# Discrete Structures

# Graphs

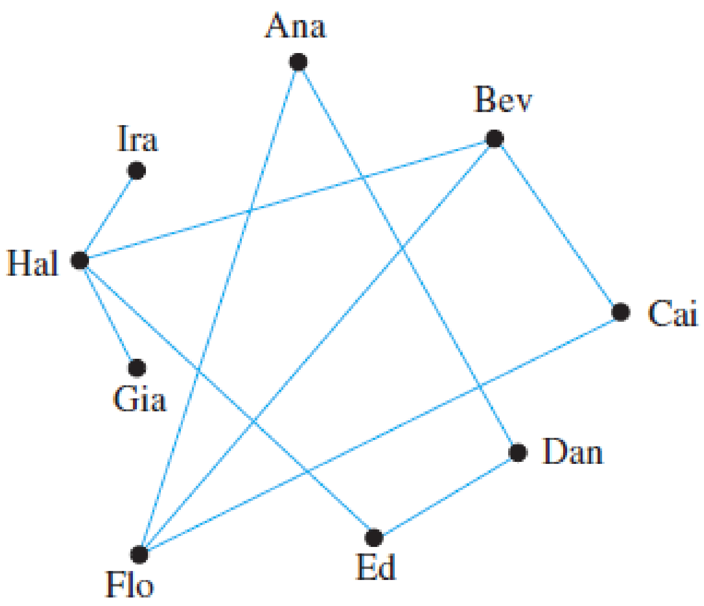
# Today’s Lecture

- Graphs
- Directed Graphs
- Simple Graphs
- Complete Graphs
- Complete Bipartite Graphs
- Subgraphs
- The Concept of Degree

# Graphs

Imagine an organization that wants to set up teams of three to work on some projects. In order to maximize the number of people on each team who had previous experience working together successfully, the director asked the members to provide names of their past partners. This information is displayed below both in a table and in a diagram.

| Name | Past Partners |
|---|---|
| Ana | Dan, Flo |
| Bev | Cai, Flo, Hal |
| Cai | Bev, Flo |
| Dan | Ana, Ed |
| Ed | Dan, Hal |
| Flo | Cai, Bev, Ana |
| Gia | Hal |
| Hal | Gia, Ed, Bev, Ira |
| Ira | Hal |



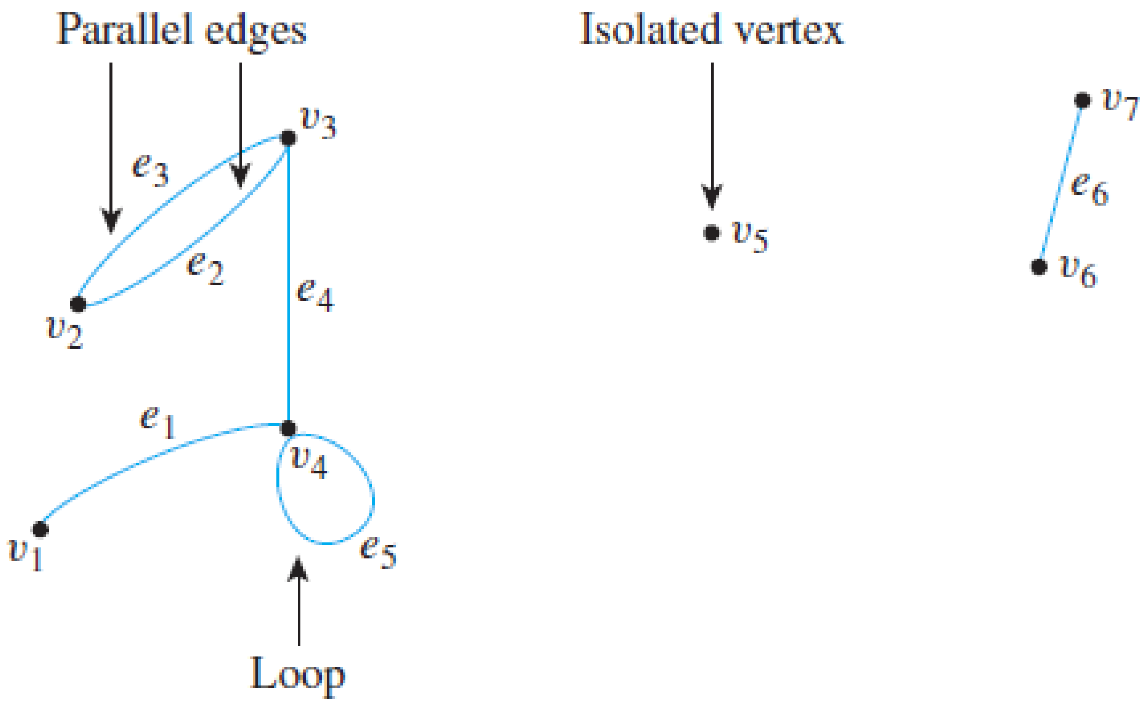
# Cont…

## Cont…

Drawings such as those shown previously are illustrations of a structure known as a *graph*. The dots are called *vertices* (plural of *vertex*) and the line segments joining vertices are called *edges*. As you can see from the drawing, it is possible for two edges to cross at a point that is not a vertex.

# Cont…

In general, a graph consists of a set of vertices and a set of edges connecting various pairs of vertices. The edges may be straight or curved and should either connect one vertex to another or a vertex to itself, as shown below.



# Cont…

In this drawing, the vertices have been labeled with *v*'s and the edges with *e*'s. When an edge connects a vertex to itself (as $e_5$ does), it is called a *loop*. When two edges connect the same pair of vertices (as $e_2$ and $e_3$ do), they are said to be *parallel*. It is quite possible for a vertex to be unconnected by an edge to any other vertex in the graph (as $v_5$ is), and in that case the vertex is said to be *isolated*.

# Definition: Graphs

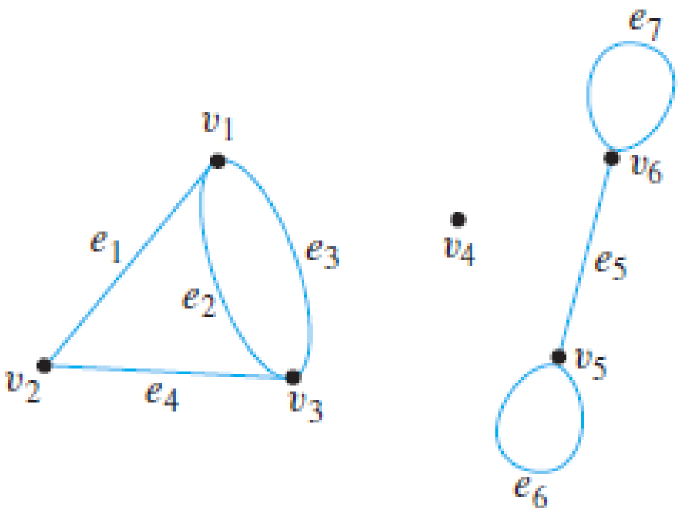
A **graph** *G* consists of two finite sets: a nonempty set *V(G)* of **vertices** and a set *E(G)* of **edges,** where each edge is associated with a set consisting of either one or two vertices called its **endpoints.** The correspondence from edges to endpoints is called the **edge-endpoint function.**

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.** 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.**

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.**

## Graphs: Examples

Example: Consider the following graph:




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

b. 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.

# Graphs: Examples (Contd….)

a. 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 |
|---|---|
| $e_1$ | $\{v_1, v_2\}$ |
| $e_2$ | $\{v_1, v_3\}$ |
| $e_3$ | $\{v_1, v_3\}$ |
| $e_4$ | $\{v_2, v_3\}$ |
| $e_5$ | $\{v_5, v_6\}$ |
| $e_6$ | $\{v_5\}$ |
| $e_7$ | $\{v_6\}$ |

b. $e_1$, $e_2$, 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.

# Drawing More Than One Picture for a Graph

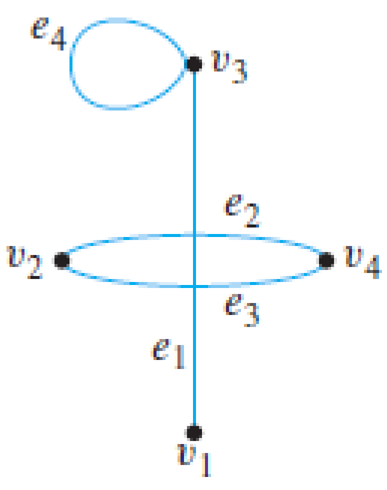
Consider the graph specified as follows:

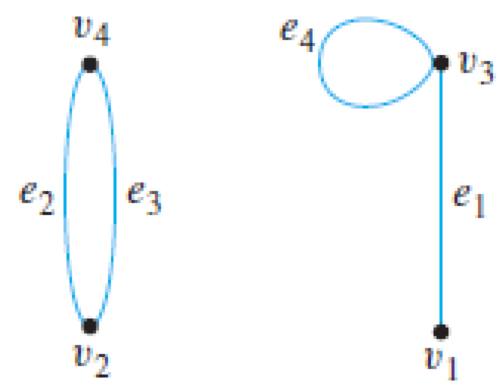
vertex set = {$v_1$, $v_2$, $v_3$, $v_4$}

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

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

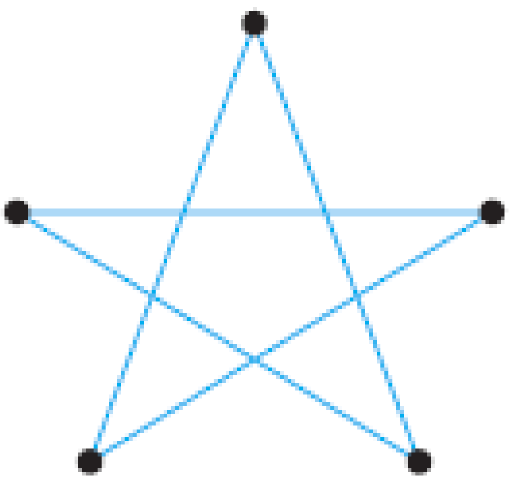
Edge-endpoint function



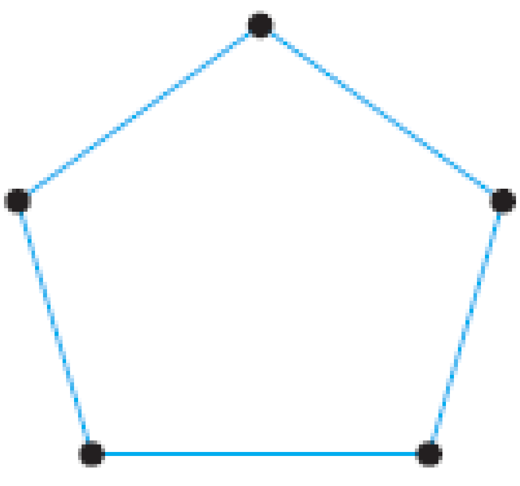



## Labeling Drawings to Show They Represent the Same Graph

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



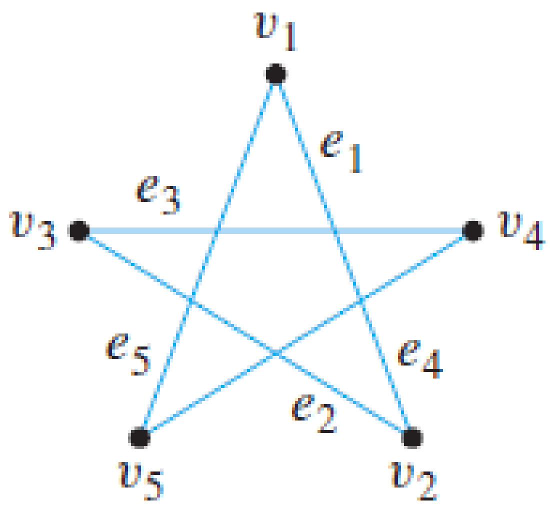
(a)



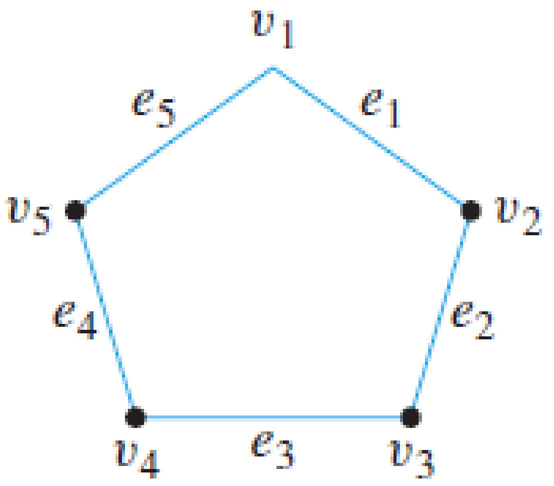
(b)

## Labeling Drawings to Show They Represent the Same Graph

Both drawings are representations of the graph with vertex set $\{v_1, v_2, v_3, v_4, v_5\}$, edge set $\{e_1, e_2, e_3, e_4, e_5\}$, and edge-endpoint function as follows:




| Edge | Endpoints |
|---|---|
| $e_1$ | $\{v_1, v_2\}$ |
| $e_2$ | $\{v_2, v_3\}$ |
| $e_3$ | $\{v_3, v_4\}$ |
| $e_4$ | $\{v_4, v_5\}$ |
| $e_5$ | $\{v_5, v_1\}$ |



# Directed Graphs

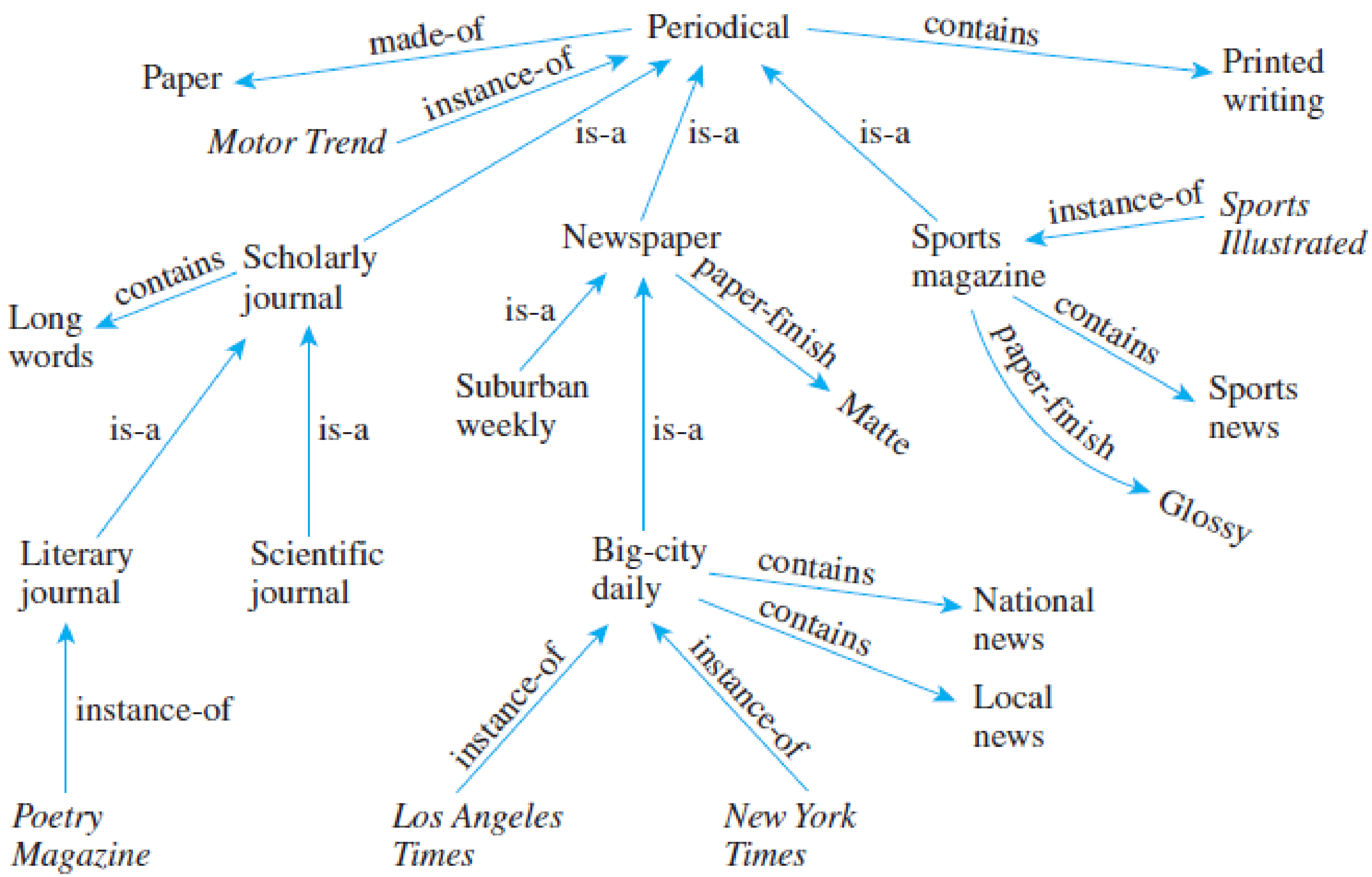
- **Definition**

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 $w$.

# Directed Graphs: *Examples of Graphs*

## Using a Graph to Represent Knowledge

What paper finish does the *New York Times* use?



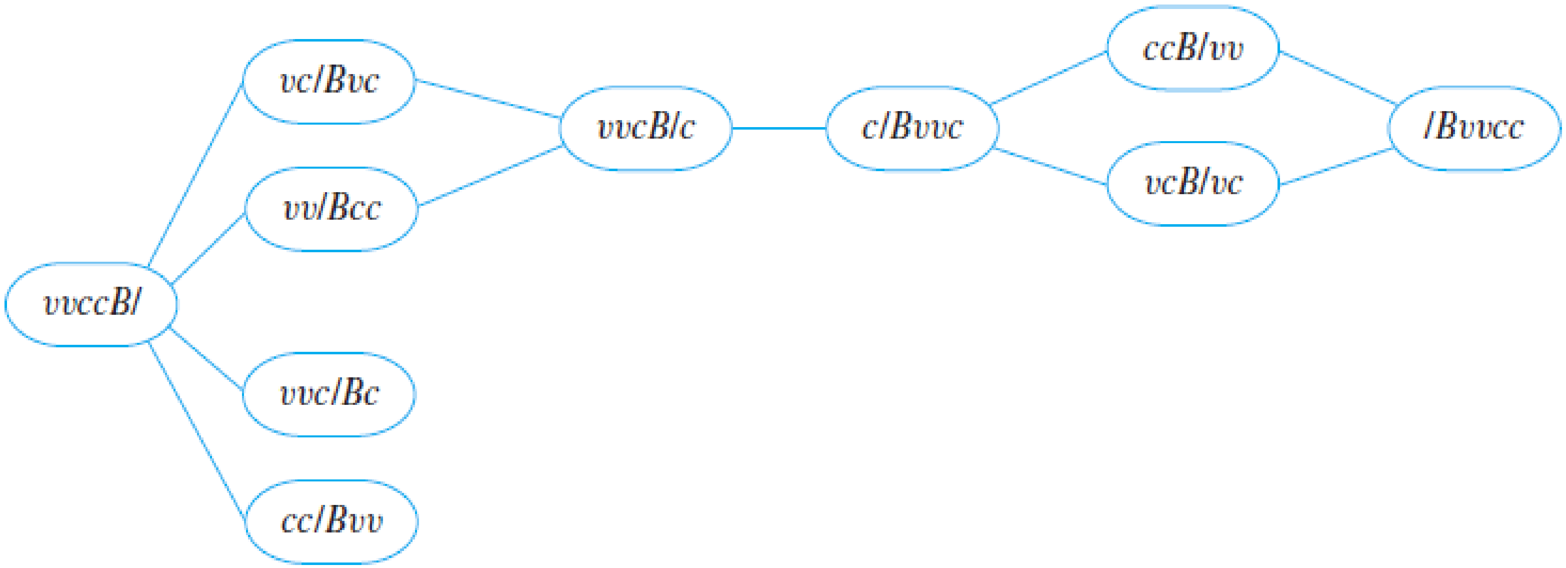
# Using a Graph to Solve a Problem: Vegetarians and Cannibals

## Using a Graph to Represent Knowledge

The following is a variation of a famous puzzle often used as an example in the study of artificial intelligence. It concerns an island on which all the people are of one of two types, either vegetarians or cannibals. Initially, two vegetarians and two cannibals are on the left bank of a river. With them is a boat that can hold a maximum of two people.

The aim of the puzzle is to find a way to transport all the vegetarians and cannibals to the right bank of the river. What makes this difficult is that at no time can the number of cannibals on either bank outnumber the number of vegetarians. Otherwise, disaster befalls the vegetarians!

# Using a Graph to Solve a Problem: Vegetarians and Cannibals



# Simple Graphs

**• Definition and Notation**

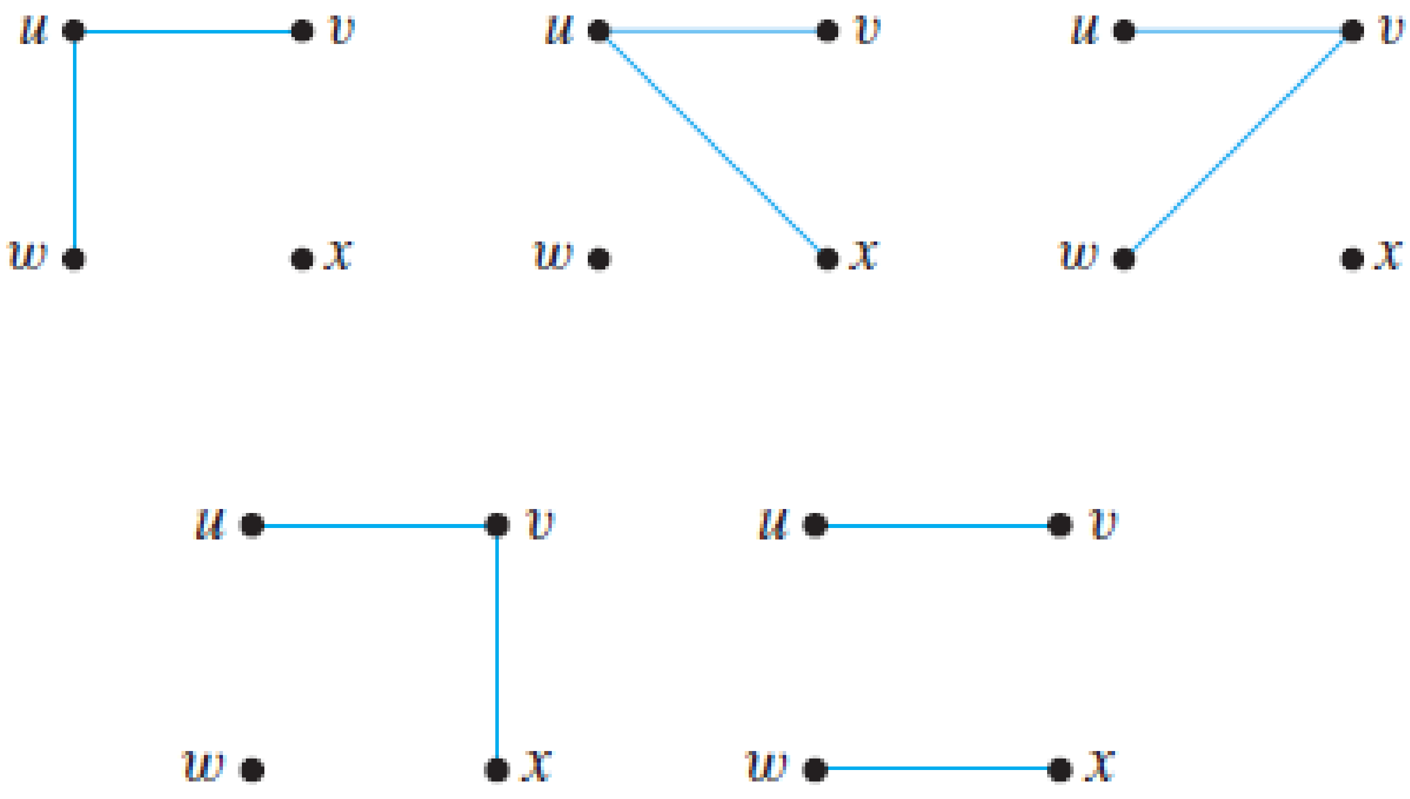
A **simple graph** is a graph that does not have any loops or parallel edges. In a simple graph, an edge with endpoints $v$ and $w$ is denoted $\{v, w\}$.

## Example

Draw all simple graphs with the four vertices $\{u, v, w, x\}$ and two edges, one of which is $\{u, v\}$.

Each possible edge of a simple graph corresponds to a subset of two vertices. Given four vertices, there are ${}^4C_2 = 6$ such subsets in all: $\{u, v\}$, $\{u, w\}$, $\{u, x\}$, $\{v, w\}$, $\{v, x\}$, and $\{w, x\}$. Now one edge of the graph is specified to be $\{u, v\}$, so any of the remaining five from this list can be chosen to be the second edge. The possibilities are shown on the next page.

# Simple Graphs



# Complete Graphs

- **Definition**

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

## Example

### Complete Graphs on $n$ Vertices: $K_1$, $K_2$, $K_3$, $K_4$, $K_5$

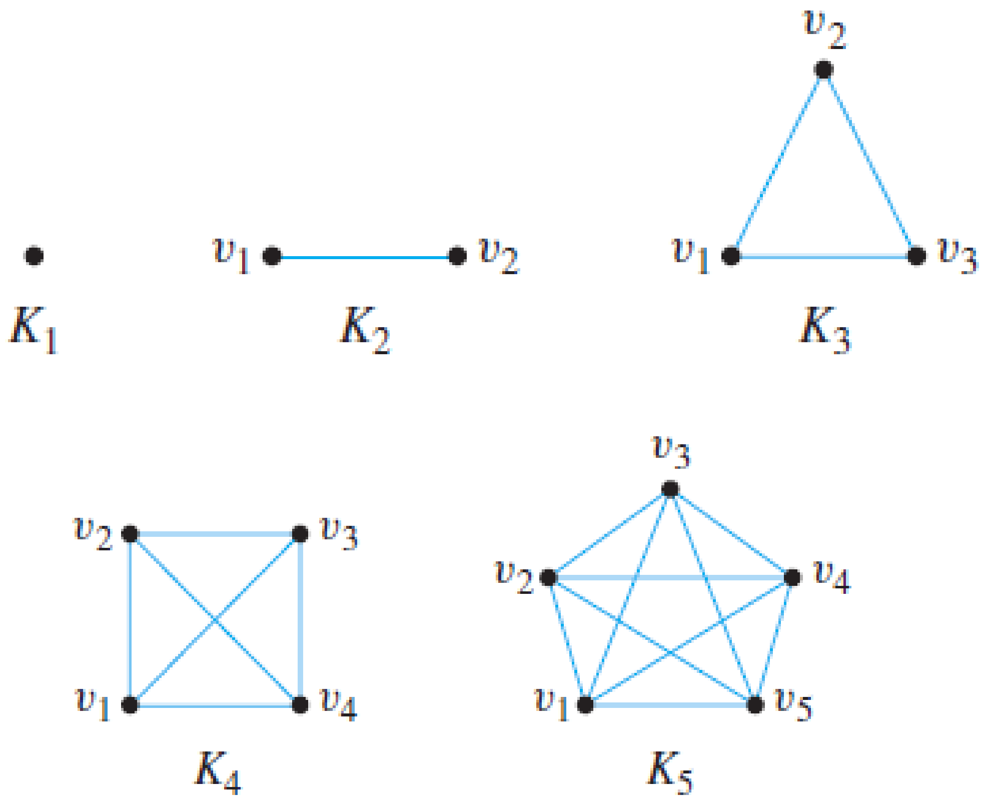
The complete graphs $K_1$, $K_2$, $K_3$, $K_4$, and $K_5$ can be drawn as follows:

# Complete Graphs

## Example

**Complete Graphs on *n* Vertices: $K_1$, $K_2$, $K_3$, $K_4$, $K_5$**

The complete graphs $K_1$, $K_2$, $K_3$, $K_4$, and $K_5$ can be drawn as follows:



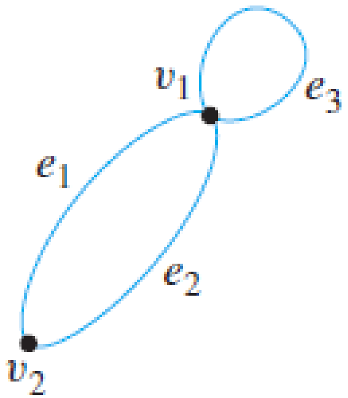
# SubGraphs

- **Definition**

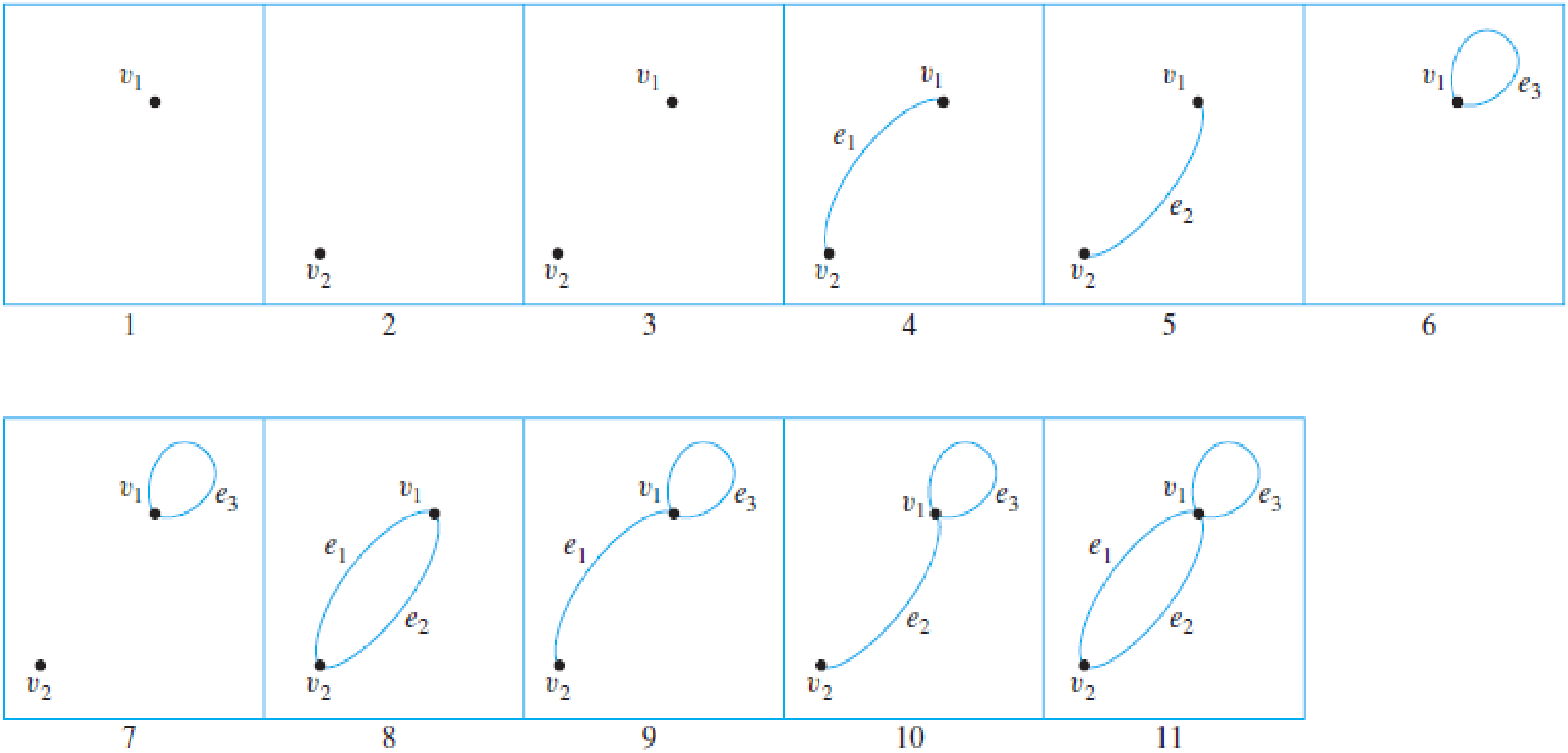
A graph $H$ is said to be a **subgraph** of a graph $G$ if, and only if, every vertex in $H$ is also a vertex in $G$, every edge in $H$ is also an edge in $G$, and every edge in $H$ has the same endpoints as it has in $G$.

## 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 $v_1$.



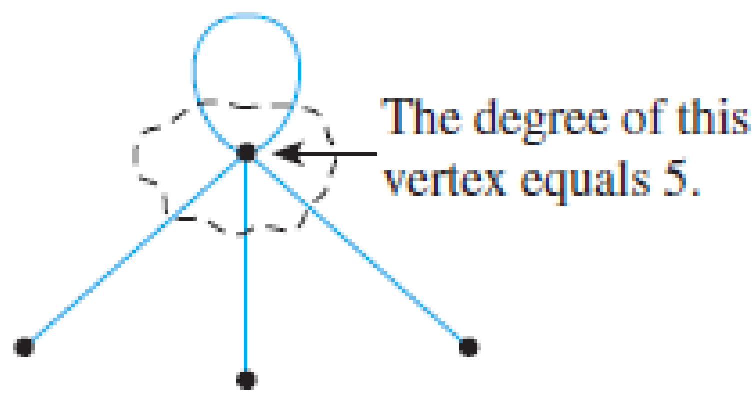
# SubGraphs



# The Concept of Degree

**• Definition**

Let $G$ be a graph and $v$ a vertex of $G$. The **degree of** $v$, denoted $\mathbf{deg}(v)$, 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 $G$.



# *The Concept of Degree*

## Degree of a Vertex and Total Degree of a Graph

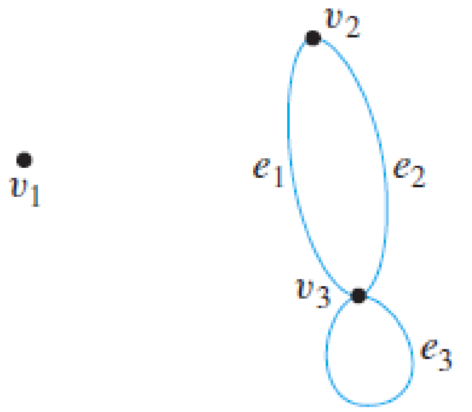
Find the degree of each vertex of the graph *G* shown below. Then find the total degree of *G*.

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



# The Concept of Degree

**Theorem 10.1.1 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

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

# The Concept of Degree

## Degree of a Vertex and Total Degree of a Graph

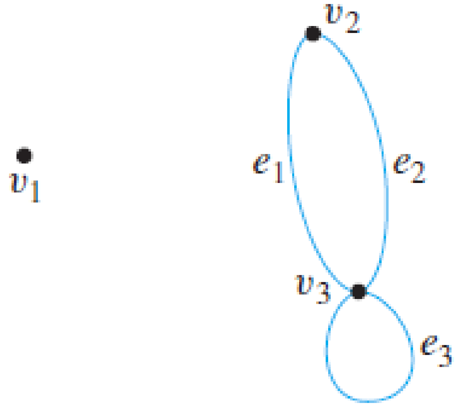
Find the degree of each vertex of the graph $G$ shown below. Then find the total degree of $G$.

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



# The Concept of Degree

**Theorem 10.1.1 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

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

## Walks

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

A **walk from *v* to *w*** is a finite alternating sequence of adjacent vertices and edges of $G$. Thus a walk has the form

$$v_0 e_1 v_1 e_2 \cdot \cdot \cdot v_{n-1} e_n v_n,$$

where the $v$'s represent vertices, the $e$'s represent edges, $v_0 = v$, $v_n = w$, and for all $i = 1, 2, . . . n$, $v_{i-1}$ and $v_i$ are the endpoints of $e_i$. The **trivial walk from *v* to *v*** consists of the single vertex $v$.

## Definitions

A **trail from *v* to *w*** is a walk from *v* to *w* that does not contain a repeated edge.

A **path from *v* to *w*** is a trail that does not contain a repeated vertex.

A **closed walk** is a walk that starts and ends at the same vertex.

A **circuit** is a closed walk that contains at least one edge and does not contain a repeated edge.

A **simple circuit** is a circuit that does not have any other repeated vertex except the first and last.

## Cont…

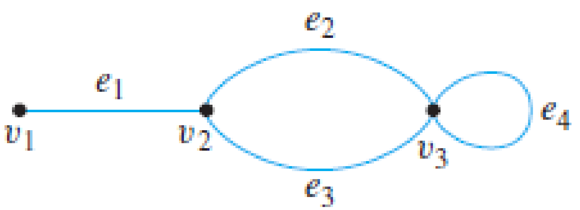
For ease of reference, these definitions are summarized in the following table:

| | Repeated Edge? | Repeated Vertex? | Starts and Ends at Same Point? | Must Contain at Least One Edge? |
|---|---|---|---|---|
| **Walk** | allowed | allowed | allowed | no |
| **Trail** | no | allowed | allowed | no |
| **Path** | no | no | no | no |
| **Closed walk** | allowed | allowed | yes | no |
| **Circuit** | no | allowed | yes | yes |
| **Simple circuit** | no | first and last only | yes | yes |

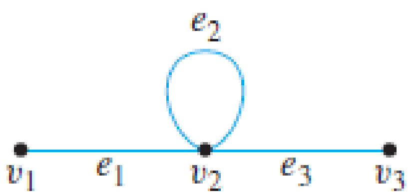
# Notations for Walk

Often a walk can be specified unambiguously by giving either a sequence of edges or a sequence of vertices.

a. 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 to refer to a walk. It could mean either $v_1e_1v_2$ or $v_2e_1v_1$.




b. In the graph of part (a), 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 $v_1e_1v_2e_2v_2e_3v_3$.



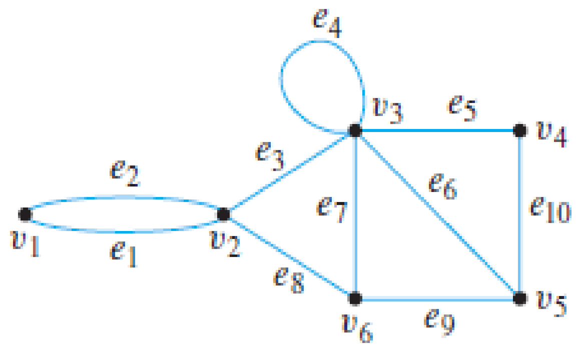
# Cont…

## Example: Trails, Paths, Circuits

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. $v_1$



# Cont…

Solution

a. This walk has a repeated vertex but does not have a repeated edge, so it is a trail from $v_1$ to $v_4$ but not a path.

b. This is just a walk from $v_1$ to $v_5$. It is not a trail because it has a repeated edge.

c. This walk starts and ends at $v_2$, contains at least one edge, and does not have a repeated edge, so it is a circuit. Since the vertex $v_3$ is repeated in the middle, it is not a simple circuit.

d. This walk starts and ends at $v_2$, contains at least one edge, does not have a repeated edge, and does not have a repeated vertex. Thus it is a simple circuit.

e. This is just a closed walk starting and ending at $v_1$. It is not a circuit because edge $e_1$ is repeated.

f. The first vertex of this walk is the same as its last vertex, but it does not contain an edge, and so it is not a circuit. It is a closed walk from $v_1$ to $v_1$. (It is also a trail from $v_1$ to $v_1$.)

# *Connectedness*

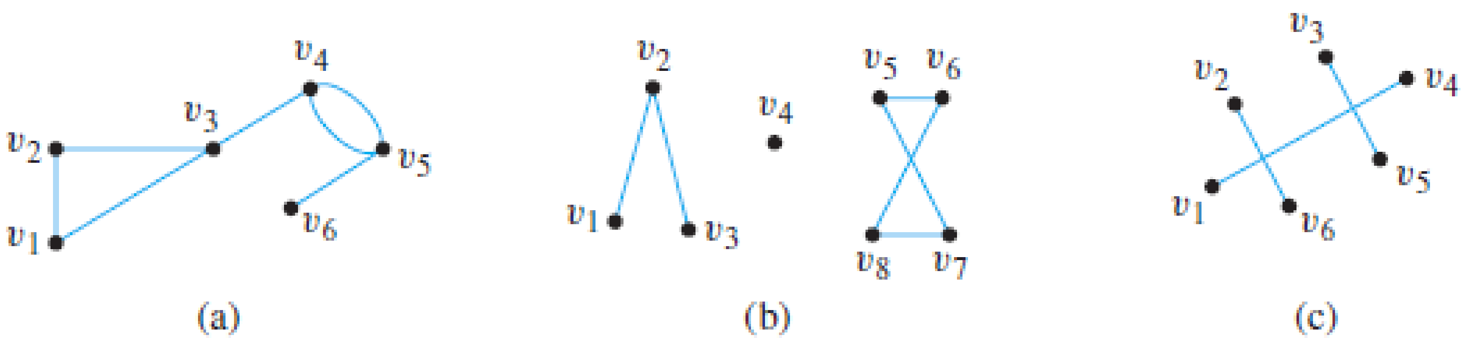
**• Definition**

Let $G$ be a graph. Two **vertices** $v$ **and** $w$ **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,

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

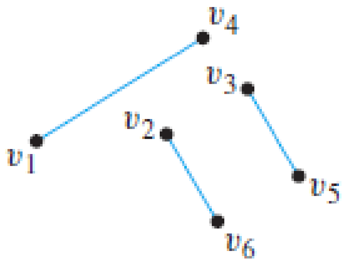
## Example

Which of the following graphs are connected?



# Contd…

**Solution** The graph represented in (a) is connected, whereas those of (b) and (c) are not. To understand why (c) is not connected, recall that in a drawing of a graph, two edges may cross at a point that is not a vertex. Thus the graph in (c) can be redrawn as follows:




## Theorem

Let *G* be a graph.

a. If *G* is connected, then any two distinct vertices of *G* can be connected by a path.

b. 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*.

c. If *G* is connected and *G* contains a circuit, then an edge of the circuit can be removed without disconnecting *G*.

# Connected Component of a Graph

- **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 $H$.

# Contd…

Find all connected components of the following graph *G*.


*G* has three connected components: $H_1$, $H_2$, and $H_3$ with vertex sets $V_1$, $V_2$, and $V_3$ and edge sets $E_1$, $E_2$, and $E_3$, where

$$V_1 = \{v_1, v_2, v_3\}, \qquad E_1 = \{e_1, e_2\},$$
$$V_2 = \{v_4\}, \qquad E_2 = \emptyset,$$
$$V_3 = \{v_5, v_6, v_7, v_8\}, \qquad E_3 = \{e_3, e_4, e_5\}.$$

# Euler Circuits

**• 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.

## Theorem

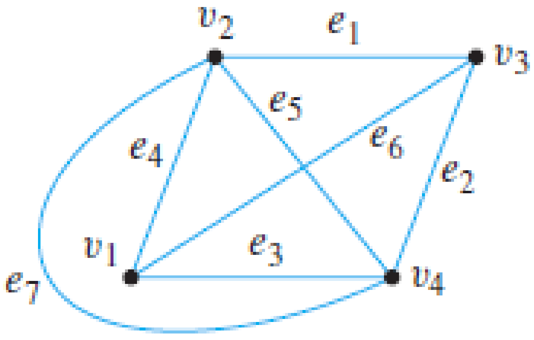
If a graph has an Euler circuit, then every vertex of the graph has positive even degree.

**Contrapositive Version of Theorem**

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

# Euler Circuits

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




Vertices $v_1$ and $v_3$ both have degree 3, which is odd. Hence by (the contrapositive form of theorem), this graph does not have an Euler circuit.

# Constructing 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.

**Constructing an Euler Circuit**

Step I: Pick any vertex $v$ of $G$ at which to start.
Step II: Pick any vertex $v$ of $G$ at which to start.
Step III: Check whether $C$ contains every edge and vertex of $G$. If so, $C$ is an Euler circuit, and we are finished. If not, perform the following steps.

# Constructing an Euler Circuit

Step IIIa: Remove all edges of *C* from *G* and also any vertices that become isolated when the edges of *C* are removed. Call the resulting subgraph *G'*.

Step IIIb: Pick any vertex *w* common to both *C* and *G*'.

Step IIIc: Pick any sequence of adjacent vertices and edges of *G*', starting and ending at *w* and never repeating an edge. Call the resulting circuit *C*'.

Step IIId: Patch *C* and *C*$ together to create a new circuit *C*$$ as follows: Start at *v* and follow *C* all the way to *w*. Then follow *C*$ all the way back to *w*. After that, continue along the untraveled portion of *C* to return to *v*.

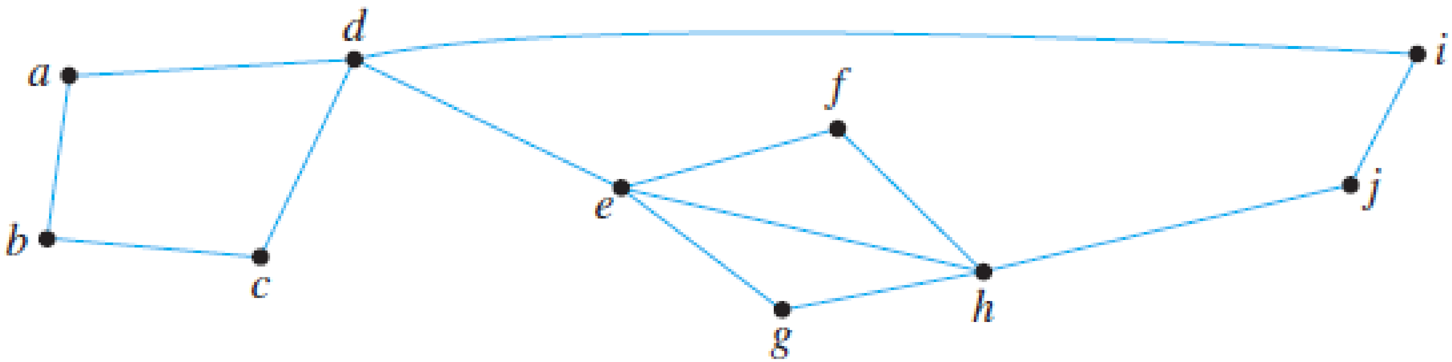
Step IIIe: Let *C* = *C*'' and go back to step 3.

Since the graph *G* is finite, execution of the steps outlined in this algorithm must eventually terminate. At that point an Euler circuit for *G* will have been constructed.

## Finding an Euler Circuit

### Example

Check that the graph below has an Euler circuit. Then use the algorithm to find an Euler circuit for the graph.




Observe that

deg$(a)$ = deg$(b)$ = deg$(c)$ = deg$(f)$ = deg$(g)$ = deg$(i)$ = deg$(j)$ = 2 and that deg$(d)$ = deg$(e)$ = deg$(h)$ = 4.

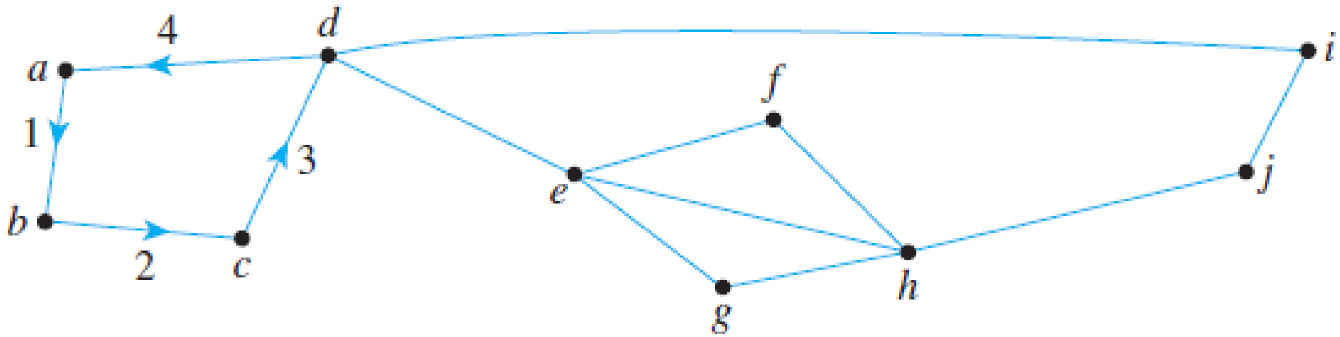
Hence all vertices have even degree. Also, the graph is connected. Thus the graph has an Euler circuit.

# Finding an Euler Circuit

To construct an Euler circuit using the algorithm, let $v = a$ and let $C$ be

$$C : abcda.$$

$C$ is represented by the labeled edges shown below.




Observe that $C$ is not an Euler circuit for the graph but that $C$ intersects the rest of the graph at $d$.

# Finding an Euler Circuit

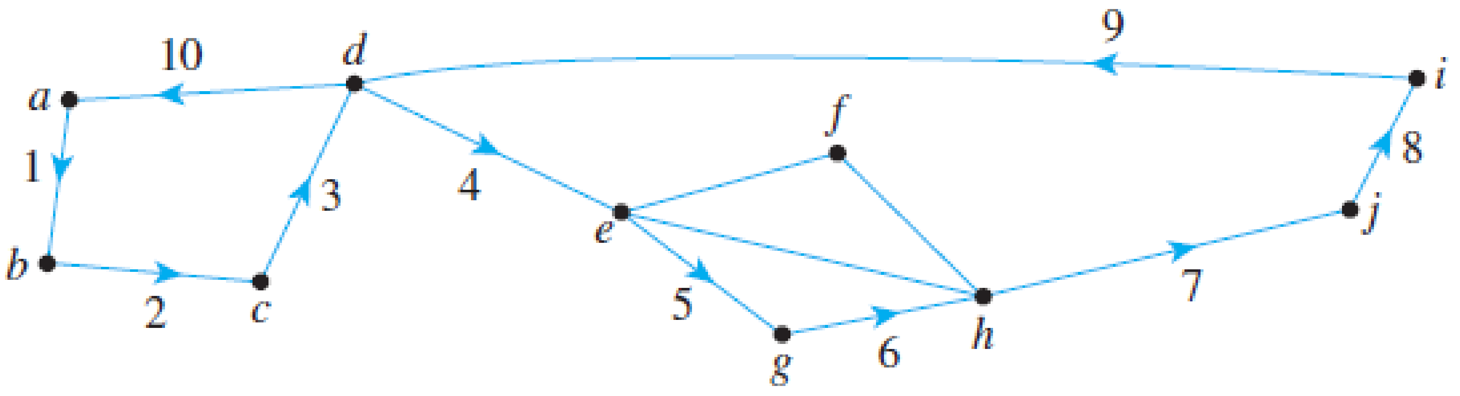
Let *C*' be

*C*': *deghjid.*

Patch *C*' into *C* to obtain

*C*'': *abcdeghjida*.

Set *C* = *C*''. Then *C* is represented by the labeled edges shown below.




Observe that *C* is not an Euler circuit for the graph but that it intersects the rest of the graph at *e*.

# Finding an Euler Circuit

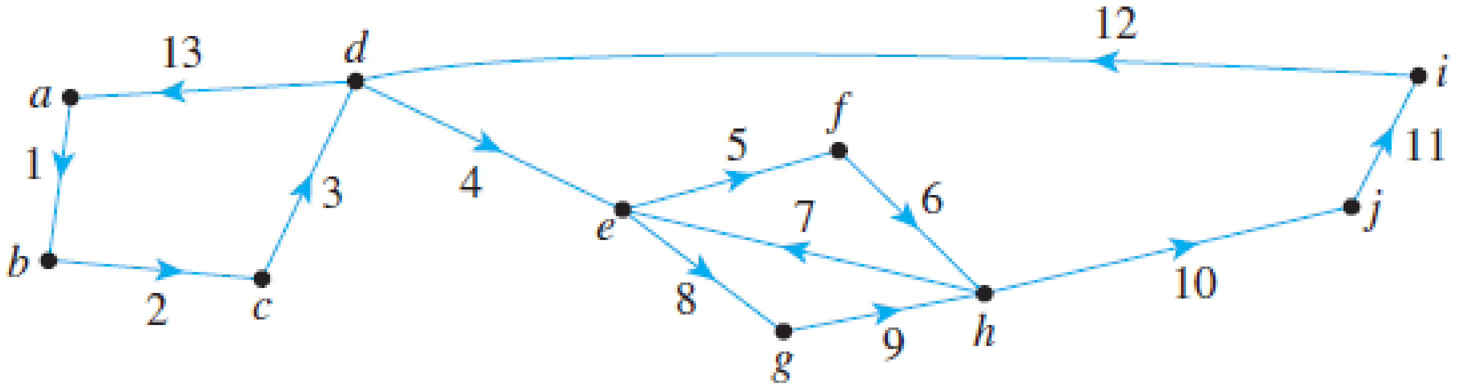
Let *C*' be

*C*': *ef he.*

Patch *C*' into *C* to obtain

*C*'': *abcde f heghjida.*

Set *C* = *C*''. Then *C* is represented by the labeled edges shown below.




Since *C* includes every edge of the graph exactly once, *C* is an Euler circuit for the graph.

# Euler Trails

## Theorem

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

**• Definition**

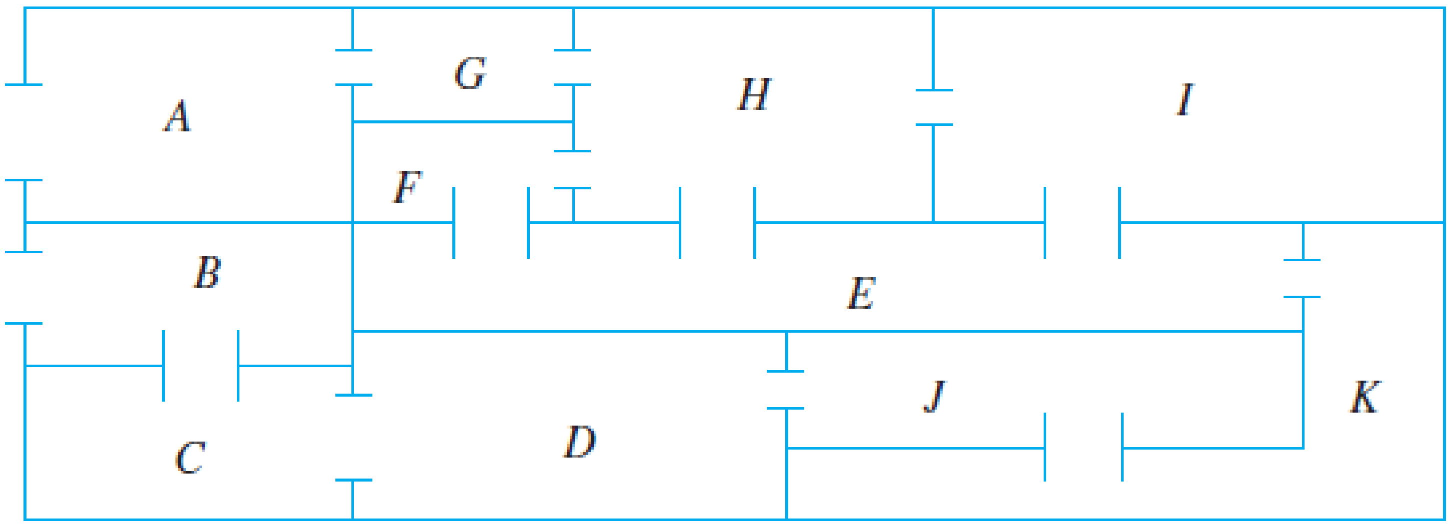
Let $G$ be a graph, and let $v$ and $w$ be two distinct vertices of $G$. An **Euler trail from $v$ to $w$** is a sequence of adjacent edges and vertices that starts at $v$, ends at $w$, passes through every vertex of $G$ at least once, and traverses every edge of $G$ exactly once.

## 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.

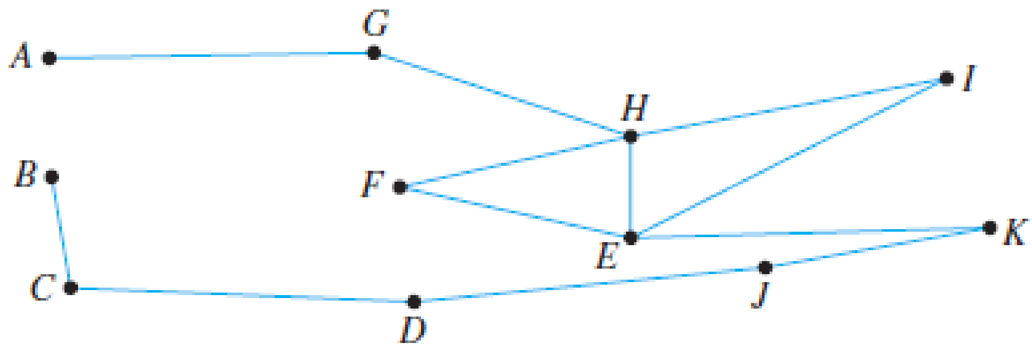
# Finding an Euler Trail

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 door-way of the house exactly once? If so, find such a trail.



# Finding an Euler Trail

Let the floor plan of the house be represented by the graph below.




Each vertex of this graph has even degree except for *A* and *B*, each of which has degree 1. Hence there is an Euler path from *A* to *B*. One such trail is

*AGHFEI HEK J DCB.*