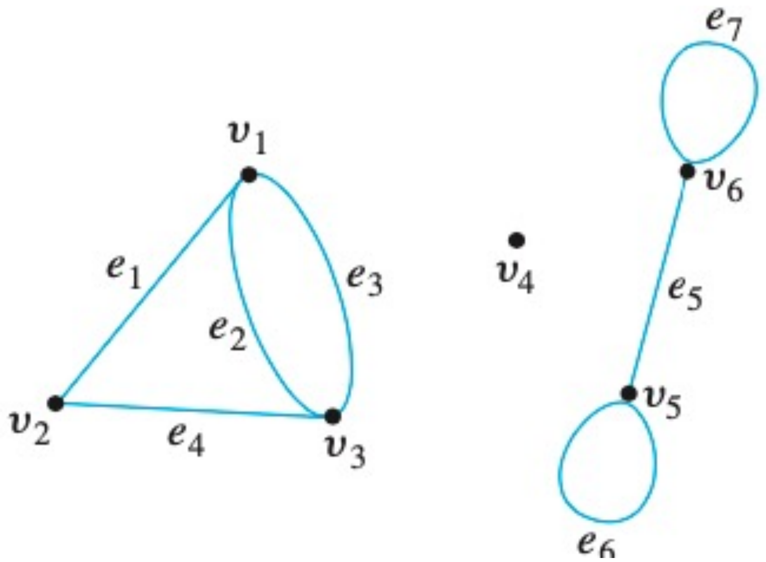
Graphs

Definition:

1. A **graph** G consists of two finite sets: a nonempty set $V(G)$ of **vertices** and a set $E(G)$ of **edges**, where each edge is associated with a set consisting of either one or two vertices called its **endpoints**. The correspondence from edges to endpoints is called the **edge-endpoint function**.
2. An edge with just one endpoint is called a **loop**, and two or more distinct edges with the same set of endpoints are said to be **parallel**.
3. An edge is said to **connect** its endpoints; two vertices that are connected by an edge are called **adjacent**; and a vertex that is an endpoint of a loop is said to be **adjacent to itself**.
4. An edge is said to be **incident on** each of its endpoints, and two edges incident on the same endpoint are called **adjacent**. A vertex on which no edges are incident is called **isolated**.

Example: Consider the following graph:

Write the vertex set and the edge set and give a table showing the edge-endpoint function.



Example: Consider the following graph:

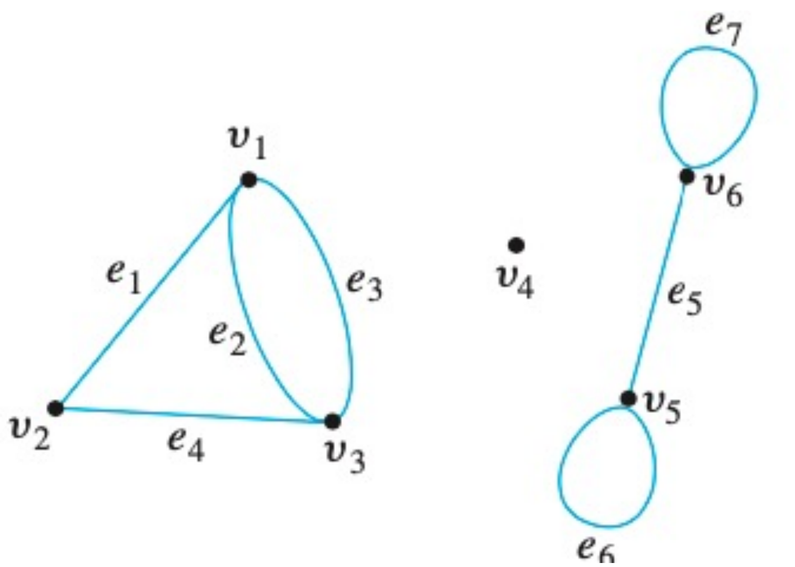
Write the vertex set and the edge set and give a table showing the edge-endpoint function.

vertex set= $\{v_1, v_2, v_3, v_4, v_5, v_6\}$

edge set= $\{e_1, e_2, e_3, e_4, e_5, e_6, e_7\}$

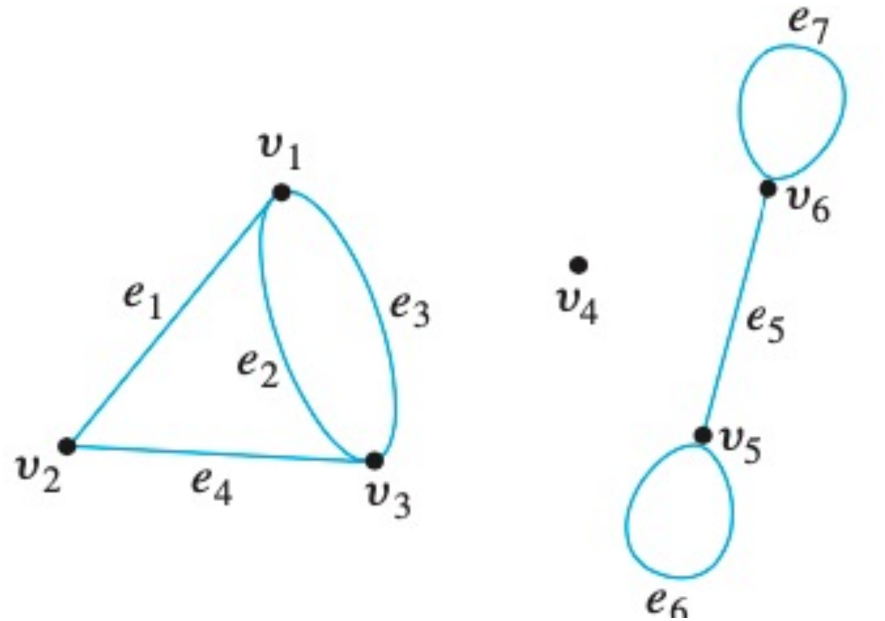
edge-endpoint function:

Edge	Endpoints
e_1	$\{v_1, v_2\}$
e_2	$\{v_1, v_3\}$
e_3	$\{v_1, v_3\}$
e_4	$\{v_2, v_3\}$
e_5	$\{v_5, v_6\}$
e_6	$\{v_5\}$
e_7	$\{v_6\}$



Example: Consider the following graph:

Find all edges that are incident on v_1, all vertices that are adjacent to v_1, all edges that are adjacent to e_1, all loops, all parallel edges, all vertices that are adjacent to themselves, and all isolated vertices.



Example: Consider the following graph:

Find all edges that are incident on v_1, all vertices that are adjacent to v_1, all edges that are adjacent to e_1, all loops, all parallel edges, all vertices that are adjacent to themselves, and all isolated vertices.

e_1, e_2, and e_3 are incident on v_1.

v_2 and v_3 are adjacent to v_1.

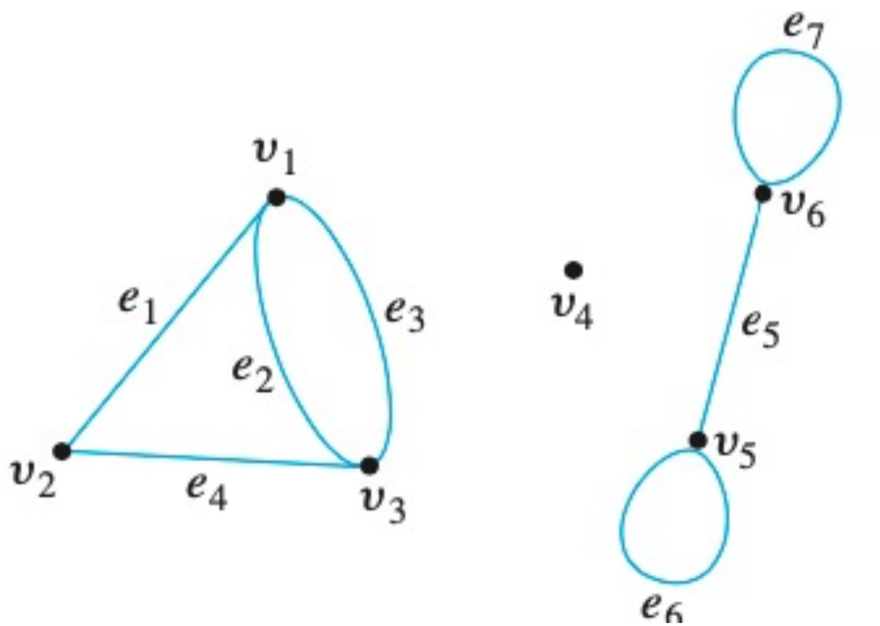
e_2, e_3, and e_4 are adjacent to e_1.

e_6 and e_7 are loops.

e_2 and e_3 are parallel.

v_5 and v_6 are adjacent to themselves.

v_4 is an isolated vertex.



Example:

Consider the graph specified as follows:

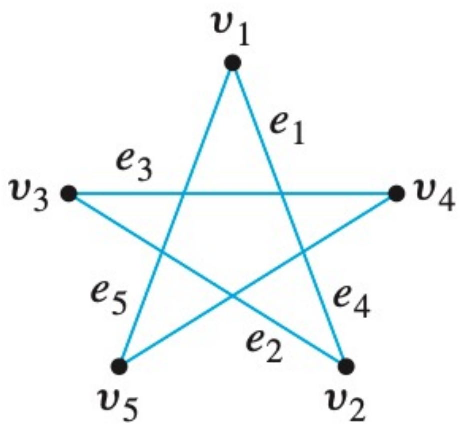
vertex set $= \{v_1, v_2, v_3, v_4\}$

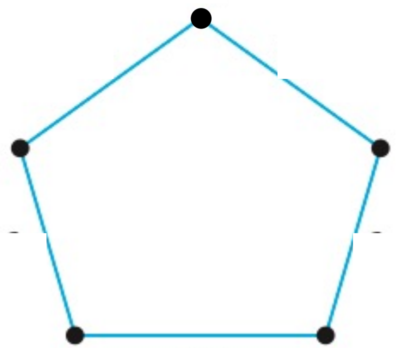
edge set $= \{e_1, e_2, e_3, e_4\}$

edge-endpoint function:

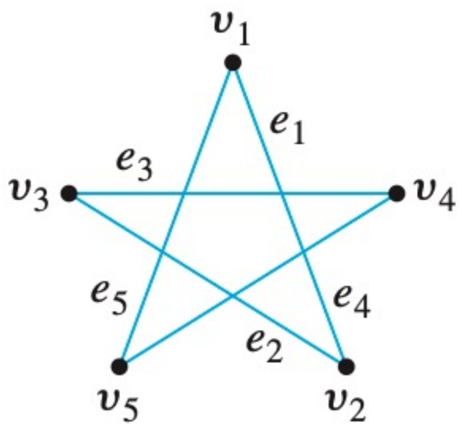
Edge	**Endpoints**
e_1	$\{v_1, v_3\}$
e_2	$\{v_2, v_4\}$
e_3	$\{v_2, v_4\}$
e_4	$\{v_3\}$

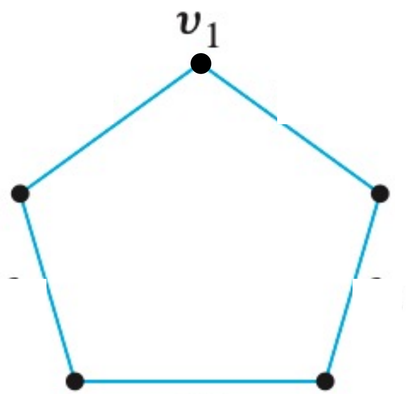
Example: Consider the two drawings shown in the figure. Label vertices and edges in such a way that both drawings represent the same graph.



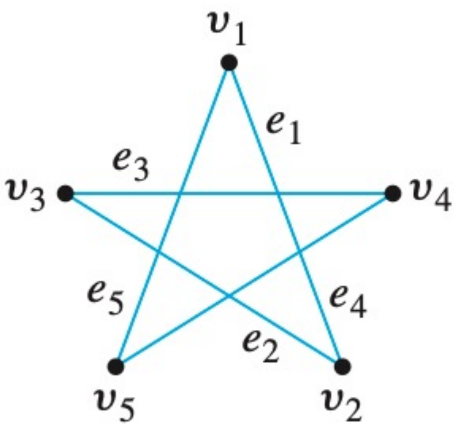


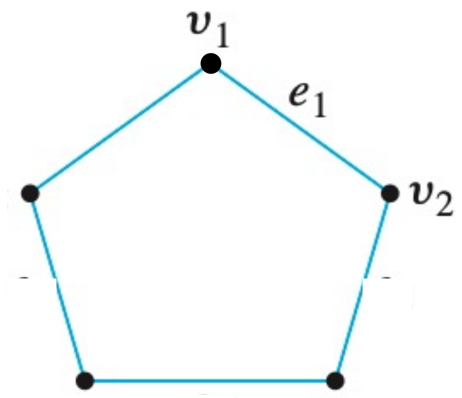
Example: Consider the two drawings shown in the figure. Label vertices and edges in such a way that both drawings represent the same graph.



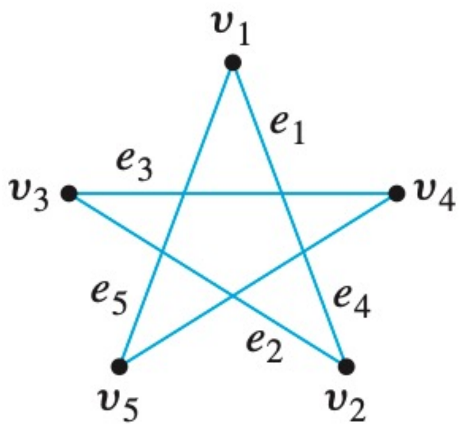


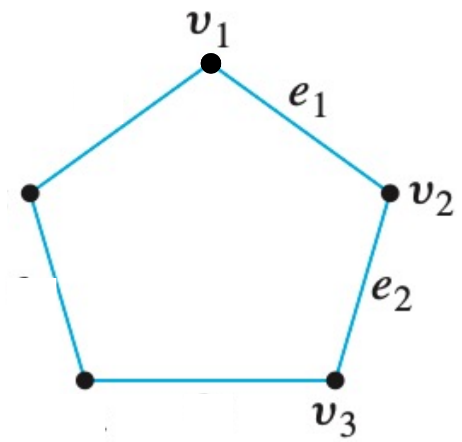
Example: Consider the two drawings shown in the figure. Label vertices and edges in such a way that both drawings represent the same graph.



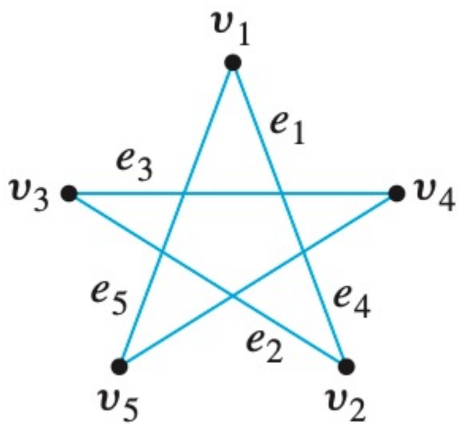


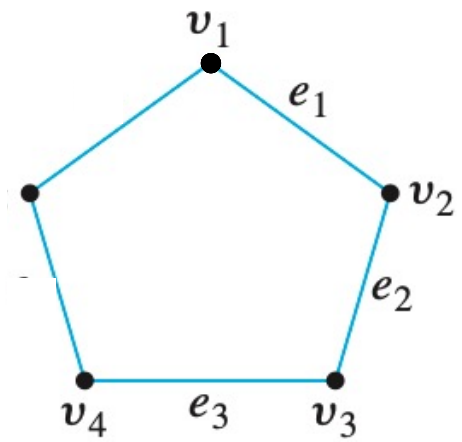
Example: Consider the two drawings shown in the figure. Label vertices and edges in such a way that both drawings represent the same graph.



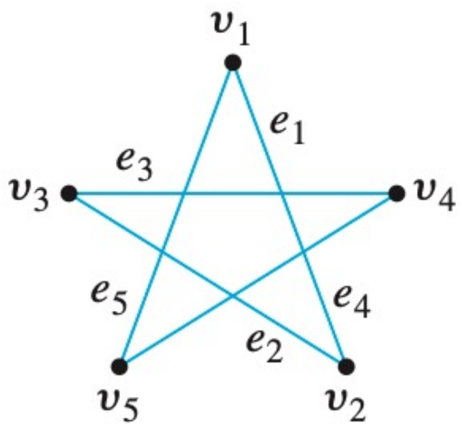


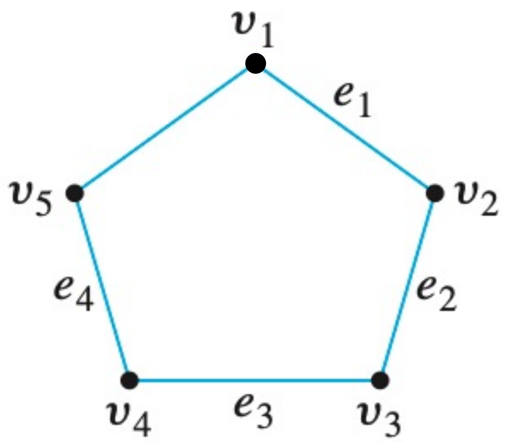
Example: Consider the two drawings shown in the figure. Label vertices and edges in such a way that both drawings represent the same graph.



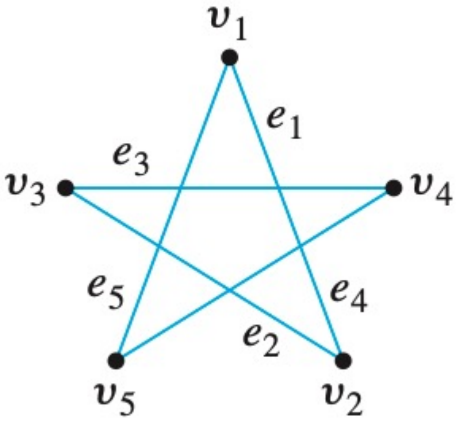


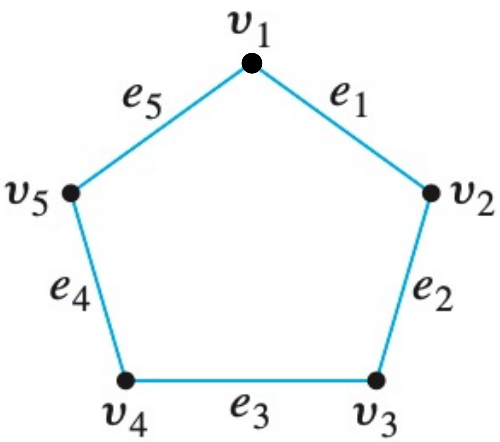
Example: Consider the two drawings shown in the figure. Label vertices and edges in such a way that both drawings represent the same graph.





Example: Consider the two drawings shown in the figure. Label vertices and edges in such a way that both drawings represent the same graph.





Definition:

A **directed graph,** or **digraph,** consists of two finite sets: a nonempty set $V(G)$ of vertices and a set $D(G)$ of directed edges, where each is associated with an ordered pair of vertices called its **endpoints.** If edge e is associated with the pair (v, w) of vertices, then e is said to be the (**directed**) **edge** from v to w.

Definition:

A **simple graph** is a graph that does not have any loops or parallel edges. In a simple graph, an edge with endpoints v and w is denoted $\{v, w\}$.

Definition:

Let n be a positive integer. A **complete graph on $\boldsymbol{n}$ vertices,** denoted $\boldsymbol{K_n}$, is a simple graph with n vertices and exactly one edge connecting each pair of distinct vertices.

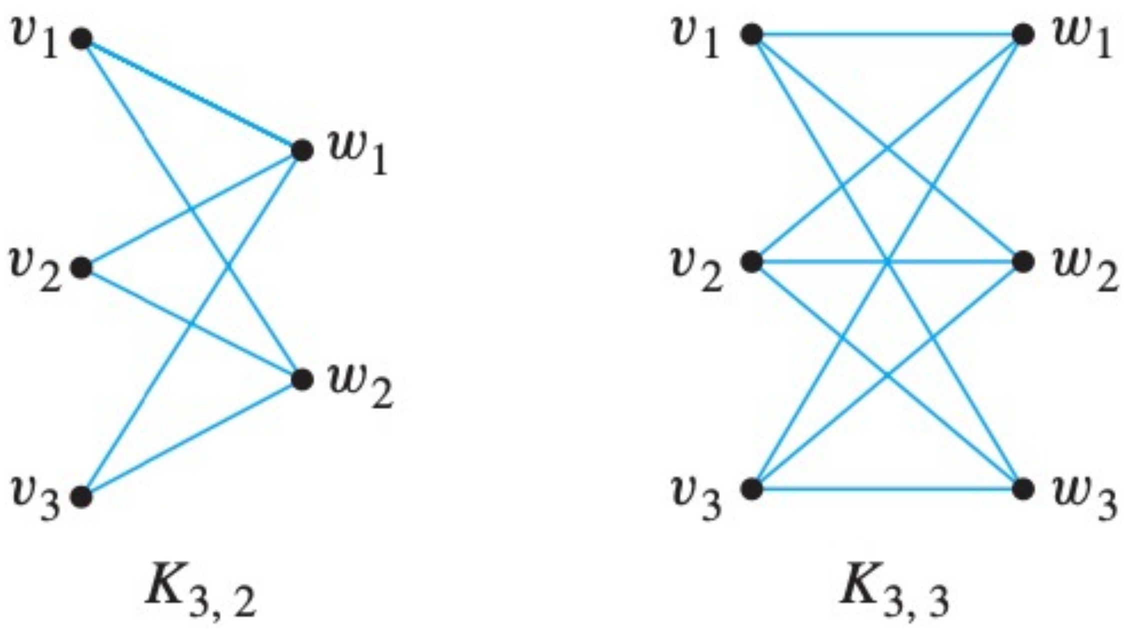
Definition:

Let m and n be positive integers. A **complete bipartite graph on** $(\boldsymbol{m}, \boldsymbol{n})$ **vertices,** denoted $\boldsymbol{K_{m,n}}$, is a simple graph with distinct vertices $v_1, v_2, \ldots, v_m$ and $w_1, w_2, \ldots, w_n$ that satisfies the following properties: For all $i, k = 1,2, \ldots, m$ and for all $j, l = 1,2, \ldots, n$,

1. There is an edge from each vertex v_i to each vertex w_j.
2. There is no edge from any vertex v_i to any other vertex v_k.
3. There is no edge from any vertex w_j to any other vertex w_l.

Example:

The complete bipartite graphs $K_{3,2}$ and $K_{3,3}$ are illustrated below.

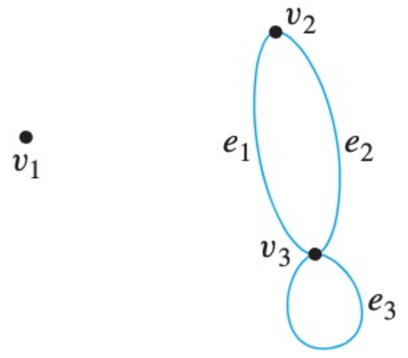


Definition:

Let *G* be a graph and v a vertex of *G*. The **degree of $\boldsymbol{v}$,** denoted **deg($\boldsymbol{v}$),** equals the number of edges that are incident on v, with an edge that is a loop counted twice. The **total degree of *G*** is the sum of the degrees of all the vertices of *G*.

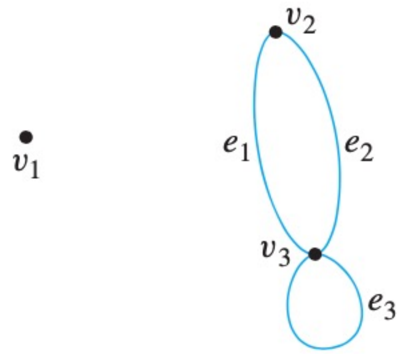
Example:

Find the degree of each vertex of the graph *G* shown below. Then find the total degree of *G*.



Example:

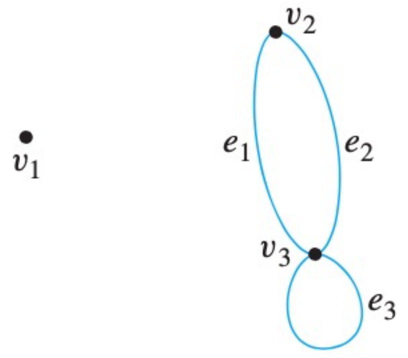
Find the degree of each vertex of the graph G shown below. Then find the total degree of G.



$\deg(v_1) = 0$ since no edge is incident on v_1 (v_1 is isolated).

Example:

Find the degree of each vertex of the graph G shown below. Then find the total degree of G.

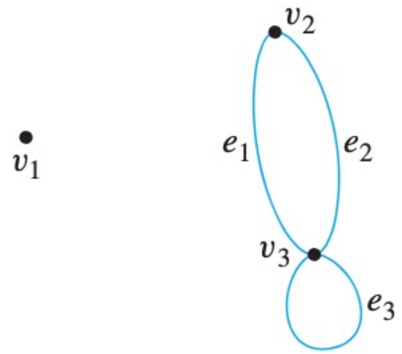


$\deg(v_1)$ = 0 since no edge is incident on v_1 (v_1 is isolated).

$\deg(v_2)$ = 2 since both e_1 and e_2 are incident on v_2.

Example:

Find the degree of each vertex of the graph G shown below. Then find the total degree of G.



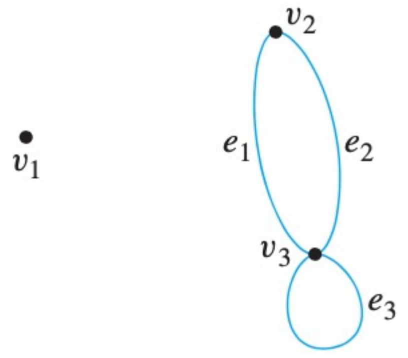
$\deg(v_1) = 0$ since no edge is incident on v_1 (v_1 is isolated).

$\deg(v_2) = 2$ since both e_1 and e_2 are incident on v_2.

$\deg(v_3) = 4$ since e_1 and e_2 are incident on v_3 and the loop e_3 is also incident on v_3 (and contributes 2 to the degree of v_3).

Example:

Find the degree of each vertex of the graph G shown below. Then find the total degree of G.



$\deg(v_1) = 0$ since no edge is incident on v_1 (v_1 is isolated).

$\deg(v_2) = 2$ since both e_1 and e_2 are incident on v_2.

$\deg(v_3) = 4$ since e_1 and e_2 are incident on v_3 and the loop e_3 is also incident on v_3 (and contributes 2 to the degree of v_3).

Total degree of G $=\deg(v_1)+\deg(v_2)+\deg(v_3)=0+2+4=6$.

Theorem: The Handshake Theorem

If *G* is any graph, then the sum of the degrees of all the vertices of *G* equals twice the number of edges of *G*. Specifically, if the vertices of *G* are $v_1, v_2, \ldots, v_n$, where n is a nonnegative integer, then

$$\begin{aligned} \text{the total degree of } G &= \deg(v_1) + \deg(v_2) + \cdots + \deg(v_n) \\ &= 2\cdot(\text{the number of edges of } G). \end{aligned}$$

Corollary:

The total degree of a graph is even.

Example:

Draw a graph with the specified properties or show that no such graph exists.

1. A graph with four vertices of degrees 1, 1, 2, and 3.
2. A graph with four vertices of degrees 1, 1, 3, and 3.

Example:

Is it possible in a group of nine people for each to be friends with exactly five others?

Proposition:

In any graph there are an even number of vertices of odd degree

Definition:

Let G be a graph, and let v and w be vertices in G.

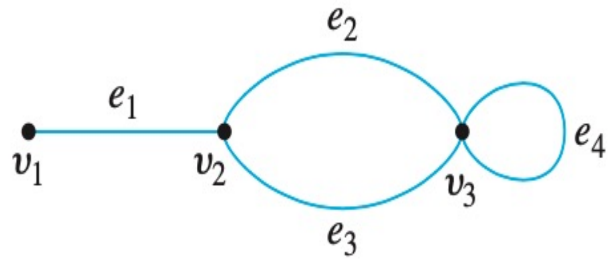
Definition:

Let *G* be a graph, and let v and w be vertices in *G*.

1. A **walk from $\boldsymbol{v}$ to $\boldsymbol{w}$** is a finite alternating sequence of adjacent vertices and edges of *G*. Thus, a walk has the form $v_0\ e_1\ v_1\ e_2 \ldots v_{n-1} e_n v_n$, where the v's represent vertices, the e's represent edges, $v_0 = v, v_n = w$, and for each $i = 1, 2, \ldots, n, v_{i-1}\ and\ v_i$ are the endpoints of e_i. The **trivial walk from $\boldsymbol{v}$ to $\boldsymbol{v}$** consists of the single vertex v.
2. A **trail from $\boldsymbol{v}$ to $\boldsymbol{w}$** is a walk from v to w that does not contain a repeated edge.
3. A **path from $\boldsymbol{v}$ to $\boldsymbol{w}$** is a trail that does not contain a repeated vertex.
4. A **closed walk** is a walk that starts and ends at the same vertex.
5. A **circuit** is a closed walk that contains at least one edge and does not contain a repeated edge.
6. A **simple circuit** is a circuit that does not have any other repeated vertex except the first and last.

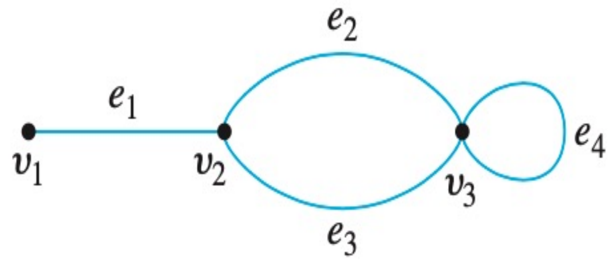
Example:

In the graph below, the notation $e_1e_2e_4e_3$ refers unambiguously to the following walk: $v_1e_1v_2e_2v_3e_4v_3e_3v_2$.



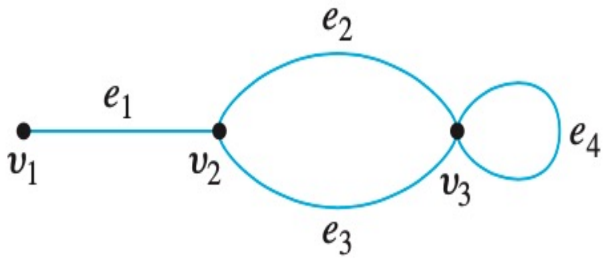
Example:

In the graph below, the notation $e_1 e_2 e_4 e_3$ refers unambiguously to the following walk: $v_1 e_1 v_2 e_2 v_3 e_4 v_3 e_3 v_2$. On the other hand, the notation e_1 is ambiguous if used by itself to refer to a walk. It could mean either $v_1 e_1 v_2$ or $v_2 e_1 v_1$.



Example:

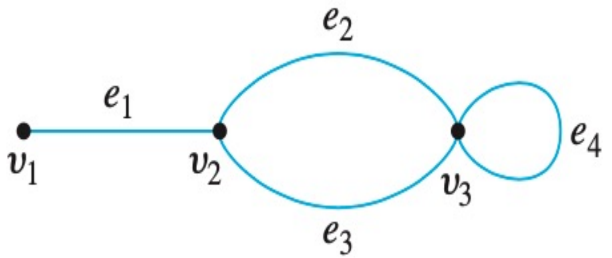
In the graph below, the notation $e_1e_2e_4e_3$ refers unambiguously to the following walk: $v_1e_1v_2e_2v_3e_4v_3e_3v_2$. On the other hand, the notation e_1 is ambiguous if used by itself to refer to a walk. It could mean either $v_1e_1v_2$ or $v_2e_1v_1$.



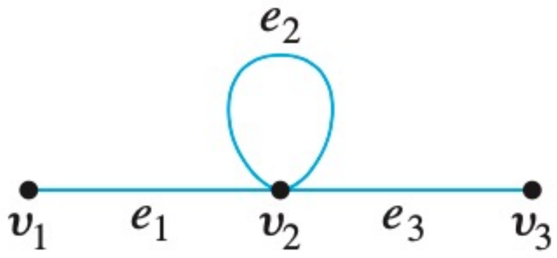
In the above graph, the notation v_2v_3 is ambiguous if used to refer to a walk. It could mean $v_2e_2v_3$ or $v_2e_3v_3$.

Example:

In the graph below, the notation $e_1e_2e_4e_3$ refers unambiguously to the following walk: $v_1e_1v_2e_2v_3e_4v_3e_3v_2$. On the other hand, the notation e_1 is ambiguous if used by itself to refer to a walk. It could mean either $v_1e_1v_2$ or $v_2e_1v_1$.



In the above graph, the notation v_2v_3 is ambiguous if used to refer to a walk. It could mean $v_2e_2v_3$ or $v_2e_3v_3$. On the other hand, in the graph below, the notation $v_1v_2v_2v_3$ refers unambiguously to the walk $v_1e_1v_2e_2v_2e_3v_3$.



Example:

In the graph below, determine which of the following walks are trails, paths, circuits, or simple circuits.

a. $v_1 e_1 v_2 e_3 v_3 e_4 v_3 e_5 v_4$

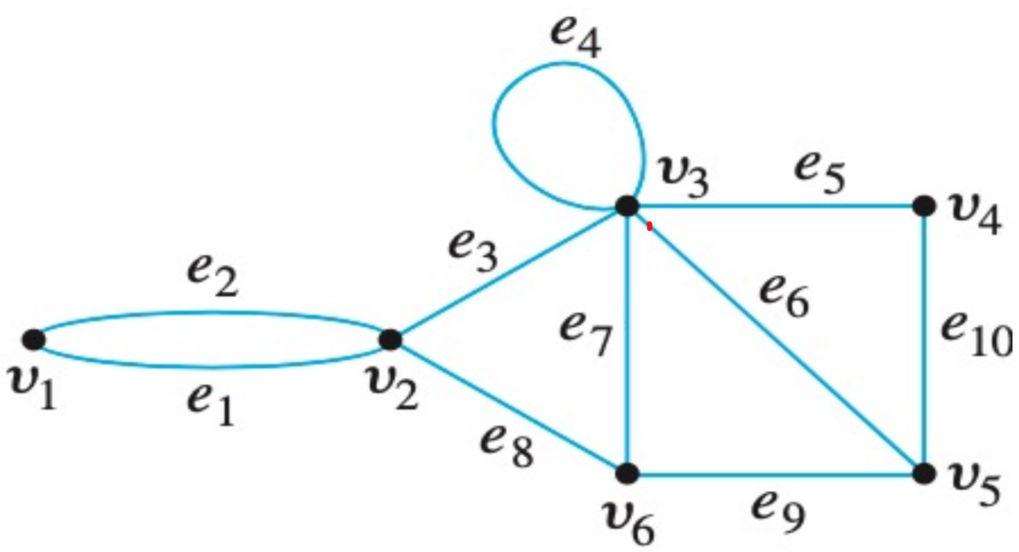
b. $e_1 e_3 e_5 e_5 e_6$.

c. $v_2 v_3 v_4 v_5 v_3 v_6 v_2$

d. $v_2 v_3 v_4 v_5 v_6 v_2$

e. $v_1 e_1 v_2 e_1 v_1$

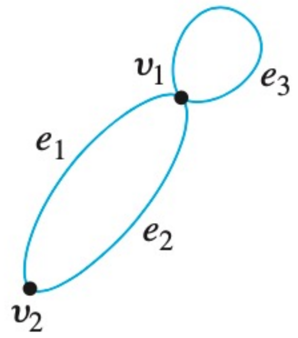
f. v_1



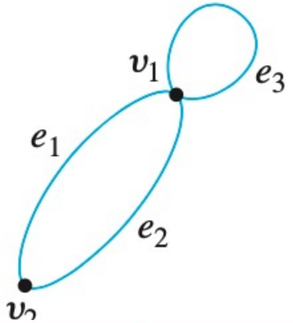
Definition:

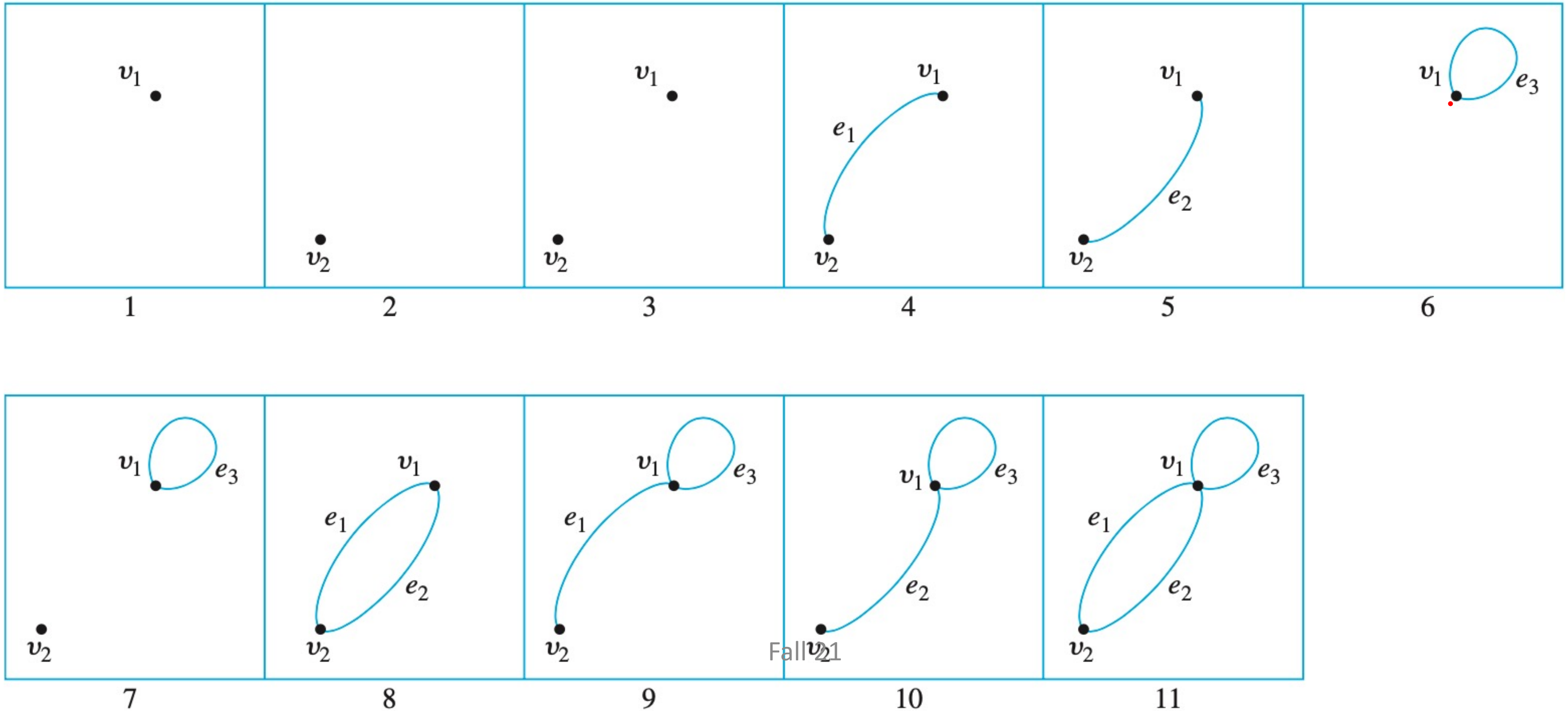
A graph *H* is said to be a **subgraph** of a graph *G* if, and only if, every vertex in *H* is also a vertex in *G*, every edge in *H* is also an edge in *G*, and every edge in *H* has the same endpoints as it has in *G*.

Example: List all subgraphs of the graph *G* with vertex set $\{v_1, v_2\}$ and edge set $\{e_1, e_2, e_3\}$, where the endpoints of e_1 are v_1 and v_2, the endpoints of e_2 are v_1 and v_2, and e_3 is a loop at v_1.



Example: List all subgraphs of the graph G with vertex set $\{v_1, v_2\}$ and edge set $\{e_1, e_2, e_3\}$, where the endpoints of e_1 are v_1 and v_2, the endpoints of e_2 are v_1 and v_2, and e_3 is a loop at v_1.





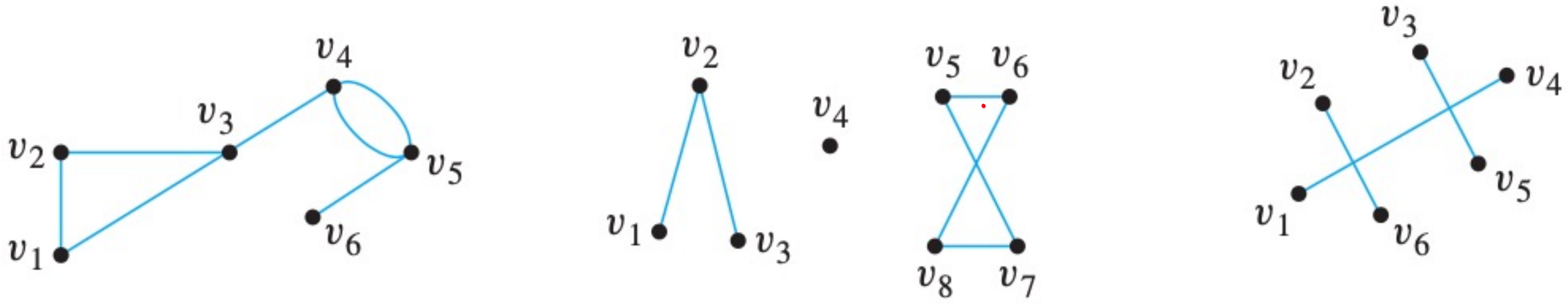
Definition:

Let G be a graph. Two **vertices $\boldsymbol{v}$ and $\boldsymbol{w}$ of *G* are connected** if, and only if, there is a walk from v to w. The **graph *G* is connected** if, and only if, given *any* two vertices v and w in G, there is a walk from v to w. Symbolically,

$$G \text{ is connected} \Leftrightarrow \forall \text{ vertices } v, w \in V(G), \exists \text{ a walk from } v \text{ to } w.$$

Example:

Which of the following graphs are connected?



(a) (b) (c)

Lemma:

Let *G* be a graph.

1. If *G* is connected, then any two distinct vertices of *G* can be connected by a path.
2. If vertices v and w are part of a circuit in *G* and one edge is removed from the circuit, then there still exists a trail from v to w in *G*.
3. If *G* is connected and *G* contains a circuit, then an edge of the circuit can be removed without disconnecting *G*.

Definition:

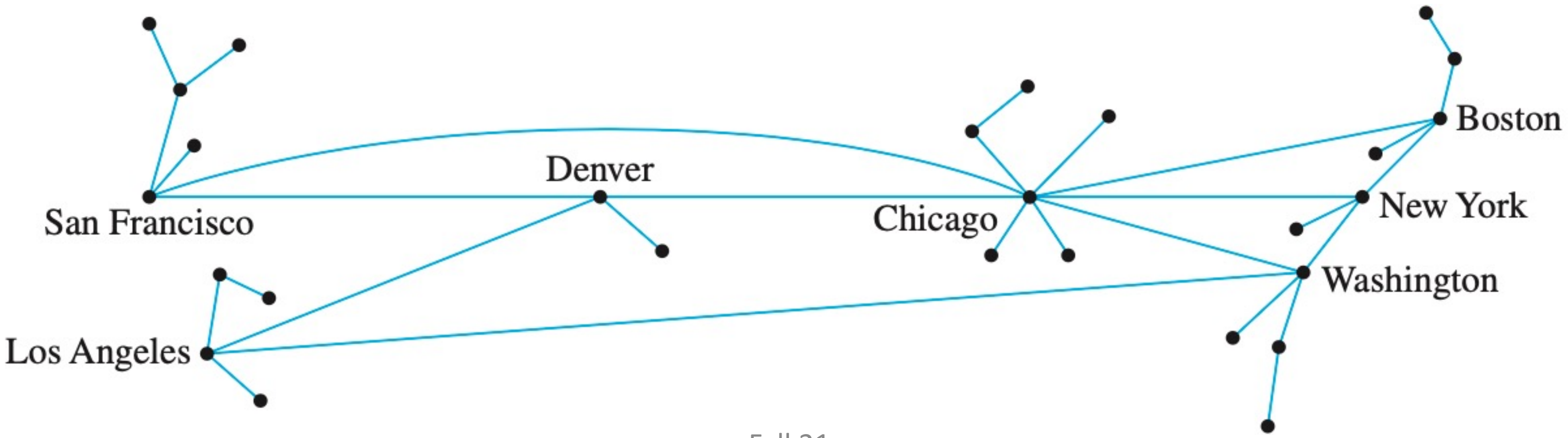
A graph *H* is a **connected component** of a graph *G* if, and only if,

1. *H* is subgraph of *G*;
2. *H* is connected; and
3. no connected subgraph of *G* has *H* as a subgraph and contains vertices or edges that are not in *H*. [no bigger connected subgraph of *G* properly contains *H* as a subgraph]

{a,b,c} properly contained by {a,b,c,d}

Using a Graph to represent a Network

Telephone, electric power, gas pipeline, and air transport systems can all be represented by graphs, as can computer networks—from small local area networks to the global Internet system that connects millions of computers worldwide. Questions that arise in the design of such systems involve choosing connecting edges to minimize cost, optimize a certain type of service, and so forth. A typical network, called a *hub-and-spoke model*, is shown below.



Using a Graph to represent Knowledge

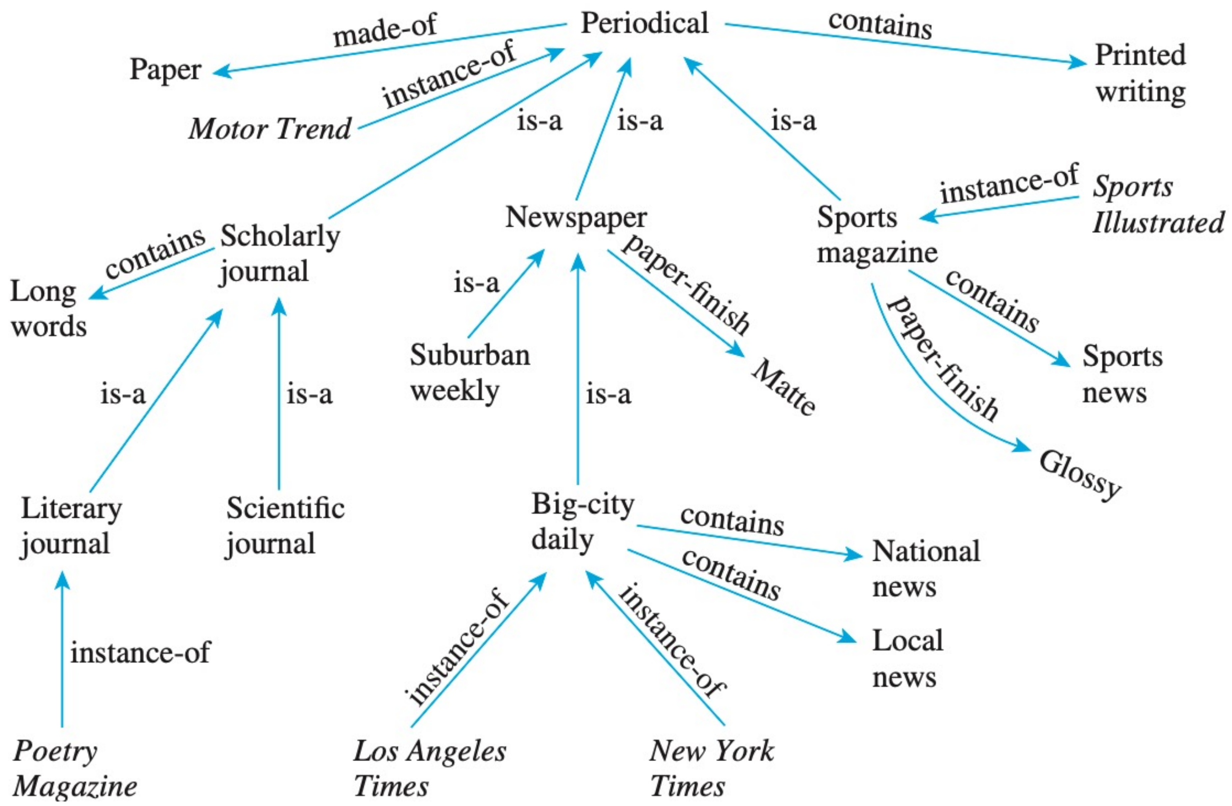
In many applications of artificial intelligence, a knowledge base of information is collected and represented inside a computer. Because of the way the knowledge is represented and because of the properties that govern the artificial intelligence program, the computer is not limited to retrieving data in the same form as it was entered; it can also derive new facts from the knowledge base by using certain built-in rules of inference.

For example, from the knowledge that the *Los Angeles Times* is a big-city daily and that a big-city daily contains national news, an artificial intelligence program could infer that the *Los Angeles Times* contains national news.

Using a Graph to represent Knowledge

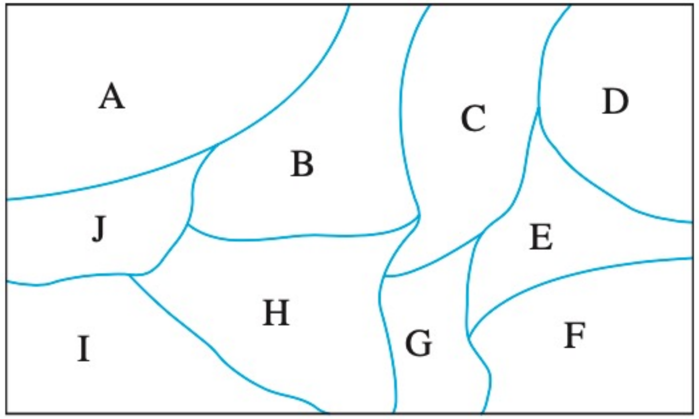
The directed graph shown in the following figure is a pictorial representation for a simplified knowledge base about periodical publications.

According to this knowledge base, what paper finish does the *New York Times* use?

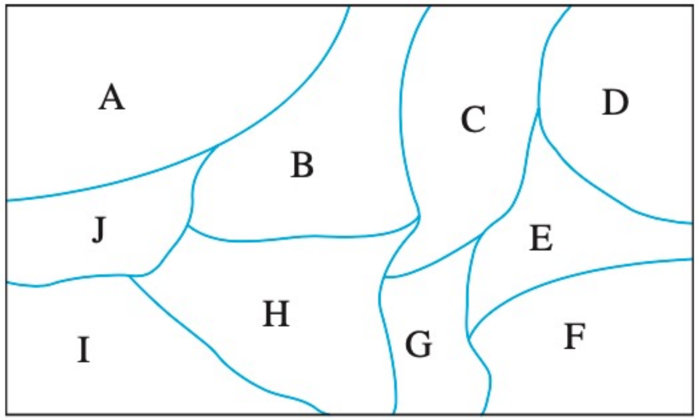


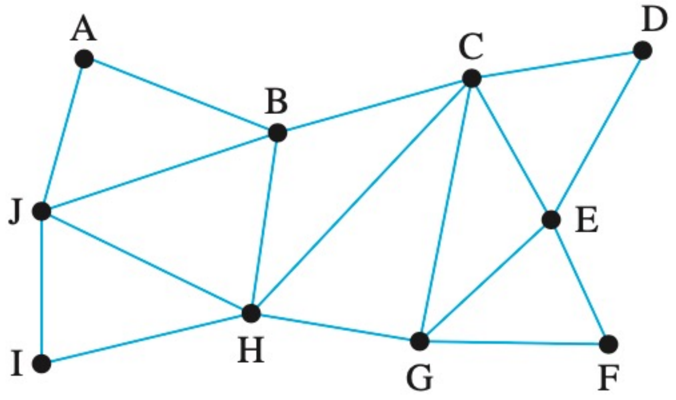
Example: **Using a Graph to Color a Map**

Imagine that the diagram shown below is a map with countries labeled *A–J*. Show that you can color the map so that no two adjacent countries have the same color.

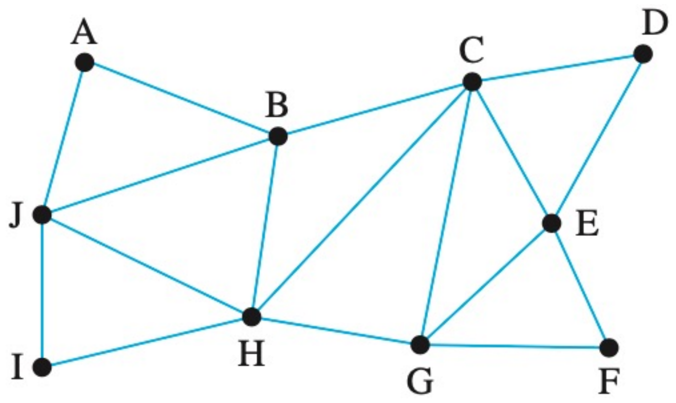


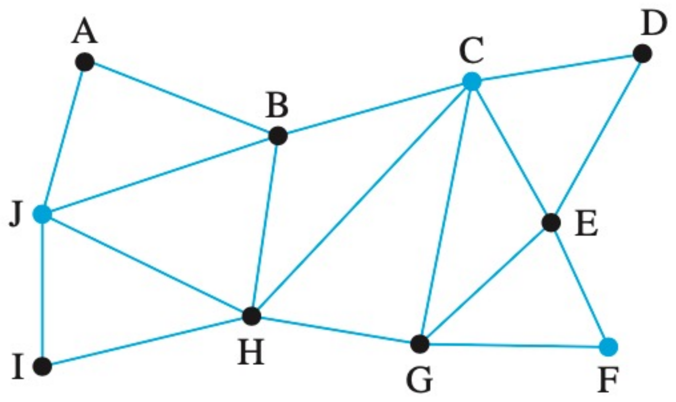
Example: Using a Graph to Color a Map



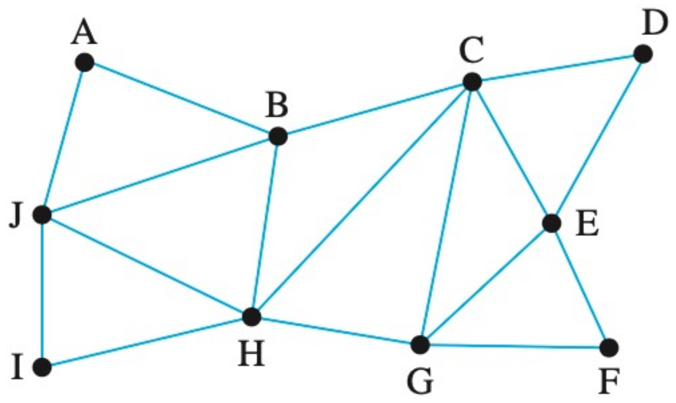


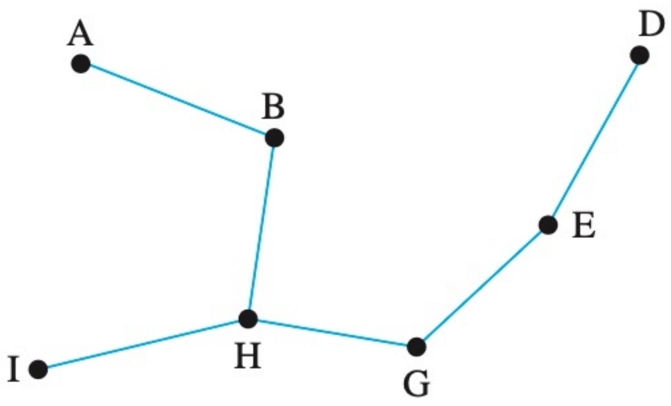
Example: Using a Graph to Color a Map

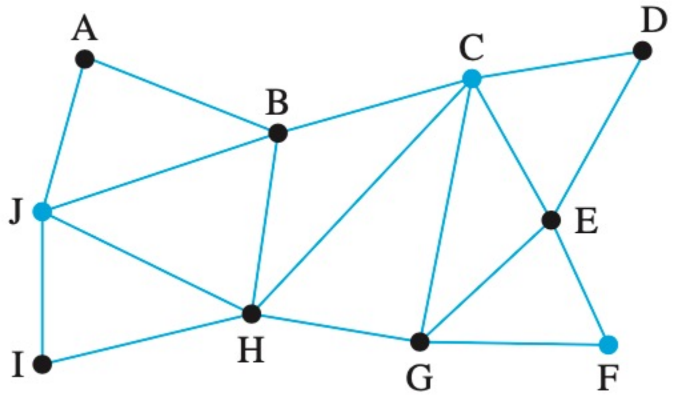




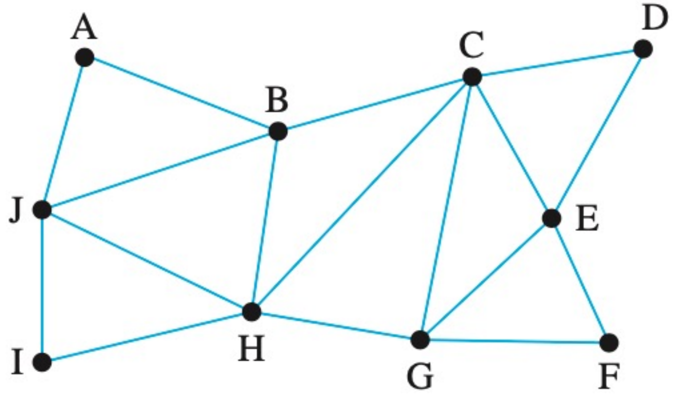
Example: **Using a Graph to Color a Map**

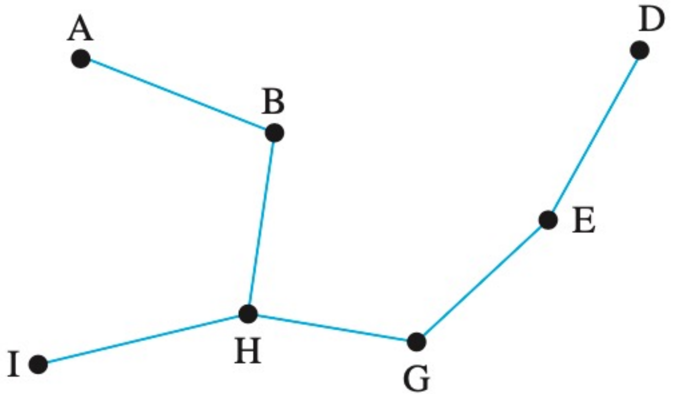


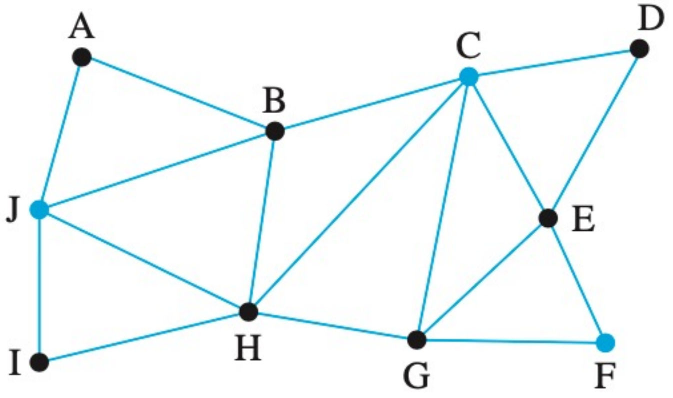


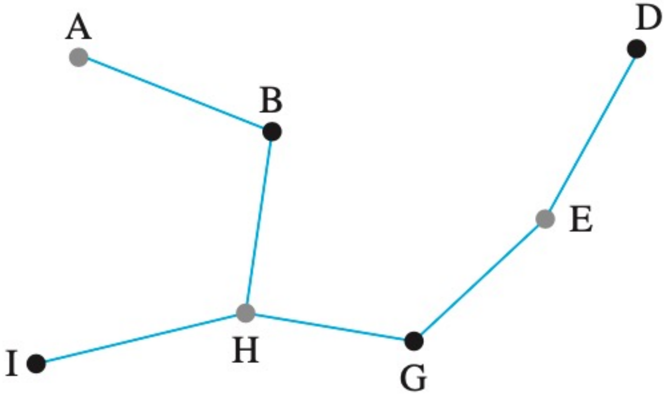


Example: Using a Graph to Color a Map

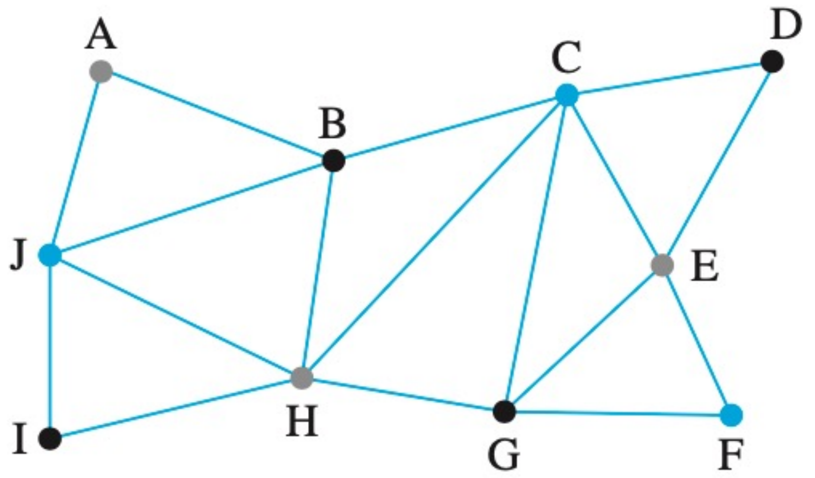




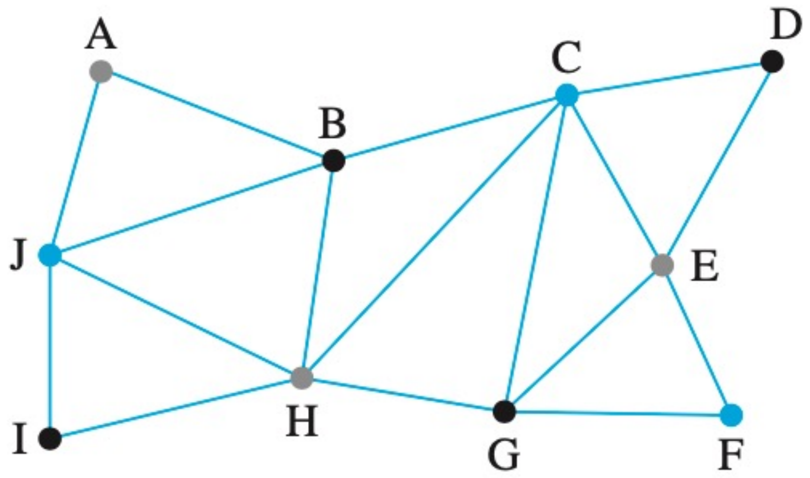


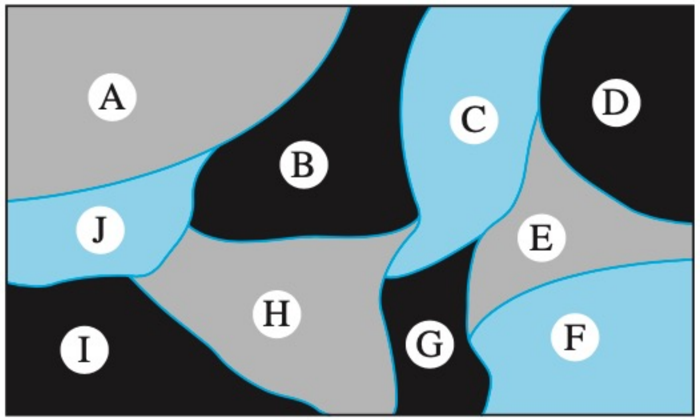


Example: Using a Graph to Color a Map



Example: Using a Graph to Color a Map

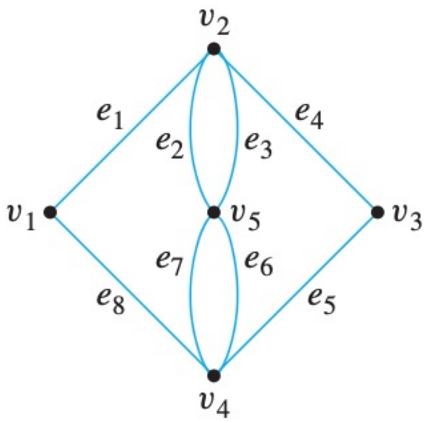




Definition:

Let *G* be a graph. An **Euler circuit** for *G* is a circuit that contains every vertex and every edge of *G*. That is, an Euler circuit for *G* is a sequence of adjacent vertices and edges in *G* that has at least one edge, starts and ends at the same vertex, uses every vertex of *G* at least once, and uses every edge of *G* exactly once.

12.



Theorem:

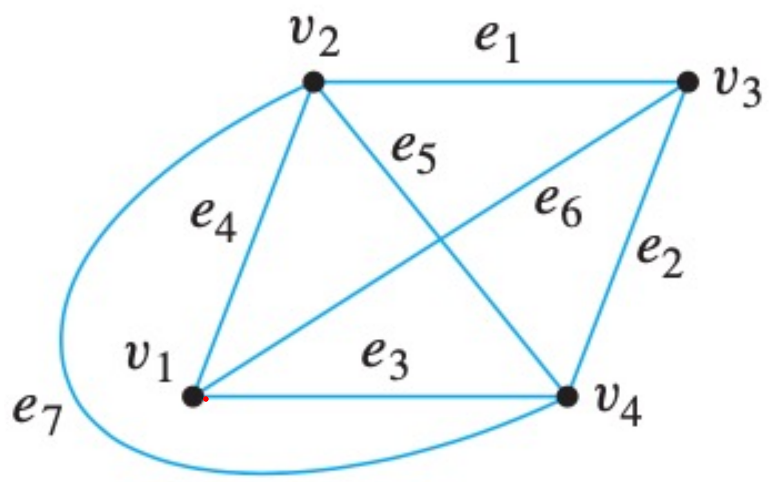
If a graph has an Euler circuit, then every vertex of the graph has positive even degree.

Or

If some vertex of a graph has odd degree, then the graph does not have an Euler circuit.

Example:

Show that the graph below does not have an Euler circuit.



Theorem:

If a graph G is connected and the degree of every vertex of G is a positive even integer, then G has an Euler circuit.

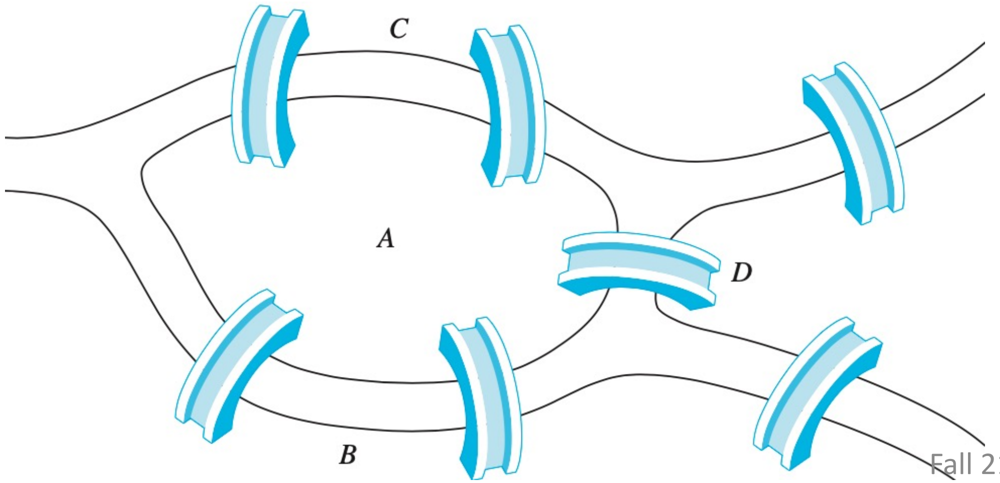
Theorem:

A graph G has an Euler circuit if, and only if, G is connected and every vertex of G has positive even degree.

Example:

The town of Königsberg in Prussia (now Kaliningrad in Russia) was built at a point where two branches of the Pregel River came together. It consisted of an island and some land along the riverbanks. These were connected by seven bridges as shown in the figure.

The question is this: Is it possible for a person to take a walk around town, starting and ending at the same location and crossing each of the seven bridges exactly once?



Definition:

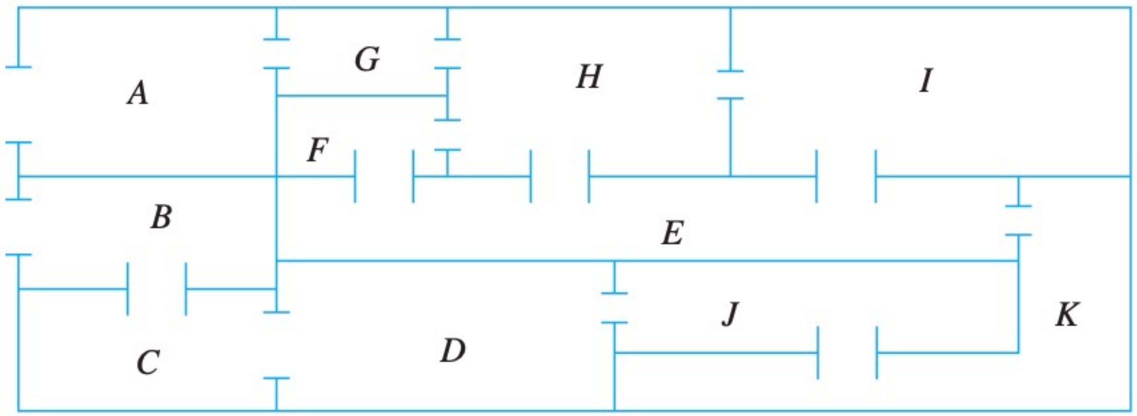
Let *G* be a graph, and let v and w be two distinct vertices of *G*. An **Euler trail from** v **to** w is a sequence of adjacent edges and vertices that starts at v, ends at w, passes through every vertex of *G* at least once, and traverses every edge of *G* exactly once.

Corollary:

Let *G* be a graph, and let v and w be two distinct vertices of *G*. There is an Euler path from v to w if, and only if, *G* is connected, v and w have odd degree, and all other vertices of *G* have positive even degree.

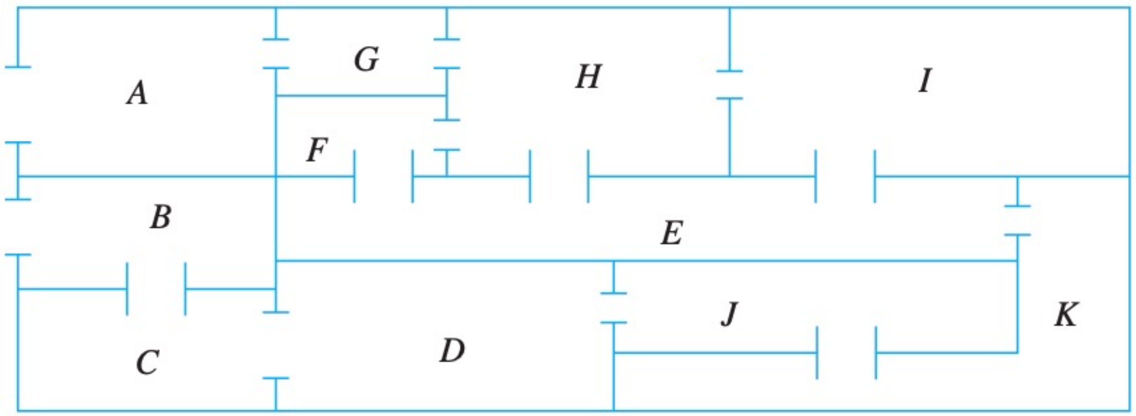
Example:

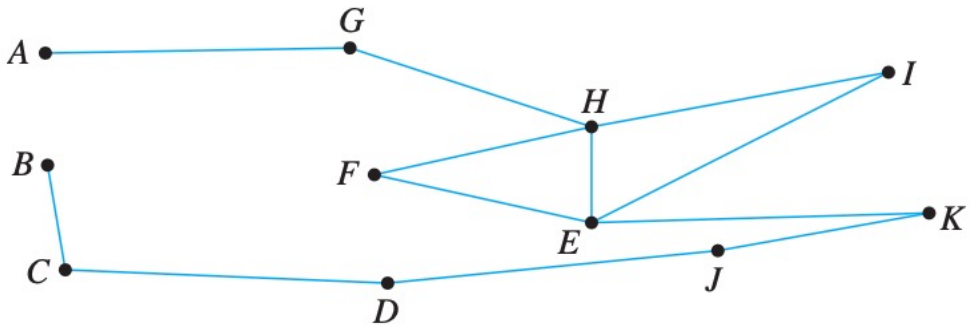
The floor plan shown below is for a house that is open for public viewing. Is it possible to find a trail that starts in room *A*, ends in room *B*, and passes through every interior doorway of the house exactly once? If so, find such a trail.



Example:

The floor plan shown below is for a house that is open for public viewing. Is it possible to find a trail that starts in room *A*, ends in room *B*, and passes through every interior doorway of the house exactly once? If so, find such a trail.

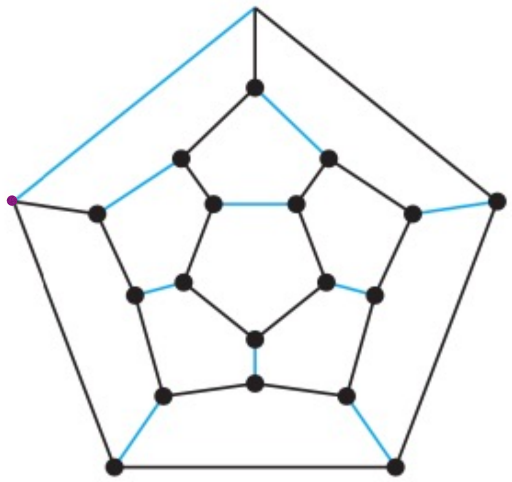




AGHIEHFEKJDCB

Definition:

Given a graph *G*, a **Hamiltonian circuit** for *G* is a simple circuit that includes every vertex of *G*. That is, a Hamiltonian circuit for *G* is a sequence of adjacent vertices and distinct edges in which every vertex of *G* appears exactly once, except for the first and the last, which are the same.



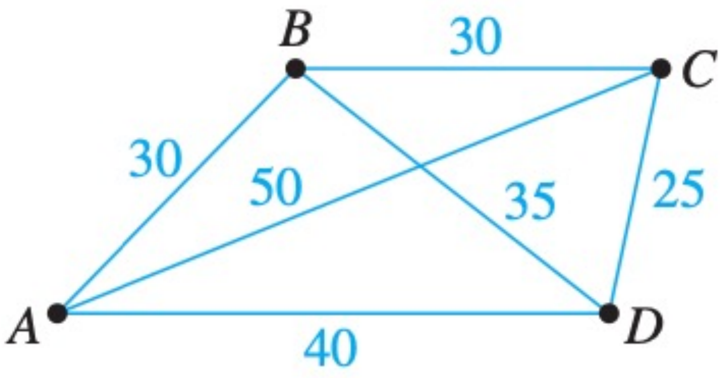
Proposition

If a graph G has a Hamiltonian circuit, then G has a subgraph H with the following properties:

1. H contains every vertex of G.
2. H is connected.
3. H has the same number of edges as vertices.
4. Every vertex of H has degree 2.

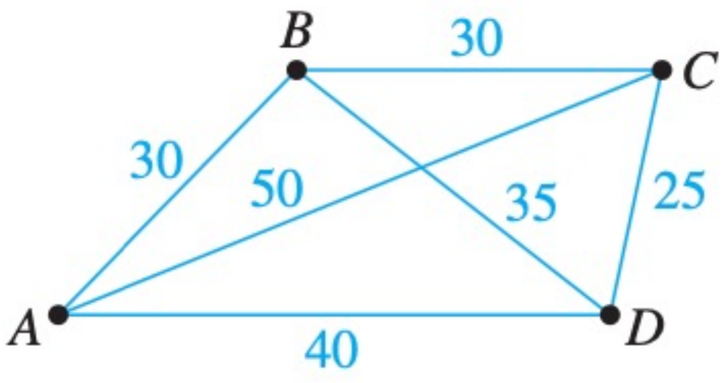
Example:

Imagine that the drawing below is a map showing four cities and the distances in kilometers between them. Suppose that a salesman must travel to each city exactly once, starting and ending in city A. Which route from city to city will minimize the total distance that must be traveled?



Example:

Imagine that the drawing below is a map showing four cities and the distances in kilometers between them. Suppose that a salesman must travel to each city exactly once, starting and ending in city *A*. Which route from city to city will minimize the total distance that must be traveled?



Route	Total Distance (In Kilometers)
$ABCDA$	$30 + 30 + 25 + 40 = 125$
$ABDCA$	$30 + 35 + 25 + 50 = 140$
$ACBDA$	$50 + 30 + 35 + 40 = 155$
$ACDBA$	140 [$ABDCA$ backwards]
$ADBCA$	155 [$ACBDA$ backwards]
$ADCBA$	125 [$ABCDA$ backwards]