergency worker would follow if he started at house *A* and needed to get to house *G*.

Q10. Using the minimum spanning tree you found in Q7, determine the path an emergency worker would follow if he started at house *E* and needed to get to house *F*.

Q11. Using the minimum spanning tree you found in Q7, determine the path an emergency worker would follow if he started at house *F* and needed to get to house *I*.

Q12. Suppose the state police headquarters are located along arc *GI*. Therefore, the government officials need to ensure that this arc is part of the minimum spanning tree. What new solution would you provide? Explain how you found this new spanning tree.

Q13. Suppose arc *BE* contains a bridge and is therefore too difficult to rebuild quickly (also assume that the spanning tree needs to include arc *GI* for the state police headquarters). What new solution would you provide? Explain how you found this new spanning tree.

Q14. Using the spanning tree you created in the previous question, determine the path an emergency worker would follow if he started at house *A* and needed to get to house *H*. How long is this path?

Q15. Suppose all of the roads in the town were rebuilt. Determine the shortest path from house *A* to house *H*. How is this question different from the previous question?

In Q15, there were many different paths from house *A* to house *H*. The shortest path may not always be obvious. In the next section, a new algorithm is introduced that makes it easier to find the shortest path between two nodes.

Section 8.2: Medical Supplies

The Triangle Medical Supplies Company, in Raleigh, North Carolina, is a supplier to many doctors' offices located in and around Raleigh, North Carolina. Sometimes emergencies come up, and they need to know the shortest distance from the company to each of the doctors' offices.

The situation in this problem is different from the situation presented in the last problem (Section 8.1). Here, the goal is to find the shortest path from the supplier's warehouse to each doctor's office; previously, the goal was to find the minimum spanning tree connecting houses in a town.

Figure 8.2.1 shows the travel alternatives to nine of the major clients served by the Triangle Medical Supplies Company and the distance between them (in miles). Note that, like in the previous section, the arc lengths are not scaled to represent the value of the weights.

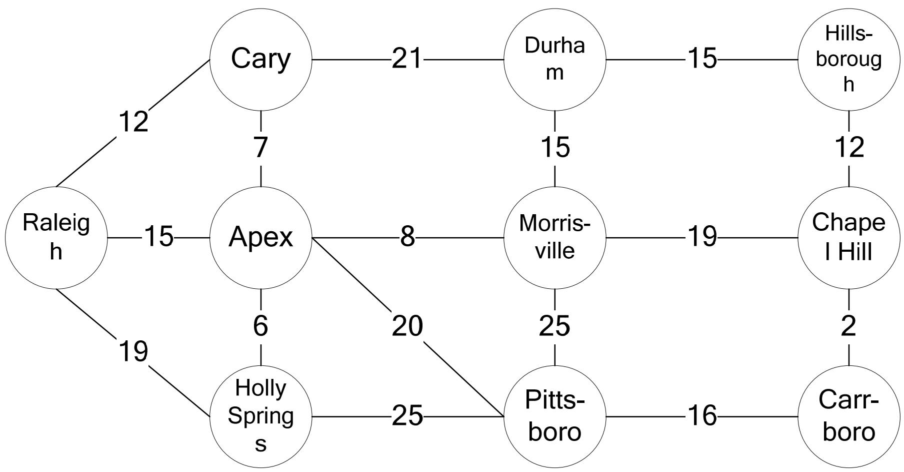


Figure 8.2.1: The medical supplies warehouse, its major clients, and the distance between them

To make this graph easier to discuss, the nodes can be represented with letters rather than the names of the cities, as shown in Figure 8.2.2.

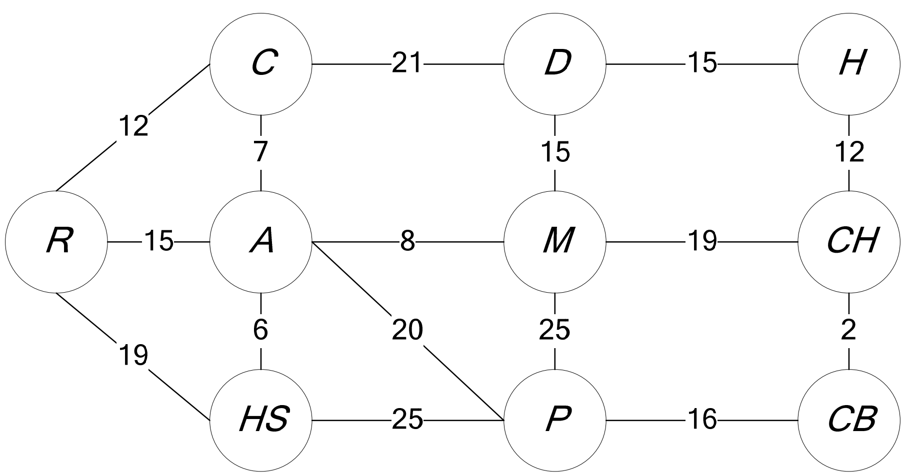


Figure 8.2.2: The supplies company graph, with nodes represented by letters

In this graph, *R* is the supplies warehouse and the shipping supply point for the Triangle Medical Supply Company. The other nodes correspond to the locations of the medical offices. The lines that connect the nodes correspond to the routes between the locations. The numbers on each line represent the travel distances in miles. The manager, Ms. Clark, wants to minimize the travel distance from the supplies warehouse to each of the customer's medical offices.

To get a feel for this, consider the travel distance from Raleigh to Carrboro (*R* to *CB*). There are several different paths, and each path has a different total travel distance. For example,

- The path *R-A-HS-P-CB* has a total distance of 15 + 6 + 25 + 16 = 62 miles
- The path *R-C-D-M-P-CB* has a total distance of 12 + 21 + 15 + 25 + 2 = 75 miles
- The path *R-HS-P-CB* has a total distance of 19 + 25 + 16 = 60 miles

Q1. Choose two other paths from Raleigh to Carrboro and calculate the travel distance.
  a. Is either of the paths you selected shorter than the ones listed above?
  b. Is either of them the shortest path? How can you be sure?

Q2. Using Figure 8.2.1, find the shortest path and the total distance from the supplies company in Raleigh (*R*) to:
  a. Morrisville (*M*)
  b. Pittsboro (*P*)

The above questions show that brute force is not necessarily an efficient way to find the shortest path between two nodes. Instead, a technique called Dijkstra's algorithm can be used to find the shortest path from the supplies warehouse to each of the medical offices.

8.2.1 Dijkstra's Algorithm

Recall that an algorithm is a step-by-step procedure. Like Kruskal's algorithm, Dijkstra's algorithm is a **greedy algorithm** because at each stage of the problem, the algorithm works by taking the best action for that stage of the problem. When this is done, the result is the shortest path from the starting node to each ending node.

Before beginning the algorithm, there are a few important things to remember:
- The word *permanent* is used to indicate when a node is "finished." A permanent node will be colored in.
- The word *temporary* is used when the node is in the "working stage."
- Throughout the algorithm, each node will be assigned a label with the following notation: [*distance value*, *preceding node*], such as [15, *C*].

Step 1: Determine the starting node. Assign this node the permanent label [0, *S*] and color the node. Because this is the starting node, there is no distance value and no preceding node. Therefore, the proper notation for the starting node's label is always [0, *S*], where *S* indicates the starting node.

In the medical supplies example, the starting node is the warehouse in Raleigh. In Figure 8.2.3, this node is labeled and colored.

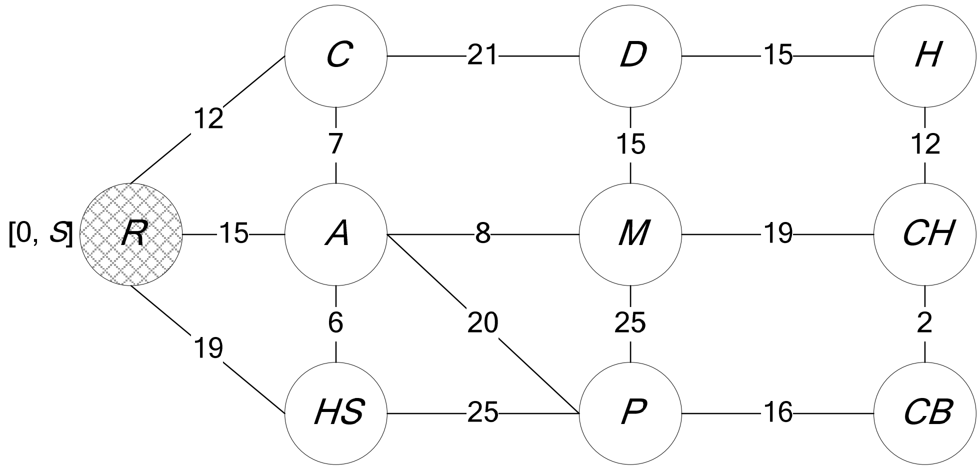


Figure 8.2.3: Identifying and labeling the starting node

Step 2: Assign temporary labels for the nodes that can be reached directly from the starting node. Each of these temporary labels will be in the form [*distance value*, *S*], where the *distance value* refers to the distance from the starting node to this node.

In Figure 8.2.3, there are three nodes that are directly connected to the starting node. In other words, there are three medical offices (Cary, Apex, and Holly Springs) that can be directly reached from the warehouse in Raleigh. In Figure 8.2.4, these three nodes have been assigned temporary labels.

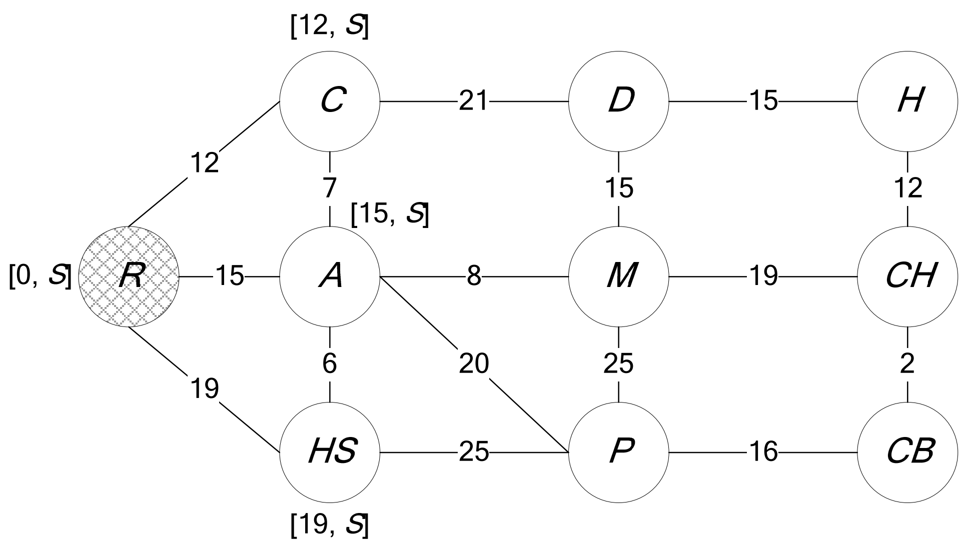


Figure 8.2.4: Computing temporary labels

Step 3: Identify the temporarily labeled node with the smallest distance value, and declare that node permanently labeled by coloring it in.
Note: If there is more than one with the same distance, you can choose either one.

In the medical supplies example, the node with the smallest distance value is *C*, which has a distance value of 12 miles. That is, Cary is the closest office to the supplies warehouse. Therefore, this node should be permanently labeled, as shown in Figure 8.2.5.

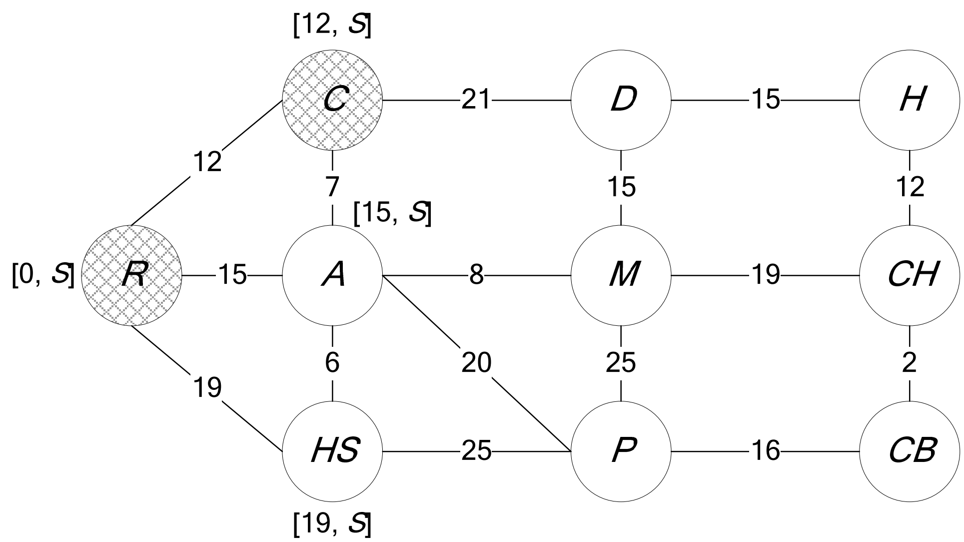


Figure 8.2.5: Changing temporarily labeled nodes to permanently labeled nodes

Step 4: Consider all nodes that are not permanently labeled and can be reached directly from the new permanently labeled node (identified in Step 3). Assign temporary labels as appropriate.
Assign labels as follows:

A. If the node does not yet have a temporary label, assign a temporary label with the following sum to as the distance value:
[distance value at the new permanently labeled node]
+ [direct distance from the new permanently labeled node to the node in question]

In the medical supplies example, the new permanently labeled node from Step 3 is node *C* (Cary). The nodes that can be directly reached from node *C* are node *D* (Durham) and node *A* (Apex). Node *D* does not have yet have a temporary label.

Since Ms. Clark is interested in the distance starting from the warehouse in Raleigh, she needs to determine the total distance from Raleigh to Durham. Therefore, she adds the distance value from Cary to Durham to the distance value from Raleigh to Cary: 21 + 12 = 33 miles. Therefore, node *D* should be temporarily labeled [33, *C*] because it takes 33 miles to get to Durham via Cary, as shown in Figure 8.2.6.

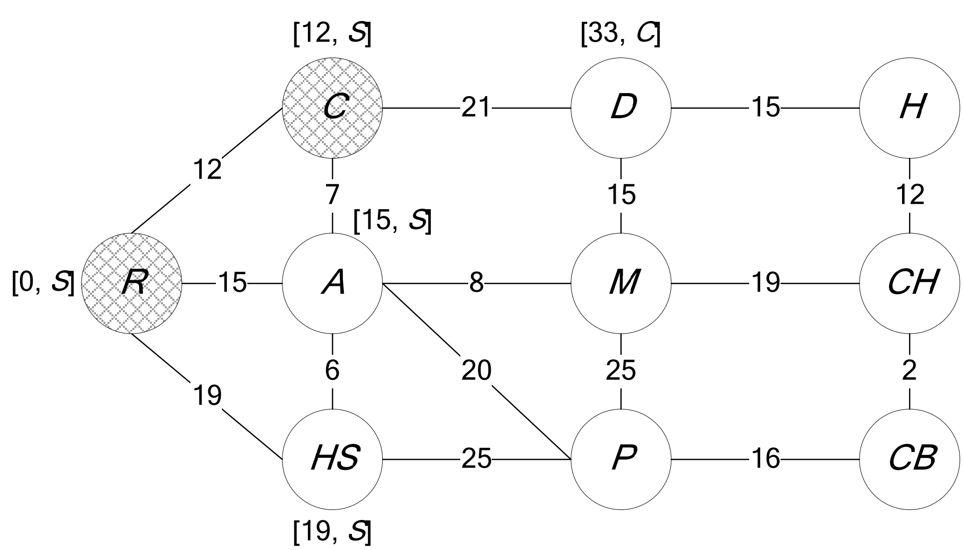


Figure 8.2.6: Computing temporary labels

B. If the node already has a temporary label, compute the same sum as in part A ([the distance value at the new permanently labeled node] + [the direct distance from the new permanently labeled node to the node in question]) and consider the following:
   1. If the sum just computed is equal to or greater than the distance value of the node in question, do nothing.
   2. If the sum just computed is less than the distance value already listed for the node in question, do the following:
      a. Change the distance value for the node in question to make it equal to the sum just computed.
      b. Change the "preceding node value" for the node in question to the letter of the new permanently labeled node.

Node *A* already has a temporary label. Therefore, Ms. Clark adds the distance from Raleigh to Cary to the distance from Cary to Apex: 12 + 7 = 19 miles. Since 19 > 15, she will do nothing. In other words, if the driver at Triangle Medical Supplies travels directly to Apex from Raleigh, it will take 15 miles. However, if the driver went through Cary first, it would take 19 miles. Since 19 is greater than 15, there is no reason for the driver to go through Cary. Thus, the temporary label for node *A* remains [15, *S*], as shown in Figure 8.2.6.

Repeat Step 3 and Step 4 until all nodes are permanently labeled.
If all nodes are permanently labeled, go to step 5.

Repeat Step 3
In the medical supplies example, the temporarily labeled node with the smallest distance value is *A*, which has a distance value of 15 miles. In other words, Apex is the next closest office to the supplies warehouse in Raleigh. Figure 8.2.7 shows this node permanently labeled.

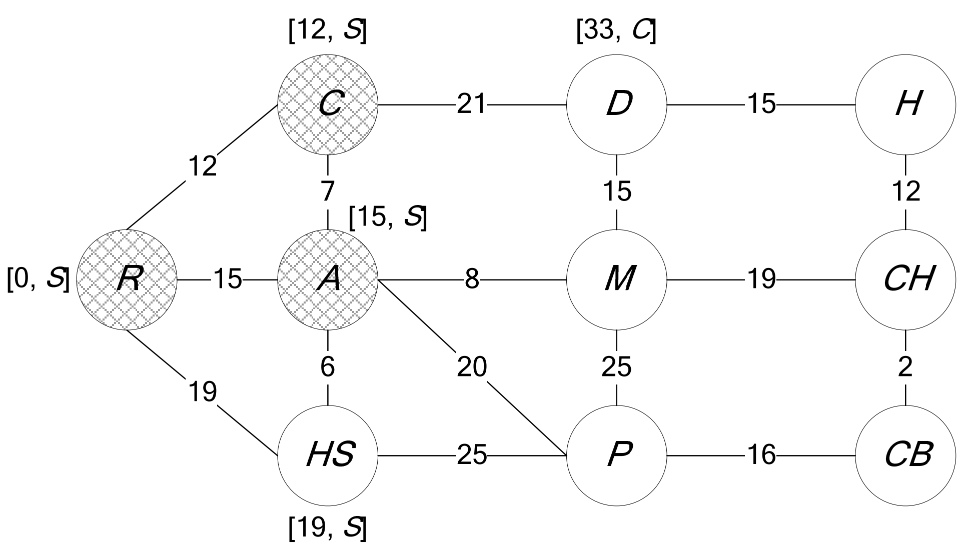


Figure 8.2.7: Changing temporarily labeled nodes into permanently labeled nodes

Repeat Step 4
The new permanently labeled node is node *A* (Apex). The temporarily labeled nodes that can be directly reached from node *A* are nodes *HS*, *M*, and *P*. Neither node *M* nor node *P* has temporary labels yet, so they are calculated by adding the distance value at *A* to the direct distance from *A*, as shown in Figure 8.2.8.

Node *HS* already has a temporary label. Computing the sum from Apex to Holly Springs, Ms. Clark finds this distance value to be 15 + 6 = 21 miles. Since the current temporary label has a distance value of 19 miles, she will not replace this label with a new one. In other words, the distance directly from Raleigh to Holly Springs is shorter (19 miles) than the distance from Raleigh to Apex to Holly Springs (21 miles). Therefore, the temporary label on node *HS* will remain [19, *S*], as shown in Figure 8.2.8.

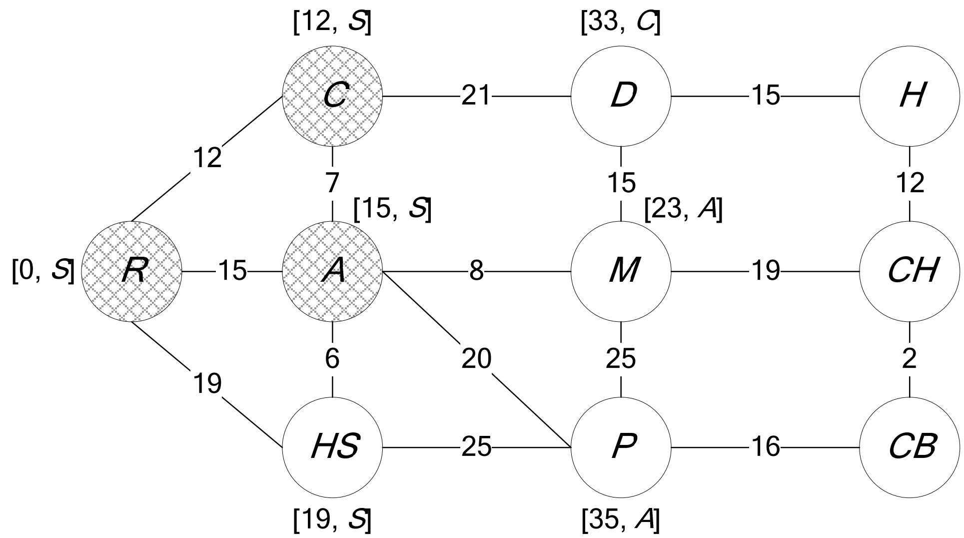


Figure 8.2.8: Computing temporary labels

Repeat Step 3
The temporarily labeled node with the next smallest distance value is node *HS* (Holly Springs). This node is permanently labeled in Figure 8.2.9.

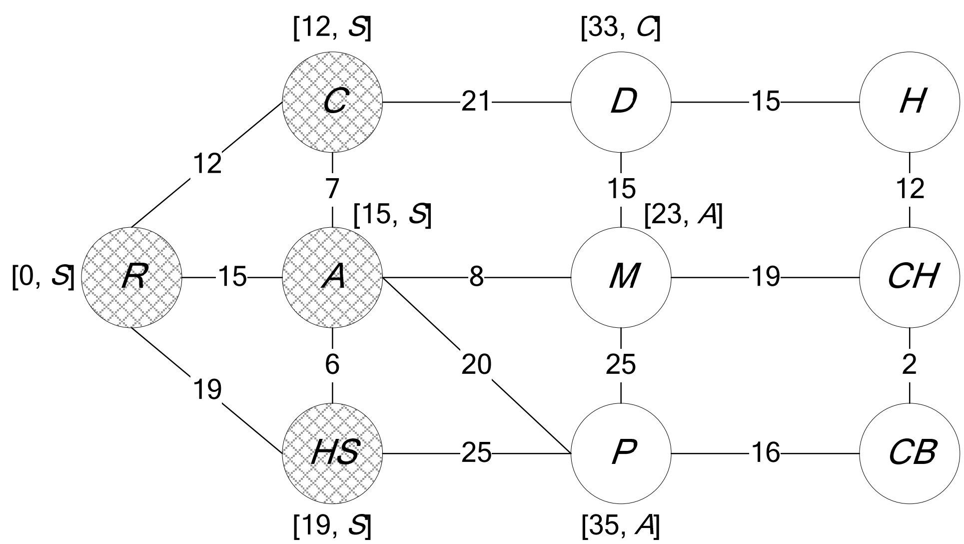


Figure 8.2.9: Changing temporarily labeled nodes into permanently labeled nodes

Repeat Step 4
The only temporarily labeled node directly connected to node *HS* is node *P*. The distance from Raleigh to Pittsboro via Holly Springs is 19 + 25 = 44 miles. Since 44 is greater than 35 miles, this temporary label should not be changed.

Repeat Step 3
The temporarily labeled node with the next smallest distance value is node *M* (Morrisville). This node is permanently labeled in Figure 8.2.10.

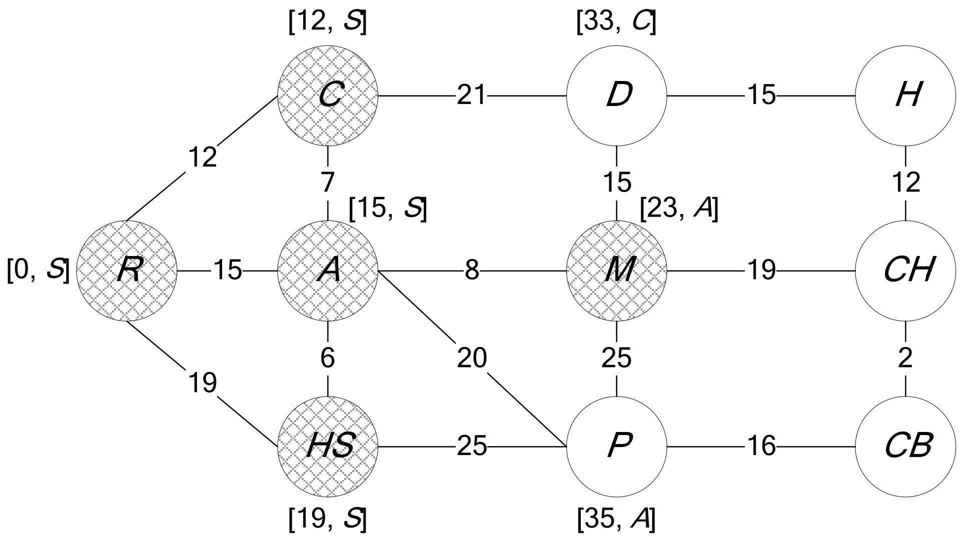


Figure 8.2.10: Changing temporarily labeled nodes into permanently labeled nodes

Repeat Step 4
Three nodes can be reached from Morrisville: Durham, Chapel Hill, and Pittsboro. Chapel Hill is the only node without a temporary label. Ms. Clark finds the distance value for Chapel Hill by adding 23 miles (the distance to Morrisville) to 19 miles (the distance from Morrisville to Chapel Hill) and obtains 42 miles. The temporary label for node *CH* is shown in Figure 8.2.11.

Nodes *D* and *P* already have temporary labels. Ms. Clark computes the total distance to Durham via Morrisville (23 + 15 = 38 miles) and the total distance to Pittsboro via Morrisville (23 + 25 = 48 miles). Neither of these distance values are less than the current distance values, so the temporary labels will be left unchanged, as shown in Figure 8.2.11.

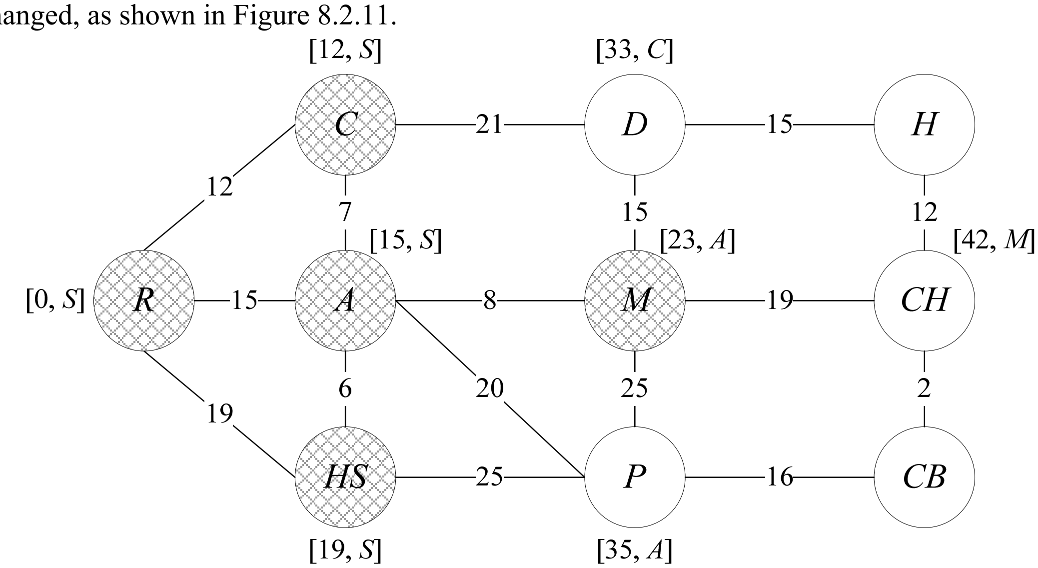


Figure 8.2.11: Computing temporary labels

Repeat Step 3
Next, node *D* (Durham) should be permanently labeled because it has the smallest distance value. This is shown in Figure 8.2.12.

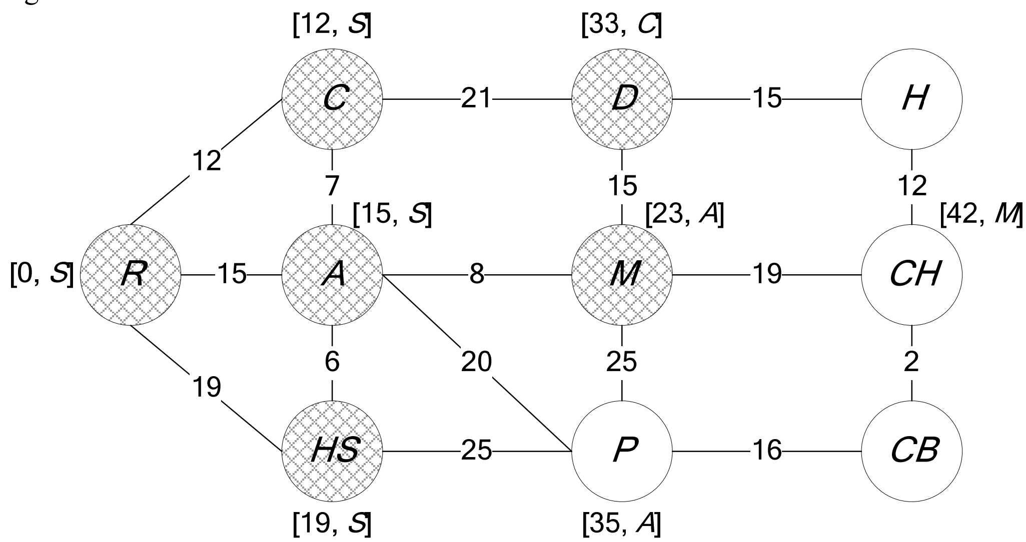


Figure 8.2.12: Changing temporarily labeled nodes into permanently labeled nodes

Repeat Step 4
The only node that can be directly reached from node *D* is node *H* (Hillsborough). Node *D* does not yet have a temporary label. Therefore, node *H* is given a temporary label with the distance value of 33 + 15 = 48 miles, as shown in Figure 8.2.13.

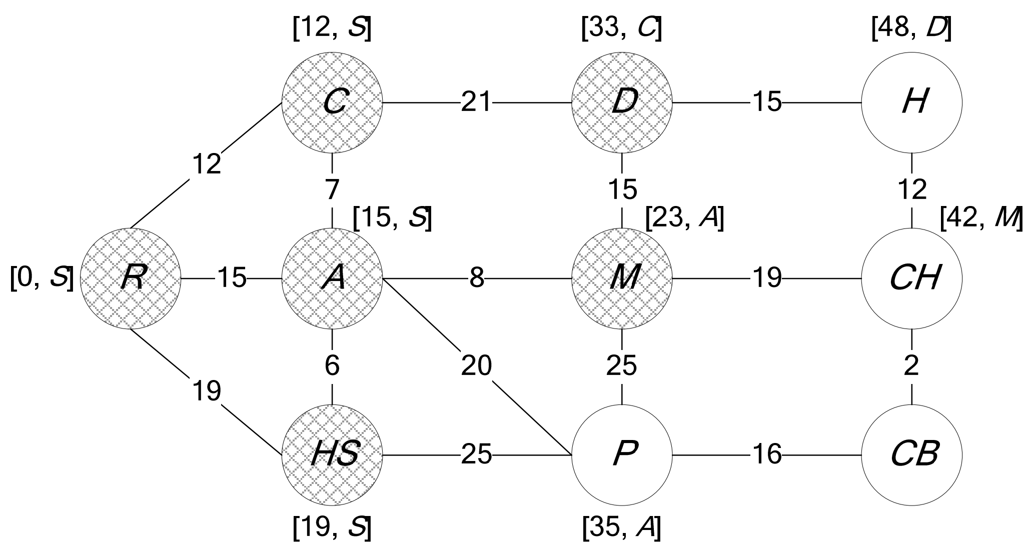


Figure 8.2.13: Computing temporary labels

Repeat Step 3
The next smallest distance value belongs to node *P* (Pittsboro). This node is permanently labeled in Figure 8.1.14.

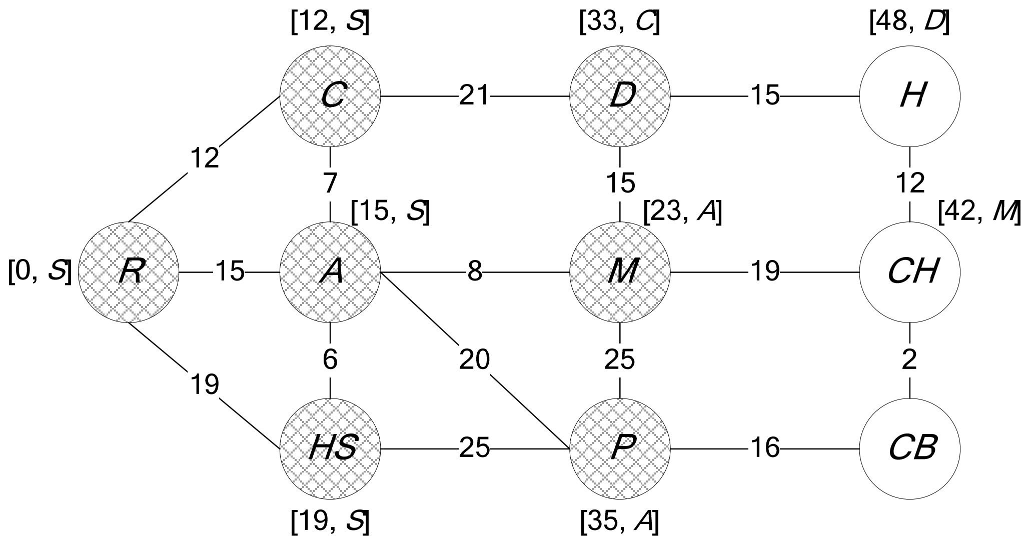


Figure 8.2.14: Changing temporarily labeled nodes into permanently labeled nodes

Repeat Step 4

The only node directly connected to node *P* is node *CB* (Carrboro). This node does not yet have a temporary label. Ms. Clark computes the distance value for this node by adding the distance value of Pittsboro (35 miles) to the distance from Pittsboro to Carrboro (16 miles). The distance value for Carrboro is therefore 51 miles, as shown in Figure 8.2.15.

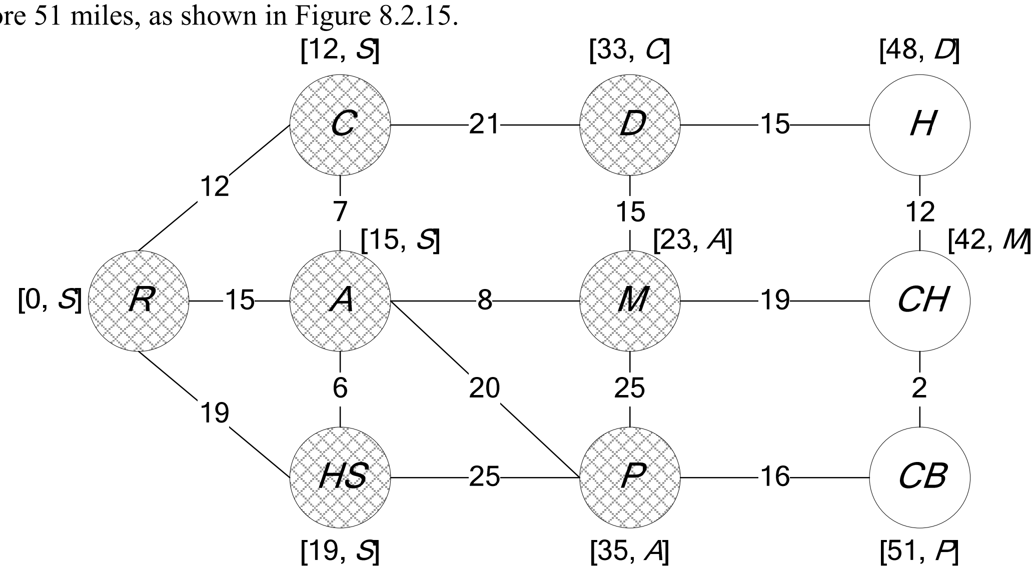


Figure 8.2.15: Computing temporary labels

Repeat Step 3

The next smallest distance value is 42 miles, which belongs to node *CH* (Chapel Hill). Therefore, this node is permanently labeled, as shown in Figure 8.2.16.

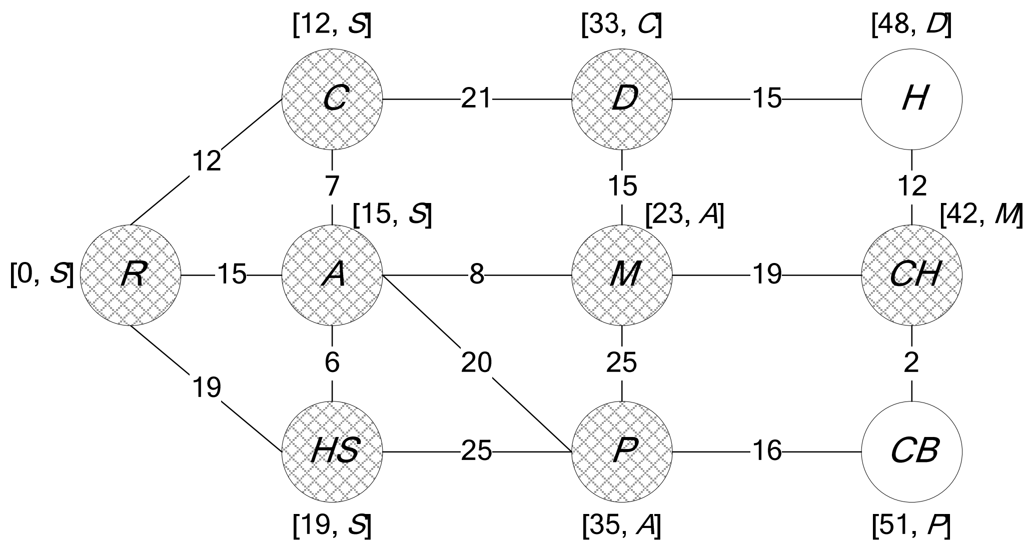


Figure 8.2.16: Changing temporarily labeled nodes into permanently labeled nodes

Repeat Step 4
From Chapel Hill, both Hillsborough and Carrboro can be directly reached. Currently, the distance value for Hillsborough is 48 miles. If the driver came from Chapel Hill, the distance value would instead be 42 + 12 = 54 miles. Since this is greater than 48, this temporary node is not changed.

However, the current distance value for Carrboro is 51 miles. If the driver drove through Chapel Hill, rather than through Pittsboro, to get to Carrboro, the distance value would instead be 42 + 2 = 44 miles. Recall from Step 4B that if this sum is less than the distance value already listed, then the distance value should be changed to this sum (44 miles), and the preceding node value should be changed to the letter of the new permanently labeled node (*CH*). These changes are shown in Figure 8.2.17.

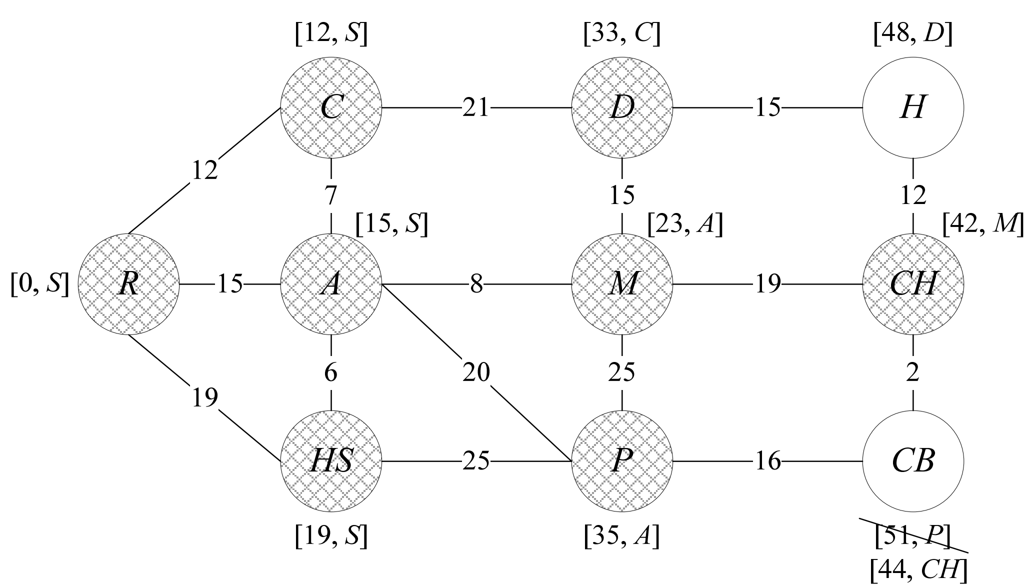


Figure 8.2.17: Computing temporary labels

Repeat Step 3
The node with the next smallest distance value is *CB* (Carrboro). Therefore, this node is permanently labeled, as shown in Figure 8.2.18.

Since no temporarily labeled nodes can be directly reached from node *CB*, Step 4 is skipped and Step 3 is repeated again. This time, node *H* (Hillsborough) is permanently labeled. Figure 8.2.18 shows all nodes are now permanently labeled.

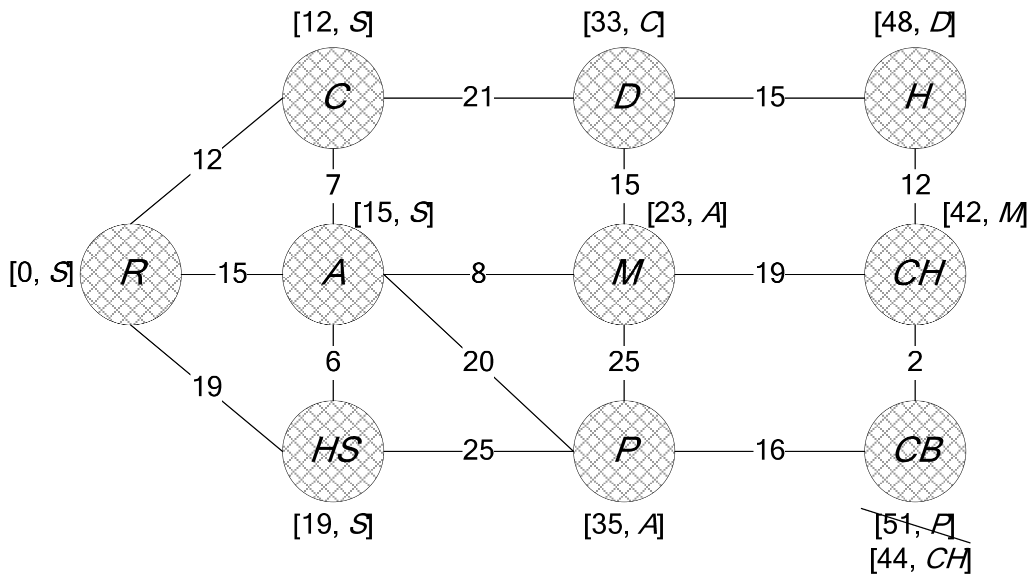


Figure 8.2.18: Graph with permanent labels for all nodes

Step 5: Identify the shortest path from the starting node to each node.
The permanent labels show the shortest distance from the starting node to each node and the preceding node on the shortest route. The shortest path to a given node can be found by starting at the given node and moving to its preceding node. Continuing this backward movement through the graph provides the shortest route from the starting node to the node in question.

For example, in Figure 8.2.18, the shortest distance from Raleigh to Pittsboro is 35 miles. Before arriving at Pittsboro, the supplies driver must go through Apex. Therefore, the shortest path from Raleigh to Pittsboro is *R*-*A*-*P* (Raleigh to Apex to Pittsboro).

Q3. Using Figure 8.2.18, find the shortest path and the total distance from the supplies company in Raleigh (*R*) to:

a. Chapel Hill (*CH*).
b. Carrboro (*CB*).
c. Hillsborough (*H*).

It seems like using Dijkstra's algorithm is a long process, but using it assures that the minimum distance from the starting node to each of the other nodes is found. Dijkstra's algorithm works for any other situation like this, and, if necessary, it can be done on a computer.

The steps of Dijkstra's algorithm are summarized below.

Dijkstra's Algorithm

Step 1: Determine the starting node. Assign this node the permanent label [0, *S*] and color the node. Because this is the starting node, there is no distance value and no preceding node. Therefore, the proper notation for the starting node's label is always [0, *S*], where *S* indicates the starting node.

Step 2: Assign temporary labels for the nodes that can be reached directly from the starting node. Each of these temporary labels will be in the form [*distance value*, *S*], where the *distance value* refers to the distance from the starting node to this node.

Step 3: Identify the temporarily labeled node with the smallest distance value, and declare that node permanently labeled by coloring it in. If all nodes are permanently labeled, go to step 5. Note: If there is more than one with the same distance, you can choose either one.

Step 4: Consider all nodes that are not permanently labeled and can be reached directly from the new permanently labeled node (identified in Step 3). Then, assign labels as follows:

A. If the node does not yet have a temporary label, compute the following sum to determine the distance value:
[distance value at the new permanently labeled node]
+ [direct distance from the new permanently labeled node to the node in question]

B. If the node already has a temporary label, compute the same sum as in part A ([the distance value at the new permanently labeled node] + [the direct distance from the new permanently labeled node to the node in question]) and consider the following:

- **1.** If the sum just computed is equal to or greater than the distance value of the node in question, do nothing.
- **2.** If the sum just computed is less than the distance value already listed for the node in question, do the following:
  - **a.** Change the distance value for the node in question to make it equal to the sum just computed.
  - **b.** Change the "preceding node value" for the node in question to the letter of the new permanently labeled node.

Repeat Step 3 and Step 4 until all nodes are permanently labeled. Then, proceed to Step 5.

Step 5: Identify the shortest path from the starting node to each node.

8.2.2 New Warehouse Location

The president of the Triangle Medical Supplies Company, Mr. Kudzik, wants to build a new warehouse next to one of their existing customers. He asks Ms. Clark, the manager, to determine the best location for the new warehouse. Mr. Kudzik wants the new warehouse to be as close as possible to all of their customers in the area so that the distances between the warehouse and the doctors' offices are minimized.

In order to solve this problem, Ms. Clark needs to find the shortest path from each doctor's office to every other doctor's office. The driver will not need to travel from the new warehouse to the existing warehouse (in Raleigh); thus, the graph only needs to include nodes for the doctors' offices, as shown in Figure 8.2.19.

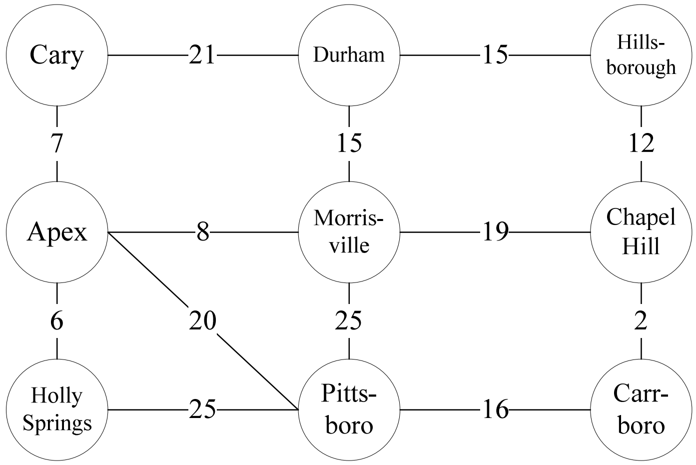


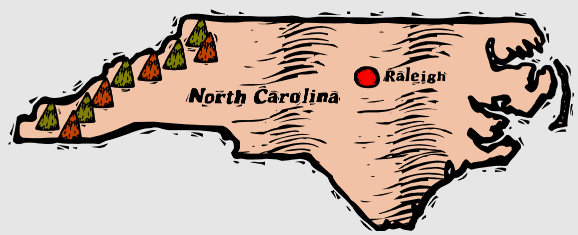
Figure 8.2.19: The major clients of the medical supplies company and the distances between them

There are two ways for Ms. Clark to think about this problem. First, she could locate the new warehouse at the doctor's office that minimizes the maximum distance to all the other offices. That is, she could find the office that, if the new warehouse was located there, the driving distance to all other offices is as small as possible. In terms of the graph, the node that *minimizes the maximum distance* to all the other nodes is called the **center location**.

Second, Ms. Clark could locate the new warehouse at the doctor's office that minimizes the average distance to all the other offices. In other words, she could find the office such that the average driving distance to each other is as small as possible. In terms of the graph, the node that is *on average closest to all of the other nodes* is called the **median location**.

Median Location

In the previous chapter, *median location* referred to the first location where the cumulative weight up to that point is at least half of the total weight of all other locations. In this chapter, *median location* has a similar but different definition; it refers to the location that is, on average, closest to the other locations.



These two locations, center and median, may or may not be the same, but they provide two different ideas of the "middle" of the graph.

To find the center and median location, Dijkstra's algorithm must be employed repeatedly. Ms. Clark uses Dijkstra's algorithm to find all the shortest distances between every node and every other node. The completed graphs are given in Figures 8.2.20, 8.2.21, and 8.2.22, where the starting nodes are Cary, Apex, and Holly Springs, respectively.

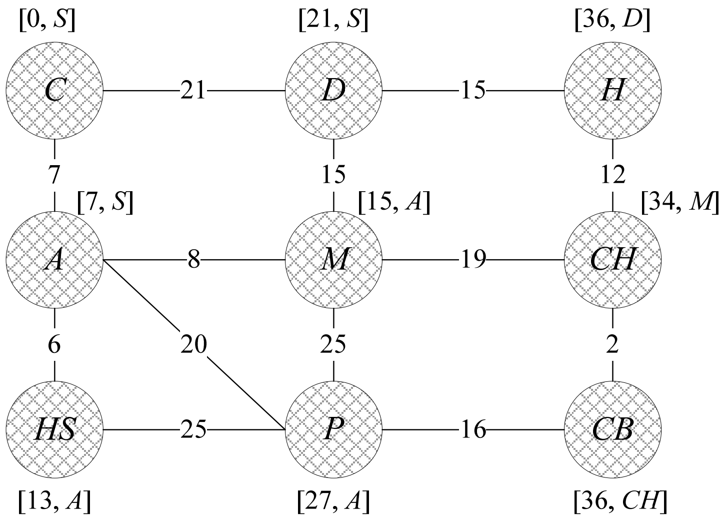


Figure 8.2.20: Final version of the graph, with Cary as the starting node

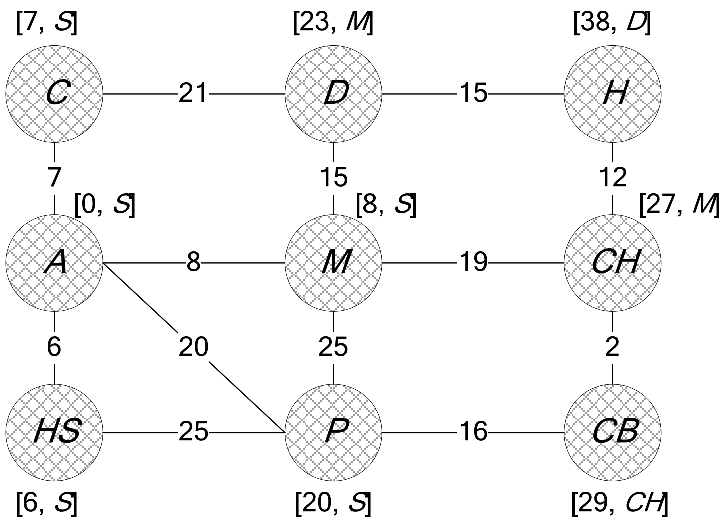


Figure 8.2.21: Final version of the graph, with Apex as the starting node

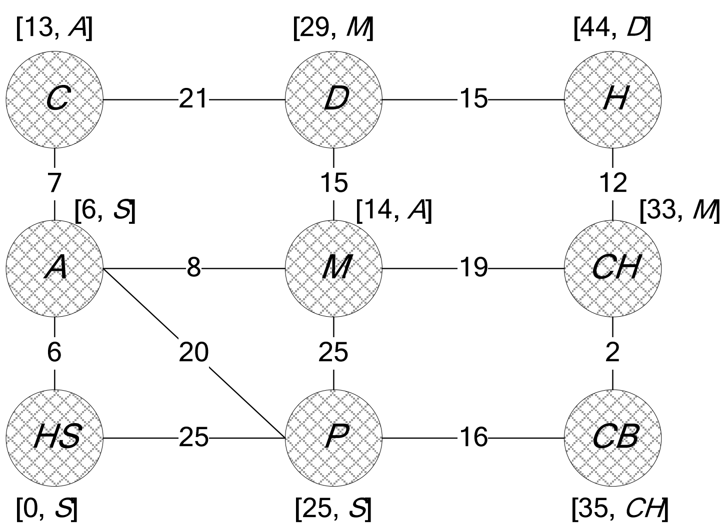


Figure 8.2.22: Final version of the graph, with Holly Springs as the starting node

The first few distances are shown in Table 8.2.1, where the leftmost column represents the starting node and the top row represents the ending node. The maximum distance and the average distance for each row are also given.

For example, if the new warehouse is located in Cary, the maximum distance from Cary to each of the offices is 36 miles (the distance from Cary to Hillsborough and also from Cary to Carrboro). The average distance from Cary to each of the offices is

$$\frac{0+7+13+21+15+27+36+34+36}{9} = 21 \text{ miles.}$$

	C	**A**	**HS**	**D**	**M**	**P**	**H**	**CH**	**CB**		**Max**	**Average**
C	0	7	13	21	15	27	36	34	36		36	21
A	7	0	6	23	8	20	38	27	29		38	17.56
HS	13	6	0	29	14	25	44	33	35		44	22.11
D												
M												
P												
H												
CH												
CB												

Table 8.2.1: The shortest paths from each office to every other office

Q4. Use Dijkstra's algorithm to complete Table 8.2.1 for the remaining offices.

Q5. What are some patterns you notice in the table?

Q6. Which office represents the center location? How do you know?

Q7. Which office represents the median location? How do you know?

Q8. Based on this information, where should Ms. Clark decide to locate the new warehouse? Why?

Q9. Suppose the doctor's office in Hillsborough has moved. Now, the distance from Durham to Hillsborough is 20 miles, and the distance from Chapel Hill is Hillsborough is 17 miles, as shown in Figure 8.2.23. Do you think this change would impact the optimal location of the new warehouse? Why or why not?

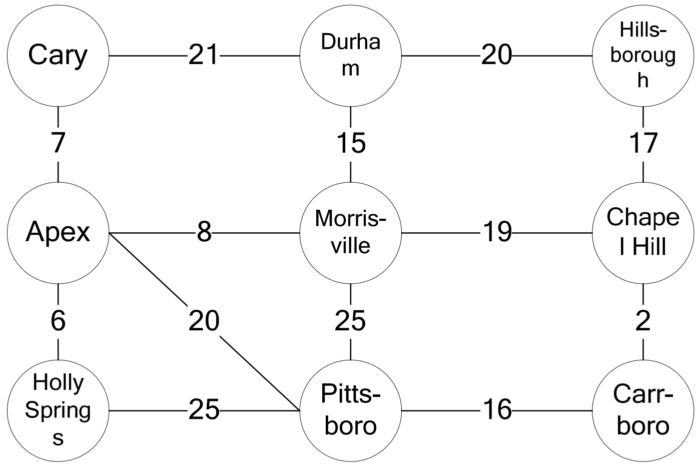


Figure 8.2.23: The major clients and the distances between them, with a new Hillsborough office

In this section, Ms. Clark first needed to determine the shortest path from the medical supplies warehouse in Raleigh to each of the doctor's offices in the area. She used Dijkstra's algorithm to determine these paths.

Then, Ms. Clark needed to find the best location for a new warehouse, next to one of the existing clients. She utilized Dijkstra's algorithm repeatedly to determine the locations that (1) minimized the maximum distance traveled and (2) minimized the average distance traveled.

In the next section, Dijkstra's algorithm is used again to find the shortest path, the center location, and the median location. However, rather than finding the distances between locations, the algorithm is used to determine how quickly a rumor would spread in a group of people.

Section 8.3: How Quickly do Rumors Spread?

Suppose there are 10 people in a class, and someone starts a rumor. Rumors tend to spread quickly, but how quickly they spread depends on how often people communicate with one another. In this section, Dijkstra's algorithm is used to determine how long it takes for a rumor to spread.

In the graph below (Figure 8.3.1), the nodes represent people and the arcs represent how often (in minutes) they communicate with each other. For example, the maximum amount of time before person *A* speaks with, sends a text message to, calls, or emails person *B* is 5 minutes, so the edge connecting *A* to *B* has a weight of 5. The goal is to examine how quickly the rumor will spread throughout the class.

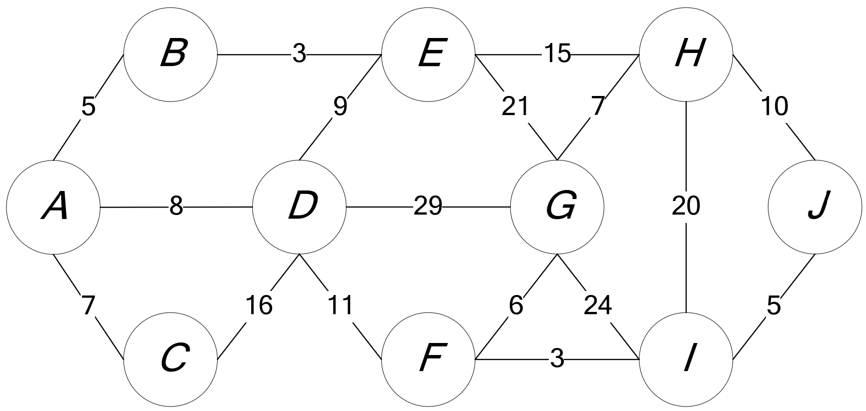


Figure 8.3.1: A graph representing a social network, where the nodes represent people and the arcs represent how often the people communicate with each other (in minutes)

Suppose person *A* begins a rumor. To determine how long it will take for the rumor to reach the most distant member in the class from *A*, the shortest path from person *A* to each person in the class must be found. Using Dijkstra's algorithm, the shortest paths from *A* to each node is given in Figure 8.3.2.

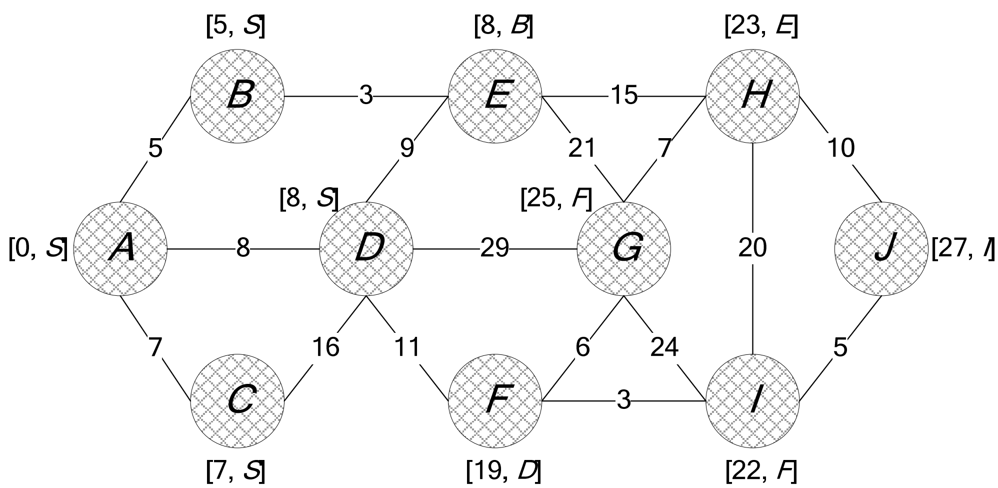


Figure 8.3.2: Final version of the graph where person *A* started the rumor

Q1. How long will it take for everyone in the class to hear the rumor? How do you know?

Q2. What is the average amount of time it takes for each member to first hear the rumor if person *A* starts the rumor?

Q3. How long would it take the rumor to spread if person *J* started it instead of person *A*? That is, find the shortest path from *J* to each node and determine which shortest path has the greatest value.

Q4. What is the average amount of time it takes for each member to first hear the rumor if person *J* starts the rumor?

Center and median locations can be used to determine which individual would need to start the rumor in order for it to spread as quickly as possible. Just like in the medical supplies problem, the center location refers to the node that minimizes the maximum distance to all the other nodes; the median location refers to the node that minimizes the average distance to all the other nodes.

Q5. Explain center and median location in terms of the rumor example.

Dijkstra's algorithm can be used to find all the shortest distances between every node and every other node These distances are given in Table 8.3.1.

	A	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**J**		**Max**	**Average**
A	0	5	7	8	8	19	25	23	22	27		27	14.4
B	5	0	12	12	3	23	24	18	26	28		28	15.1
C	7	12	0	15	15	26	32	30	29	34		34	20
D	8	12	15	0	9	11	17	24	14	19		24	12.9
E	8	3	15	9	0	20	21	15	23	25		25	13.9
F	19	23	26	11	20	0	6	13	3	8		26	12.9
G	25	24	32	17	21	6	0	7	9	14		32	15.5
H	23	18	30	24	15	13	7	0	15	10		30	15.5
I	22	26	29	14	23	3	9	15	0	5		29	14.6
J	27	28	34	19	25	8	14	10	5	0		34	17

Table 8.3.1: The shortest paths from each person to every other person

Q6. Determine which individual's node is the center. What does this mean in terms of the problem?

Q7. Determine which individual's node is the median. What does this mean in terms of the problem?

Q8. If you were in this class and you wanted a rumor to spread quickly, who would you want to start the rumor? Why?

Q9. Suppose person *D* is absent. Determine the center and median locations in this new scenario (see Figure 8.3.3).

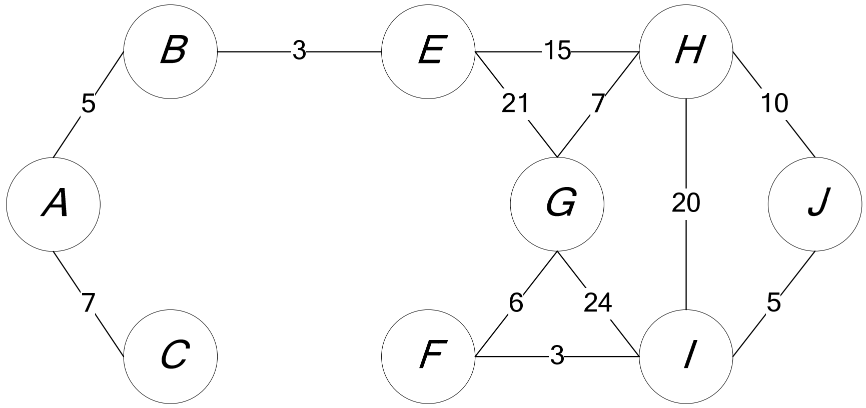


Figure 8.3.3: A graph representing a social network, with person *D* absent

Although this rumor example was quite different from the medical supplies example, both could be solved using Dijkstra's algorithm. Furthermore, both used the ideas of center and median to get an idea about which node is the "middle" of the graph.

The concepts discussed in this chapter (i.e., minimum spanning tree, shortest path, center, and median) give an introduction into the large field known as graph theory.

Section 8.4: Chapter 8 (Graph Theory) Homework Questions

1. Find the Bacon number, using imdb.com, of the following actors/actresses. Create a graph with at least **three** paths from the actor/actress to Kevin bacon; write the movie and actor/actress for each connection.
    a. Jodie Foster
    b. Natalie Portman
    c. Harrison Ford
    d. Nicole Kidman

2. A company has a business with several offices in the New York City area. The owner needs to set up phone lines connecting the offices. The phone company charges the company based on distances of the phone line connections. The following mileage chart shows the distances between each of the offices. Using this chart, create a tree representing the offices and the distance between them. Then use Kruskal's algorithm to find the minimum spanning tree connecting the offices.

	Manhattan	Bronx	Brooklyn	Queens	Newark, NJ
Manhattan	**	8.1	8.0	10.2	13.4
Bronx	8.1	**	15.6	14.0	21.8
Brooklyn	8.0	15.6	**	14.1	13.9
Queens	10.2	14.0	14.1	**	19.8
Newark, NJ	13.4	21.8	13.9	19.8	**

Table 8.4.1: Distances between cities in miles

3. The figure below represents a plan for a network of streets. Each node represents a place that needs water, i.e. a residence, a fire hydrant, a commercial property, etc., and the distance between nodes is measured in miles. To minimize the cost of laying down pipes, use Kruskal's algorithm to find the minimum spanning tree connecting the nodes.

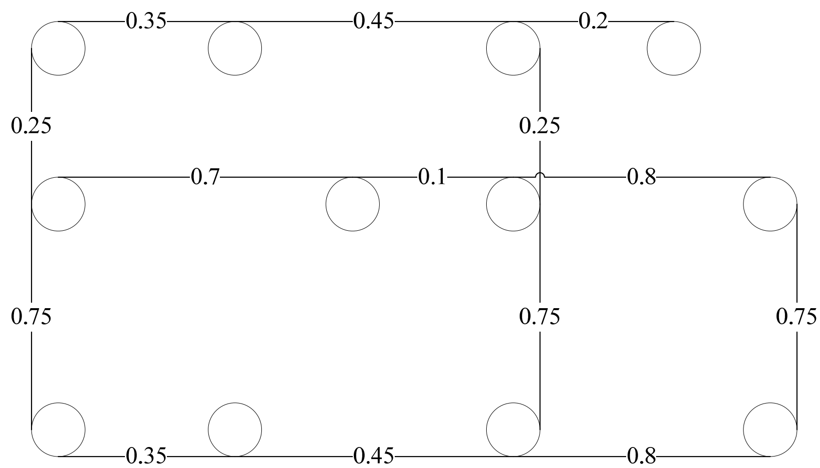


Figure 8.4.1: Network of streets

4. Here is another medical supply graph. The *O* is the warehouse, and the other letters are locations needing emergency supplies. The numbers on the arcs represent miles. Using Dijkstra's algorithm, we can create the final version below it. Identify the shortest path as identified for each of the locations, *A*, *B*, *C*, *D*, *E*, and *T*.

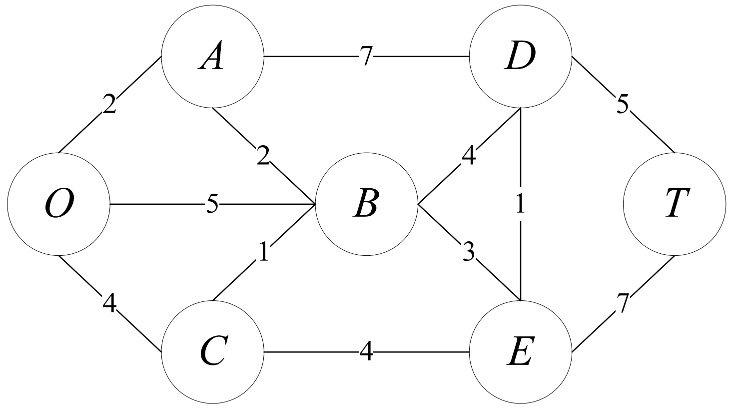


Figure 8.4.2: Graph for example medical supply problem

a. Find the Shortest Paths and the distances from O to each of the medical supply houses.

Where to	Path	Shortest Distance in miles
To A		
To B		
To C		
To D		
To E		
To T		

b. Would the answer for the shortest path to T change if the distance from E to T were 3 instead of 7? How would it change?

c. Would the answer for the shortest path to E change in this case? How would it change?

5. Steve would like to visit some of his friends at college. Due to money and time constraints, he can only travel 150 miles on any given trip. Use Dijkstra's algorithm on the graph below to find the shortest distance between Steve's home and each of his friends' colleges, where the nodes are labeled with various colleges and the arcs are labeled with the distance between the colleges, in miles. Based on this graph, which colleges could he visit? Which of these trips include more than one stop?

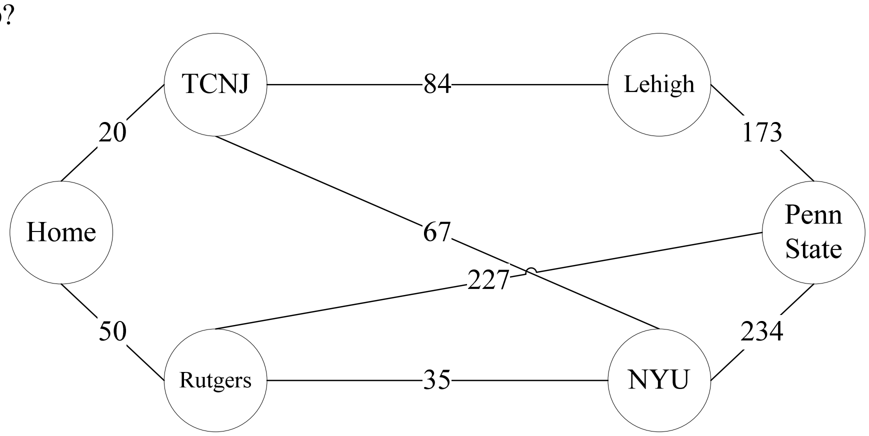


Figure 8.4.3: Distances between colleges

6. Below is a graph of a neighborhood. The nodes represent houses and the arcs represent roads. The weights on the arcs represent the length of time it takes to walk from one house to the next. Rochelle needs to take her little sister trick-or-treating in the neighborhood. The node labeled T represents the house the gives the best and the most candy. Use Dijkstra's algorithm to find the shortest path from home to node T. How many houses will they visit along the way?

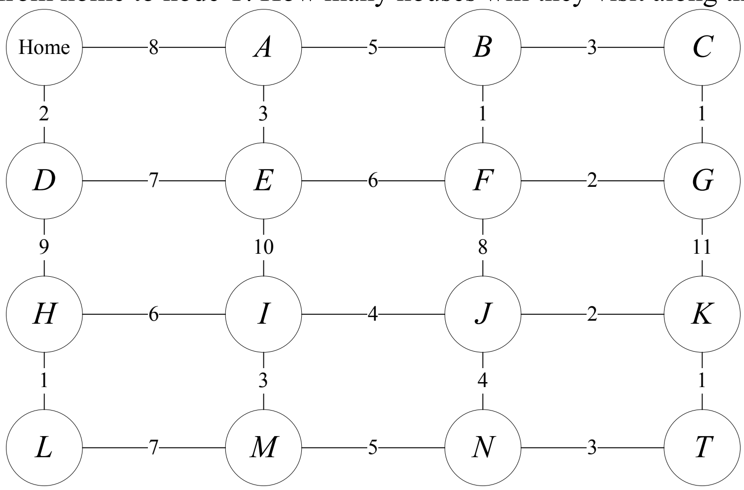


Figure 8.4.4: Graph of a neighborhood

7. Use Dijkstra's algorithm to find the shortest path from each node to every other node in the graphs below. Then determine the median and the center for each graph.
   a.

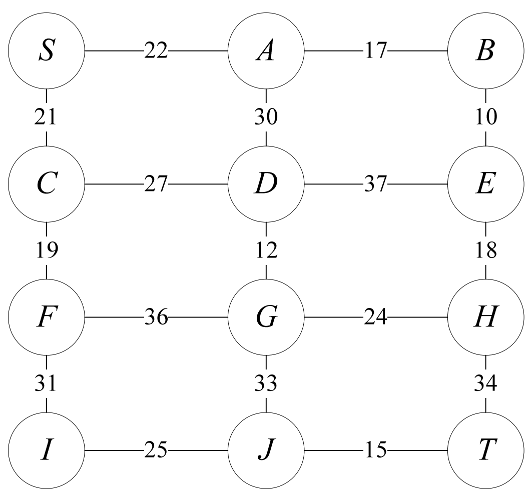


Figure 8.4.5: Dijkstra's algorithm example

b.

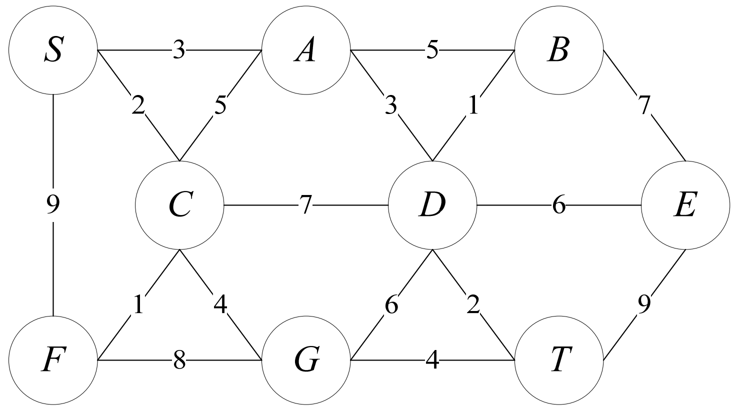


Figure 8.4.6: Dijkstra's algorithm example

Terms

Algorithm	A step-by-step process for solving a problem
Arc	A line segment on a graph
Bacon Number	The minimum number of degrees of separation between two individuals when playing the six degrees of Kevin Bacon game
Center Location	The node in a graph that minimizes that maximum distance to all other nodes in the graph
Circuit	A path in a graph that begins and ends at the same node
Dijkstra's Algorithm	An algorithm created by Edsger Dijkstra that finds the shortest path between two points in a graph
Graph	A set of nodes and arcs
Graph Theory	The study of graphs
Greedy Algorithm	A method of finding an optimal solution, where at each step, the best action is taken for that step of the problem
Kruskal's Algorithm	An algorithm created by Joseph Kruskal that finds the minimum spanning tree of a given graph
Median Location	The node in a graph that is on average closest to all of the other nodes in the graph
Minimum Spanning Tree	The spanning tree of a graph with the least amount of total weight
Node	Points on a graph
Path	A connection between two nodes along arcs
Shortest Path Problems	Problems where the goal is to find the shortest path between two nodes in a graph
Six Degrees of Kevin Bacon	A game in which actors are connected, through movies, to the actor Kevin Bacon with as few connections as possible; this game is an application of the "six degrees of separation" idea
Six Degrees of Separation	The idea that, through social networks, every person is connected to every other person by an average of six people
Spanning Tree	A tree that connects all of the nodes

Tree	A graph with no circuits
Weights	A number assigned to an arc that represents the cost, distance, or time related to that arc

Chapter 8 (Graph Theory) Objectives

You should be able to:

- Investigate and define graphs, arcs, nodes, networks, spanning trees, and weights and be able to identify them in a contextual problem.

- Given a contextual problem, find the minimum spanning tree using Kruskal’s algorithm and analyze and make decisions based on the results.

- Given a contextual problem, find the shortest path using Dijkstra’s algorithm and analyze and make decisions based on the results.

References

Anderson, D. R., Sweeney, D. J., & Williams, T. A. (1991). *An introduction to management science: Quantitative approaches to decision making* (6th ed.). St. Paul, MN: West Publishing Company.

Films And TV. (2010). Retrieved January 12, 2011, from http://www.filmsandtv.com/

The Internet Movie Database (IMDb). (2011). Retrieved January 12, 2011, from http://www.imdb.com/

Reynolds, P. (2010). The Oracle of Bacon. Retrieved January 12, 2011, from http://oracleofbacon.org/

SixDegrees.org | It's a small world. You can make a difference. (n.d.). Retrieved January 12, 2011, from http://www.sixdegrees.org/