3 GRAPH THEORY INTRODUCTION UNIT

3.1 Introduction

A **graph** G is a finite nonempty set of objects called **vertices** together with a set of unordered pairs of distinct vertices called **edges**. We denote the vertex set of a graph G by $V(G)$ and the edge set by $E(G)$. The number of elements in the vertex set of a graph G is called the **order** of G, denoted n, and the number of elements in the edge set of a graph G is called the **size** of G, denoted m. A pair of vertices v_i and v_j in V are **adjacent** if they are connected by an edge; otherwise, v_i and v_j are **nonadjacent**. The **degree of v**, denoted $\mathbf{deg}(v)$, is the number of vertices adjacent to v. Note that a vertex of degree zero is called an **isolated vertex**. The **minimum degree** of G, denoted $\delta(G)$, is the minimum degree among the vertices of G and the **maximum degree** of G, denoted $\Delta(G)$, is the maximum degree among the vertices of G. A vertex u is said to be **connected** to a vertex v in a graph G if there exists a sequence of edges from u ro v in G. A graph G is **connected** if every two of its vertices are connected.

3.2 In Class Examples

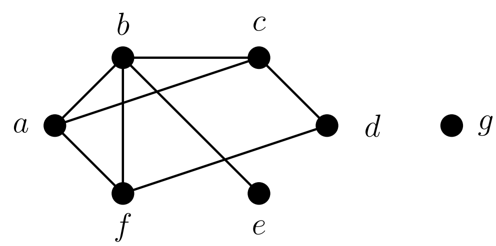


Figure 3: A graph G

1. List the vertex set and edge set of G.

2. List the degrees of the vertices of G.

3. What is the minimum degree of G?

4. What is the maximum degree of G?

5. Is the graph G connected?

Solution

1. $V(G) = \{a, b, c, d, e, f, g\}$ and $E(G) = \{ab, ac, af, bf, be, bc, cd, df\}$

2. The degrees are as follows: $\deg(a) = 3$, $\deg(b) = 4$, $\deg(c) = 3$, $\deg(d) = 2$, $\deg(e) = 1$, $\deg(f) = 3$, and $\deg(g) = 0$.

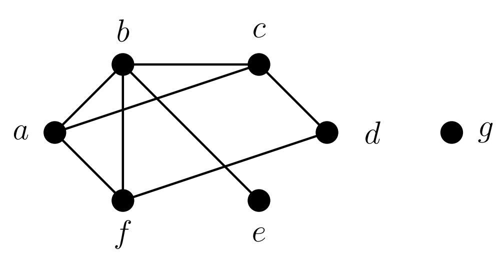
3. The minimum degree of G is 0.

4. The maximum degree of G is 4.

5. No, the graph G is not connected. The vertex g is not connected to any vertex in the graph.

For a graph G with vertex set $V(G)$ and edge set $E(G)$, we call a graph H a **subgraph** of G if the vertex set $V(H)$ and edge set $E(H)$ are subsets of $V(G)$ and $E(G)$ where each edge $e = uv$ is in $E(H)$, and both u and v are in $V(H)$. We can obtain H from G by deleting edges and/or vertices from G. Note that when you remove each vertex v, all edges incident with v must also be removed.

Example



A graph G

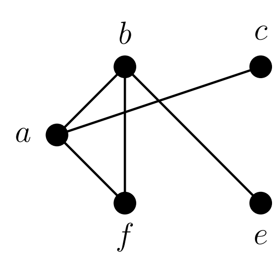


Figure 4: A graph H; H is a subgraph of G.

A graph is called a **complete** graph if every two of its vertices are adjacent, denoted K_n.

Examples

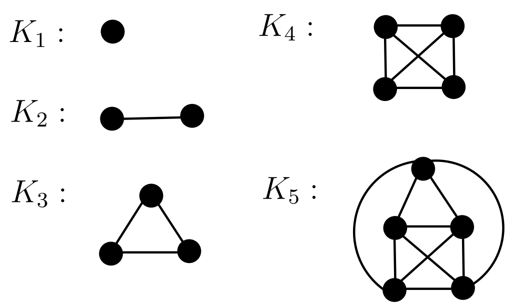


Figure 5: Complete Graphs

A **path** is graph in which all but two vertices have degree 2, and the other two vertices have degree 1, denoted P_n.

Examples

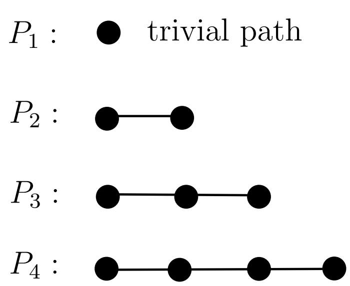


Figure 6: Paths

A **cycle** is a graph in which every vertex has degree 2, denoted C_n. Note that cycles must have at least three vertices.

Examples

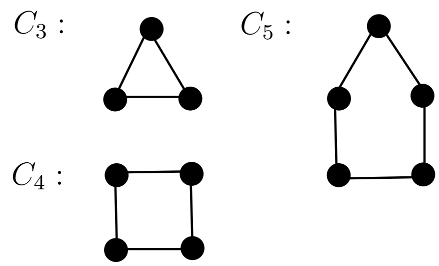
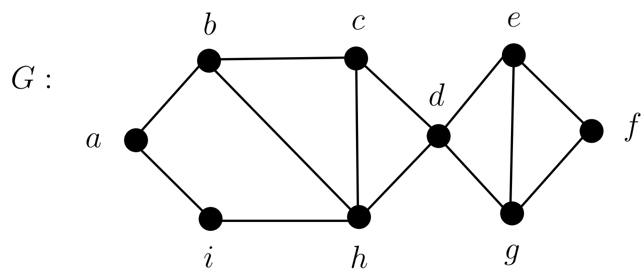


Figure 7: Cycles

3.3 Homework Exercises

Use the following graph to answer the questions.



1. List the vertex set and edge set of G.

2. List the degrees of the vertices of G.

3. What is the minimum degree of G?

4. What is the maximum degree of G?

5. Is the graph connected?

6. Give an example of a subgraph of G consisting of four vertices.

Draw the following graphs.

7. P_8

8. K_6

9. C_7

4 VERTEX COLORING UNIT

4.1 Introduction

In this unit we will examine a concept in Graph Theory called **vertex coloring**. This concept can be very useful in real life applications, such as how to mange conflicts of interest. For example, we will later see how graph coloring techniques can be applied to assigning frequencies to radio stations, scheduling club meetings, and coloring the countries of a map. By a **coloring** of a graph G, we mean the assignment of colors (numbers) to the vertices of G, one color to each vertex, so that adjacent vertices are assigned different colors. A k-**coloring** of G is a coloring of G using k colors. For example, Figure 8 shows a 5-coloring of the graph G_1, as well as a 4-coloring of the graph G_2.

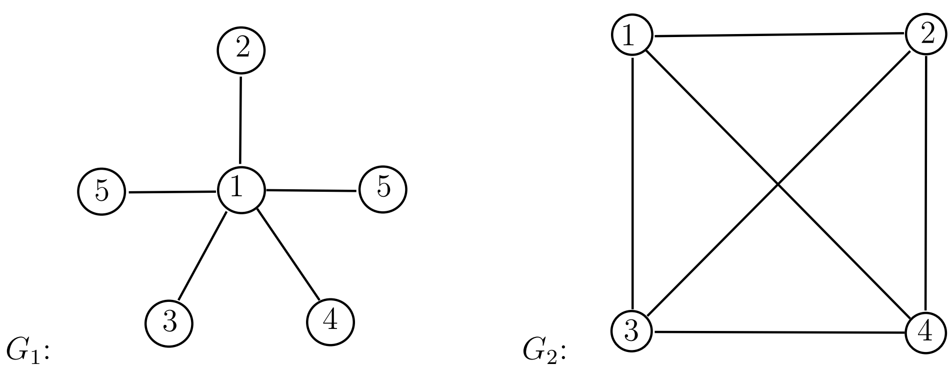


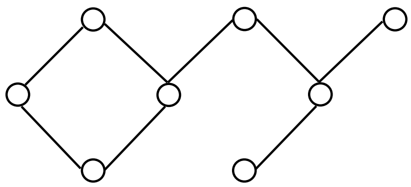
Figure 8: Vertex Coloring Example

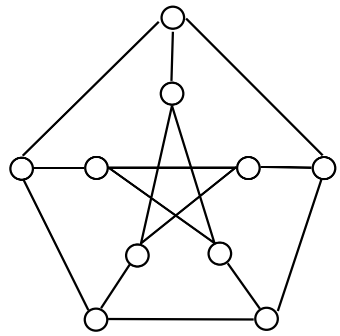
Notice that a graph G with n vertices can always have an n-coloring. But what if we wanted to determine the smallest number of colors needed for a graph? In other words, we want to find a smallest number k, where $k \leq n$, for which a k-coloring of G exists. Looking back at Figure 8, we can find a 4-coloring, 3-coloring, and a 2-coloring for the graph G_1. However, there is no k-coloring where $k < 4$ for the graph of G_2 since $G_2 = K_4$ and all edges are present between all vertices.

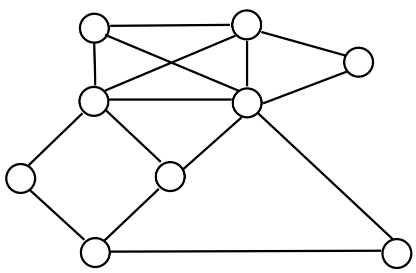
This brings us to the question of finding the **chromatic number** of a graph G. The chromatic number of a graph G is the minimum value k for which a k-coloring of G exists. The chromatic number of G is denoted $\chi(G)$. Thus, if we look once more at the graphs in Figure 8, we can see that $\chi(G_1) = 2$ and $\chi(G_2) = 4$. Note that we have only given the chromatic numbers of the these two graphs, we have not proved that we know the chromatic numbers.

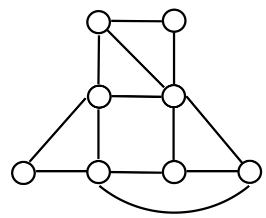
4.2 In Class Exercises

Find the chromatic number of the graphs below.

(a) 

(b) 

(c) 

(d) 

Let us explore further how to show that we have determined the chromatic number, in other words show that we cannot find a smaller number of colors. For example find the chromatic number of H.

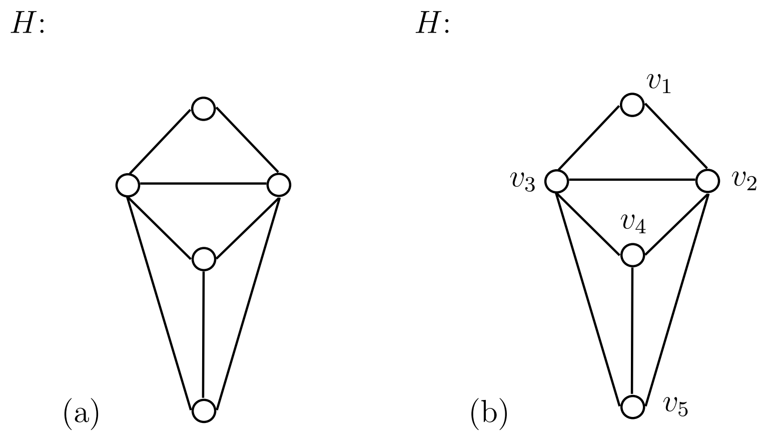


Figure 9: Chromatic Number Example

We claim that the chromatic number is at most 4 because we can find a 4-coloring. Suppose the vertices of H are labeled as in Figure 9(b). For example, v_1 and v_4 can be colored B-blue, v_2 colored R-red, v_3 colored G-green, and v_5 colored Y-yellow. Thus, we know that the chromatic number of H is at most 4, $\chi(H) \leq 4$. Now, if we can show that there is no 3-coloring of H, then we can without a doubt say that the chromatic number of H is 4. Notice that the vertices v_1, v_2, and v_3 form a triangle, therefore three colors are required to color these vertices. Say that we assign color B to v_1, color R to v_2, and color G to v_3. Since v_4 is adjacent to both v_2 and v_3, we cannot assign R or G to v_4. However, v_4 can be colored B. We can now see that v_5 is adjacent to a vertex colored R, a vertex colored B, and a vertex colored G. Thus, we need to introduce a new color to color v_5, so there exists no 3-coloring of H. Hence, we have shown that $\chi(H) = 4$.

4.4 Applications

Now that we have an understanding of the coloring concept, how can we use vertex coloring in day-to-day situations?

Potential Problems

1. A club scheduling conflict where some students are members of more than one club.

2. A radio station conflict where frequencies would interfere with each other if the stations were too close.

3. A map conflict where two countries need to be painted differently if they share a border.

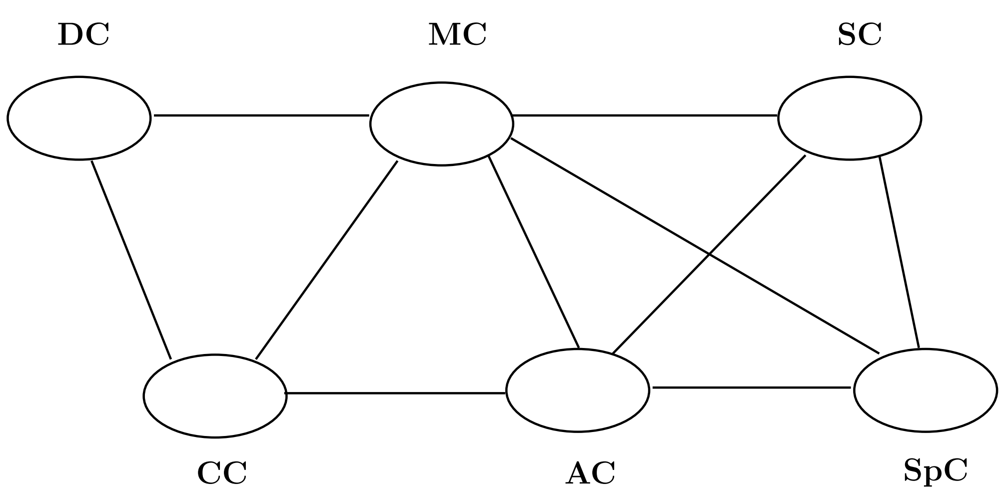
Let us do an example of Problem 1. Suppose that you were given the responsibility to schedule the meeting times of all the clubs in your High School. The first problem is that some students belong to more than one club, so not all of the clubs can meet on the same day. Secondly, the school does not want to be open every day for after-school clubs. Thus, you need to schedule as few days of the week for the clubs as possible. Below is the list of clubs and club members who belong to more than one club.

Clubs and Members

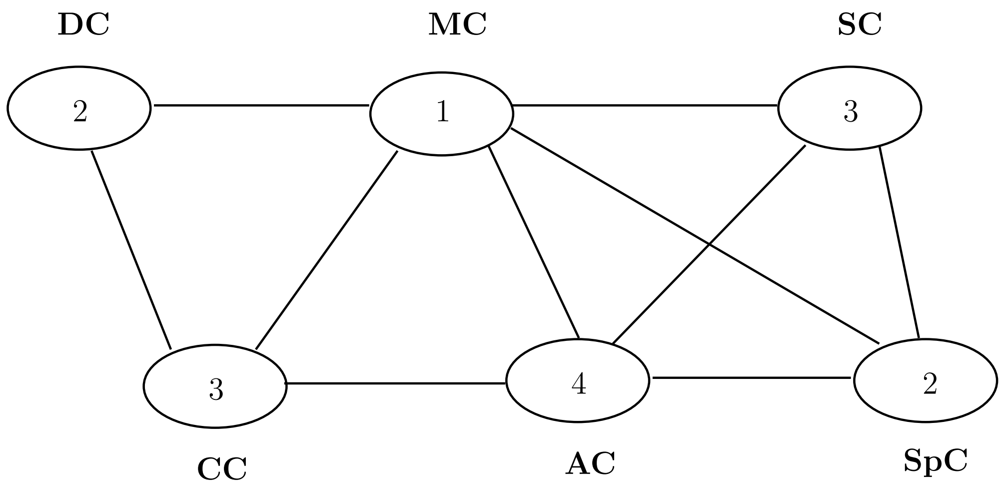
Clubs	Students in Multi-Clubs
Math Club	Dayna, Dale, Kristy
Debate Club	Kristy, Dayna, Travis
Science Club	Dale
Computer Club	Kristy, Rachel, Travis
Art Club	Rachel, Dale
Spanish Club	Dale

The question is: What is the minimum number of days needed so that no two clubs sharing a member meet on the same day?

Remembering our process of mathematical modeling, we can see that we are given a real-world situation, so now we need to represent our problem with a graph model. Let us have the vertices of our graph represent the different clubs, where two vertices are adjacent if the clubs they represent share a member. For example, the Debate Club would be adjacent to the Math Club because Dayna and Kristy are in both clubs. Here is a mathematical model where MC - Math Club, DC - Debate Club, SC - Science Club, CC - Computer Club, AC - Art Club, and SpC - Spanish Club:



Now we want to use as few colors as possible to color the graph, that is we want to assign a color to each vertex so that any two adjacent vertices have different colors. Let us have the colors represent the days of the week, so Day 1 is color 1, Day 2 is color 2 and so on.



We can see that four colors are needed to color the graph. So, let us interpret our solution in terms of our real-world problem. We can see that four days are needed in order for every club to meet once a week. Note that because of Dale's involvement in so many clubs, four days is the absolute fewest. Here is a chart to see the schedule more clearly.

Day 1	Day 2	Day 3	Day 4
Math Club	Debate Club Spanish Club	Science Club Computer Club	Art Club

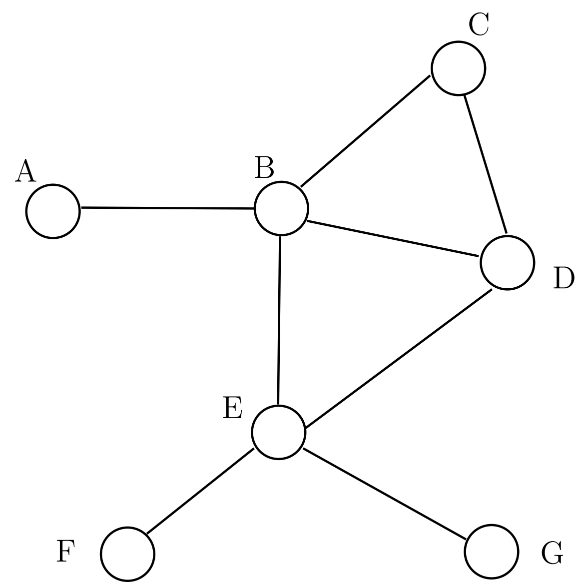
Thus, the minimum number of colors used to color the graph gives the minimum number of days needed to resolve all time conflicts.

Now let us do an example of Problem 2. The Federal Communications Commission (FCC) makes sure that the broadcast from one radio station does not interfere with the broadcast from any other radio station. Assigning an appropriate frequency to each station does this. The FCC requires that stations within transmitting range of each other must use different frequencies. Suppose that the FCC approves a new law where stations within 500 miles of each other must be assigned different frequencies. The locations of the seven stations are given in the grid below with the distances between the stations in miles.

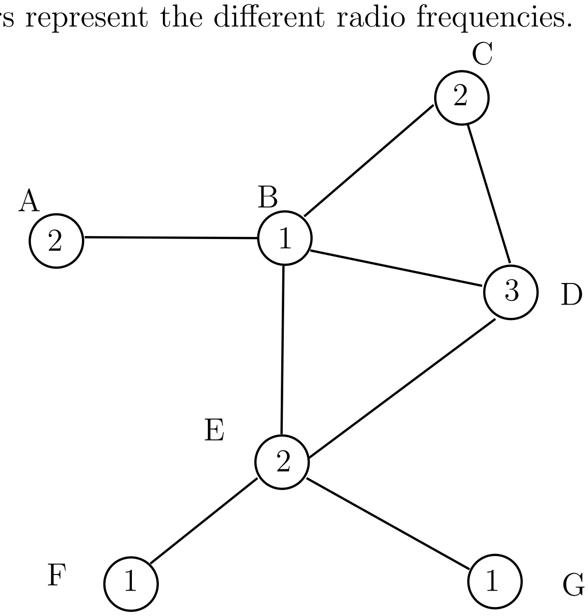
	A	B	C	D	E	F	G
A	-	450	550	700	600	850	900
B	450	-	500	300	250	600	750
C	550	500	-	100	530	800	900
D	700	300	100	-	470	650	700
E	600	250	530	470	-	350	490
F	850	600	800	650	350	-	530
G	900	750	900	700	490	530	-

The FCC wants you to assign a frequency to each station so that no two stations interfere with each other. The FCC also wants you to assign the fewest possible number of frequencies.

Now we need to translate the above grid into a graph model. Since the stations within 500 miles of each other must be assigned different frequencies, we only need to concern ourselves with the stations that are 500 miles or less apart. Suppose the vertices represent the stations and two vertices are adjacent whenever the stations they represent are 500 miles or less apart.



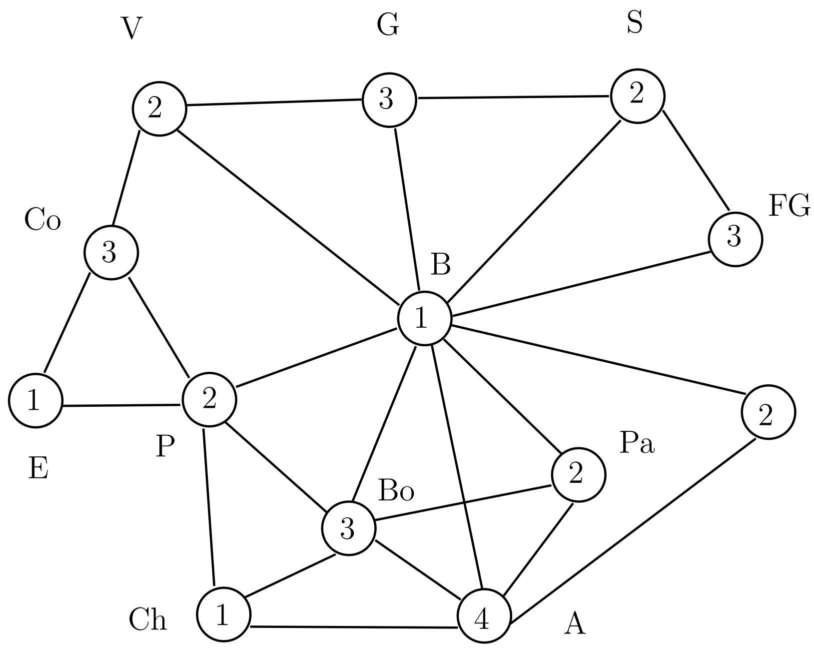
Now we want to use as few colors as possible to color the graph, that is, we want to assign a color to each vertex so that any two adjacent vertices have different colors. Let us have the colors represent the different radio frequencies.



We can see that three colors are needed to color the graph. So, let us interpret our solution in terms of our real-world problem. We can see that three radio frequencies are needed so that stations within 500 miles of each other get different frequencies.

Let us conclude this section by doing an example of Problem 3. You may have noticed in your geography or social studies class that maps are always colored so that neighboring countries do not have the same color. This is done so that the countries are easily distinguished and do not blend into each other. Suppose that a mapmaker gives you a map of South America.

In this given real-world situation we can represent our problem with a graph model. Let each country be a vertex where vertices are adjacent if they share a border.



We can see that four colors are needed to color this map.

4.5 Addendum

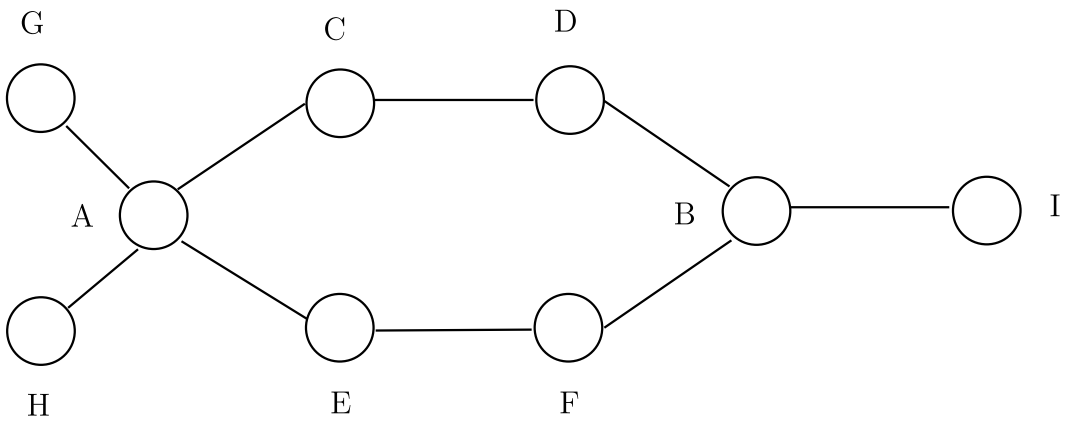
Note that as the number of vertices in a graph increases, it will become more difficult to label the vertices. Thus, you might think that a coloring algorithm would be the answer. However, there is no known efficient algorithm for coloring any graph with the fewest number of colors possible. You might try one of the following approaches:

- Color the vertex of highest degree first and keep coloring, trying to color as many vertices with a given color as possible. Make sure that adjacent vertices have different colors.
- Use the Welsh and Powell algorithm.

Welsh and Powell Algorithm

1. Begin by making a list of all the vertices starting with those of highest degree and ending with those of lowest degree.
2. Color the highest uncolored vertex on your list with an unused color.
3. Go down the list coloring as many uncolored vertices with the current color as you can.
4. If all the vertices are now colored, you are finished. If not, go back to Step 2.

However, the Welsh and Powell algorithm does not <u>always</u> give the fewest number of colors. For example look at the graph below. We will first use the Welsh and Powell algorithm, and then color the graph on our own.



Using the Welsh and Powell algorithm three colors will be used.

Step 1: List the vertices according to degree: A, B, C, D, E, F, G, H and I.

Step 2: A is at the top of the list. Color A green.

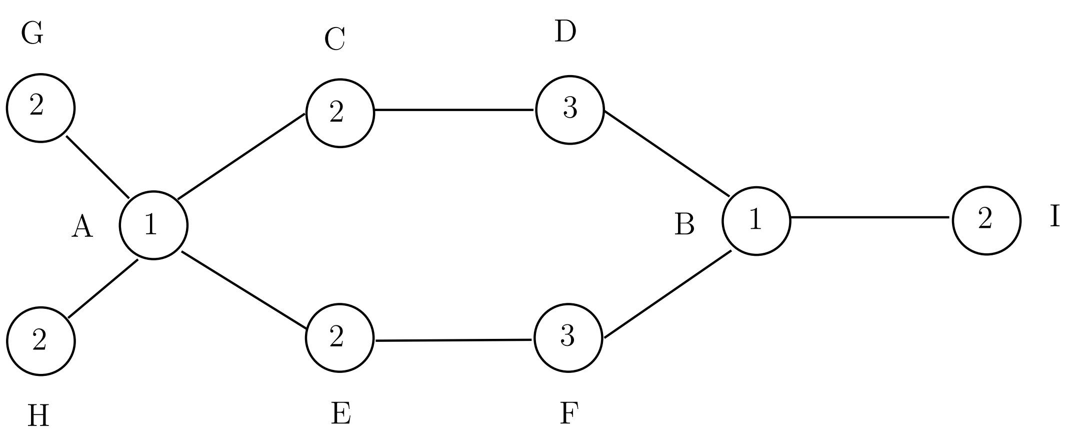
Step 3: Going down the list, we color B green.

Step 4: Choose a new color, red, for the highest uncolored vertex, C.

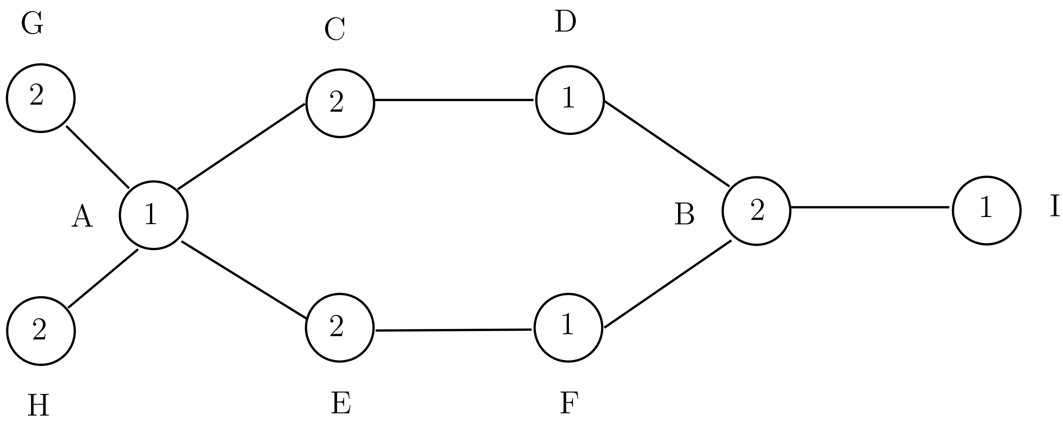
Step 5: Going down the list, we color E, G, H, and I red.

Step 6: Choose a new color, blue, for the highest uncolored vertex, D.

Step 7: Going down the list, we color F blue.



However, we can color the graph with two colors. Let A, D, F, and I be red and C, E, G, H, and B, be blue.

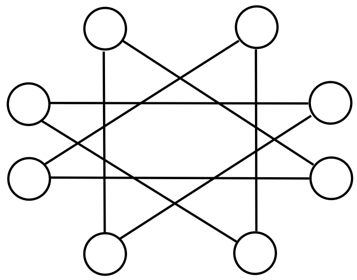


In fact, finding such a method for using an algorithm to find the chromatic number is a famous unsolved problem! [3]

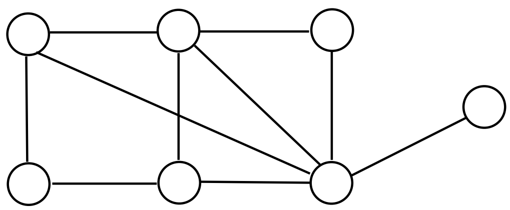
Mathematical Computations

1. Find the chromatic number for each of the graphs below.

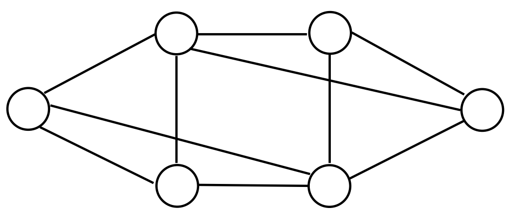
(a)



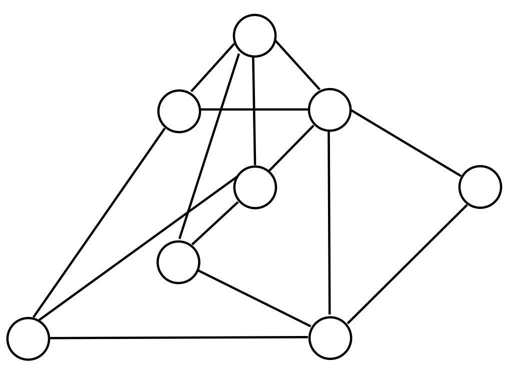
(b)



(c)



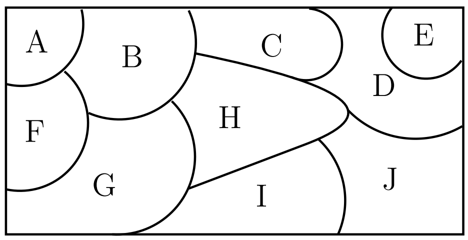
(d)



2. **(a)** Draw a connected graph that has five vertices and a chromatic number of four.

   (b) Draw a connected graph that has five vertices and a chromatic number of two.

3. Color the following a map using only three colors.



1. Below is a list of chemicals together with a list of other chemicals with which each cannot be stored.

Chemicals	Cannot Be Stored With
1	2,5,7
2	1,3,5,4
3	2,4,6
4	2,3,7
5	1,2,6,7
6	5,3
7	1,4,5

How many different storage facilities are necessary in order to keep all seven chemicals?

2. A local zoo wants to take visitors on animal feeding tours. They offer the following tours:

Tour 1: Visit lions, elephants, and giraffes.

Tour 2: Visit monkeys, hippos, and flamingos.

Tour 3: Visit elephants, flamingos, and bears.

Tour 4: Visit hippos, reptiles, and bears.

Tour 5: Visit kangaroos, monkeys, and reptiles.

The animals should not be fed more than once a day. Also, there is only room for one tour group at a time at any one site. Can these tours be scheduled using only Monday, Wednesday, and Friday?

3. Draw graphs to represent the maps below. Color the graphs and find the minimum number of colors needed to color each map.

(a)

(b)

Problem Solving Computations

1. Prove that $\chi(G_1) = 2$ for the graph G_1 of Figure 6.

2. Prove that $\chi(G_2) = 4$ for the graph G_2 of Figure 6.

5 MINIMUM SPANNING TREE UNIT

5.1 Introduction

In this unit we will examine a concept in Graph Theory called **minimum spanning trees**. This concept can be very useful in learning how to find the best network. We will later see how minimum spanning trees can be applied to optimizing a computer network, optimizing a road network, and optimizing cost. But first we need to define what a tree is and define its properties. The definition of a **walk** is the course taken from one vertex to another vertex along edges of a graph. A **path** is a walk in which no vertex nor edge is repeated. Hence, in a **cycle**, or a closed path, no edges may be repeated and only the beginning and ending vertices may be the same. A **tree** is a connected graph with no cycles. For example, which graphs in Figure 10 are trees and why?

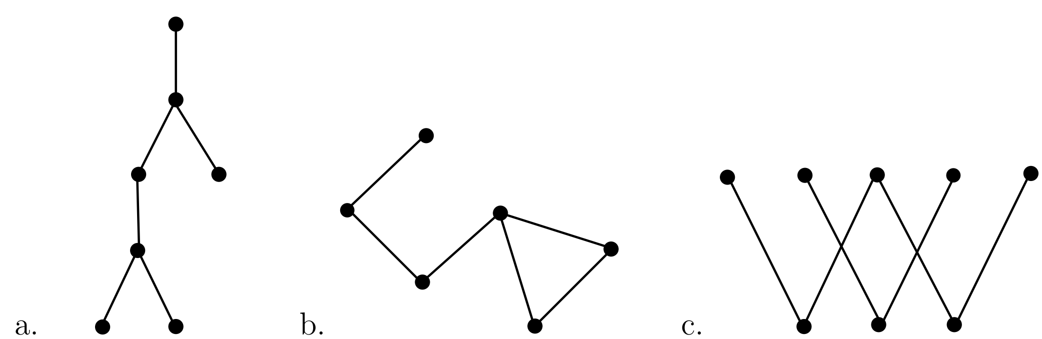


Figure 10: Trees

We can see that Figure 10a is a tree because it is a connected graph with no cycles. Figure 10b is not a tree because it has a cycle, and Figure 10c is not a tree because it is not connected.

This now brings us to define a spanning tree. A **spanning tree** of a connected graph G is a tree that is a subgraph of G and contains every vertex of G.

5.2 In Class Exercises

How many different spanning trees can you find for the graph below?

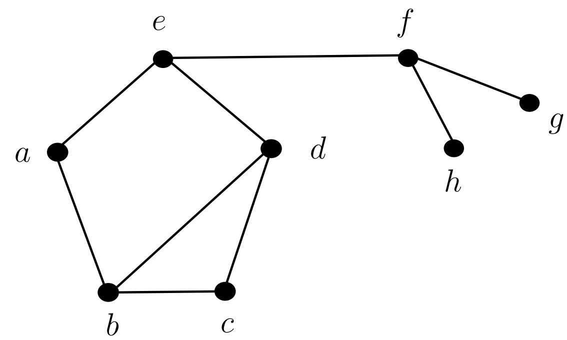
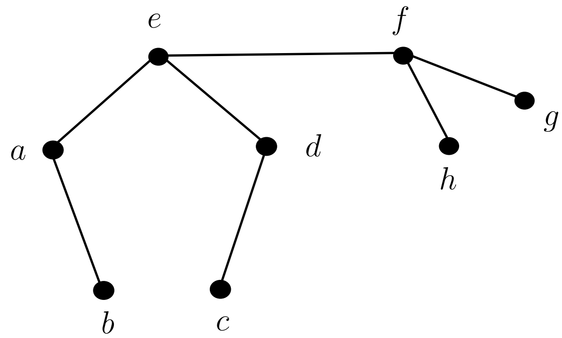
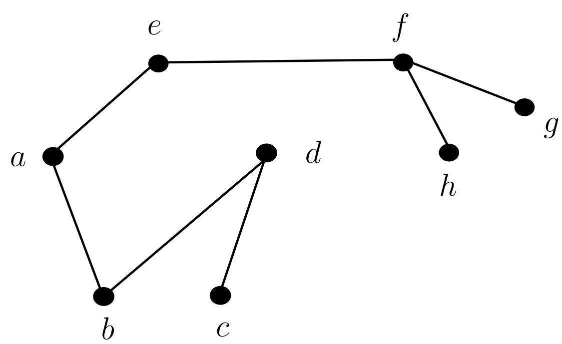


Figure 11: Spanning Trees

To get you started, we will walk through the construction of two spanning trees. For our first spanning tree, let us begin at vertex b. Do not forget that we want to connect all the vertices in the graph such that we have no cycles, i.e., using the smallest number of edges possible. For this spanning tree, we will connect b to a and then a to e, creating a ba edge and an ae edge. Now we have another choice: we can either take the ed edge, or the ef edge. Let us take the ef edge. In order to include the vertices g and h, both the fg edge and the fh edge must be included. Thus, the only vertices that are not in our spanning tree thus far are d and c. We can connect d by choosing the db edge or the de edge. Let us take the de edge. To connect c we can choose the cb edge. Let us take the cd edge; hence we have constructed our first spanning tree with 7 edges. Note that this spanning tree is not unique. Our choices of edges determined it.



Now let us construct another spanning tree. This time we will begin with vertex c. Observe that we can either take the cb edge or the cd edge. For this spanning tree, let us take the cd edge. We are now left with two choices for d, either the de edge or the db edge, let us take the db edge. Next we will take the ba edge followed by the ae edge. Notice that we took the ba edge because c is already in our spanning tree, so the only vertex adjacent to b in our original graph is a, thus resulting in the ba edge choice. Now, the remaining vertices that need to be connected to our spanning tree are f, g, and h. Thus, we can add the ef edge, fg edge, and the fh edge and our second spanning tree with 7 edges is constructed.



See if you can find the remaining 9 spanning trees for the graph in Figure 11!

Spanning Trees

From the previous example, we can see that it is not always easy to find a spanning tree, especially for a large graph. Thankfully, we have a couple of algorithms to help us find a spanning tree for a graph if one exists. The first method is called the **breadth-first search algorithm**.

Breadth-First 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.

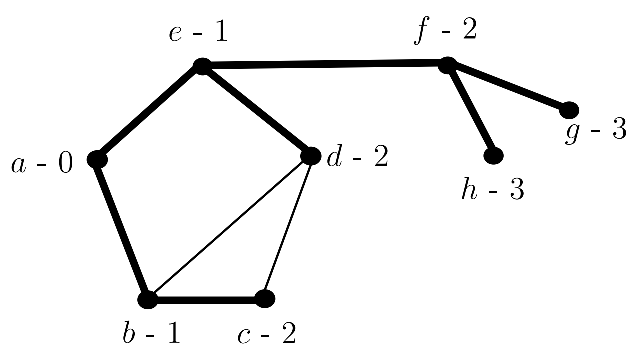
3. For each vertex labeled with a 1, find an edge that connects it with the vertex labeled 0. Darken those edges.

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 that one edge exists, choose one arbitrarily.

5. Continue this process until there are no more unlabeled vertices adjacent to labeled ones. If not all of the 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.

Example using the Breadth-First Algorithm

Let us return to Figure 11 and apply the Breadth-First Algorithm. Under Step 1, we can pick vertex a to be our starting vertex, label it 0. For Step 2, we can see that the only vertices adjacent to a are e and b, so we can label them with a 1. Step 3 tells us to darken the ae edge and the ab edge. Through Step 4 we will label d, c, and f with a 2 because they are adjacent to vertices labeled as a 1, e and b. So we can darken the ef edge and the bc edge. Now we have a choice, d is adjacent to e and b, which are both labeled with a 1, so we can either darken the ed edge or the bd edge. Let us choose the ed edge. Under Step 5, label g and h with a 3 and darken the fg edge and the fh edge, thus constructing a spanning tree. Note, this spanning tree should appear as one of your 11 spanning trees from the in-class example. If you apply the same steps with a different vertex, or choose a different edge when you have a choice, you will construct a different spanning tree.



Minimum Spanning Tree

In applications involving spanning trees, there are sometimes numbers called **weights**, associated with each edge of a graph. Thus, a **minimum spanning tree** is a spanning tree for the graph for which the total of the weights in the tree is minimum. Note that a graph can have more than one minimum spanning tree, but all of the minimum spanning trees must have the same minimum weight sum. Let us now see how we can construct a minimum spanning tree from a graph.

We will take the same graph in Figure 11 and add weights to the edges.

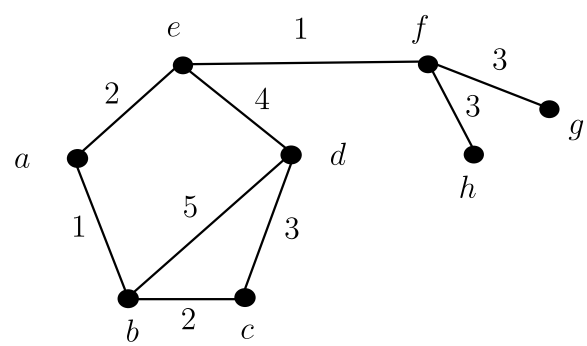


Figure 12: Minimum Spanning Trees

One algorithm for finding a minimum spanning tree for a graph is known as **Prim's Algorithm**.

Prim's Minimum Spanning Tree Algorithm

1. Find the edge with the smallest weights in the graph. Darken it and circle its two vertices. Ties are broken arbitrarily.

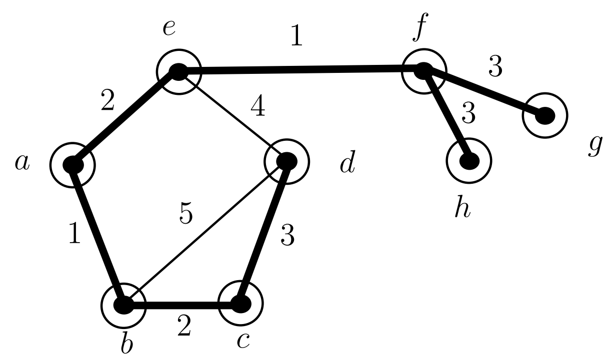
2. Find the edge with the smallest weight from the remaining undarkened edges having one circled vertex and one uncircled vertex. Darken this edge and circle its uncircled vertex.

3. Repeat Step 2 until all vertices are circles.

Example using Prim's Algorithm

Let us look at Figure 12 and apply Prim's Algorithm. Under Step 1, we can see that there are two edges that have the smallest weight; the ef edge and the ab edge both have a weight of 1. We will decide to use the ef edge, hence darken the ef edge and circle f and e. Under Step 2, we can see that the ae edge with weight 2, ae - 2, is the smallest weight among the remaining undarkened edges have one circled vertex and one uncircled vertex. Therefore, we can darken the ae edge and circle a. Step 3 tells us to repeat Step 2, so next we can darken the ab - 1 edge and circle b. Now darken the bc - 2 edge and circle c. Notice that we are now given a choice, the fg - 3 edge, fh - 3 edge, and the cd - 3 edge are all possible choices with the same weight. It does not matter which order you decide to add them to our construction of a minimum spanning tree. Let us darken the cd edge and circle d, then the fg edge and circle g, followed by the fh edge and circle h. All of the vertices have been circled and are connected, thus we have a minimum spanning tree with a minimal weight of 3 + 3 +

$1 + 2 + 1 + 2 + 3 = 15$.



Another algorithm for finding a minimum spanning tree for a graph is known as **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 the smallest weight to the largest weight. Ties are broken arbitrarily.

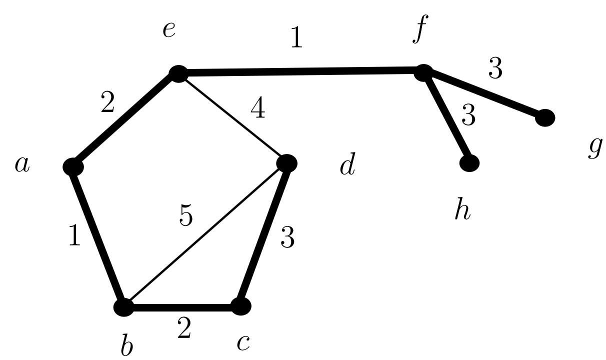
3. Darken the first edge on the list.

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.

Example using Kruskal's Algorithm

Let us return to Figure 12 and apply Kruskal's Algorithm. Step 1 tells us to make sure that our graph is connected, which we can easily see. Under Step 2 we can create a list: $ef-1, ab-1, ae-2, bc-2, cd-3, fg-3fh-3, ed-4$, and $bd-5$. Note, for the edges that have the same weight, it does not matter which edge comes first in your listing. Through Step 3 and 4 we can darken ef, ab, ae, bc, cd, fg, and fh. We do not want to darken the ed edge or the bd edge because we will create a cycle. From Step 5, we can see that we have constructed our minimum spanning tree with a minimal weight of 15 because Figure 12 has 8 vertices, and we have selected 7 edges to be in our minimum spanning tree.



5.4 Applications

Now that we have an understanding of the minimum spanning tree concept, how can we use minimum spanning trees in day-to day situations?

Potential Problems

1. Optimizing a computer network using the least amount of wire.

2. Optimizing a road network for the least amount of mileage.

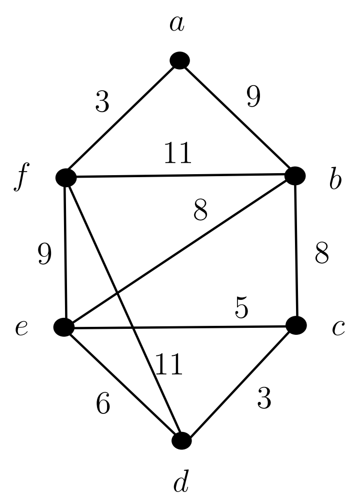
3. Optimizing a cell phone network for the least total cost.

Let us do an example of Problem 1. Suppose that at your high school, six computers in six different offices need to be networked. Your school wants the "best" possible network, i.e. use the least amount of wire to link all the computers. Note that the connection between two computers can either be linked directly or indirectly through another computer. The grid below shows which computers can be linked directly as well as how much wire in meters is needed, where the computers are letters and the distances are in meters.

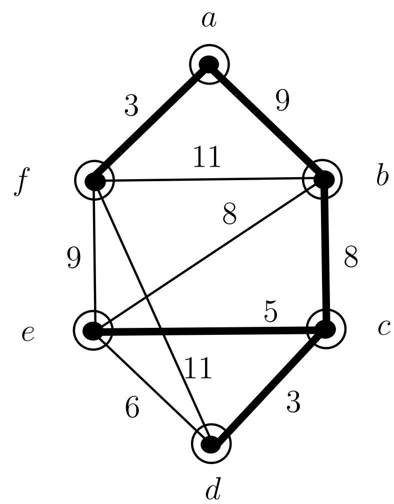
	A	B	C	D	E	F
A	-	9	-	-	-	3
B	9	-	8	-	8	11
C	-	8	-	3	5	-
D	-	-	3	-	6	11
E	-	8	5	6	-	9
F	3	11	-	11	9	-

The question is: What is the minimum amount of wire needed to connect all six computers so that every computer is linked directly or indirectly to every other computer?

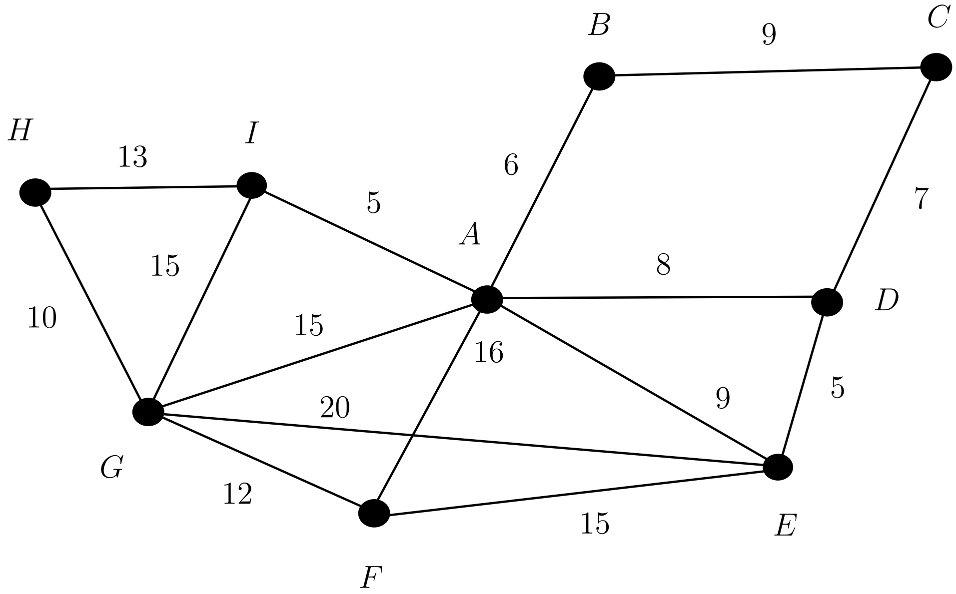
Remembering our process of mathematical modeling, we begin with a given real-world situation, so now we need to represent our problem with a graph model. Let us have the vertices of our graph represent the computers, where two vertices are connected by an edge if the computers have a direct connection. The weights attached to each edge are the distance between the two points on the grid.



Now we can solve our mathematical problem by using Prim's Algorithm or Kruskal's Algorithm to find the minimum spanning tree. Let us decide on Prim's algorithm. Using this algorithm, a possible spanning tree is dc, ce, cb, ba, and the af edge with a minimum weight of 28. Another consists of dc, ce, eb, ef, and the fa edge with a minimum weight of 28. Now let us interpret our solution in terms of our real world problem. We have found two minimum spanning trees, which mean that we have two computer networks that use the minimum amount of wire, 28 meters, to connect all six computers.



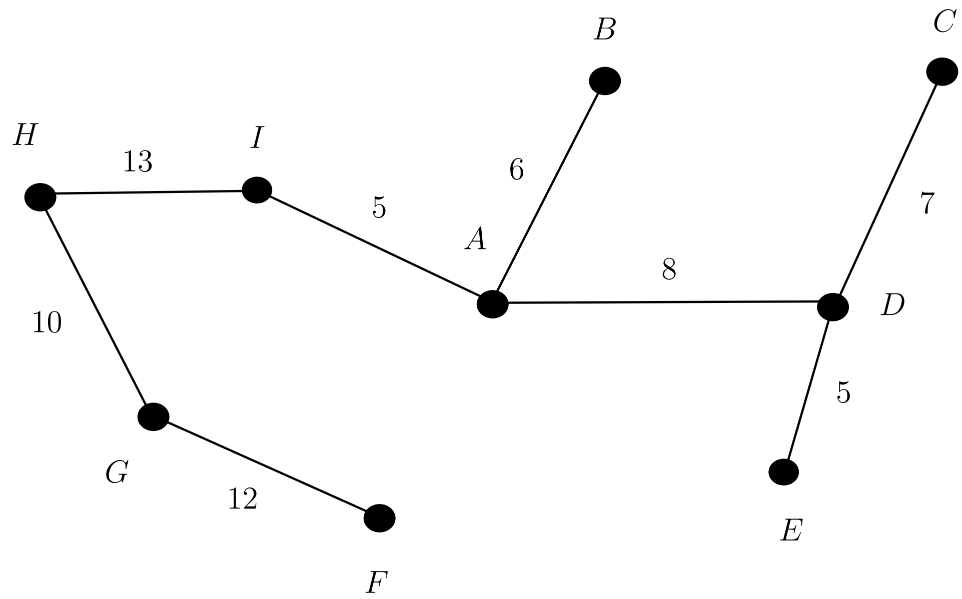
Now we will do an example of Problem 2. Suppose that, in making plans for winter storms, your local county government needs a design for repairing the county roads in case of an emergency. You are given a map of the towns in your county and the existing major roads between them.



Map of your county with mileage

You are then asked to devise a plan that repairs the least number of miles of road but keeps a route open between each pair of towns.

We begin with our real-world situation. For our graph model, we can simply use the given map. Let us solve our mathematical problem this time by using Kruskal's Algorithm to find a minimum spanning tree. This algorithm creates the list: $AI-5, DE-5, AB-6, DC-7, AD-8, BC-9, AE-9, GH-10, GF-12, HI-13, GI-15, GA-15, FE-15, AF-16, GE-20$. Using this algorithm, a possible minimum spanning tree consists of $AI, DE, AB, DC, AD, GH, GF$, and the HI edge with a minimum weight of 66. Now let us interpret our solution in terms of our real world problem. Through our minimum spanning tree we have discovered a plan that connects the towns with the minimum possible number of miles of road, 66 miles.

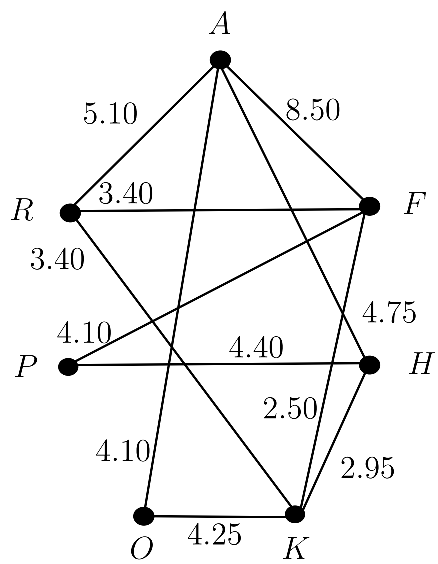


We will now conclude this section by doing an example of Problem 3. Suppose that a family with seven members in different parts of the country has a relative serving overseas. The family wants to set up a cell phone calling network so everyone will know the latest news about the overseas relative, for the least total cost. The grid below shows the cost for a 15-minute phone call between each pair of family members.

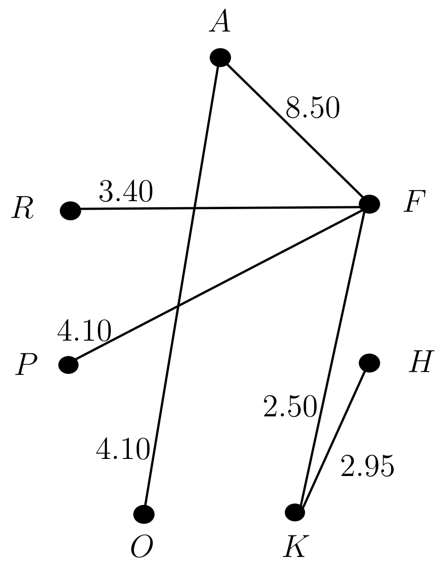
	Alice	Faith	Hillard	Kristy	Owen	Peter	Rachel
Alice	-	\$3.50	\$4.75	-	\$4.10	-	\$5.10
Faith	\$3.50	-	-	\$2.50	-	\$4.10	\$3.40
Hillard	\$4.75	-	-	\$2.95	-	\$4.40	-
Kristy	-	\$2.50	\$2.95	-	\$4.25	-	\$3.40
Owen	\$4.10	-	-	\$4.25	-	-	\$3.20
Peter	-	\$4.10	\$4.40	-	-	-	-
Rachel	\$5.10	\$3.40	-	\$3.40	-	-	-

What is the total cost of the least expensive calling network they can set up?

In this given real-world situation we need to represent our problem with a graph model. Let us have the vertices of our graph represent a family member using the first letter of their name, where two vertices are connected by an edge if the family members have a direct connection. The weight attached to each edge is the amount of money between the two callers on the grid.

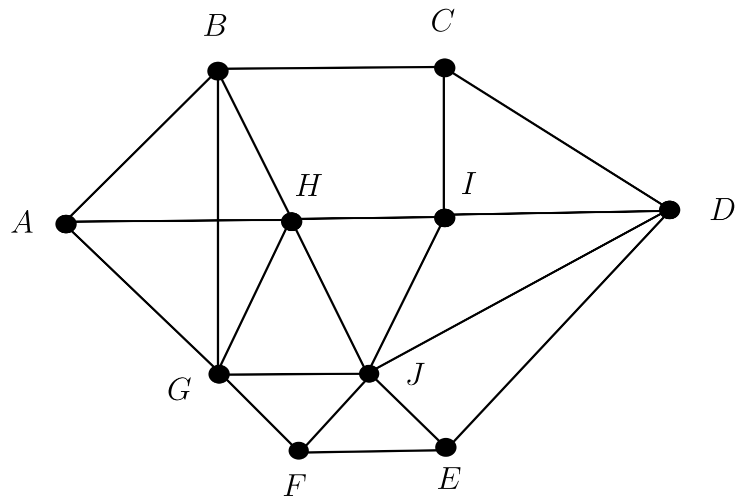


Let us use Kruskal's algorithm to find a minimum spanning tree. Using this algorithm, a possible spanning tree is KF, KH, FR, FA, FP, and the AO edge with a minimum cost of \$20.55. If we interpret our solution in terms of our real world problem, we can see that Kristy calls Faith and Hillard, Faith calls Rachel, Alice, and Peter, and Alice calls Owen for a total minimum cost of \$20.55.

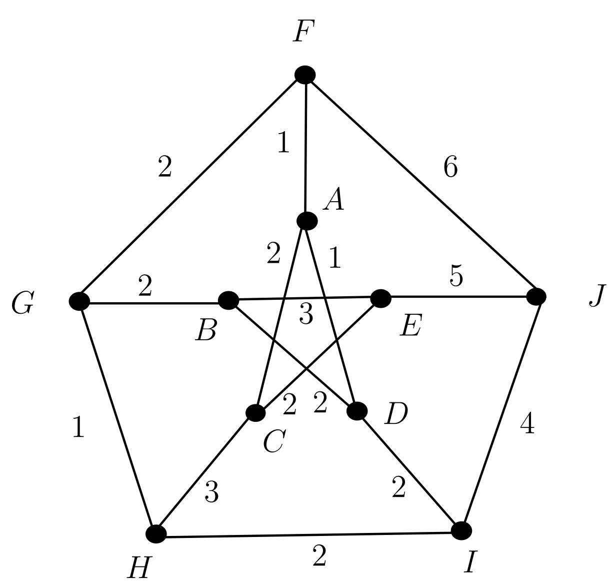

A
8.50
R
3.40
F
4.10
P
H
2.50
4.10
2.95
O
K

Mathematical Computations

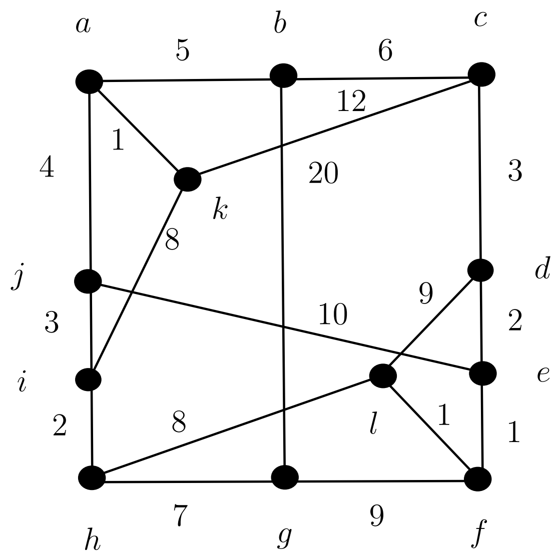
1. Use the breadth-first algorithm to find a spanning tree for this graph. Begin at vertex C.



2. For the following graph, first use Prim's Algorithm to find a minimum spanning tree, then use Kruskal's Algorithm to find a minimum spanning tree. What is the minimum weight in each case?



3. Find a minimum spanning tree for this weighted graph using your favorite method.



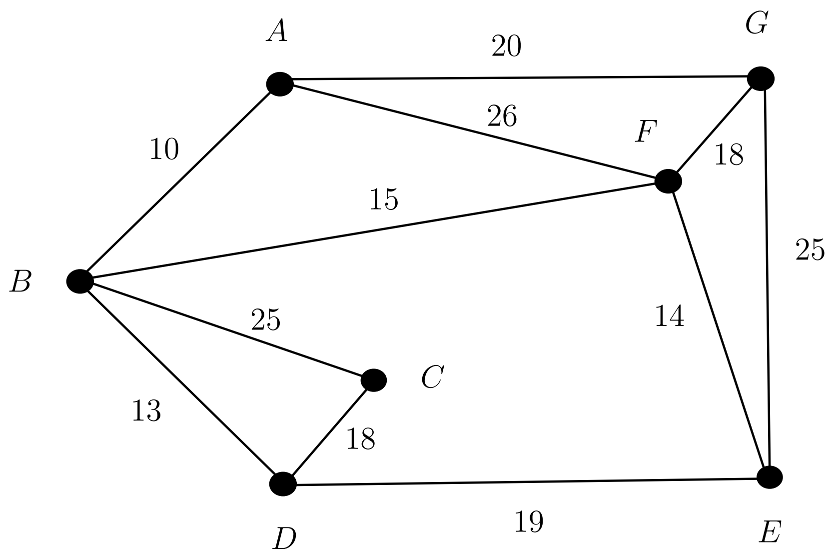
1. A local restaurant has opened an outdoor patio for the summer. The owner wants to hang nine festive light fixtures at designated locations on the overhead latticework. Because of the layout of the patio and the latticework, it is not possible to install wiring between every pair of lights. The grid below shows the distances in feet between lights that can be linked directly. The owner wants to use the minimum amount of wire to get all nine lights connected.

	A	B	C	D	E	F	G	H	I
A	-	16	-	-	15	15	-	-	-
B	16	-	16	12	-	-	-	-	-
C	-	16	-	-	-	-	12	-	-
D	-	12	-	-	-	-	10	-	-
E	15	-	-	-	-	7	-	-	-
F	15	-	-	-	7	-	-	-	-
G	-	-	12	10	-	-	-	18	-
H	-	-	-	-	-	-	18	-	8
I	-	-	-	-	-	-	-	8	-

What is the minimum amount of wire needed to connect all nine lights?

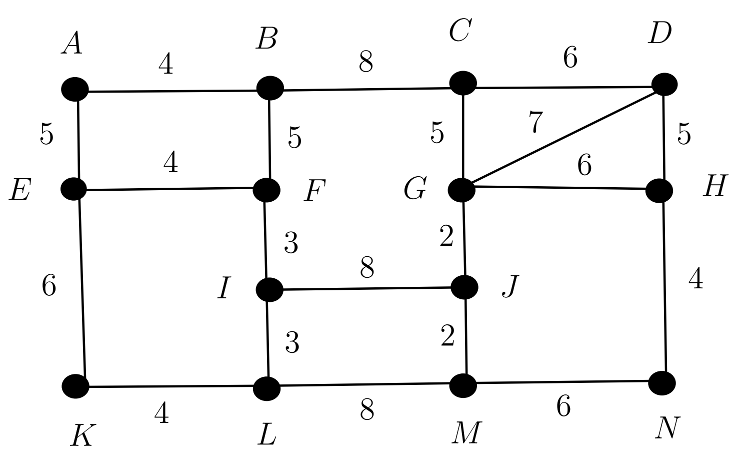
2. There are seven small towns in Madison County that are connected to each other by gravel roads, as in the following diagram. The distances are given in miles. The county wants to pave some of the roads so that people can get from town to town on paved roads, either directly or indirectly. However, Madison

County is on a tight budget so the total number of miles paved needs to be minimum.

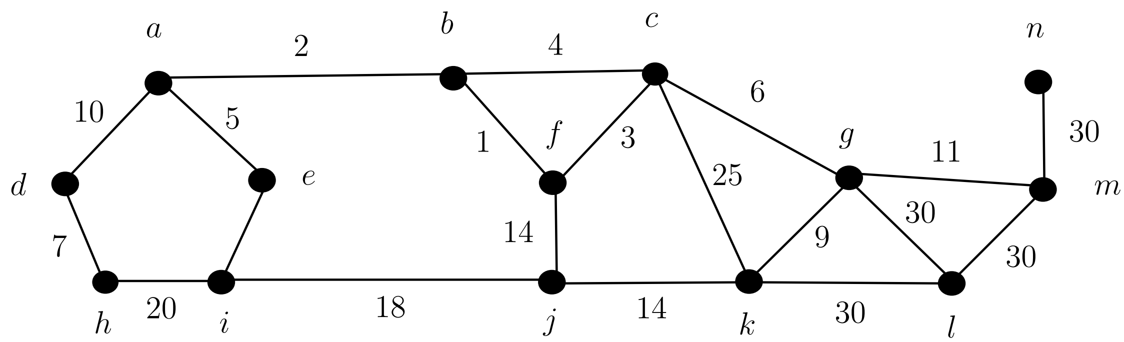


Find and draw a network of paved roads that will fulfill the county's requirements.

3. The computers in each of the offices at Earl of March High School need to be linked by cable. The map below shows the cost of each link in hundreds of dollars. What is the minimum cost of linking the 14 offices?



1. Suppose that for the graph below the edges represent possible highways that may be built to join pairs of cities. Let us assume that these weights are the projected costs in millions of dollars for the highways. The idea is to build the cheapest highway network to keep the state's budget down.



Now suppose that the governor lives in town k. He expects to make a lot of trips to towns c and l and has used his political clout to force the construction of the direct routes ck and kl even though those highways will be rather expensive. Find a minimum spanning tree for the graph subject to the restriction that edges ck and kl are in the tree. Explain why there can be no spanning tree containing ck and kl that has smaller cost than the tree that you have found.

6 DOMINATION UNIT

6.1 Introduction

In this unit we will examine a concept in Graph Theory called **domination**. This concept can be very useful in real life applications. For example, we will later see how domination techniques can be applied in placing a minimum number of security stations in a county, determining the fewest number of stops a bus driver needs to make, and constructing the least amount of radio stations in an area. A set of vertices S in graph G is a **dominating set** of G if every vertex of G is either in S or adjacent to a vertex in S. The **domination number** of G, denoted by $\gamma(G)$, of a graph G is the smallest number of vertices in any dominating set of G. Let us look at Figure 13 and determine the minimum dominating sets and the domination number.

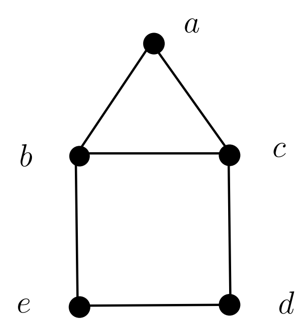


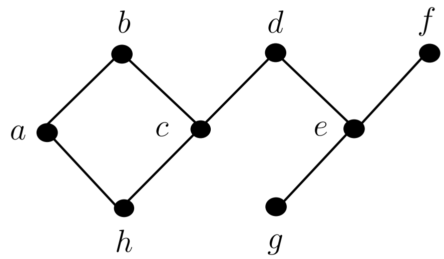
Figure 13: Domination Example

First, we can clearly notice that all the vertices form a dominating set, but we we want to find the least number. Notice that we could choose $\{b, d\}$ as a dominating set, since b dominates itself and the vertices adjacent to it, a, c, and e. Next d

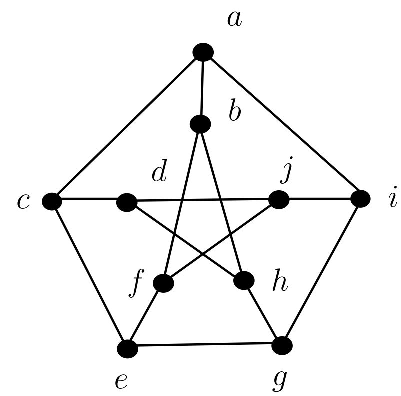
dominates itself and its neighbors c and e. Other minimum dominating sets are $\{a, e\}, \{a, d\}, \{c, e\}, \{b, e\}, \{b, c\}$, and $\{c, d\}$. So we know that we need at most two vertices to dominate the graph, $\gamma(G) \leq 2$. To see that $\gamma(G) = 2$, we must show that one vertex can not dominate the graph. To see this, note that no one vertex is adjacent to every vertex in the graph. Hence $\gamma(G) = 2$.

Find the domination number of the graphs below.

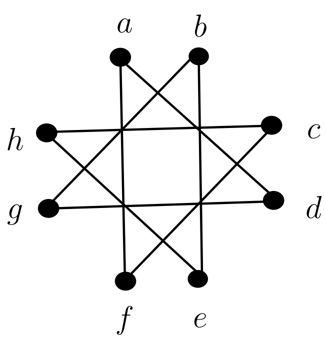
(a)



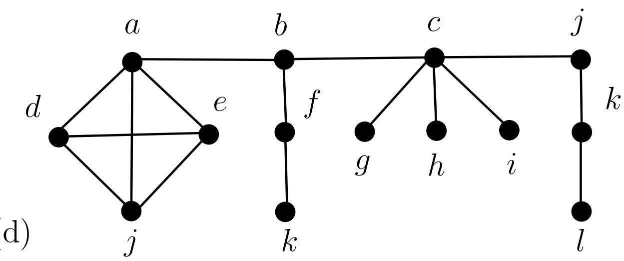
(b)



(c)



(d)



History of Domination

The concept of domination appears to have originated while playing a game of chess, where the idea is to cover or dominate squares of a chessboard with certain chess pieces. In the game of chess, a queen can move horizontally, vertically, or diagonally over any unoccupied squares. For example in Figure 14, the queen can move to (or attack, or dominate) all of the squares marked with an "X". In 1862, de Jaenisch tried to figure out all of the minimum number of queens that can be placed on a chessboard so that every square is either occupied by a queen or being attacked by at least one queen. His question is now commonly known as **The Five Queens Problem**, since it has been proven that the answer is 5 [10, 5]. One possible solution is shown in Figure 14.

X		X		X			
	X	X	X				
X	X	**Q**	X	X	X	X	X
	X	X	X				
X		X		X			
		X			X		
		X				X	
		X					X

					Q		
	Q						
						Q	
			Q				
Q							

Figure 14: The Five Queens Graph

You might be wondering what the connection is between the above queens problem and dominating sets in graphs. The connection can be easily seen through a model of this problem. We can let the 64 squares of a chessboard be the vertices of our graph G where two vertices (squares) are adjacent in G if each square can be reached by a queen on the other square in a single move. This graph G, which we have just constructed, is generally referred to as the **Queen's Graph**. Hence, the smallest number of queens that dominate all the squares of a chessboard is the domination number of G, $\gamma(G)$.

6.4 Applications

Now that we have an understanding of the domination concept, how can we use domination in day-to-day situations?

Potential Problems

1. Placing the smallest number of security stations in a county so that every high school is protected.

2. Determining the fewest number of stops a bus driver needs to make.

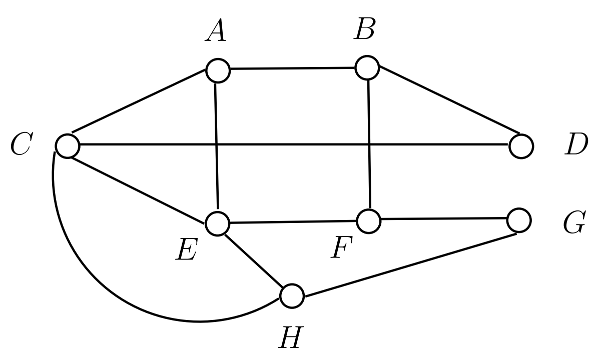
3. Constructing the least amount of radio stations in an area.

Let us do an example of Problem 1. Suppose that a county contains eight high schools connected by roads. The table below tells us which high schools are connected by a road, where a Y is given if the distance between the two schools is less than five miles. The Board of Education is this county wants to upgrade their security so that each high school either has a security station within their high school or is within 5 miles of a high school that does have one, so that if an emergency arises security can get to the scene quickly. Due to budgetary constraints only a minimum number of security stations can be built.

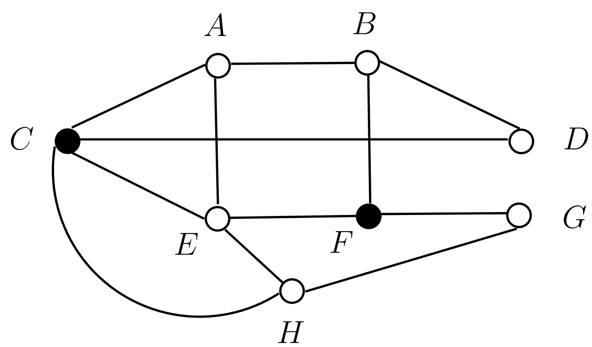
	A	B	C	D	E	F	G	H
A	-	Y	Y	-	Y	-	-	-
B	Y	-	-	Y	-	Y	-	-
C	Y	-	-	Y	Y	-	-	Y
D	-	Y	Y	-	-	-	-	-
E	Y	-	Y	-	-	Y	-	Y
F	-	Y	-	-	Y	-	Y	-
G	-	-	-	-	-	Y	-	Y
H	-	-	Y	-	Y	-	Y	-

What is the smallest number of stations that must be built? Give a set where these could be placed. In addition, in which high schools should security stations be placed if there is already one at B, the largest high school in the county?

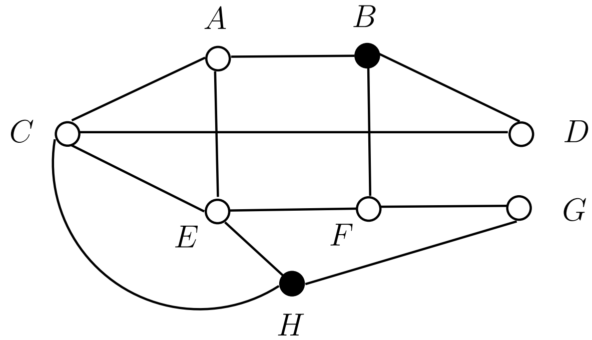
Now we need to translate the above grid into a graph model. Let us have the vertices of our graph represent the different high school, where two vertices are adjacent if the high schools they represent are less than five miles apart.



To answer the first question, we need to find the smallest dominating set of the graph. The dominating set $\{C, F\}$ will dominate G. Vertex C dominates itself, A, D, E, and H. Vertex F dominates B, G, E and itself. Thus $\gamma(G) \leq 2$. To see that $\gamma(G) = 2$, we must show that one vertex can not dominate the graph. Since we do not have a vertex that is adjacent to every vertex in the graph, we know that $\gamma(G) = 2$. So, now let us interpret our solution in terms of our real-world problem. We can see that two security stations, located at the high schools C and F, are needed so that each of the remaining high schools has a security station nearby.

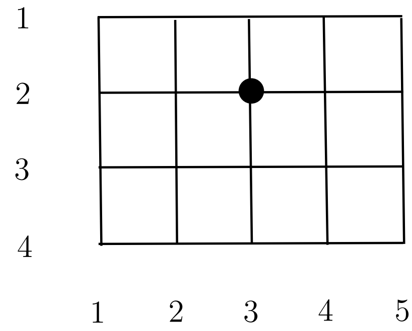


To answer the second question, we are still looking for the smallest dominating set, only this time we must include the vertex B in our dominating set. Subsequently, B, will dominate itself, A, F, and D. To dominate the remaining vertices C, E, H, and G, we can include H in our dominating set. Thus, we have found a second smallest dominating set that contains $\{B,H\}$. Now, let us interpret our solution in terms of our real-world problem. We can see that two security stations, located at the largest high school, B, and one at H, are needed to that each of the remaining high schools have a security station nearby.

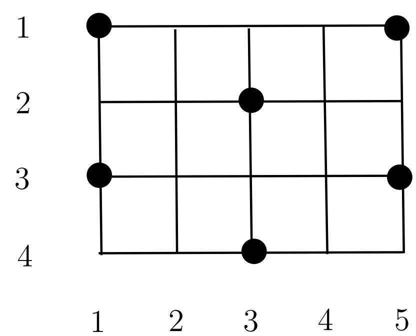


Notice that the first smallest dominating set that we found, $\{C,F\}$, is not the only smallest dominating set. The only way to have an algorithm to determine the smallest dominating number of a graph is to try every possible set of vertices to see if it dominates. However, if you have a large graph this will be impossible in a reasonable amount of time, even using the world's fastest computers. Also, please note that if you were forced to include a different vertex from the beginning, say vertex D, the domination set will not be a minimum dominating set. A possible dominating set could be $\{D, E, G\}$ and the domination number is 3, which is not the smallest.

Now let us do an example of Problem 2. Suppose that a school district passed a rule that no child in elementary school shall have to walk more than one block to school or to a school bus stop. Due to increase of the price of gas, the school buses need to minimize their number of stops. Determine the fewest number of stops the bus driver would have to make so that for each intersection in the graph below there is either a bus stop or there is a bus stop (or school) no more than one block away.



We begin with our real-world situation. For our graph model, we can simply use the given map above. We need to find a dominating set that consists of the smallest number of intersections for school bus stops. The domination number is 5 and minimum dominating set consists of the vertices {(1,1), (1,5), (2,3), (3,1), (3,5), and (4,3)}. Now, let us interpret our solution in terms our real-world problem. We can see that five bus stops need to be installed at intersections (1.1), (1.5), (3,1), and (3,5), so that the students are either one block away from the school or one block away from a bus stop.



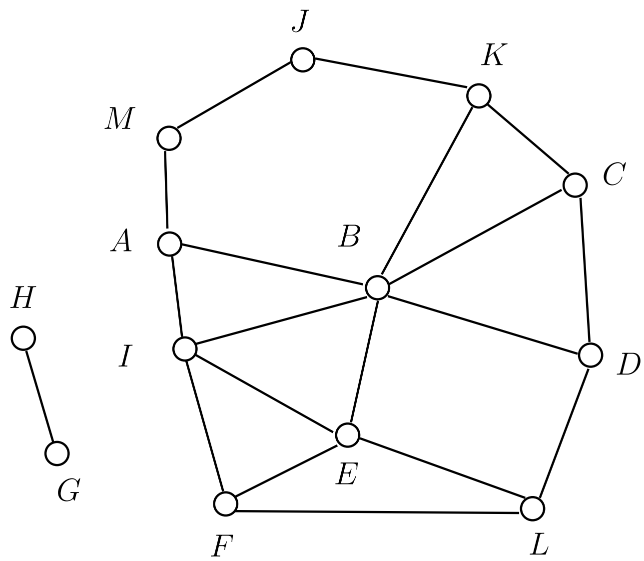
Let us conclude this section by doing an example of Problem 3. Suppose that we have a collection of small towns along the Appalachian Mountains in the areas of the White Mts., Green Mts., Bershire Hills, Catskill Mts., Blue Ridge Mts., and

Cumberland Plateau. We would like to establish radio stations in some of the towns so that messages can be broadcast to all of the towns in the area. Since each radio station has a limited broadcasting range, fifty miles, we need to use several stations to reach all towns. But, radio stations are costly, so we need to construct as few as possible. The locations of the thirteen towns are given in the grid below with the distances between the town in miles.

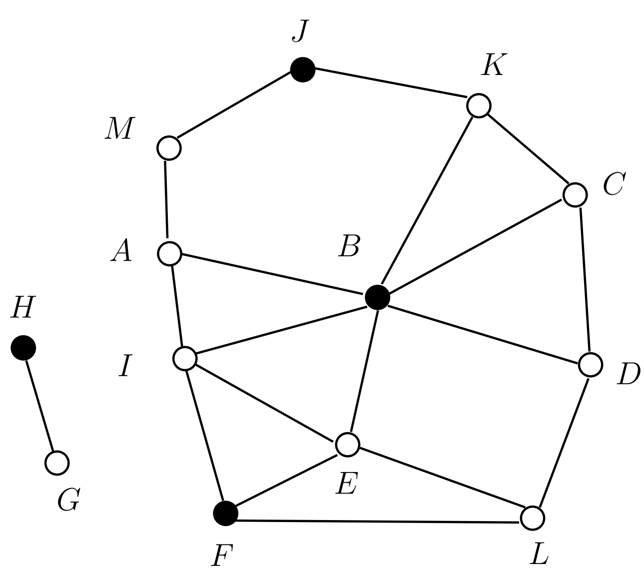
	A	B	C	D	E	F	G	H	I	J	K	L	M
A	-	39	60					54	27	54			24
B	39	-	28	32	32				38		44		
C	60	28	-	25						54	25		
D		32	25	-	52							45	
E		32		52	-	15			23			37	
F					15	-	53		26			42	
G						53	-	42	54				
H	54						42	-	53				65
I	27	38			23	26	54	53	-				
J	54		54							-	33		36
K		44	25							33	-		
L				45	37	42						-	
M	24							65		36			-

What is the fewest number of stations that need to be constructed?

Now we need to translate the above grid into a graph model. Since we know that that a radio station has a broadcast range of only fifty miles, we can disregard towns that are more than fifty miles apart. Therefore, we can now have the vertices represent towns and connect two vertices by an edge whenever the towns they represent are 50 miles or less apart. This gives us the graph below.



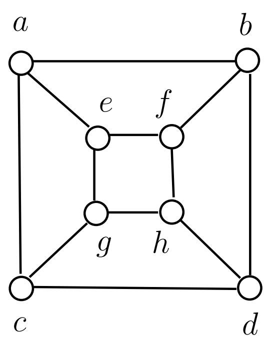
We want to find a set of the least number of stations which dominate all other vertices. The domination number is 4 and a minimum dominating set consists of $\{B, F, H, J\}$. Now let us interpret our solution in terms of our real-world problem. We only need to construct four radio stations in the towns of B, F, H, and J, so that all of the other towns can be reached.



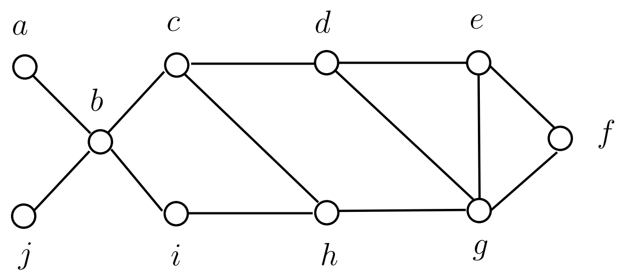
Mathematical Computations

1. Find the domination number of the graphs below.

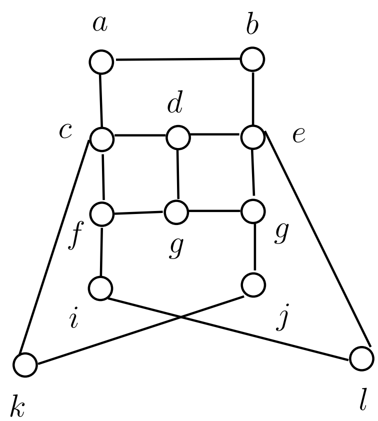
(a)



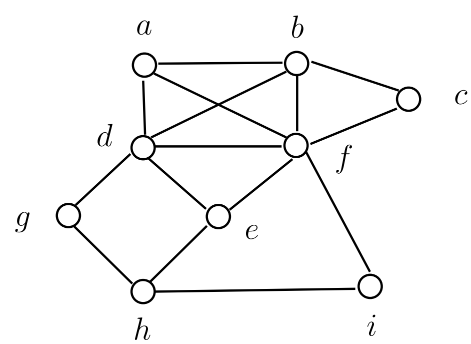
(b)

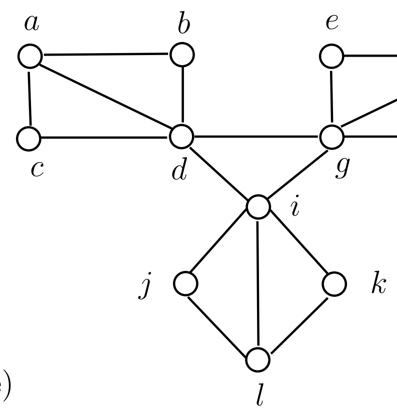


(c)

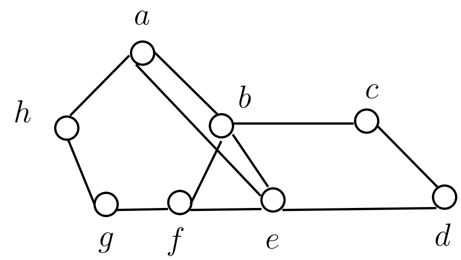


(d)



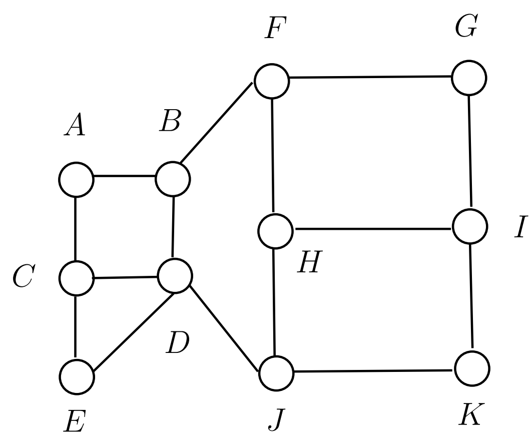


(e)



(f)

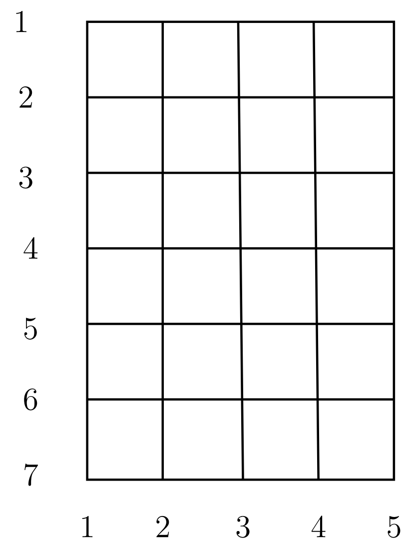
1. Suppose that a company contains eleven offices connected by hallways as indicated in the graph below. The manager of the company wants to install top of the line photocopy machines so that each office has a copier within their office or is near an office that has one. Unfortunately, the company is new and funds are limited, thus only a minimum number of photocopiers can be installed.



Tell the manager the minimum number of photocopier machines that need to be purchased and in which offices to place them.

2. Suppose that a contractor is building a new subdivision. The last decision that the contractor has to make is where to place the waste receptacles. Regrettably, the contractor went over budget building the community center, so not every intersection can have a waste receptacle. The contractor would like for you to determine the number of receptacles that are needed so that, for each

intersection, there is either a receptacle or there is one at an intersection one block away. The following figure is a street grid of the city.



Street grid of the city

3. Suppose that we have a collection of small villages in Alaska. We would like to locate radio stations in some of these villages so that messages can be broadcast to all of the villages in the region. Since each radio station has a limited broadcasting range, fifty miles, we need to use several stations to reach all the villages. The locations of the ten villages are given in the grid below with the distances between the villages in miles.

	A	B	C	D	E	F	G	H	I	J
A		40	45	40					65	
B	40		40		30	30				
C	45	40		40					50	
D	40		40				60			
E		30				30				70
F		30			30		45	45		
G				60		45		45		
H						45	45		50	50
I	65		50					50		50
J					70			50	50	

What is the fewest number of stations that need to be constructed?

Problem Solving Computations

1. Find a dominating set of five queens, all in the same row, for an 8x8 chessboard.

2. Find a dominating set of five queens, where no two queens attack each other, for an 8x8 chessboard.

3. Find a dominating set of five queens, all on the main diagonal, for an 8x8 chessboard.

7 HAMILTONIAN PATHS AND CYCLES UNIT

7.1 Introduction

In this unit, we will examine a concept in Graph Theory called **Hamiltonian paths and cycles**. This concept can be very useful in real life applications, such as how to solve transportation problems. For example, we will later see how Hamiltonian paths and cycles can be applied to determine tournament rankings and to model a well-known problem called the traveling salesperson problem. First, we need to give definitions for a Hamiltonian path and cycle. A **Hamiltonian path** is a path that visits each vertex of a graph *exactly once*. A **Hamiltonian cycle** is a Hamiltonian path that starts and ends at the same vertex. Try to find a Hamiltonian cycle in each of the graphs below in Figure 15.

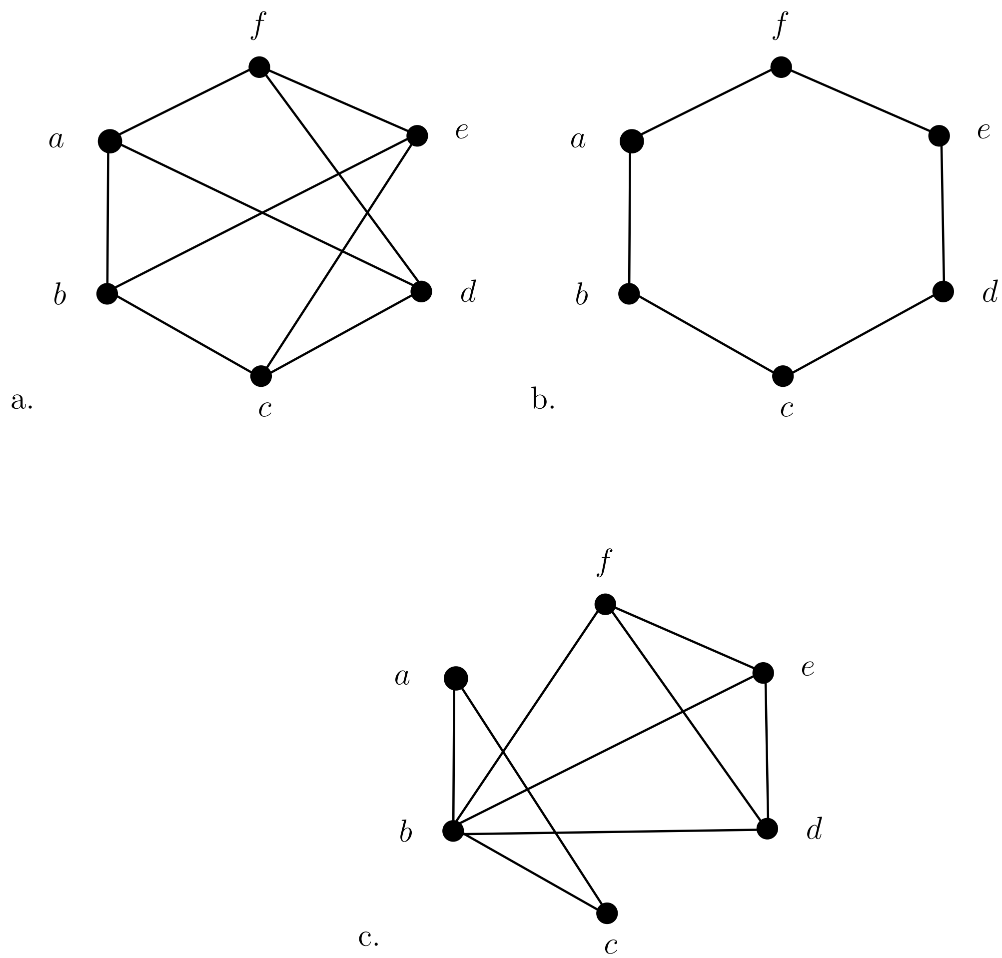


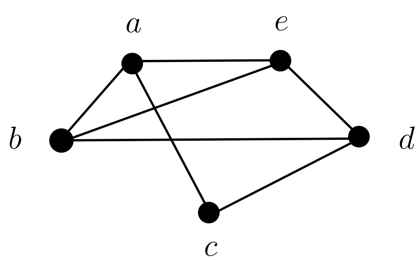
Figure 15: Hamiltonian Paths and Cycles

We can see that the graph in Figure 15a has a Hamiltonian cycle. One such cycle can be listed as a, b, e, c, d, f, a, another as d, f, e, c, b, a, d. Notice that it is not required that every edge of the graph be used when visiting each vertex exactly once. The graph in Figure 15b also has a Hamiltonian cycle. One such cycle can be listed as a, b, c, d, e, f, a. The graph in Figure 15c does not have a Hamiltonian cycle.

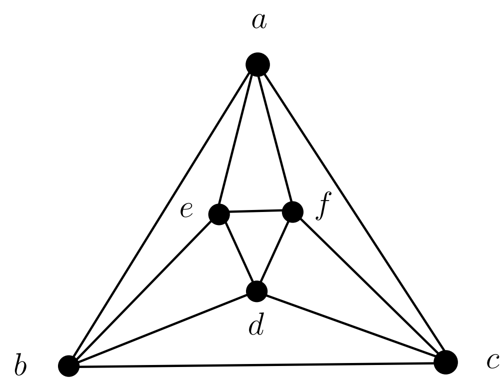
We can easily observe that both Figure 15a and Figure 15b also have a Hamiltonian path, because by definition a Hamiltonian cycle is a Hamiltonian path that starts and ends at the same vertex. In the above examples, drop the last vertex listed in the given cycles and a path is then shown, for example Figure 15a has the path, a, d, e, c, d, f. Thus, every graph that has a Hamiltonian cycle has a Hamiltonian path. However, sometimes there are graphs that have a Hamiltonian path but do not have a Hamiltonian cycle. For example, Figure 15c has a Hamiltonian path but no cycle. One such path can be listed as c, a, b, d, e, f.

7.2 In Class Exercises

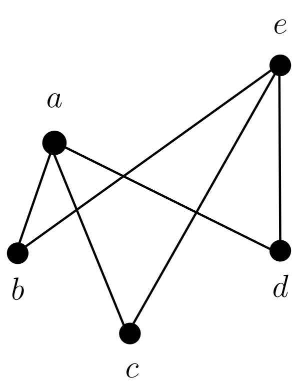
Determine if the following graphs below contain a Hamiltonian cycle. If the graph does not contain a Hamiltonian cycle, does it contain a Hamiltonian path?



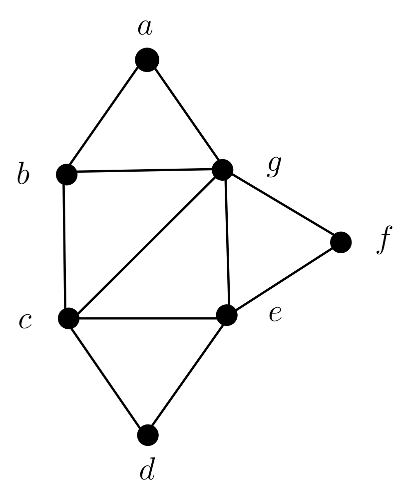
(a)



(b)



(c)



(d)

Hamiltonian cycles are interesting to mathematicians because there is no straightforward test for determining if a graph has a Hamiltonian cycle. Later we will examine three different approaches for trying to determine if a graph has a Hamiltonian cycle. For now, the different approaches are the best that we can do when trying to find a Hamiltonian cycle. Mathematicians have concluded that finding a general approach that will work for any graph may be impossible.

However, mathematicians have been able to come up with several conditions that guarantee that a given graph has a Hamiltonian cycle. The following theorem guarantees the existence of a Hamiltonian cycle for certain types of graphs.

> If a connected graph, G, has n vertices, where $n \geq 3$ and every vertex in G has a degree of at least $\frac{n}{2}$, then G has a Hamiltonian cycle [9].

Let us now go back to Figure 15a and check the degree of each vertex. Since each of the six vertices of the graph has a degree of at least $\frac{6}{2} = 3$, the graph has a Hamiltonian cycle. Thus, before we even tried to find a Hamiltonian cycle on our own, this theorem guarantees that we will find at least one. Unfortunately, this theorem does not tell us how to find an actual cycle in the graph.

If a graph has some vertices with a degree less than $\frac{n}{2}$, then the theorem does not apply. This does not automatically mean that the graph does not have a Hamiltonian cycle; the graph may or may not have a Hamiltonian cycle. Both Figure 15b and

Figure 15c has vertices with a degree less than 3, thus so far no conclusions can be drawn. However, after further inspection we did discover that Figure 15b did have a Hamiltonian cycle and Figure 15c did not.

History of Hamiltonian graphs

In 1857, Sir William Rowan Hamiltonian, a well-known Irish mathematician, invented a game consisting of a regular solid dodecahedron made out of wood and some string. Note that a dodecahedron has 20 vertices, 30 edges and 12 faces, where each face is shaped like a regular pentagon. Hamilton had a peg placed at each vertex to represent a famous city of the time. The object of the game was for the player to travel to each city exactly once; the player starts and ends at the same vertex and can only travel from one city to the next if an edge exits between the vertices. The string was used to visually show the route that the player traveled by having the player wind the string around the pegs as he/she was playing the game. Unfortunately, since the wooden dodecahedron was difficult to travel with, the game was not very popular [4, 9].

Subsequently, Hamilton decided to make another version of the game called the Icosian Game of the Around the World game. He “flattened” the dodecahedron into a vertex-edge graph, and had holes represent the vertices, so that a peg could be placed in the holes and moved around the graph in search of the tour that started and ended at the same vertex. Unfortunately, the game was still not a huge success, probably because this game is not that difficult to solve [4, 9].

Below, in Figure 16, is a graph of the dodecahedron. Play Hamilton’s Icosian game by finding an around the world tour. Note that the around the world tour is simply a Hamiltonian cycle.

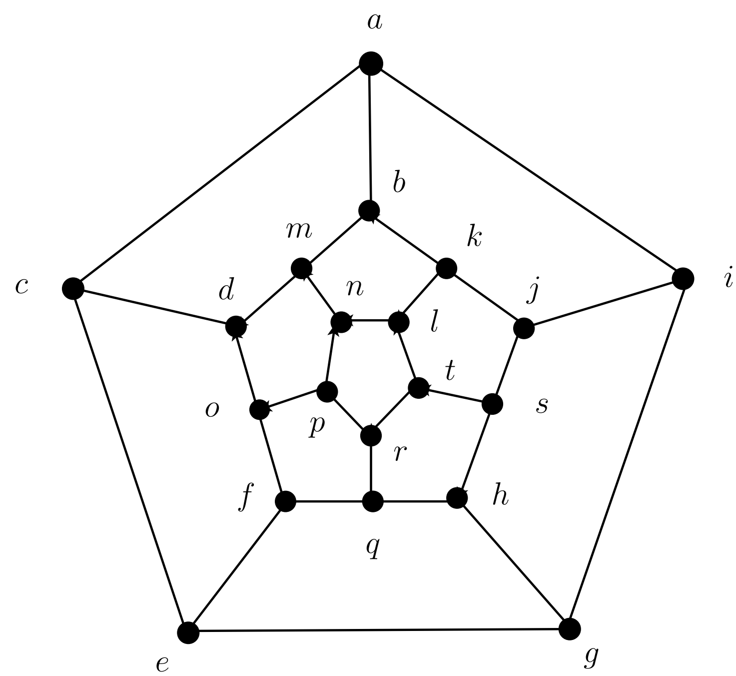


Figure 16: Hamilton's Icosian Game

7.4 Application

Now that we have an understanding of the concept of Hamiltonian paths and cycles, how can we use Hamiltonian paths and cycles in day-to-day situations?

Potential Problems

1. Determine the ranks of a team in a competition where each team plays every other team.

2. The Traveling Salesperson Problem.

Before we jump into looking at an example of Problem 1, we need to first define another graph theory term. We know that it is often useful for the edges of a graph to have direction. Think about a competition where each player plays every other player. We can use graph theory to illustrate this idea. A complete graph, where the vertices represent the players and a directed edge from vertex A to vertex B shows that player A defeats player B, is a type of digraph known as a **tournament**. A remarkable fact about this kind of digraph is that every tournament contains a Hamiltonian path. This means that, at the end of a tournament, it is possible to rank the teams in order from winner to loser.

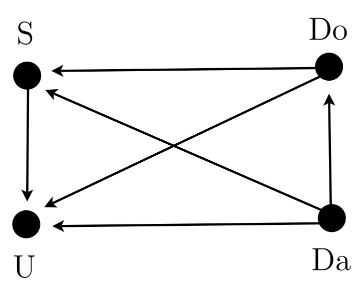
Now let us do an example of Problem 1. Suppose four soccer teams play in the high school round robin tournament. The matrix below shows the results of the tournament. Give the rankings of the teams in the soccer round robin tournament.

High School Tournament Results

	S	Do	Da	U
Science Hill (S)	0	0	0	1
Dobyns Bennett (Do)	1	0	0	1
Daniel Boone (Da)	1	1	0	1
Unicoi Co (U)	0	0	0	0

Remember that the matrix is read from row to column, with a "1" representing a win. For example, the "1" in the Daniel Boone-Dobyns Bennett entry means that Daniel Boone beat Dobyns Bennett.

Through our process of mathematical modeling, we begin with a given real-world situation. Now we need to translate the above grid into a graph model. Let us have the vertices of our graph represent the different high schools. Using the information of which team won in each meet, a tournament digraph can be constructed.



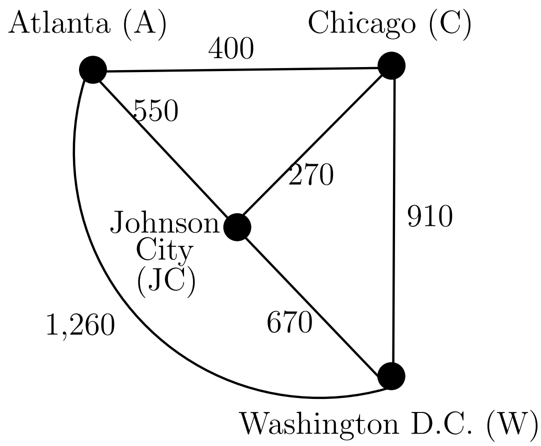
After examining the digraph, we can see that the only Hamiltonian path is Da-Do-S-U. So, let us interpret our solution in terms of our real-world problem. The Hamiltonian path gives a sequence of teams where each team beats the next. Thus, the Hamiltonian path Da-Do-S-U also serves as a ranking. Daniel Boone High School placed first, Dobyns Bennett High School placed second, Science Hill third, and Unicoi Co High School last.

Now we will do an example of Problem 2. Suppose that you are a salesperson who lives in Johnson City. You want to travel to several different cities, say Chicago, Atlanta, Washington D.C., exactly once and then return home to Johnson City. The list below represents the trips that are available to you and the costs of making a trip between the cities.

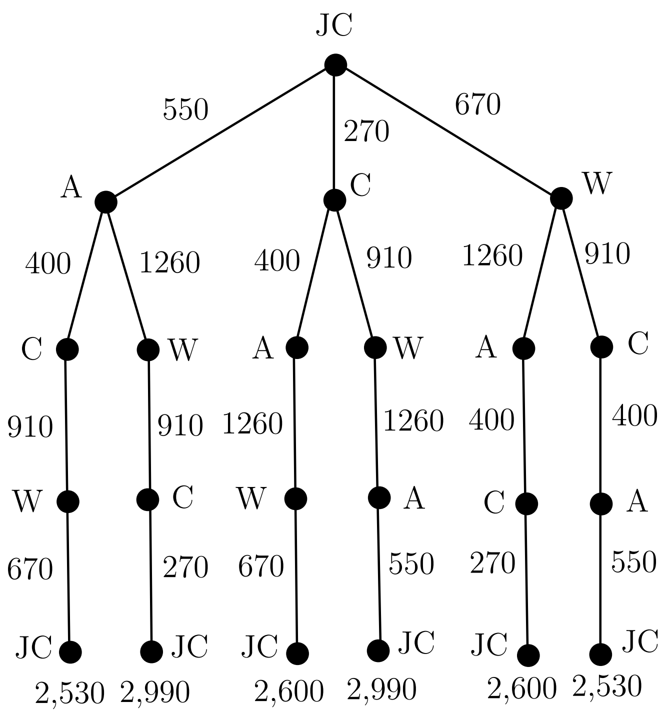
Cities	Cost
Atlanta - Chicago	$400
Atlanta - Johnson City	$550
Atlanta - Washington D.C.	$1,260
Chicago - Johnson City	$270
Chicago - Washington D.C.	$910
Johnson City - Washington D.C.	$670

Since you own your business, it is important to use the least amount of money for your trip. To save money, try to find the least expensive route that begins in Johnson City, visits each of the other cities exactly once, and returns to Johnson City.

Now we need to translate the above list into a graph model. Let us have the vertices of our graph represent the different cities. Two vertices are connected by an edge if a trip can be made between the two cities. The weight attached to each edge is the amount of money it will cost to travel between the two cities.



The first way that we will try to solve the problem of finding minimum total weight is to list every possible cycle, along with its cost. Using a tree diagram like the one below will help us sort out all of the possible cycles.



After looking at the diagram it is easy to see that out of all the possible routes, the best solution comes from the cycle that consists of JC, A, C, W, JC or the cycle in the reverse order, JC, W, C, A, JC. Therefore the least expensive route that begins in Johnson City, visits each of the other cities exactly once, and returns to Johnson City costs $2,530, where the trip itinerary goes as follows: Johnson City to Washington D.C. to Chicago to Atlanta back to Johnson City, or the reverse.

The method that we just used is called the **brute-force method**. You might think that, since this method guarantees a solution, we have found a general solution to the Traveling Salesperson Problem. However, we can quickly see that as the number of vertices increases, checking all the possible routes becomes almost impossible! Even with the world's fastest computer, it would take millions of years to compute the weights of every cycle for a graph with 25 vertices. [9] Thus, we have found a method that guarantees a solution but is not very efficient because it can only be used on graphs with a small number of vertices.

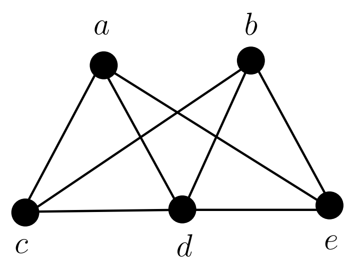
Another way that we will try to solve the problem of finding minimum total weight is to begin at a vertex, look for the nearest vertex, move to it, and so on until you complete the cycle. This method is called the **nearest-neighbor method**. Looking back, let us start at JC, and then move to the nearest neighboring vertex, then to the nearest vertex not yet visited, and return to JC when all of the other cities have been visited. In this case the nearest vertex is the vertex that costs the least amount of money. Therefore, the cycle would start at Johnson City, then Chicago, Atlanta, Washington D.C., and then back to Johnson City. So in this case the minimum weight is $270 + 400 + 1,260 + 670 = 2,600$. Thus, the trip would cost your business $2,600.

But, we already discovered that the cheapest round trip would cost $2,530. So, even though the nearest-neighbor method gives a very quick solution it will not always give the correct solution. Now you can understand why there is not a general solution for the traveling salesperson problem that will work in all situations. Either the brute-force method is chosen, which guarantees the best route but is prohibitively slow for large graphs or the nearest-neighbor method, which is quick but does not guarantee the best solution.

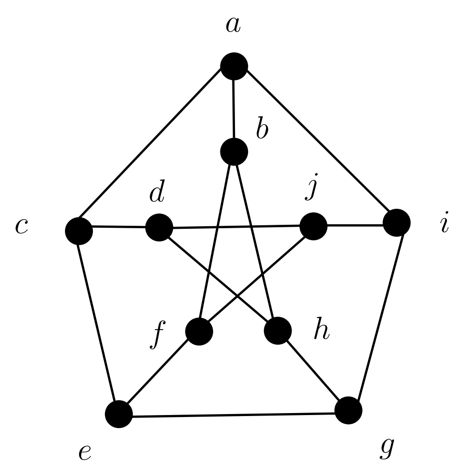
Mathematical Computations

1. Determine if the following graphs below contain a Hamiltonian cycle. If the graph does not contain a Hamiltonian cycle, does it contain a Hamiltonian path? (Do not forget to use the theorem that we learned about some Hamiltonian graphs!)

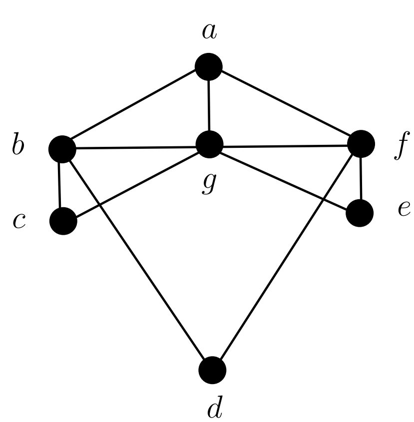
(a)



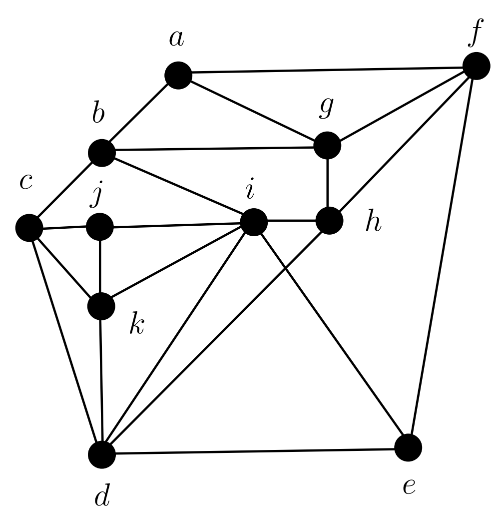
(b)



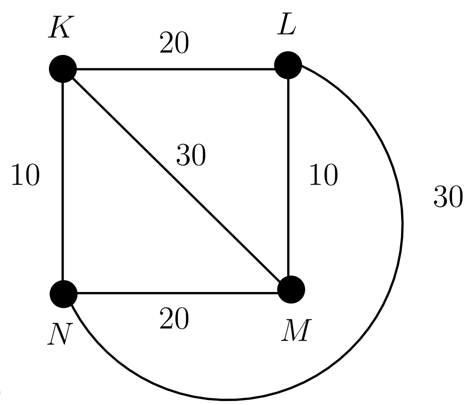
(c)



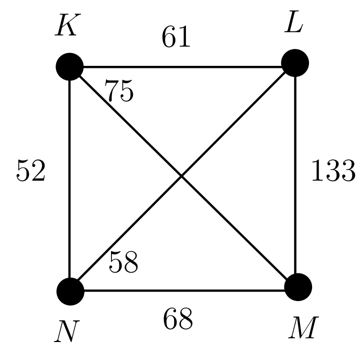
(d)



2. Find a minimum-weight Hamiltonian cycle that begins and ends at K. First find the minimum-weight Hamiltonian cycle by using the brute-force method, and then use the nearest neighbor method. Does the nearest-neighbor method give the shortest cycle possible?



(a)



(b)

1. Suppose four girls play in the Elida High School round robin girls' tennis tournament. The matrix below shows the results of the tournament. Give the rankings of the girls in the round robin tennis tournament.

Elida High School Tournament Results

	A	B	C	D	E
Amy(A)	0	1	1	0	1
Beth(B)	0	0	1	0	0
Cathy(C)	0	0	0	0	0
Dana(D)	1	1	1	0	1
Emily(E)	0	1	1	0	0

2. Suppose that you are a salesperson who lives in Johnson City. You need to travel to several different cities in Tennessee, Knoxville, Chattanooga, and Memphis exactly once and then return home to Johnson City. The list below represents the trips that are available to you and the miles between the cities.

Cities	Miles
Chattanooga - Johnson City	217
Chattanooga - Knoxville	112
Chattanooga - Memphis	344
Johnson City - Knoxville	107
Johnson City - Memphis	497
Knoxville - Memphis	392

Since you are taking the company's car, it is important to travel the minimum amount of miles on your trip. To save mileage, try to find the minimum route that begins in Johnson City, visits each of the other cities exactly once, and returns to Johnson City.

First find the minimum-weight Hamiltonian route by using the brute-force method, and then use the nearest-neighbor method. Does the nearest-neighbor method give the shortest possible route?

Problem Solving Computations

1. Draw a connected Hamiltonian graph that has five vertices. Label your vertices and provide the Hamiltonian cycle in your graph.

2. Draw a tournament with five players, where player A beats everyone, C beats everyone except A, B is beaten by everyone, and D beats E.

3. Draw a tournament with five players, A, B, C, D, and E, where there is a three-way tie between A, D, and E for first place.