Discrete Mathematics, KOM1062
Lecture #8

Instructor: Dr. Yavuz Eren

Lecture Book: **"Discrete Mathematics, Seventh Edt., Kenneth H. Rosen, 2007, McGraw Books Discrete Mathematics and Applications, Susanna S. Epp, Brooks, 4th Edt., 2011".**

Spring 2024

Connectivity

Many problems can be modeled with paths formed by traveling along the edges of graphs.

For instance, the problem of determining whether a message can be sent between two computers using intermediate links can be studied with a graph model.

Problems of efficiently planning routes for mail delivery, garbage pickup, diagnostics in computer networks, and so on can be solved using models that involve paths in graphs.

Informally, a **path** is a sequence of edges that begins at a vertex of a graph and travels from vertex to vertex along edges of the graph.

As the path travels along its edges, it visits the vertices along this path, that is, the endpoints of these edges.

Definition: Let *n be a nonnegative integer and G an undirected graph. A path of length n from u* to *v in G is a sequence of n edges* $e_1, \ldots, e_n$ *of G for which there exists a sequence*

$$x_0 = u, x_1, \ldots, x_{n-1}, x_n = v$$

of vertices such that e_i *has, for* $i = 1, \ldots, n$, *the endpoints* x_{i-1} and x_i .

When it is not necessary to distinguish between multiple edges, we will denote a path

$$e_1, e_2, \ldots, e_n,$$

where e_i is associated with $\{x_{i-1}, x_i\}$ for $i = 1, 2, \ldots, n$ by its vertex sequence $x_0, x_1, \ldots, x_n$. This notation identifies a path only as far as which vertices it passes through.

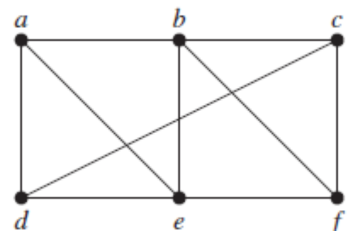
Consequently, it does not specify a unique path when there is more than one path that passes through this sequence of vertices, which will happen if and only if there are multiple edges between some successive vertices in the list. Note that a path of length zero consists of a single vertex.

Ex.: In the simple graph shown in Figure below, *a, d, c, f, e is a simple path of length 4, because {a, d}, {d, c}, {c, f }, and {f, e} are all edges.*

However, d, e, c, a is not a path, because {e, c} is not an edge.

Note that *b, c, f, e, b is a circuit of length 4 because {b, c}, {c, f }, {f, e}, and {e, b} are* edges, and this path begins and ends at *b.*

The path a, b, e, d, a, b, which is of length 5, is not simple because it contains the edge *{a, b} twice*.

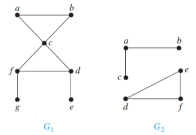


Definition: Let *n be a nonnegative integer and G a directed graph. A path of length n from u to v in G is a* sequence of edges $e_1, e_2, \ldots, e_n$ *of G such that* e_1 *is associated with* (x_0, x_1), e_2 *is associated* with (x_1, x_2), *and so on, with* e_n *associated with* (x_{n-1}, x_n), *where* $x_0 = u$ *and* $x_n = v$. *When* there are no multiple edges in the directed graph, this path is denoted by its vertex sequence $x_0, x_1, x_2, \ldots, x_n$.

A path of length greater than zero that begins and ends at the same vertex is called a *circuit or cycle. A path or circuit is called simple if it does not contain the same* edge more than once.

Definition : An undirected graph is called *connected if there is a path between every pair of distinct* vertices of the graph. An undirected graph that is not *connected is called disconnected. We* say that we *disconnect a graph when we remove vertices or edges, or both, to produce a* disconnected subgraph.

Ex. :The graph G_1 *in the Figure below is connected, because for every pair of distinct vertices there is a* path between them. However, the graph G_2 *in the Figure is not* connected. For instance, there is no path in G_2 *between vertices a and d.*

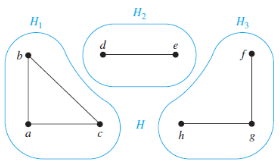


A connected component of a graph G is a connected subgraph of G that is not a proper subgraph of another connected subgraph of G.

That is, a connected component of a graph G is a maximal connected subgraph of G. A graph G that is not connected has two or more connected components that are disjoint and have G as their union.

Ex.: What are the connected components of the graph H shown in Figure below?

Solution: The graph H is the union of three disjoint connected subgraphs H1, H2, and H3, shown in figure below. These three subgraphs are the connected components of H.



The level of connectedness: Suppose that a graph represents a computer network. Knowing that this graph is connected tells us that any two computers on the network can communicate.

However, we would also like to understand how reliable this network is.

For instance, will it still be possible for all computers to communicate after a router or a communications link fails?

To answer this and similar questions, we now develop some new concepts.

Sometimes the removal from a graph of a vertex and all incident edges produces a subgraph with more connected components.

Such vertices are called cut vertices (or articulation points). The removal of a cut vertex from a connected graph produces a subgraph that is not connected. Analogously, an edge whose removal produces a graph with more connected components than in the original graph is called a cut edge or bridge.

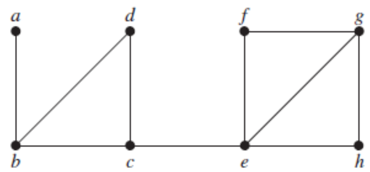
Note that in a graph representing a computer network, a cut vertex and a cut edge represent an essential router and an essential link that cannot fail for all computers to be able to communicate.

Ex. :Find the cut vertices and cut edges in the graph G_1 *shown in Figure below.*

***Solution**:* The cut vertices of G_1 *are b, c, and e.*

The removal of one of these vertices (and its adjacent edges) disconnects the graph.

The cut edges are *{a, b} and {c, e}. Removing either one* of these edges disconnects G_1.



G_1

Vertex Connectivity: Not all graphs have cut vertices.

For example, the complete graph K_n*, where* $n \geq 3$*, has no cut vertices.*

When you remove a vertex from K_n *and all edges* incident to it, the resulting subgraph is the complete graph K_{n-1}*, a connected graph.*

Connec*ted* graphs without cut vertices are called non-separable graphs, and can be thought of as more connected than those with a cut vertex.

We can extend this notion by defining a more granulated measure of graph connectivity based on the minimum number of vertices that can be removed to disconnect a graph.

We define the vertex connectivity of a non-complete graph *G, denoted by* $\kappa(G)$*, as the minimum number of vertices in a vertex* cut.

When *G is a complete graph, it has no vertex cuts, because removing any subset of its vertices* and all incident edges still leaves a complete graph.

Consequently, we cannot define $\kappa(G)$ *as the* minimum number of vertices in a vertex cut when *G is complete. Instead, we set* $\kappa(K_n) = n - 1$*,* the number of vertices needed to be removed to produce a graph with a single vertex.

The larger $\kappa(G)$ *is, the more connected we consider G to be.*

Disconnected graphs and K_1 have $\kappa(G) = 0$*, connected graphs with cut vertices and* K_2 *have* $\kappa(G) = 1$*, graphs without cut* vertices that can be disconnected by removing two vertices and K_3 *have* $\kappa(G) = 2$*, and so* on.

We say that a graph is *k-connected (or k-vertex-connected), if* $\kappa(G) \geq k$*. A graph G is 1-*connected if it is connected and not a graph containing a single vertex; a graph is 2-connected, or biconnected, if it is nonseparable and has at least three vertices.

Note that if *G is a k-connected* graph, then *G is a j -connected graph for all j with* $0 \leq j \leq k$*.*

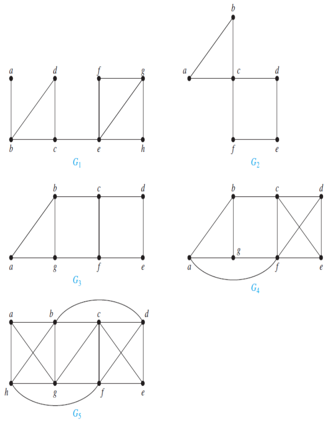
Ex.: Find the vertex connectivity for each of the graphs in figure below.

Solution: Because G_1 *is a connected graph with a cut* vertex, we know that $\kappa(G_1) = 1$.

Similarly , $\kappa(G_2) = 1$*, because c is a* cut vertex of G_2. G_3 *has no cut vertices, but that {b, g} is a vertex cut. Hence,* $\kappa(G_3) = 2$.

Similarly, because G_4 *has a vertex cut of size two, {c, f }, but no cut vertices. It follows that* $\kappa(G_4) = 2$.

G_5 *has no vertex cut of size two, but {b, c, f }* is a vertex cut of G_5. *Hence,* $\kappa(G_5) = 3$.



Edge Connectivity: We can also measure the connectivity of a connected graph G = (V ,E) in terms of the minimum number of edges that we can remove to disconnect it.

If a graph has a cut edge, then we need only remove it to disconnect G. If G does not have a cut edge, we look for the smallest set of edges that can be removed to disconnect it.

A set of edges E is called an edge cut of G if the subgraph G − E is disconnected.

The edge connectivity of a graph G, denoted by $\lambda(G)$, is the minimum number of edges in an edge cut of G.

This defines $\lambda(G)$ for all connected graphs with more than one vertex because it is always possible to disconnect such a graph by removing all edges incident to one of its vertices.

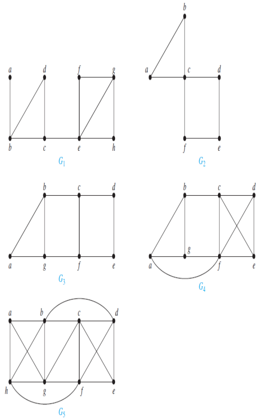
Note that $\lambda(G) = 0$ if G is not connected. We also specify that $\lambda(G) = 0$ if G is a graph consisting of a single vertex.

It follows that if G is a graph with n vertices, then $0 \le \lambda(G) \le n - 1$.

We can show that $\lambda(G) = n - 1$ where G is a graph with n vertices if and only if $G = K_n$, which is equivalent to the statement that $\lambda(G) \le n - 2$ when G is not a complete graph.

Ex.: Find the edge connectivity of each of the graphs in Figure of previous example.

Solution : Each of the five graphs in Figure above is connected and has more than one vertex, so we know that all of them have positive edge connectivity.



As we saw in the previous Example, G_1 *has a cut* edge, so *$\lambda(G_1) = 1$.*

The graph G_2 *has no cut edges, but the removal of the two edges* {*a, b*} *and* {*a, c*} *disconnects it. Hence,* $\lambda(G_2) = 2$.

Similarly, $\lambda(G_3) = 2$, *because* G_3 *has no* cut edges, but the removal of the two edges {*b, c*} *and* {*f, g*} *disconnects it.*

It can be verified that the removal of no two edges disconnects G_4, *but the removal* of the three edges {*b, c*}, {*a, f* }, *and* {*f, g*} *disconnects it. Hence,* $\lambda(G_4) = 3$.

Finally, $\lambda(G_5) = 3$, *because the removal of any two of its edges does not disconnect* it, but the removal of {*a, b*}, {*a, g*}, *and* {*a, h*} *does.*

AN INEQUALITY FOR VERTEX CONNECTIVITY AND EDGE CONNECTIVITY

When $G = (V, E)$ *is a non-complete connected graph with at least three vertices, the minimum* degree of a vertex of *G is an upper bound for both the vertex connectivity of G and the edge* connectivity of *G.*

That is, $\kappa(G) \leq \min_{v \in V} deg(v)$ *and* $\lambda(G) \leq \min_{v \in V} deg(v)$. *To see this,* observe that deleting all the neighbors of a fixed vertex of minimum degree disconnects *G, and* deleting all the edges that have a fixed vertex of minimum degree as an endpoint disconnects *G.*

$$\kappa(G) \leq \lambda(G) \leq \min_{v \in V} deg(v)$$

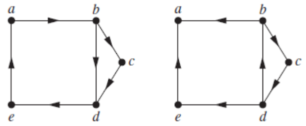
Connectedness in Directed Graphs

Definition: A directed graph is *strongly connected if there is a path from a to b and from b to a whenever a and b are vertices in the graph.*

Definition: A directed graph is *weakly connected if there is a path between every two vertices in the* underlying undirected graph.

Ex. : Are the directed graphs *G and H shown in figure below strongly connected? Are they weakly* connected?
Solution: *G* is strongly connected because there is a path between any two vertices in this directed graph. Hence, *G* is also weakly connected.

The graph *H* is not strongly connected. There is no directed path from *a* to *b* in this graph. However, *H* is weakly connected, because there is a path between any two vertices in the underlying undirected graph of *H* (the reader should verify this).



G H

Paths and Isomorphism

There are several ways that paths and circuits can help determine whether two graphs are isomorphic.

For example, the existence of a simple circuit of a particular length is a useful invariant that can be used to show that two graphs are not isomorphic.

In addition, paths can be used to construct mappings that may be isomorphisms.

A useful isomorphic invariant for simple graphs is the existence of a simple circuit of length *k, where k is a positive integer greater than 2.*

Ex.: Determine whether the graphs *G and H shown in Figure below are isomorphic.*

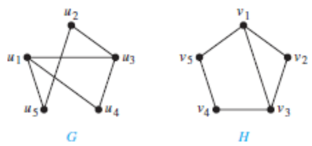
Solution:

Both *G* and *H* have five vertices and six edges,

both have a simple circuit of length three, a simple circuit of length four, and a simple circuit of length five.

both have two vertices of degree three and three vertices of degree two, and

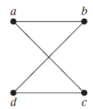
Because all these isomorphic invariants agree, *G* and *H* may be isomorphic.



Counting Paths Between Vertices

The number of paths between two vertices in a graph can be determined using its adjacency matrix.

Ex.: How many paths of length four are there from *a to d in the simple graph G in Figure below?*



Solution*:* The adjacency matrix of *G* (ordering the vertices as *a, b, c, d*) is

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

Hence, the number of paths of length four from *a* to *d* is the *(1, 4)*th entry of $\mathbf{A}^4$. Because

$$A^4 = \begin{bmatrix} 8 & 0 & 0 & 8 \\ 0 & 8 & 8 & 0 \\ 0 & 8 & 8 & 0 \\ 8 & 0 & 0 & 8 \end{bmatrix}$$

there are exactly eight paths of length four from *a* to *d*. By inspection of the graph, we see that *a, b, a, b, d*; *a, b, a, c, d*; *a, b, d, b, d*; *a, b, d, c, d*; *a, c, a, b, d*; *a, c, a, c, d*; *a, c, d, b, d*; and *a, c, d, c, d* are the eight paths of length four from *a* to *d*.

Theorem: Let *G be a graph with adjacency matrix A with respect to the ordering* $v_1, v_2, \ldots,$ v_n *of* the vertices of the graph (with directed or undirected edges, with multiple edges and loops allowed). The number of different paths of length r from v_i *to* v_j, *where r is a positive integer,* equals the *(i, j)th entry of* A^r.

Euler and Hamilton Paths

Can we travel along the edges of a graph starting at a vertex and returning to it by traversing each edge of the graph exactly once?

Similarly, can we travel along the edges of a graph starting at a vertex and returning to it while visiting each vertex of the graph exactly once?

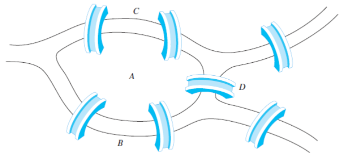
Although these questions seem to be similar,

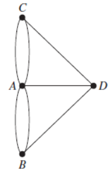
the first question, which asks whether a graph has an ***Euler circuit****, can be easily answered simply by examining the degrees of the vertices of the graph,*

while the second question, which asks whether a graph has a ***Hamilton circuit****, is quite difficult* to solve for most graphs.

Euler Paths and Circuits

The town of Königsberg, Prussia was divided into four sections by the branches of the Pregel River. These four sections included the two regions on the banks of the Pregel, Kneiphof Island, and the region between the two branches of the Pregel. In the eighteenth century seven bridges connected these regions. The following Figure depicts these regions and bridges.





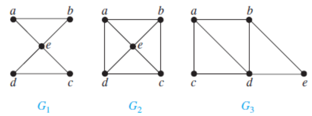
The townspeople took long walks through town on Sundays. They wondered whether it was possible to start at some location in the town, travel across all the bridges once without crossing any bridge twice, and return to the starting point.

The Swiss mathematician Leonhard Euler solved this solution, published in 1736, may be the first use of graph theory. Euler studied this problem using the multigraph obtained when the four regions are represented by vertices and the bridges by edges. This multigraph is shown in Figure above(rightmost).

The problem of traveling across every bridge without crossing any bridge more than once can be rephrased in terms of this model. The question becomes: Is there a simple circuit in this multigraph that contains every edge?

Definition: An *Euler circuit* in a graph *G* is a simple circuit containing every edge of *G*. An *Euler path* in *G* is a simple path containing every edge of *G*.

Ex.: Which of the undirected graphs in Figure below have an Euler circuit? Of those that do not, which have an Euler path?



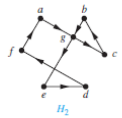
Solution:

The graph *G*1 has an Euler circuit, for example, *a, e, c, d, e, b, a*.

Neither of the graphs *G*2 or *G*3 has an Euler circuit (the reader should verify this).

*G*2 does not have an Euler path (as the reader should verify).

Ex. : Which of the directed graphs in Figure below have an Euler circuit? Of those that do not, which have an Euler path?



Solution:

The graph *H*2 has an Euler circuit, for example, *a, g, c, b, g, e, d, f, a*.

Neither *H*1 nor *H*3 has an Euler circuit (as the reader should verify).

*H*3 has an Euler path, namely, *c, a, b, c, d, b*, but *H*1 does not (as the reader should verify).

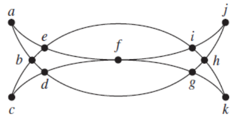
NECESSARY AND SUFFICIENT CONDITIONS FOR EULER CIRCUITS AND PATHS

There are simple criteria for determining whether a multigraph has an Euler circuit or an Euler path. Euler discovered them when he solved the famous Königsberg bridge problem. We will assume that all graphs discussed in this section have a finite number of vertices and edges.

What can we say if a connected multigraph has an Euler circuit? What we can show is that every vertex must have even degree. To do this, first note that an Euler circuit begins with a vertex *a and continues with an edge incident with a, say {a, b}. The edge {a, b} contributes one* to deg*(a). Each time the circuit passes through a vertex it contributes two to the vertex's degree,* because the circuit enters via an edge incident with this vertex and leaves via another such edge. Finally, the circuit terminates where it started, contributing one to deg*(a). Therefore, deg(a)* must be even, because the circuit contributes one when it begins, one when it ends, and two every time it passes through *a (if it ever does). A vertex other than a has even degree because* the circuit contributes two to its degree each time it passes through the vertex. We conclude that if a connected graph has an Euler circuit, then every vertex must have even degree.

Theorem : A connected multigraph with at least two vertices has an Euler circuit if and only if each of its vertices has even degree.

Ex.: Many puzzles ask you to draw a picture in a continuous motion without lifting a pencil so that no part of the picture is retraced. We can solve such puzzles using Euler circuits and paths. For example, can *Mohammed's scimitars, shown in Figure, be drawn in this way, where the* drawing begins and ends at the same point?



G

Solution:
We can solve this problem because the graph *G* shown in figure above has an Euler circuit.

It has such a circuit because all its vertices have even degree.

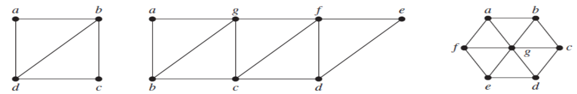
First, we form the circuit *a, b, d, c, b, e, i, f, e, a*. We obtain the subgraph *H* by deleting the edges in this circuit and all vertices that become isolated when these edges are removed.

Then we form the circuit *d, g, h, j, i, h, k, g, f, d* in *H*. After forming this circuit we have used all edges in *G*. Splicing this new circuit into the first circuit at the appropriate place produces the Euler circuit *a, b, d, g, h, j, i, h, k, g, f, d, c, b, e, i, f, e, a*.

This circuit gives a way to draw the scimitars without lifting the pencil or retracing part of the picture.

Theorem: A connected multigraph has an Euler path but not an Euler circuit if and only if it has exactly two vertices of odd degree.

Ex.: Which graphs shown in Figure below have an Euler path?



G_1 G_2 G_3

Solution:
*G*1 contains exactly two vertices of odd degree, namely, *b* and *d*.

Hence, it has an Euler path that must have *b* and *d* as its endpoints. One such Euler path is *d, a, b, c, d, b*.

Similarly, *G*2 has exactly two vertices of odd degree, namely, *b* and *d*.

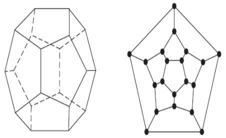
So it has an Euler path that must have *b* and *d* as endpoints. One such Euler path is *b, a, g, f, e, d, c, g, b, c, f, d*.

*G*3 has no Euler path because it has six vertices of odd degree.

Hamilton Paths and Circuits

A simple path in a graph G that passes through every vertex exactly once is called a *Hamilton path*, and a simple circuit in a graph G that passes through every vertex exactly once is called a *Hamilton circuit*. That is, the simple path $x_0, x_1, \ldots, x_{n-1}, x_n$ in the graph $G = (V, E)$ is a Hamilton path if $V = \{x_0, x_1, \ldots, x_{n-1}, x_n\}$ and $x_i \neq x_j$ for $0 \leq i < j \leq n$, and the simple circuit $x_0, x_1, \ldots, x_{n-1}, x_n, x_0$ (with $n > 0$) is a Hamilton circuit if $x_0, x_1, \ldots, x_{n-1}, x_n$ is a Hamilton path.

This terminology comes from a game, called the *Icosian puzzle, invented in 1857 by the* Irish mathematician Sir William Rowan Hamilton. It consisted of a wooden dodecahedron(Figure a), with a peg at each vertex of the dodecahedron, and string. The 20 vertices of the dodecahedron were labeled with different cities in the world. The object of the puzzle was to start at a city and travel along the edges of the dodecahedron, visiting each of the other 19 cities exactly once, and end back at the first city. The circuit traveled was marked off using the string and pegs.



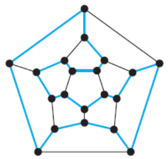
(a) (b)

Because the author cannot supply each reader with a wooden solid with pegs and string, we will consider the equivalent question:

Is there a circuit in the graph shown in Figure above(b) that passes through each vertex exactly once?

This solves the puzzle because this graph is isomorphic to the graph consisting of the vertices and edges of the dodecahedron.

A solution of Hamilton’s puzzle is shown in Figure below.

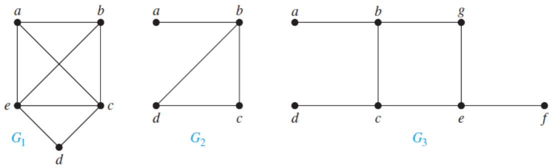


Ex.: Which of the simple graphs in figure below have a Hamilton circuit or, if not, a Hamilton path?

Solution:
*G*1 has a Hamilton circuit: *a, b, c, d, e, a*.

There is no Hamilton circuit in *G*2 (this can be seen by noting that any circuit containing every vertex must contain the edge {*a, b*} twice), but *G*2 does have a Hamilton path, namely, *a, b, c, d*.

*G*3 has neither a Hamilton circuit nor a Hamilton path, because any path containing all vertices must contain one of the edges {*a, b*}, {*e, f* }, and {*c, d*} more than once.



CONDITIONS FOR THE EXISTENCE OF HAMILTON CIRCUITS

Is there a simple way to determine whether a graph has a Hamilton circuit or path? At first, it might seem that there should be an easy way to determine this, because there is a simple way to answer the similar question of whether a graph has an Euler circuit.

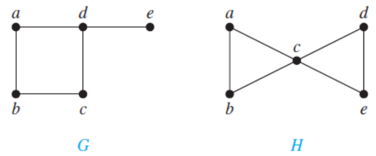
Surprisingly, there are no known simple necessary and sufficient criteria for the existence of Hamilton circuits. However, many theorems are known that give sufficient conditions for the existence of Hamilton circuits.

Also, certain properties can be used to show that a graph has no Hamilton circuit. For instance, a graph with a vertex of degree one cannot have a Hamilton circuit, because in a Hamilton circuit, each vertex is incident with two edges in the circuit.

Moreover, if a vertex in the graph has degree two, then both edges that are incident with this vertex must be part of any Hamilton circuit.

Also, note that when a Hamilton circuit is being constructed and this circuit has passed through a vertex, then all remaining edges incident with this vertex, other than the two used in the circuit, can be removed from consideration. Furthermore, a Hamilton circuit cannot contain a smaller circuit within it.

Ex. :Show that neither graph displayed in Figure below has a Hamilton circuit.



Solution:

There is no Hamilton circuit in *G* because *G* has a vertex of degree one, namely, *e*.

Now consider *H*. Because the degrees of the vertices *a*, *b*, *d*, and *e* are all two, every edge incident with these vertices must be part of any Hamilton circuit.

It is now easy to see that no Hamilton circuit can exist in *H*, for any Hamilton circuit would have to contain four edges incident with *c*, which is impossible.

DIRAC'S THEOREM: If *G is a simple graph with n vertices with n ≥ 3 such that the* degree of every vertex in *G is at least n/2, then G has a Hamilton circuit.*

ORE'S THEOREM: If *G is a simple graph with n vertices with n ≥ 3 such that* deg*(u) + deg(v) ≥ n for every pair of nonadjacent vertices u and v in G, then G has a* Hamilton circuit.

Applications of Hamilton Circuits

Hamilton paths and circuits can be used to solve practical problems.

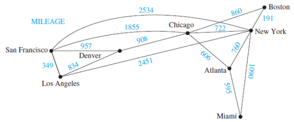
For example, many applications ask for a path or circuit that visits each road intersection in a city, each place pipelines intersect in a utility grid, or each node in a communications network exactly once.

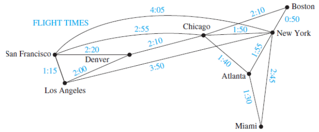
Finding a Hamilton path or circuit in the appropriate graph model can solve such problems. The famous traveling salesperson problem asks for the shortest route a traveling salesperson should take to visit a set of cities.

This problem reduces to finding a Hamilton circuit in a complete graph such that the total weight of its edges is as small as possible

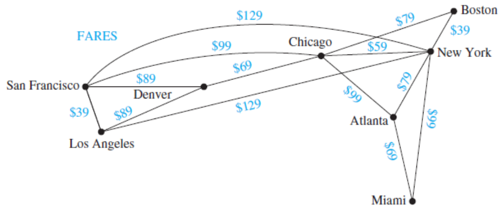
Shortest-Path Problems

Many problems can be modeled using graphs with weights assigned to their edges. As an illustration, consider how an airline system can be modeled. We set up the basic graph model by representing cities by vertices and flights by edges. Problems involving distances can be modeled by assigning distances between cities to the edges. Problems involving flight time can be modeled by assigning flight times to edges. Problems involving fares can be modeled by assigning fares to the edges. Following figures displays three different assignments of weights to the edges of a graph representing distances, flight times, and fares, respectively.





33



Graphs that have a number assigned to each edge are called weighted graphs.

Several types of problems involving weighted graphs arise frequently.

Determining a path of least length between two vertices in a network is one such problem.

To be more specific, let the length of a path in a weighted graph be the sum of the weights of the edges of this path.

Note that this use of the term *length is different from the use of length to denote the* number of edges in a path in a graph without weights.

34

The question is: What is a shortest path, that is, a path of least length, between two given vertices?

For instance, in the airline system represented by the weighted graph shown in the Figure, what is a shortest path in air distance between Boston and Los Angeles?

What combinations of flights has the smallest total flight time (that is, total time in the air, not including time between flights) between Boston and Los Angeles?

What is the cheapest fare between these two cities?

Another important problem involving weighted graphs asks for a circuit of shortest total length that visits every vertex of a complete graph exactly once.

This is the famous *traveling salesperson problem, which asks for an order in which a salesperson should visit each of the* cities on his route exactly once so that he travels the minimum total distance.

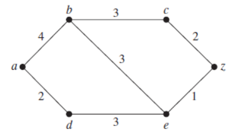
A Shortest-Path Algorithm

There are several different algorithms that find a shortest path between two vertices in a weighted graph.

We will present a greedy algorithm discovered by the Dutch mathematician Edsger Dijkstra in 1959.

The version we will describe solves this problem in undirected weighted graphs where all the weights are positive. It is easy to adapt it to solve shortest-path problems in directed graphs.

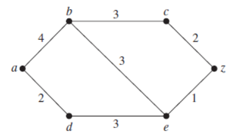
Ex.: What is the length of a shortest path between *a and z in the weighted graph shown in Figure below?*



Solution: Although a shortest path is easily found by inspection, we will develop some ideas useful in understanding Dijkstra's algorithm. We will solve this problem by finding the length of a shortest path from *a* to successive vertices, until *z* is reached.

The only paths starting at *a* that contain no vertex other than *a* are formed by adding an edge that has *a* as one endpoint. These paths have only one edge. They are *a, b* of length 4 and *a, d* of length 2. It follows that *d* is the closest vertex to *a*, and the shortest path from *a* to *d* has length 2.

We can find the second closest vertex by examining all paths that begin with the shortest path from *a* to a vertex in the set {*a, d*}, followed by an edge that has one endpoint in {*a, d*} and its other endpoint not in this set. There are two such paths to consider, *a, d, e* of length 7 and *a, b* of length 4. Hence, the second closest vertex to *a* is *b* and the shortest path from *a* to *b* has length 4.



To find the third closest vertex to *a*, we need examine only the paths that begin with the shortest path from *a* to a vertex in the set {*a, d, b*}, followed by an edge that has one endpoint in the set {*a, d, b*} and its other endpoint not in this set. There are three such paths, *a, b, c* of length 7, *a, b, e* of length 7, and *a, d, e* of length 5. Because the shortest of these paths is *a, d, e*, the third closest vertex to *a* is *e* and the length of the shortest path from *a* to *e* is 5.

Let us to find the fourth closest vertex to *a*, we need examine only the paths that begin with the shortest path from *a* to a vertex in the set {*a, d, b, e*}, followed by an edge that has one endpoint in the set {*a, d, b, e*} and its other endpoint not in this set. There are two such paths, *a, b, c* of length 7 and *a, d, e, z* of length 6. Because the shorter of these paths is *a, d, e, z*, the fourth closest vertex to *a* is *z* and the length of the shortest path from *a* to *z* is 6.

Dijkstra's algorithm proceeds by finding the length of a shortest path from a to a first vertex, the length of a shortest path from a to a second vertex, and so on, until the length of a shortest path from a to z is found.

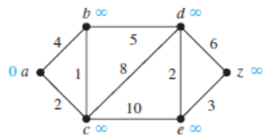
As a side benefit, this algorithm is easily extended to find the length of the shortest path from a to all other vertices of the graph, and not just to z.

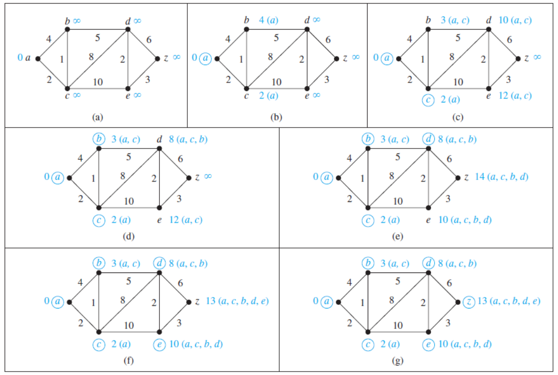
We will now consider the general problem of finding the length of a shortest path between a and z in an undirected connected simple weighted graph. Dijkstra's algorithm proceeds by finding the length of a shortest path from a to a first vertex, the length of a shortest path from a to a second vertex, and so on, until the length of a shortest path from a to z is found.

This algorithm is easily extended to find the length of the shortest path from a to all other vertices of the graph, and not just to z. The algorithm relies on a series of iterations. A distinguished set of vertices is constructed by adding one vertex at each iteration.

A labeling procedure is carried out at each iteration. In this labeling procedure, a vertex w is labeled with the length of a shortest path from a to w that contains only vertices already in the distinguished set. The vertex added to the distinguished set is one with a minimal label among those vertices not already in the set.

Ex.: Use Dijkstra's algorithm to find the length of a shortest path between the vertices *a and z in the* weighted graph displayed in Figure(a) below.



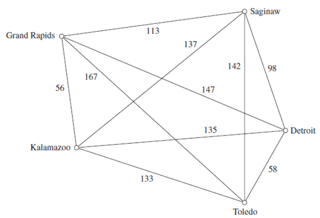


The steps used by Dijkstra's algorithm to find a shortest path between *a and z are shown* in Figure above. At each iteration of the algorithm the vertices of the set S_k *are circled. A shortest* path from *a to each vertex containing only vertices in* S_k *is indicated for each iteration. The* algorithm terminates when *z is circled. We find that a shortest path from a to z is a, c, b, d, e, z,* with length 13.

The Traveling Salesman Problem

We now discuss an important problem involving weighted graphs. Consider the following problem: A traveling salesperson wants to visit each of *n cities exactly once and return to his starting* point.

For example, suppose that the salesperson wants to visit Detroit, Toledo, Saginaw, Grand Rapids, and Kalamazoo. In which order should he visit these cities to travel the minimum total distance?

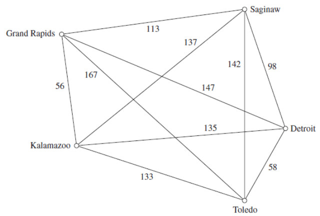


Route	*Total Distance (miles)*
Detroit–Toledo–Grand Rapids–Saginaw–Kalamazoo–Detroit	610
Detroit–Toledo–Grand Rapids–Kalamazoo–Saginaw–Detroit	516
Detroit–Toledo–Kalamazoo–Saginaw–Grand Rapids–Detroit	588
Detroit–Toledo–Kalamazoo–Grand Rapids–Saginaw–Detroit	458
Detroit–Toledo–Saginaw–Kalamazoo–Grand Rapids–Detroit	540
Detroit–Toledo–Saginaw–Grand Rapids–Kalamazoo–Detroit	504
Detroit–Saginaw–Toledo–Grand Rapids–Kalamazoo–Detroit	598
Detroit–Saginaw–Toledo–Kalamazoo–Grand Rapids–Detroit	576
Detroit–Saginaw–Kalamazoo–Toledo–Grand Rapids–Detroit	682
Detroit–Saginaw–Grand Rapids–Toledo–Kalamazoo–Detroit	646
Detroit–Grand Rapids–Saginaw–Toledo–Kalamazoo–Detroit	670
Detroit–Grand Rapids–Toledo–Saginaw–Kalamazoo–Detroit	728

To solve this problem we can assume the salesperson starts in Detroit (because this must be part of the circuit) and examine all possible ways for him to visit the other four cities and then return to Detroit (starting elsewhere will produce the same circuits).

There are a total of 24 such circuits, but because we travel the same distance when we travel a circuit in reverse order, we need only consider 12 different circuits to find the minimum total distance he must travel.

We list these 12 different circuits and the total distance traveled for each circuit. As can be seen from the list, the minimum total distance of 458 miles is traveled using the circuit Detroit–Toledo–Kalamazoo–Grand Rapids–Saginaw–Detroit (or its reverse).



Route	*Total Distance (miles)*
Detroit–Toledo–Grand Rapids–Saginaw–Kalamazoo–Detroit	610
Detroit–Toledo–Grand Rapids–Kalamazoo–Saginaw–Detroit	516
Detroit–Toledo–Kalamazoo–Saginaw–Grand Rapids–Detroit	588
Detroit–Toledo–Kalamazoo–Grand Rapids–Saginaw–Detroit	458
Detroit–Toledo–Saginaw–Kalamazoo–Grand Rapids–Detroit	540
Detroit–Toledo–Saginaw–Grand Rapids–Kalamazoo–Detroit	504
Detroit–Saginaw–Toledo–Grand Rapids–Kalamazoo–Detroit	598
Detroit–Saginaw–Toledo–Kalamazoo–Grand Rapids–Detroit	576
Detroit–Saginaw–Kalamazoo–Toledo–Grand Rapids–Detroit	682
Detroit–Saginaw–Grand Rapids–Toledo–Kalamazoo–Detroit	646
Detroit–Grand Rapids–Saginaw–Toledo–Kalamazoo–Detroit	670
Detroit–Grand Rapids–Toledo–Saginaw–Kalamazoo–Detroit	728

The traveling salesperson problem asks for the circuit of minimum total weight in a weighted, complete, undirected graph that visits each vertex exactly once and returns to its starting point.

This is equivalent to asking for a Hamilton circuit with minimum total weight in the complete graph, because each vertex is visited exactly once in the circuit.

The most straightforward way to solve an instance of the traveling salesperson problem is to examine all possible Hamilton circuits and select one of minimum total length.

How many circuits do we have to examine to solve the problem if there are *n vertices in the graph?*

Once a starting point is chosen, there are *(n − 1)! different Hamilton circuits to examine, because there* are *n − 1 choices for the second vertex, n − 2 choices for the third vertex, and so on.*

Because a Hamilton circuit can be traveled in reverse order, we need only examine *(n − 1)!/2 circuits to find* our answer. Note that *(n − 1)!/2 grows extremely rapidly. Trying to solve a traveling salesperson* problem in this way when there are only a few dozen vertices is impractical.

For example, with 25 vertices, a total of 24!*/2 (approximately 3.1 × 1023) different Hamilton circuits would have* to be considered. If it took just one nanosecond (10−9 second) to examine each Hamilton circuit, a total of approximately ten million years would be required to find a minimum-length Hamilton circuit in this graph by exhaustive search techniques.

Because the traveling salesperson problem has both practical and theoretical importance, a great deal of effort has been devoted to devising efficient algorithms that solve it.

However, no algorithm with polynomial worst-case time complexity is known for solving this problem.

A practical approach to the traveling salesperson problem when there are many vertices to visit is to use an approximation algorithm.

These are algorithms that do not necessarily produce the exact solution to the problem but instead are guaranteed to produce a solution that is close to an exact solution.

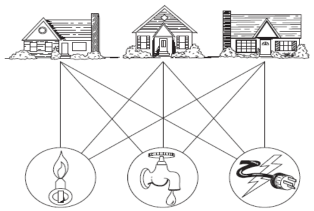
Planar Graphs

Consider the problem of joining three houses to each of three separate utilities, as shown in Figure below. Is it possible to join these houses and utilities so that none of the connections cross?

This problem can be modeled using the complete bipartite graph $K_{3,3}$*. The original question* can be rephrased as: Can $K_{3,3}$ *be drawn in the plane so that no two of its edges cross?*

In this lecture we will study the question of whether a graph can be drawn in the plane without edges crossing. In particular, we will answer the houses-and-utilities problem.

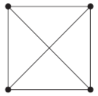
There are always many ways to represent a graph. When is it possible to find at least one way to represent this graph in a plane without any edges crossing?



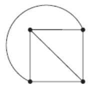
Definition: A graph is called *planar if it can be drawn in the plane without any edges crossing (where* a crossing of edges is the intersection of the lines or arcs representing them at a point other than their common endpoint). Such a drawing is called a *planar representation of the graph.*

A graph may be planar even if it is usually drawn with crossings, because it may be possible to draw it in a different way without crossings.

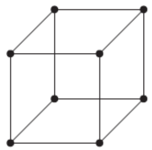
Ex. : Is K_4 *(shown in Figure below with two edges crossing) planar?*



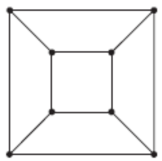
Solution: *K*4 is planar because it can be drawn without crossings, as shown



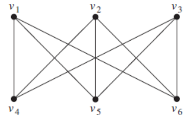
Ex.: Is *Q3, shown in Figure below, planar?*



Solution: *Q*3 is planar, because it can be drawn without any edges crossing, as shown



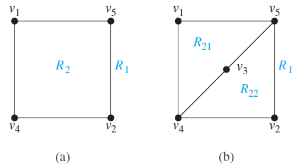
Ex. : Is $K_{3,3}$ *shown in Figure below, planar?*



Solution: Any attempt to draw $K3,3$ in the plane with no edges crossing is doomed.
We now show why. In any planar representation of $K3,3$, the vertices $v1$ and $v2$ must be connected to both $v4$ and $v5$.

These four edges form a closed curve that splits the plane into two regions, $R1$ and $R2$, as shown in figure a below. The vertex $v3$ is in either $R1$ or $R2$.

When $v3$ is in $R2$, the inside of the closed curve, the edges between $v3$ and $v4$ and between $v3$ and $v5$ separate $R2$ into two subregions, $R21$ and $R22$, as shown in figure b.



Note that: Planarity of graphs plays an important role in the design of electronic circuits.

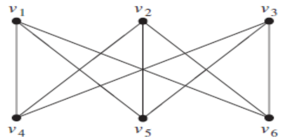
We can model a circuit with a graph by representing components of the circuit by vertices and connections between them by edges.

We can print a circuit on a single board with no connections crossing if the graph representing the circuit is planar. When this graph is not planar, we must turn to more expensive options. For example, we can partition the vertices in the graph representing the circuit into planar subgraphs. We then construct the circuit using multiple layers.

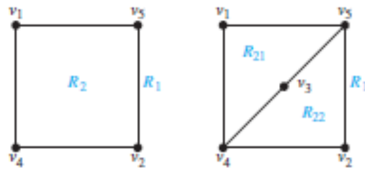
The planarity of graphs is also useful in the design of road networks. Suppose we want to connect a group of cities by roads.

We can model a road network connecting these cities using a simple graph with vertices representing the cities and edges representing the highways connecting them.We can built this road network without using underpasses or overpasses if the resulting graph is planar.

Ex.: Is $K_{3,3}$ *shown in Figure below, planar?*



Solution: Any attempt to draw *K*3,3 in the plane with no edges crossing is doomed. In any planar representation of *K*3,3, the vertices *v*1 and *v*2 must be connected to both *v*4 and *v*5. These four edges form a closed curve that splits the plane into two regions, *R*1 and *R*2, as shown in Figure 7(a). The vertex *v*3 is in either *R*1 or *R*2. When *v*3 is in *R*2, the inside of the closed curve, the edges between *v*3 and *v*4 and between *v*3 and *v*5 separate *R*2 into two subregions, *R*21 and *R*22, as shown in Figure 7(b).



(a) (b)

51

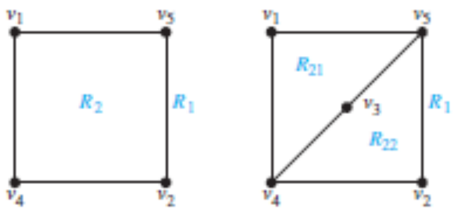
51

Next, note that there is no way to place the final vertex *v*6 without forcing a crossing.

For if *v*6 is in *R*1, then the edge between *v*6 and *v*3 cannot be drawn without a crossing.

If *v*6 is in *R*21, then the edge between *v*2 and *v*6 cannot be drawn without a crossing. If *v*6 is in *R*22, then the edge between *v*1 and *v*6 cannot be drawn without a crossing.

A similar argument can be used when *v*3 is in *R*1. The completion of this argument is left for the reader. It follows that *K*3,3 is not planar.



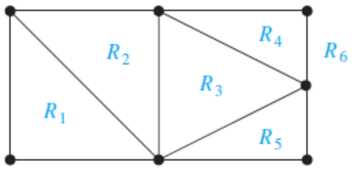
(a) (b)

Euler's Formula

A planar representation of a graph splits the plane into regions, including an unbounded region. For instance, the planar representation of the graph shown in Figure below splits the plane into six regions. These are labeled in the Figure.

Euler showed that all planar representations of a graph split the plane into the same number of regions. He accomplished this by finding a relationship among the number of regions, the number of vertices, and the number of edges of a planar graph.

Teorem(Euler's Formula): Let *G be a connected planar simple graph with e edges and v* vertices. Let *r be the number of regions in a planar representation of G. Then r = e − v + 2.*



Ex.: Suppose that a connected planar simple graph has 20 vertices, each of degree 3. Into how many regions does a representation of this planar graph split the plane?

Solution: This graph has 20 vertices, each of degree 3, so $v = 20$. Because the sum of the degrees of the vertices, $3v = 3*20 = 60$, is equal to twice the number of edges, $2e$, we have $2e = 60$, or $e = 30$.

Consequently, from Euler's formula, the number of regions is

$r = e - v + 2 = 30 - 20 + 2 = 12$.

Euler's formula can be used to establish some inequalities that must be satisfied by planar graphs. (following Corollaries).

Corollary 1: If *G is a connected planar simple graph with e edges and v vertices, where* $v \geq 3$, *then* $e \leq 3v - 6$.

Corollary 2: If *G is a connected planar simple graph, then G has a vertex of degree not exceeding five.*

This corollary can be used to demonstrate that K_5 *is non-planar.*

Ex. : Show that K_5 *is nonplanar using Corollary 1.*

Solution: The graph $K5$ has five vertices and 10 edges.

However, the inequality $e \leq 3v - 6$ is not satisfied for this graph because $e = 10$ and $3v - 6 = 9$.

Therefore, $K5$ is not planar.

It was previously shown that $K3,3$ *is not planar. Note, however, that this graph has six vertices* and nine edges. This means that the inequality $e = 9 \leq 12 = 3*6 - 6$ *is satisfied. Consequently,* the fact that the inequality $e \leq 3v - 6$ *is satisfied does not imply that a graph is planar. However,* the following corollary of Theorem(Euler's Formula) can be used to show that $K_{3,3}$ *is nonplanar.*

Corollary 3: If a connected planar simple graph has *e edges and v vertices with* $v \geq 3$ *and no circuits of* length three, then $e \leq 2v - 4$.

Ex.: Use this Corollary to show that $K_{3,3}$ *is nonplanar.*
Solution:

Because $K3,3$ has no circuits of length three (this is easy to see because it is bipartite), corollary can be used. $K3,3$ has six vertices and nine edges. Because $e = 9$ and $2v - 4 = 8$, Corollary 3 shows that $K3,3$ is nonplanar.

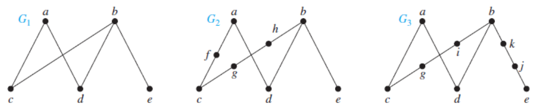
Kuratowski's Theorem

We have seen that $K_{3,3}$ *and* K_5 *are not planar. Clearly, a graph is not planar if it contains either* of these two graphs as a subgraph.

Surprisingly, all nonplanar graphs must contain a subgraph that can be obtained from $K_{3,3}$ *or* K_5 *using certain permitted operations.* If a graph is planar, so will be any graph obtained by removing an edge *{u, v} and adding a* new vertex *w together with edges {u,w} and {w, v}. Such an operation is called an elementary* subdivision.

The graphs $G_1 = (V_1,E_1)$ *and* $G_2 = (V_2,E_2)$ *are called homeomorphic if they* can be obtained from the same graph by a sequence of elementary subdivisions.

Ex. : Show that the graphs *G1, G2, and G3 displayed in Figure below are all homeomorphic.*



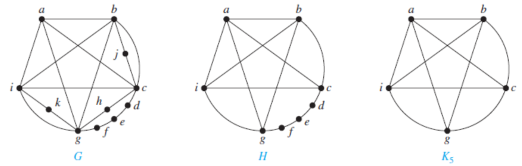
Solution:
These three graphs are homeomorphic because all three can be obtained from *G*1 by elementary subdivisions.

*G*1 can be obtained from itself by an empty sequence of elementary subdivisions. To obtain *G*2 from *G*1 we can use this sequence of elementary subdivisions:
(i) remove the edge {*a, c*}, add the vertex *f* , and add the edges {*a, f* } and {*f, c*};
(ii) remove the edge {*b, c*}, add the vertex *g*, and add the edges {*b, g*} and {*g, c*}; and
(iii) remove the edge {*b, g*}, add the vertex *h*, and add the edges {*g, h*} and {*b, h*}.

We leave it to the reader to determine the sequence of elementary subdivisions needed to obtain *G*3 from *G*1.

Theorem : A graph is nonplanar if and only if it contains a subgraph homeomorphic to $K_{3,3}$ *or* K_5.

Ex. : Determine whether the graph *G shown in Figure below is planar.*



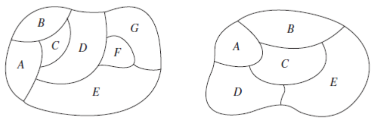
Solution:

G has a subgraph *H* homeomorphic to *K*5. *H* is obtained by deleting *h, j* , and *k* and all edges incident with these vertices.

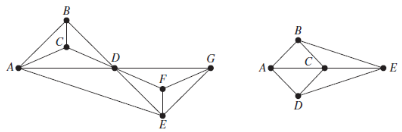
H is homeomorphic to *K*5 because it can be obtained from *K*5 (with vertices *a*, *b*, *c*, *g*, and *i*) by a sequence of elementary subdivisions, adding the vertices *d*, *e*, and *f* . (The reader should construct such a sequence of elementary subdivisions.) Hence, *G* is nonplanar.

Graph Coloring

Consider the problem of determining the least number of colors that can be used to color a map so that adjacent regions never have the same color. For instance, for the map shown on the left in Figure below, four colors suffice, but three colors are not enough. (The reader should check this.) In the map on the right in Figure below, three colors are sufficient (but two are not).



Each map in the plane can be represented by a graph. To set up this correspondence, each region of the map is represented by a vertex. Edges connect two vertices if the regions represented by these vertices have a common border. Two regions that touch at only one point are not considered adjacent. The resulting graph is called the dual graph of the map. Following Figure displays the dual graphs that correspond to the maps shown in Figure above.



Definition: A *coloring of a simple graph is the assignment of a color to each vertex of the graph so that* no two adjacent vertices are assigned the same color.

Note that**,** a graph can be colored by assigning a different color to each of its vertices. However, for most graphs a coloring can be found that uses fewer colors than the number of vertices in the graph. What is the least number of colors necessary?

Definition: The *chromatic number of a graph is the least number of colors needed for a coloring of* this graph. The chromatic number of a graph *G is denoted by χ(G).*

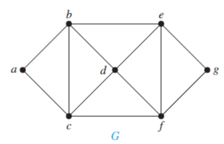
Note that asking for the chromatic number of a planar graph is the same as asking for the minimum number of colors required to color a planar map so that no two adjacent regions are assigned the same color.

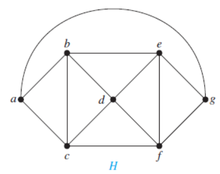
Theorem : The chromatic number of a planar graph is no greater than four.

Note that the four color theorem applies only to planar graphs.

Two things are required to show that the chromatic number of a graph is *k. First, we must* show that the graph can be colored with *k colors. This can be done by constructing such a* coloring. Second, we must show that the graph cannot be colored using fewer than *k colors.*

Ex. : What are the chromatic numbers of the simle graphs *G and H shown in figure below?*
Solution:





Ex. : What is the chromatic number of K_n?
Solution:

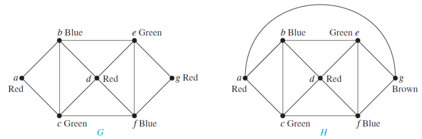
The chromatic number of G is at least three, because the vertices a, b, and c must be assigned different colors. To see if G can be colored with three colors, assign red to a, blue to b, and green to c.

Then, d can (and must) be colored red because it is adjacent to b and c. Furthermore, e can (and must) be colored green because it is adjacent only to vertices colored red and blue, and f can (and must) be colored blue because it is adjacent only to vertices colored red and green.

Finally, g can (and must) be colored red because it is adjacent only to vertices colored blue and green. This produces a coloring of G using exactly three colors. Following figure displays such a coloring. The graph H is made up of the graph G with an edge connecting a and g.

Any attempt to color H using three colors must follow the same reasoning as that used to color G, except at the last stage, when all vertices other than g have been colored.

Then, because g is adjacent (in H) to vertices colored red, blue, and green, a fourth color, say brown, needs to be used. Hence, H has a chromatic number equal to 4. A coloring of H is shown in Figure 4.



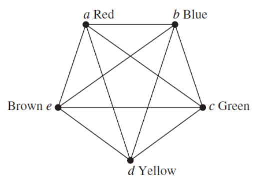
63

Ex. : What is the chromatic number of K_n?

Solution:
A coloring of *Kn* can be constructed using *n* colors by assigning a different color to each vertex. Is there a coloring using fewer colors? The answer is no.

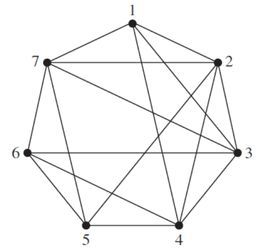
No two vertices can be assigned the same color, because every two vertices of this graph are adjacent.

Hence, the chromatic number of *Kn* is *n*. That is, $\chi(Kn) = n$. (Recall that *Kn* is not planar when $n \geq 5$, so this result does not contradict the four color theorem.) A coloring of *K*5 using five colors is shown in Figure 5.



64

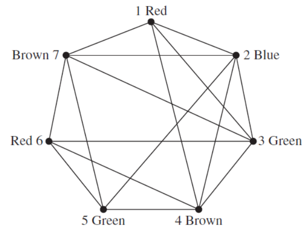
Ex. : Scheduling Final Exams How can the final exams at a university be scheduled so that no student has two exams at the same time?
Solution:



This scheduling problem can be solved using a graph model, with vertices representing courses and with an edge between two vertices if there is a common student in the courses they represent. Each time slot for a final exam is represented by a different color.

A scheduling of the exams corresponds to a coloring of the associated graph. For instance, suppose there are seven finals to be scheduled. Suppose the courses are numbered 1 through 7. Suppose that the following pairs of courses have common students: 1 and 2, 1 and 3, 1 and 4, 1 and 7, 2 and 3, 2 and 4, 2 and 5, 2 and 7, 3 and 4, 3 and 6, 3 and 7, 4 and 5, 4 and 6, 5 and 6, 5 and 7, and 6 and 7. In Figure 8 the graph associated with this set of classes
is shown. A scheduling consists of a coloring of this graph.

Because the chromatic number of this graph is 4 (the reader should verify this), four time slots are needed.Acoloring of the graph using four colors and the associated schedule are shown in figure.



Time Period	Courses
I	1, 6
II	2
III	3, 5
IV	4, 7