CS201 DISCRETE MATHEMATICS FOR COMPUTER SCIENCE

Dr. QI WANG

Department of Computer Science and Engineering
Office: Room903, Nanshan iPark A7 Building
Email: wangqi@sustech.edu.cn

Graph Concepts

- $G = (V, E)$, *simple* graph, *multigraph*, *pseudograph*
- *Undirected*, *directed* graph
- Special graphs

  K_n, C_n, W_n, Q_n, $K_{m,n}$

  Hall's Marriage Theorem on *bipartite* graphs

Graph Concepts

- $G = (V, E)$, *simple* graph, *multigraph*, *pseudograph*
- *Undirected*, *directed* graph
- Special graphs

  K_n, C_n, W_n, Q_n, $K_{m,n}$

  Hall's Marriage Theorem on *bipartite* graphs
- Representation of graphs

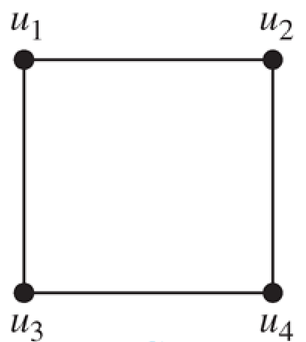
  adjacency list, *adjacency matrix*, *incidence matrix*

Isomorphism of Graphs

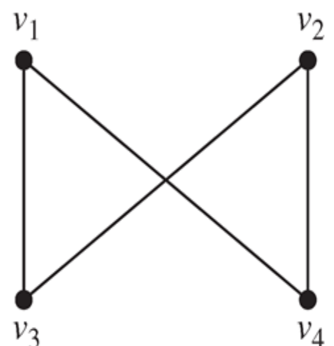
- **Definition** The simple graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are *isomorphic* if there is a one-to-one and onto function from V_1 to V_2 with the property that a and b are adjacent in G_1 if and only if $f(a)$ and $f(b)$ are adjacent in G_2, for all a and b in V_1. Such a function is called an *isomorphism*.

Isomorphism of Graphs

- **Definition** The simple graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are *isomorphic* if there is a one-to-one and onto function from V_1 to V_2 with the property that a and b are adjacent in G_1 if and only if $f(a)$ and $f(b)$ are adjacent in G_2, for all a and b in V_1. Such a function is called an *isomorphism*.



G

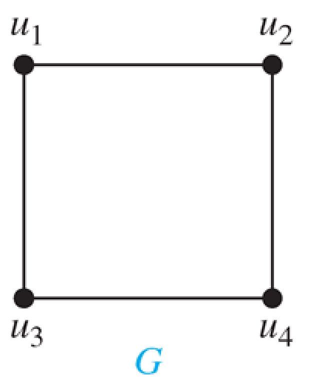


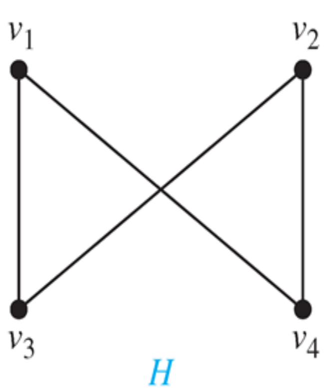
H

Are the two graphs isomorphic?

Isomorphism of Graphs

- **Definition** The simple graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are *isomorphic* if there is a one-to-one and onto function from V_1 to V_2 with the property that a and b are adjacent in G_1 if and only if $f(a)$ and $f(b)$ are adjacent in G_2, for all a and b in V_1. Such a function is called an *isomorphism*.





Are the two graphs isomorphic?

Define a one-to-one correspondence:
$f(u_1) = v_1$, $f(u_2) = v_4$, $f(u_3) = v_3$, and $f(u_4) = v_2$

Isomorphism of Graphs

- It is usually difficult to determine whether two simple graphs are isomorphic using brute force since there are $n!$ possible one-to-one correspondences.

Isomorphism of Graphs

- It is usually difficult to determine whether two simple graphs are isomorphic using brute force since there are $n!$ possible one-to-one correspondences.

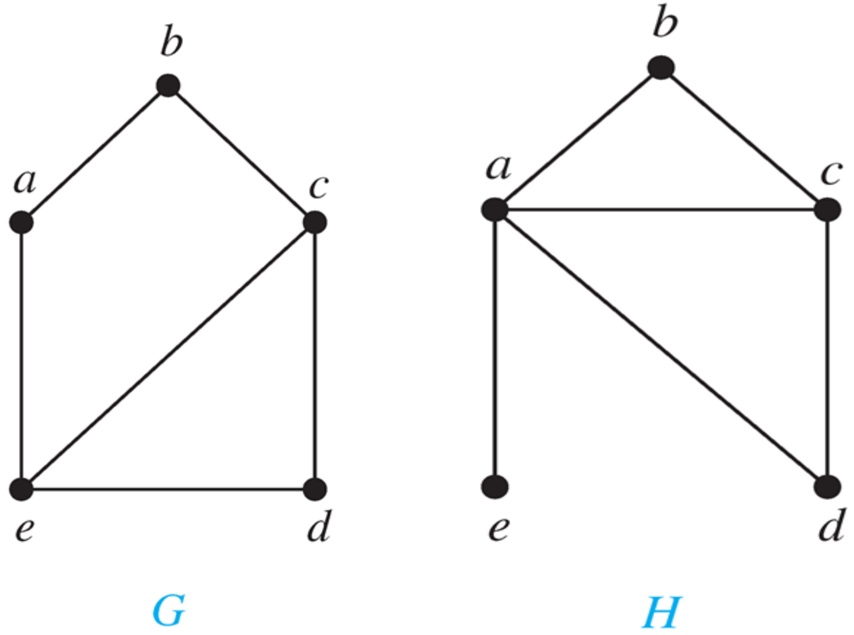
- Sometimes it is not difficult to show that two graphs are not isomorphic. We can achieve this by checking some *graph invariants*.

Isomorphism of Graphs

- It is usually difficult to determine whether two simple graphs are isomorphic using brute force since there are $n!$ possible one-to-one correspondences.
- Sometimes it is not difficult to show that two graphs are not isomorphic. We can achieve this by checking some *graph invariants*.
- Useful graph invariants include the number of vertices, number of edges, degree sequence, etc.

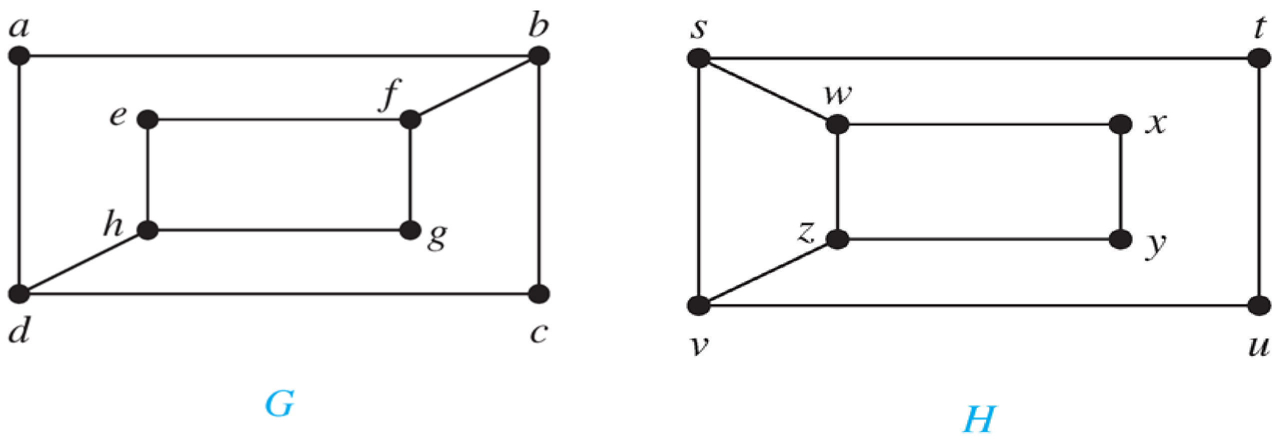
Isomorphism of Graphs

- **Example** Determine whether these two graphs are isomorphic.

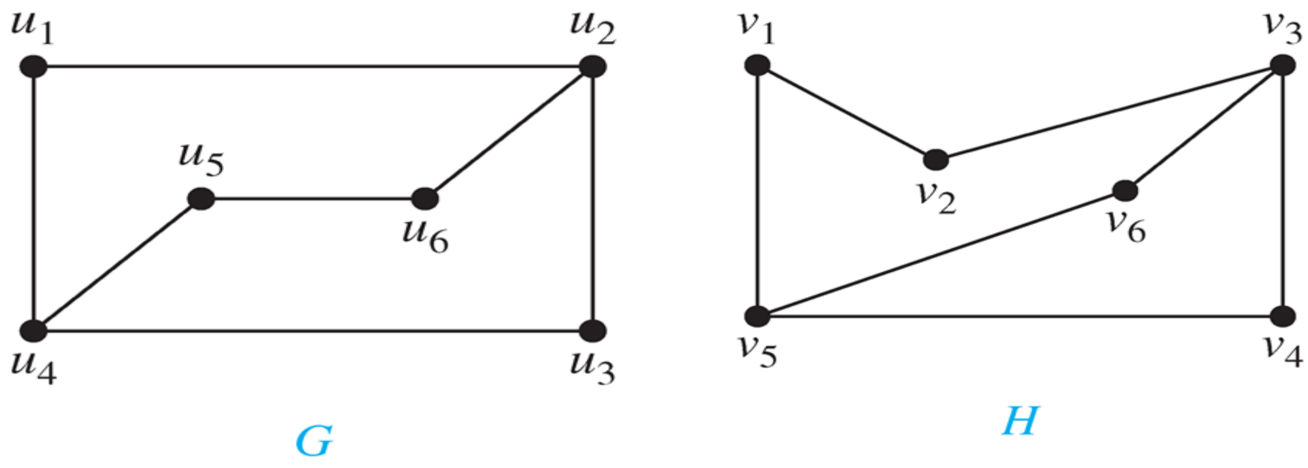


Isomorphism of Graphs

- **Example** Determine whether these two graphs are isomorphic.



- **Example** Determine whether these two graphs are isomorphic.



Path

- **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 $e_1, e_2, \ldots, e_n$ 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 the endpoints x_{i-1} and x_i for $i = 1, \ldots, n$. The path is a *circuit* if it begins and ends at the same vertex, i.e., if $u = v$ and has length greater than zero. A path or circuit is *simple* if it does not contain *repeating vertices*.

Path

- **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 $e_1, e_2, \ldots, e_n$ 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 the endpoints x_{i-1} and x_i for $i = 1, \ldots, n$. The path is a *circuit* if it begins and ends at the same vertex, i.e., if $u = v$ and has length greater than zero. A path or circuit is *simple* if it does not contain *repeating vertices*.

◇ it starts and ends with a vertex

◇ each edge joins the vertex before it in the sequence to the vertex after it in the sequence

◇ no edge appears more than once in the sequence

Path

- **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 $e_1, e_2, \ldots, e_n$ 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 the endpoints x_{i-1} and x_i for $i = 1, \ldots, n$. The path is a *circuit* if it begins and ends at the same vertex, i.e., if $u = v$ and has length greater than zero. A path or circuit is *simple* if it does not contain *repeating vertices*.

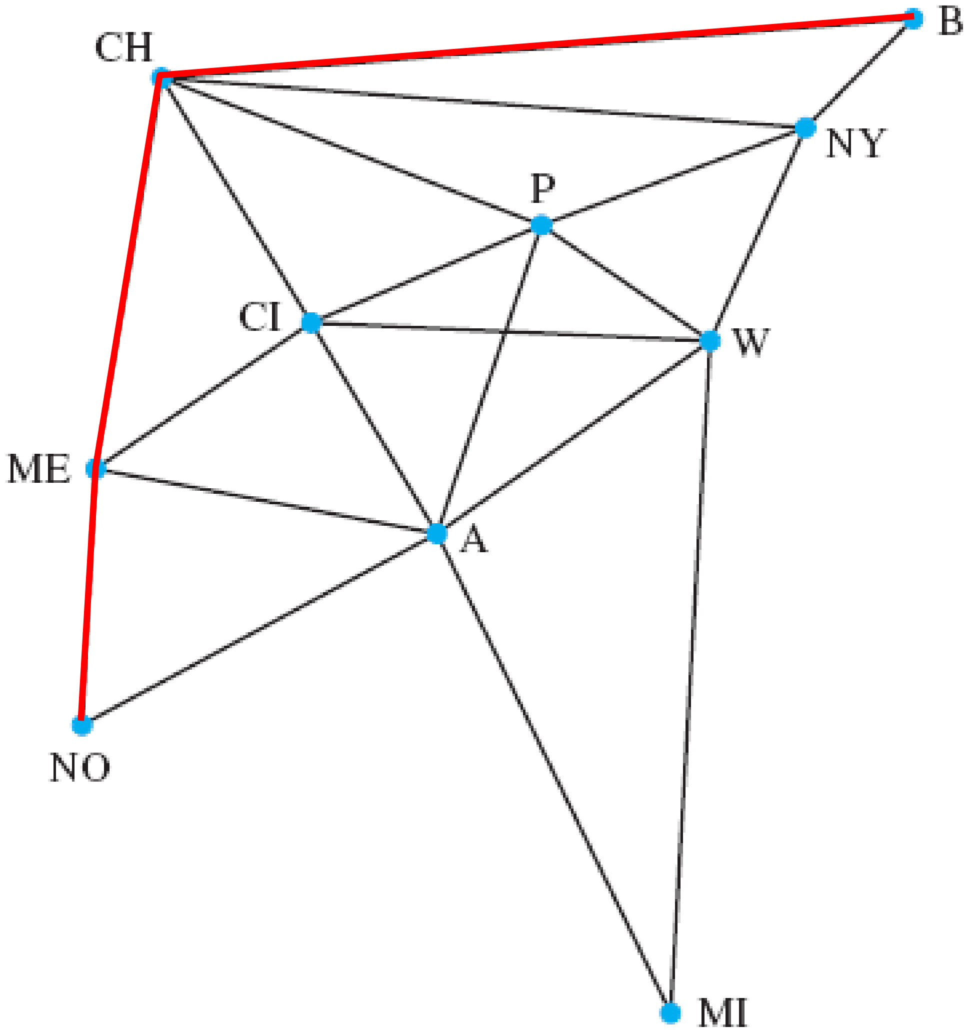
$\diamond$ it starts and ends with a vertex

$\diamond$ each edge joins the vertex before it in the sequence to the vertex after it in the sequence

$\diamond$ no edge appears more than once in the sequence

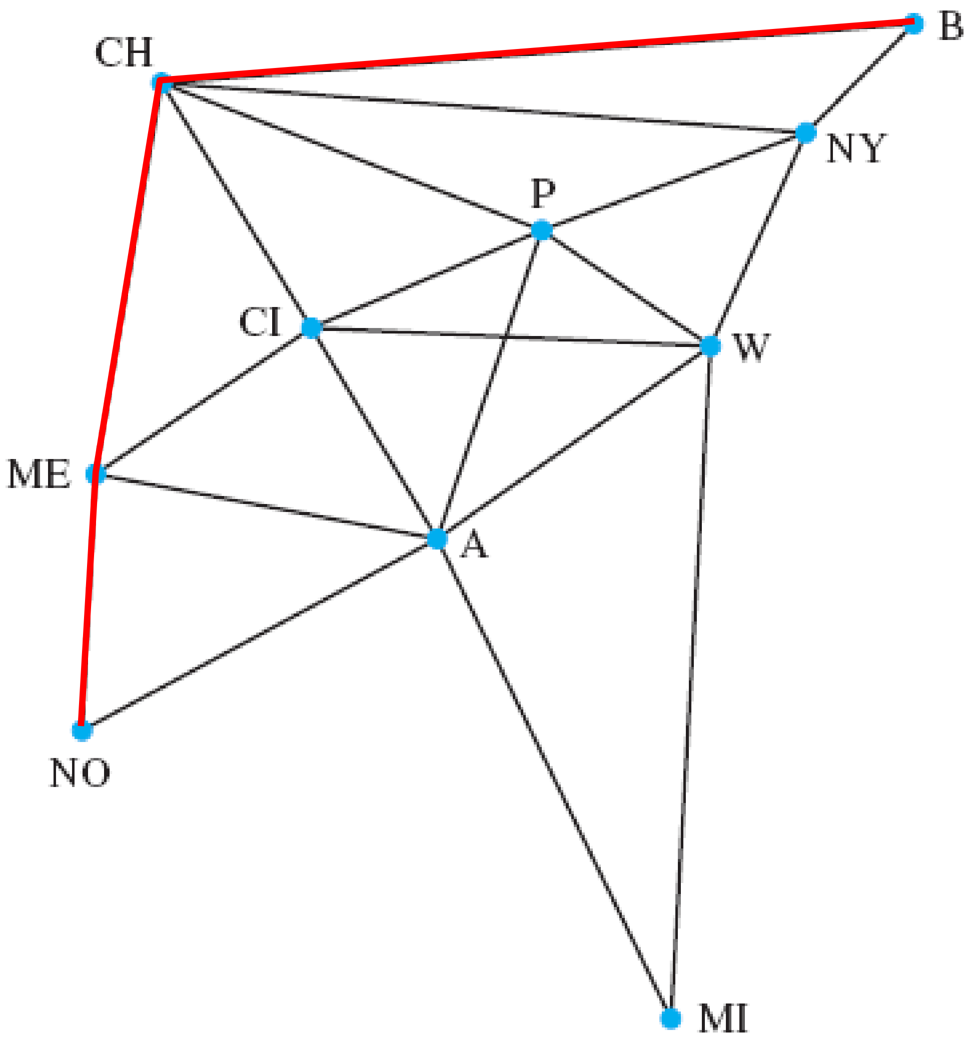
Length of a path $=$ # of edges on path

Path



Path

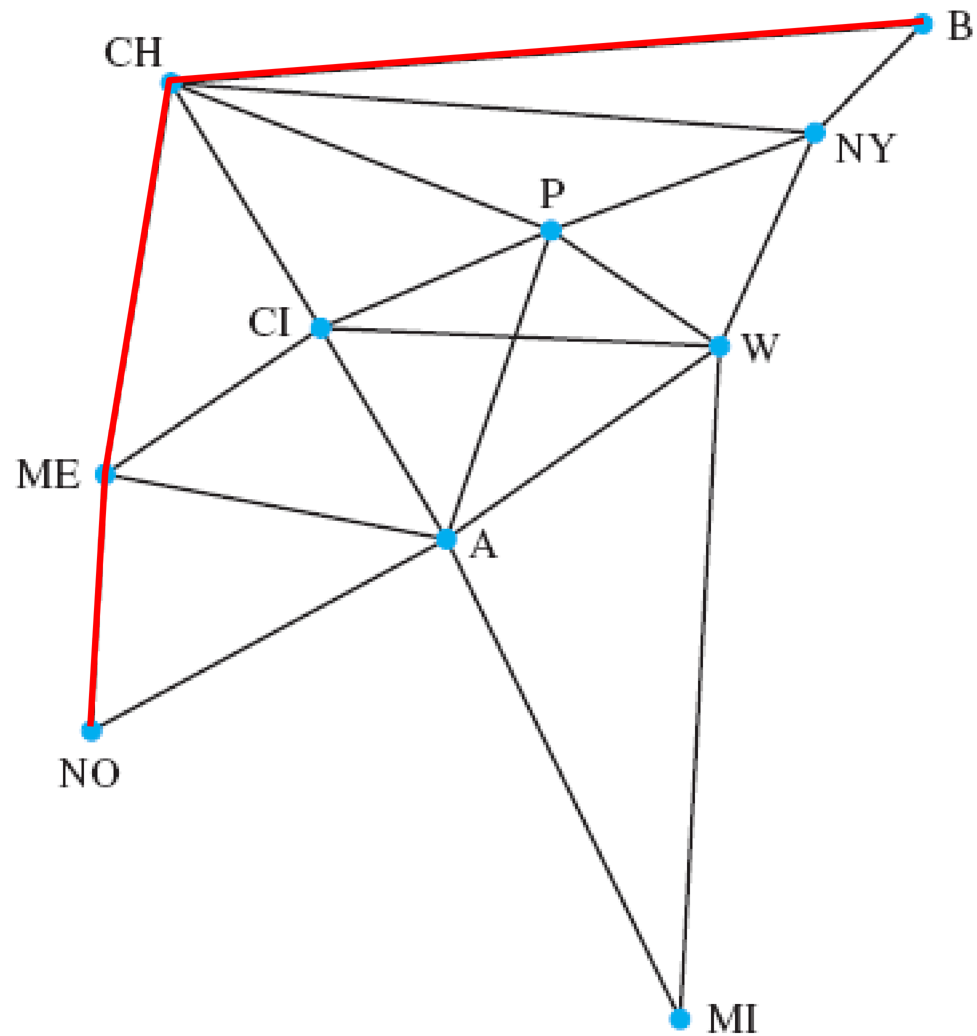
Path from Boston to New Orleans is B, CH, ME, NO



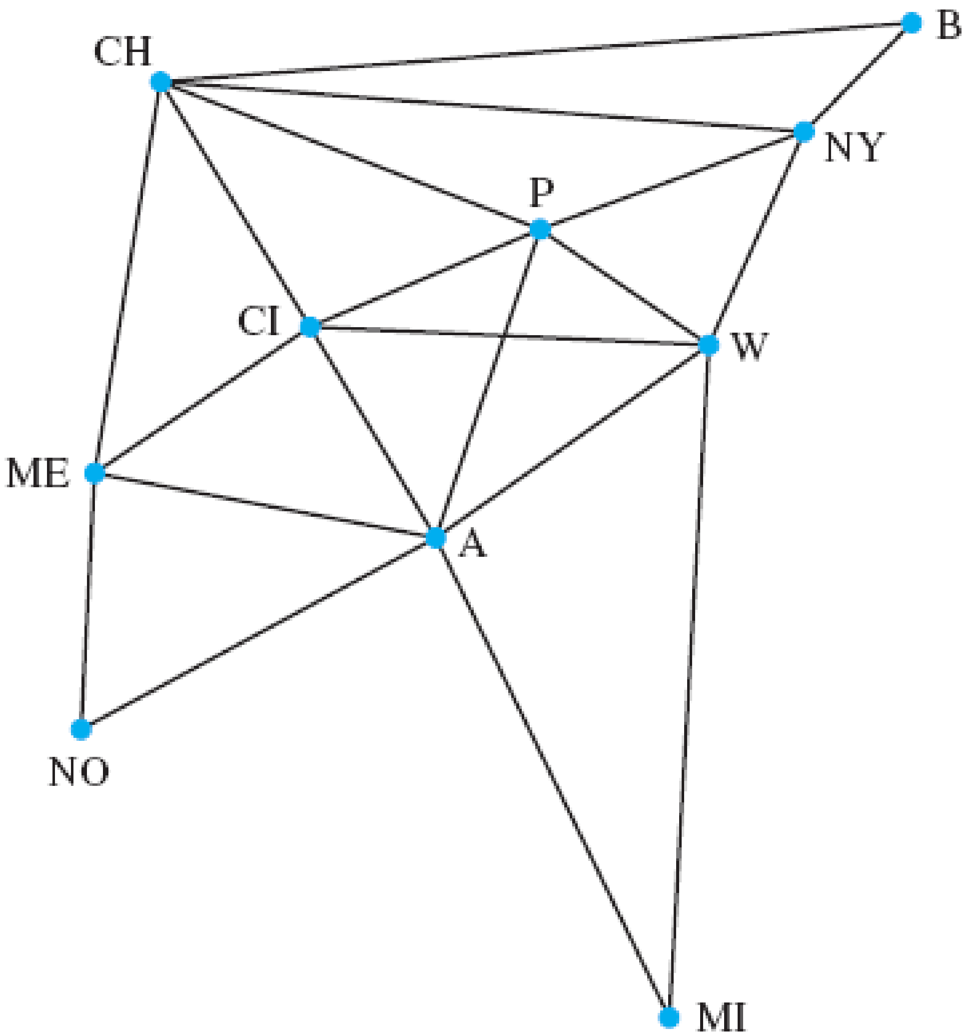
Path

Path from Boston to New Orleans is B, CH, ME, NO

This path has length 3.



Connectivity



Company decides to lease only minimum number of communication lines it needs to be able to send a message from any city to any other city by using any number of intermediate cities.

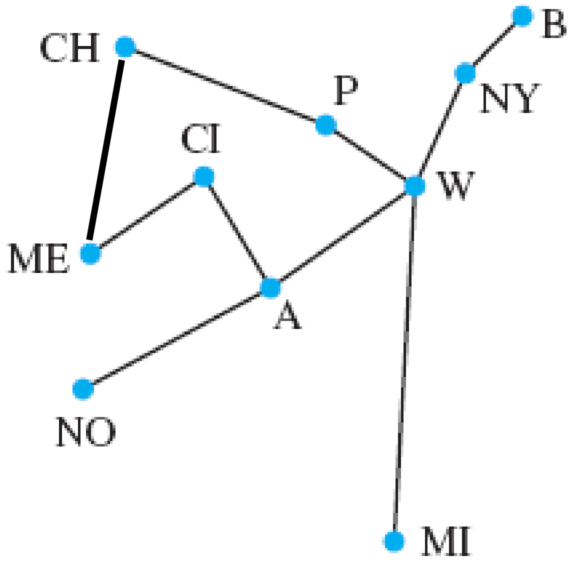
What is the minimum number of lines it needs to lease?

Connectivity

- Choosing 10 edges?

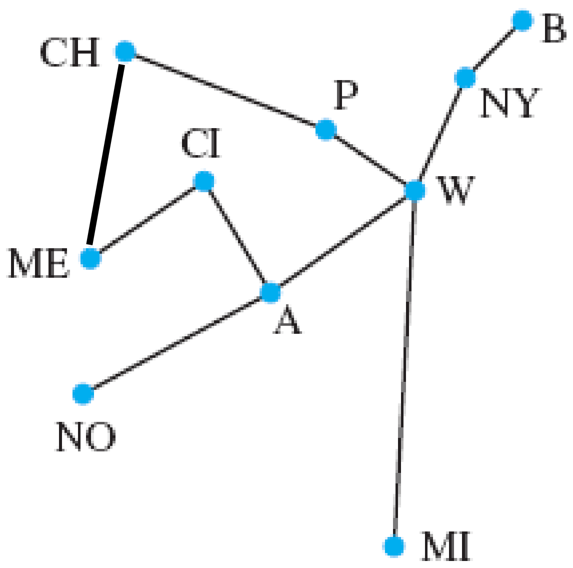
Connectivity

- Choosing 10 edges?



Connectivity

- Choosing 10 edges?

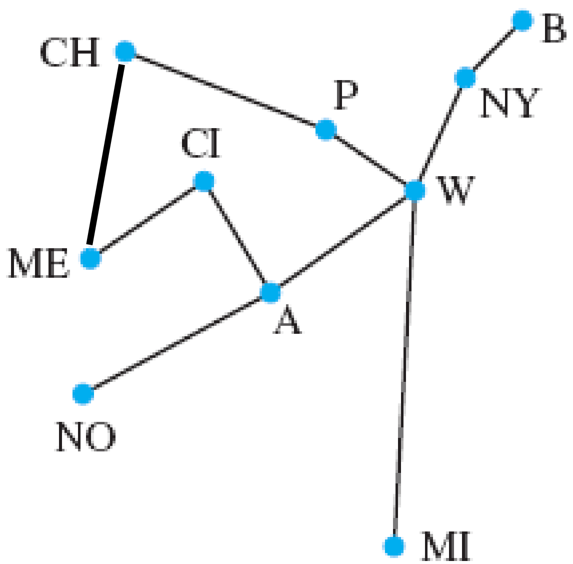


Too many.

Could throw away edge CI, A, and still have a solution.

Connectivity

- Choosing 10 edges?

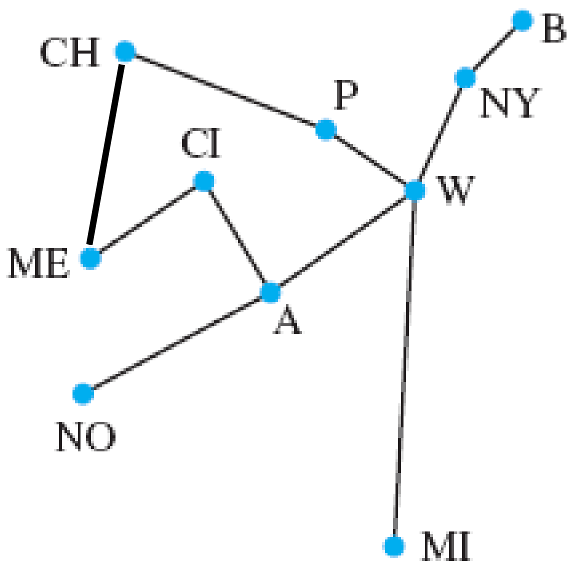


Choosing 8 edges?

Too many.

Could throw away edge CI, A, and still have a solution.

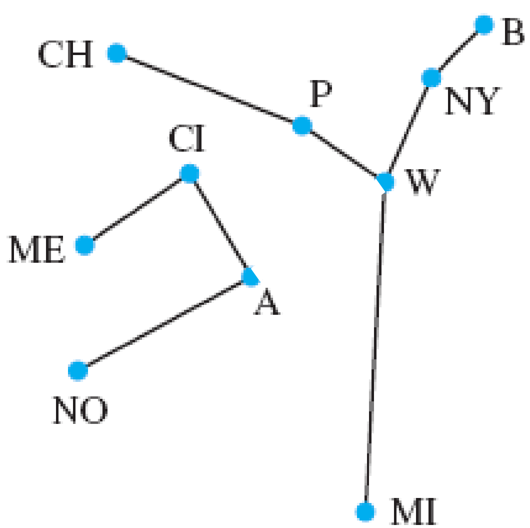
- Choosing 10 edges?



Too many.

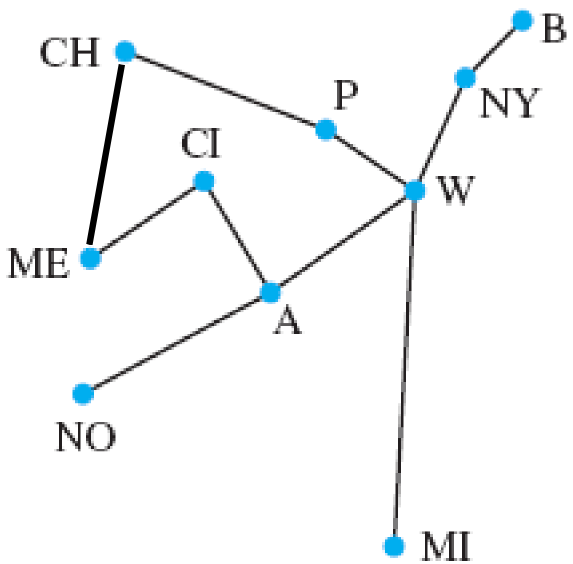
Could throw away edge CI, A, and still have a solution.

Choosing 8 edges?



Connectivity

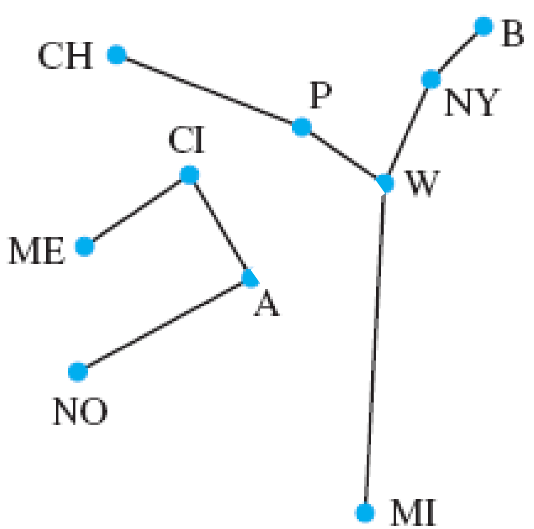
- Choosing 10 edges?



Too many.

Could throw away edge CI, A, and still have a solution.

Choosing 8 edges?



Not enough.

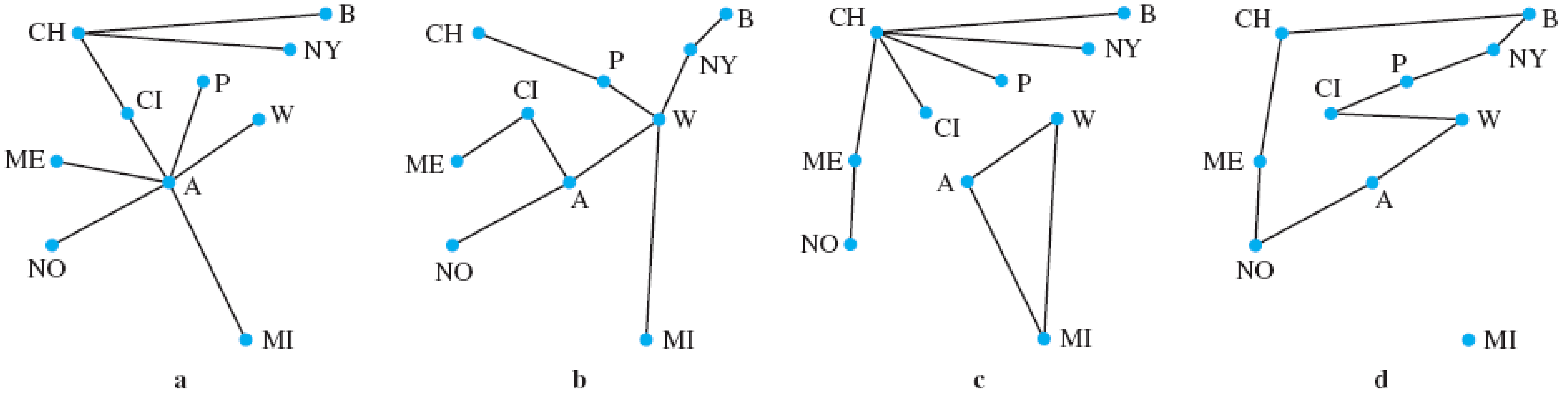
There is no path from, e.g., NO to B.

Connectivity

- Choosing 9 edges:

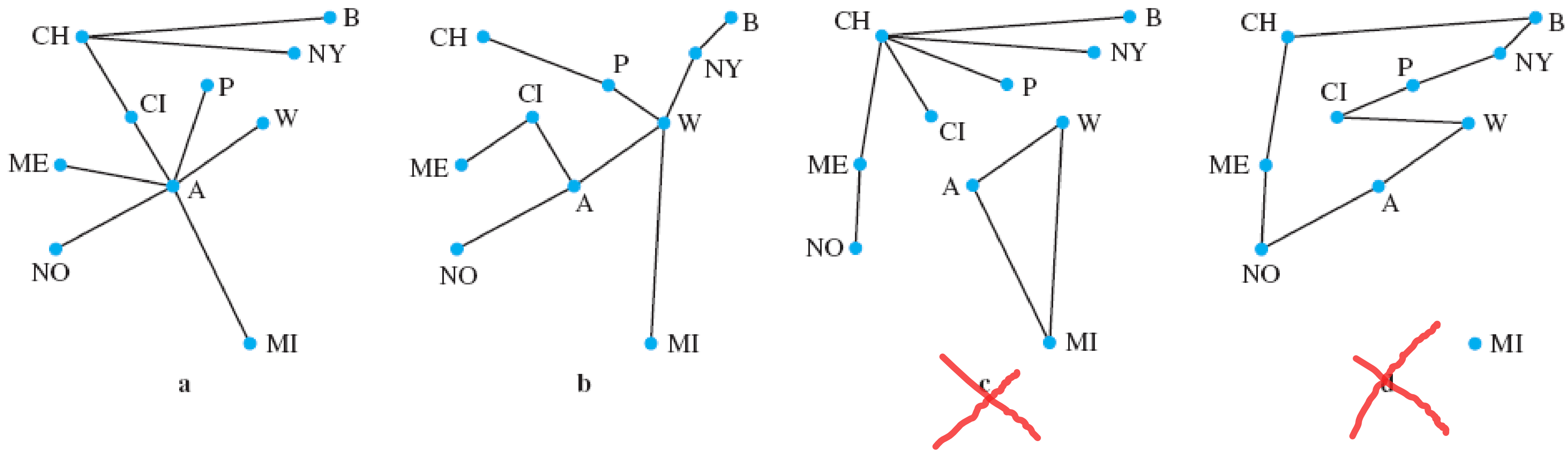
Connectivity

- Choosing 9 edges:



Connectivity

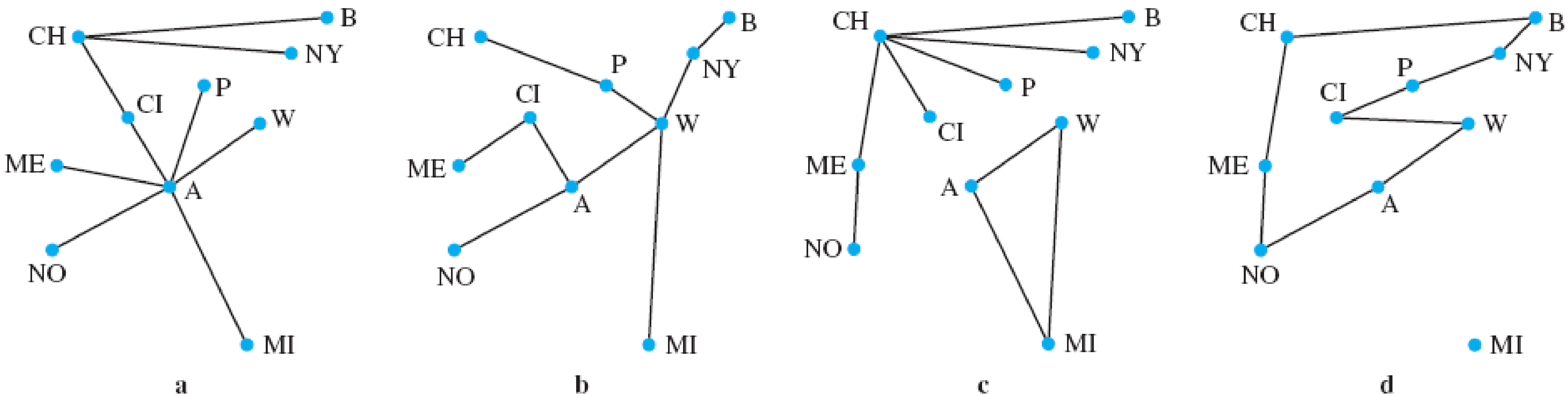
- Choosing 9 edges:



Two vertices are *connected* if there is a path between them.

Connectivity

- Choosing 9 edges:

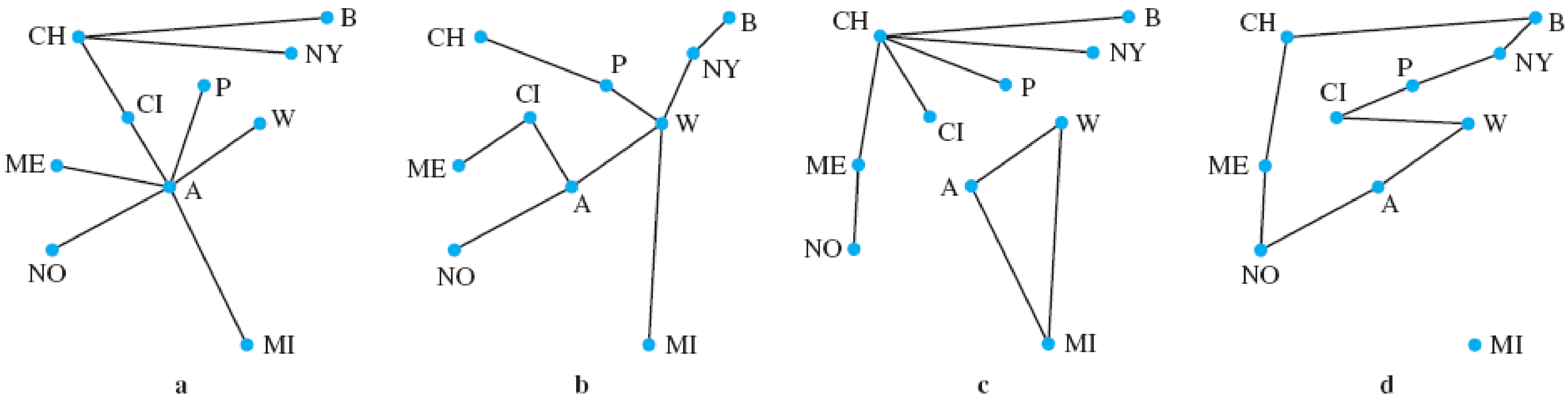


Two vertices are *connected* if there is a path between them.

Example: W, B are connected in (b), but are disconnected in (c).

Connectivity

- Choosing 9 edges:



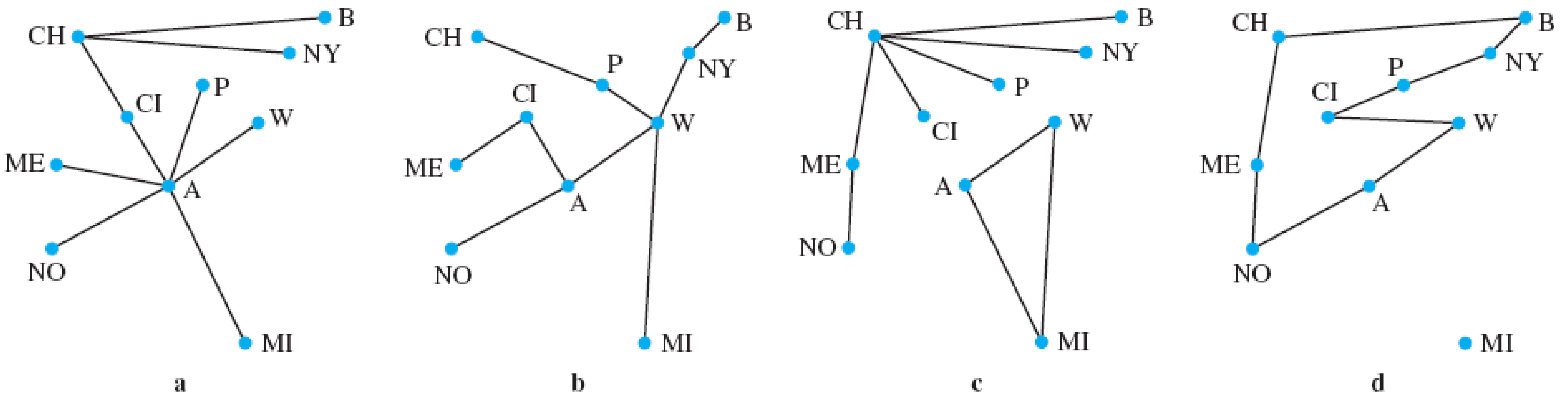
Two vertices are *connected* if there is a path between them.

Example: W, B are connected in (b), but are disconnected in (c).

Definition An undirected graph is called *connected* if there is a path between every pair of distinct vertices of the graph.

Connectivity

- Choosing 9 edges:



Two vertices are *connected* if there is a path between them.

Example: W, B are connected in (b), but are disconnected in (c).

Definition An undirected graph is called *connected* if there is a path between every pair of distinct vertices of the graph.

Example: (a) and (b) are connected, (c) and (d) are disconnected.

Path

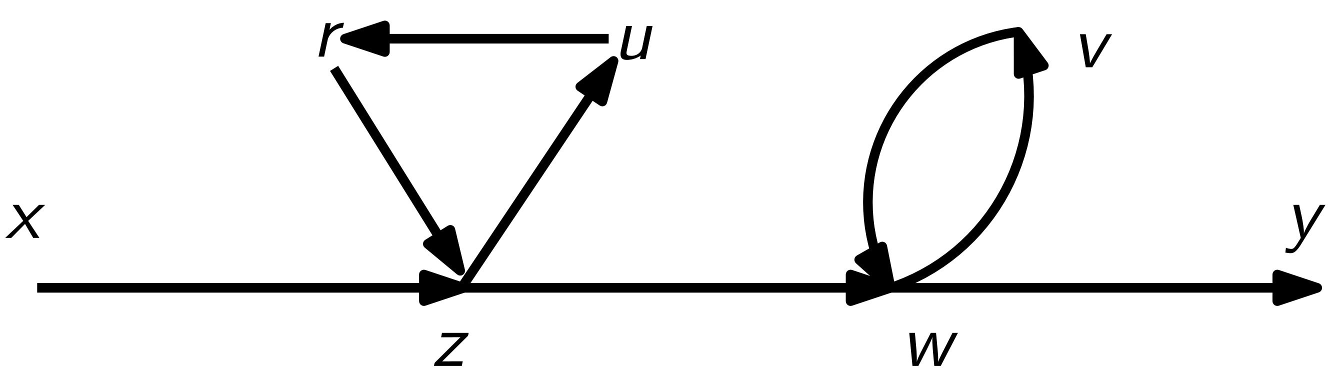
- **Lemma** If there is a path between two distinct vertices x and y of a graph G, then there is a simple path between x and y in G.

Path

- **Lemma** If there is a path between two distinct vertices x and y of a graph G, then there is a simple path between x and y in G.
  Proof Just delete cycles (loops).

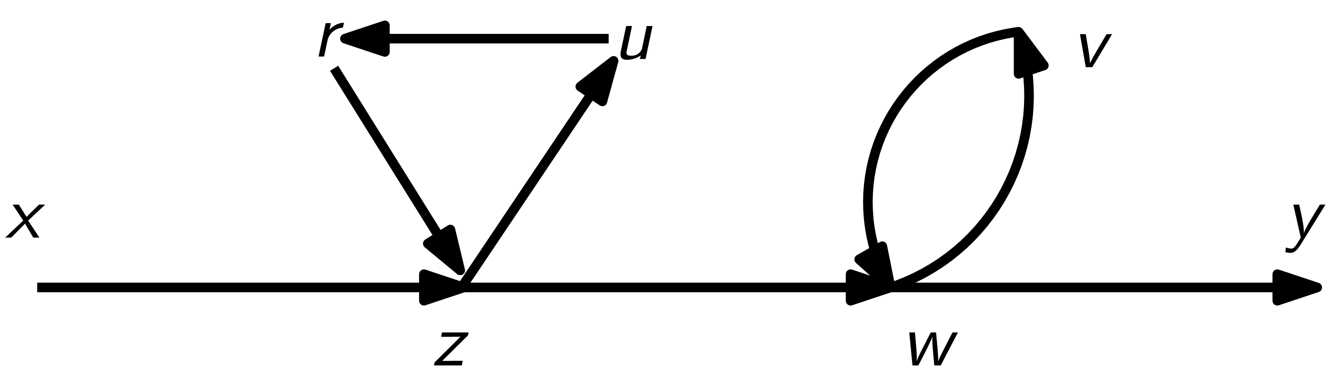
Path

- **Lemma** If there is a path between two distinct vertices x and y of a graph G, then there is a simple path between x and y in G.
  Proof Just delete cycles (loops).



Path

- **Lemma** If there is a path between two distinct vertices x and y of a graph G, then there is a simple path between x and y in G.
  Proof Just delete cycles (loops).

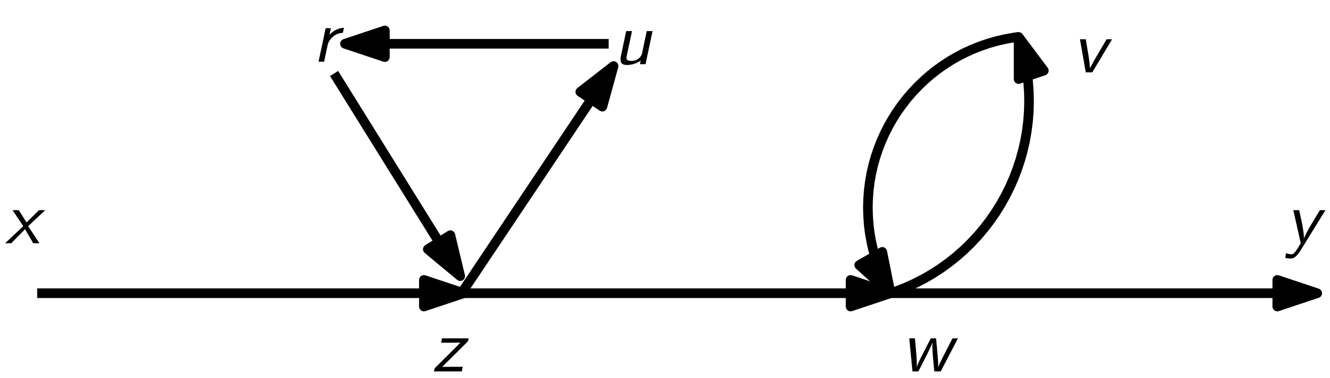


Path from x to y

$x, z, u, r, z, w, v, w, y$.

Path

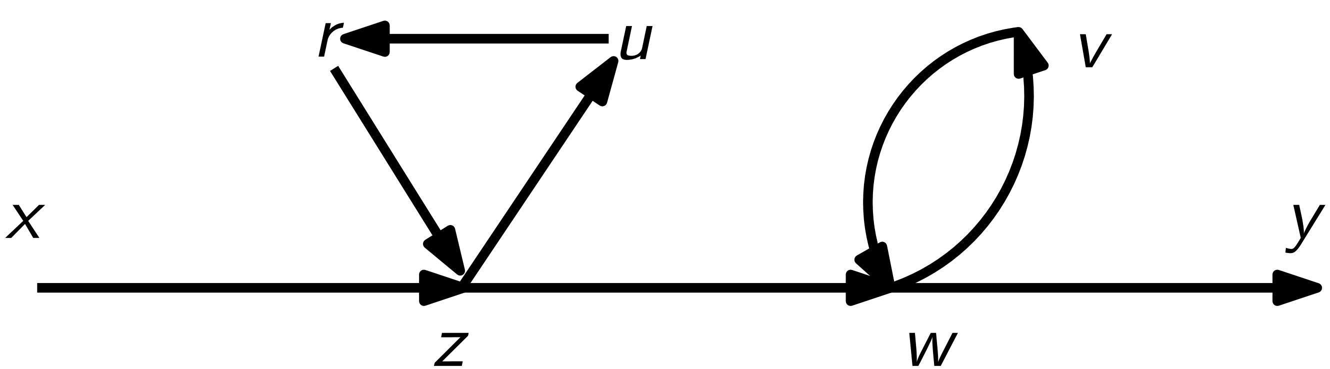
- **Lemma** If there is a path between two distinct vertices x and y of a graph G, then there is a simple path between x and y in G.
  Proof Just delete cycles (loops).



Path from x to y $x, z, u, r, z, w, v, w, y$. → Path from x to y x, z, w, y.

Path

- **Lemma** If there is a path between two distinct vertices x and y of a graph G, then there is a simple path between x and y in G.
  Proof Just delete cycles (loops).



Path from x to y
$x, z, u, r, z, w, v, w, y$.

Path from x to y
x, z, w, y.

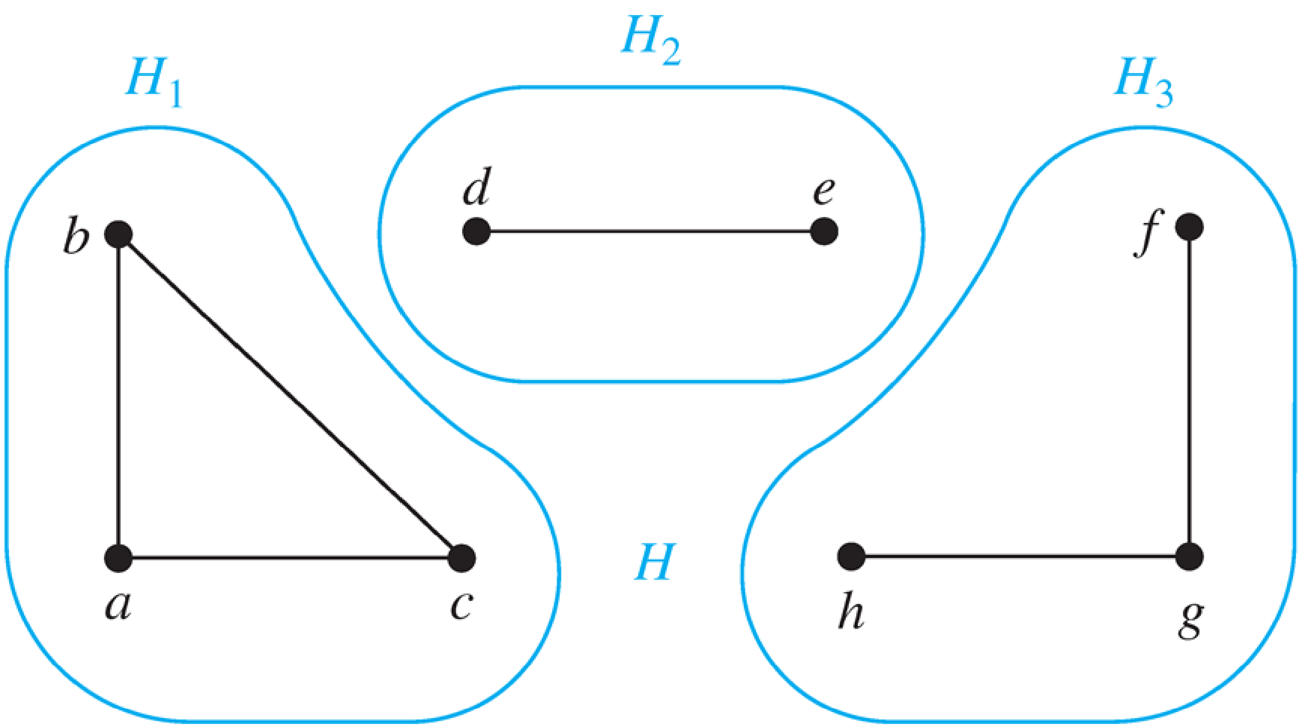
Theorem There is a simple path between every pair of distinct vertices of a connected undirected graph.

Connected Components

- **Definition** 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.

Connected Components

- **Definition** 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.



Connectedness in Directed Graphs

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

Connectedness in Directed Graphs

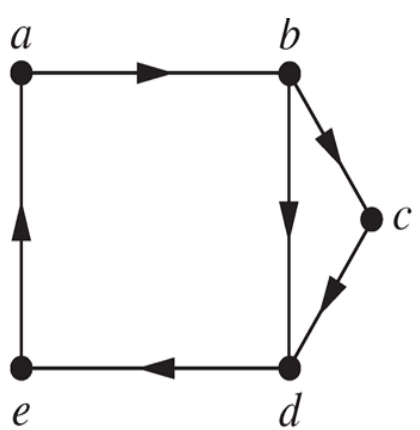
- **Definition** A directed graph is *strongly connected* if there is a path from a to b and a path from b to a whenever a and b are vertices in the graph.

  Definition A directed graph is *weakly connected* if there is a path between every two vertices in the underlying undirected graph, which is the undirected graph obtained by ignoring the directions of the edges in the directed graph.

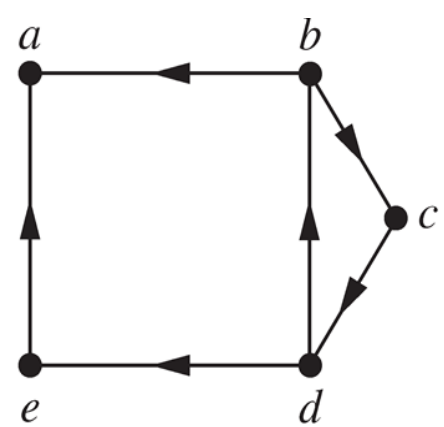
Connectedness in Directed Graphs

- **Definition** A directed graph is *strongly connected* if there is a path from a to b and a path 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, which is the undirected graph obtained by ignoring the directions of the edges in the directed graph.



G



H

Cut Vertices and Cut Edges

- Sometimes the removal from a graph of a vertex and all incident edges disconnect the graph. Such vertices are called *cut vertices*. Similarly we may define *cut edges*.

Cut Vertices and Cut Edges

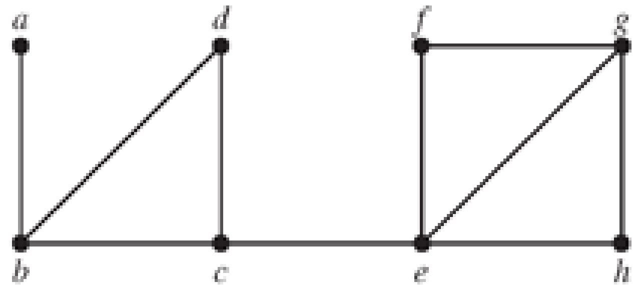
- Sometimes the removal from a graph of a vertex and all incident edges disconnect the graph. Such vertices are called *cut vertices*. Similarly we may define *cut edges*.

  A set of edges E' is called an *edge cut* of G if the subgraph $G - E'$ is disconnected. The *edge connectivity* $\lambda(G)$ is the minimum number of edges in an edge cut of G.

Cut Vertices and Cut Edges

- Sometimes the removal from a graph of a vertex and all incident edges disconnect the graph. Such vertices are called *cut vertices*. Similarly we may define *cut edges*.

  A set of edges E' is called an *edge cut* of G if the subgraph $G - E'$ is disconnected. The *edge connectivity* $\lambda(G)$ is the minimum number of edges in an edge cut of G.



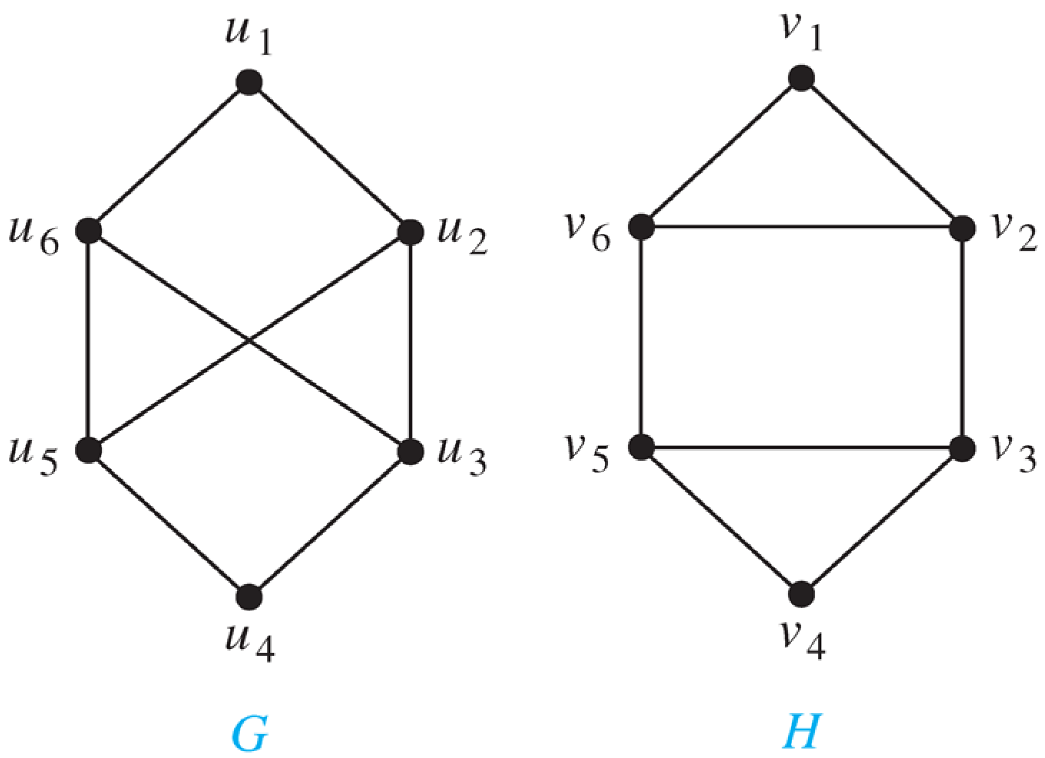
G_1

Paths and Isomorphism

- The existence of a simple circuit of length k is isomorphic invariant. In addition, paths can be used to construct mappings that may be isomorphisms.

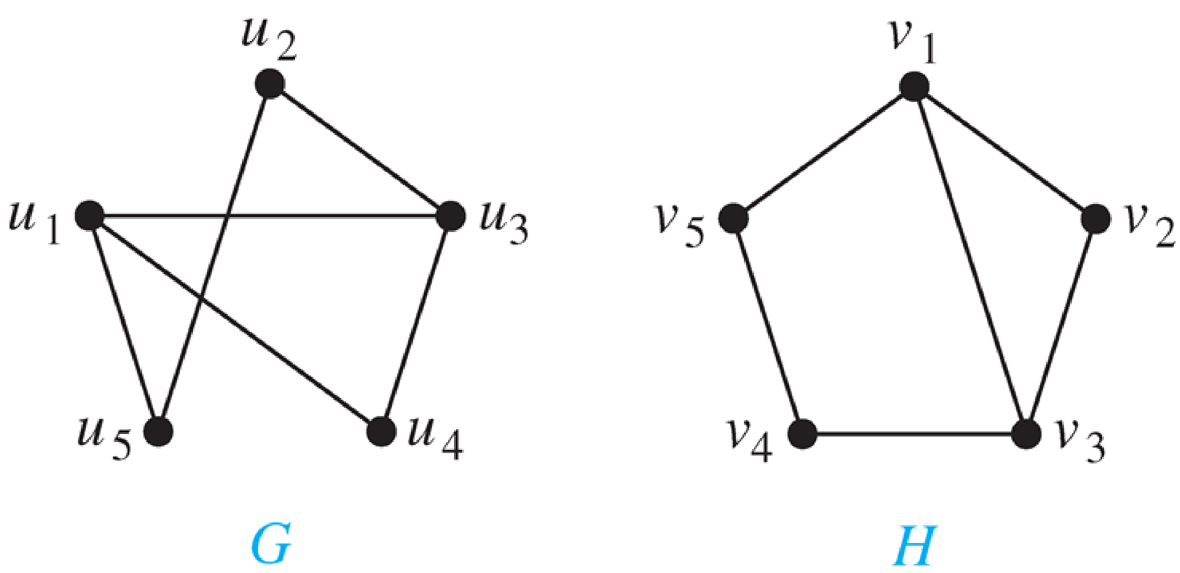
Paths and Isomorphism

- The existence of a simple circuit of length k is isomorphic invariant. In addition, paths can be used to construct mappings that may be isomorphisms.



Paths and Isomorphism

- The existence of a simple circuit of length k is isomorphic invariant. In addition, paths can be used to construct mappings that may be isomorphisms.



Counting Paths between Vertices

- **Theorem** Let G be a graph with adjacency matrix $\mathbf{A}$ with respect to the ordering $v_1, v_2, \ldots, v_n$ of vertices. The number of different paths of length r from v_i to v_j, where $r > 0$ is positive, equals the (i, j)-th entry of $\mathbf{A}^r$.

Counting Paths between Vertices

- **Theorem** Let G be a graph with adjacency matrix $\mathbf{A}$ with respect to the ordering $v_1, v_2, \ldots, v_n$ of vertices. The number of different paths of length r from v_i to v_j, where $r > 0$ is positive, equals the (i,j)-th entry of $\mathbf{A}^r$.

Proof (by **induction**)

Counting Paths between Vertices

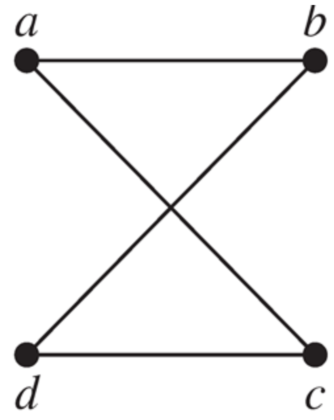
- **Theorem** Let G be a graph with adjacency matrix $\mathbf{A}$ with respect to the ordering $v_1, v_2, \ldots, v_n$ of vertices. The number of different paths of length r from v_i to v_j, where $r > 0$ is positive, equals the (i, j)-th entry of $\mathbf{A}^r$.

Proof (by **induction**)

$\mathbf{A}^{r+1} = \mathbf{A}^r\mathbf{A}$, the (i, j)-th entry of $\mathbf{A}^{r+1}$ equals $b_{i1}a_{1j} + b_{i2}a_{2j} + \cdots + b_{in}a_{nj}$, where b_{ik} is the (i, k)-th entry of $\mathbf{A}^r$.

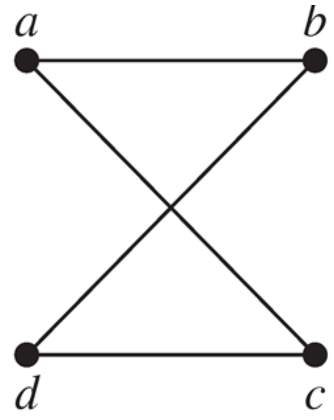
Counting Paths between Vertices

- **Example** How many paths of length 4 are there from a to d in the graph G?



Counting Paths between Vertices

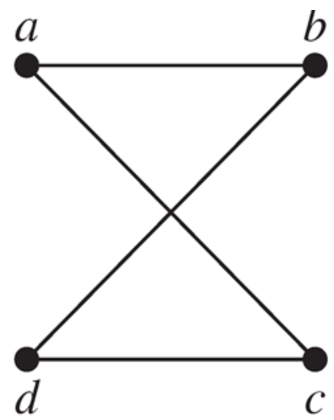
- **Example** How many paths of length 4 are there from a to d in the graph G?



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

Counting Paths between Vertices

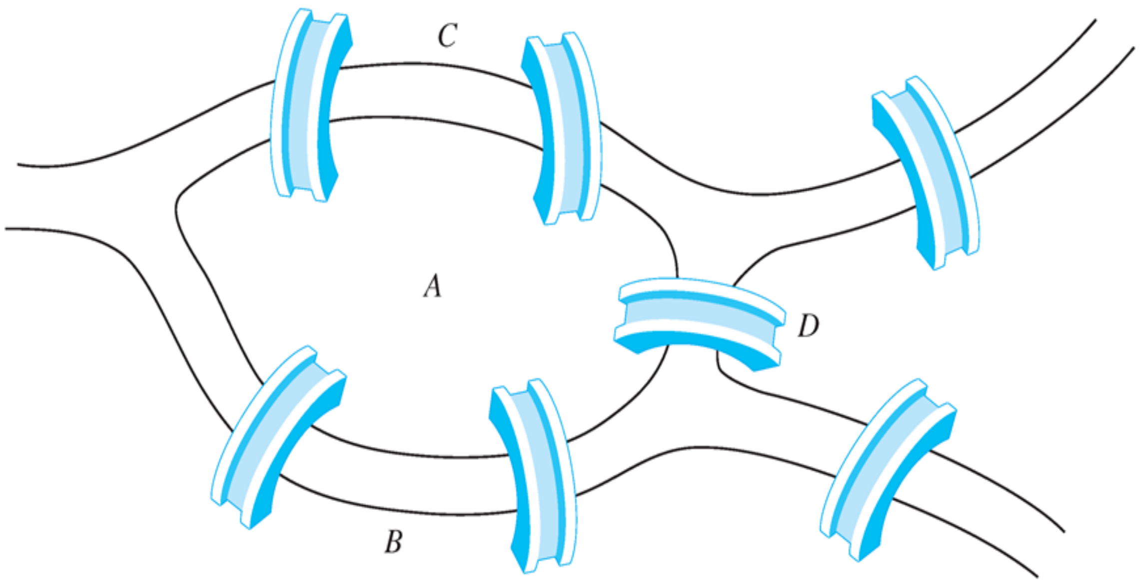
- **Example** How many paths of length 4 are there from a to d in the graph G?



$$\begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \qquad \begin{bmatrix} 8 & 0 & 0 & 8 \\ 0 & 8 & 8 & 0 \\ 0 & 8 & 8 & 0 \\ 8 & 0 & 0 & 8 \end{bmatrix}$$

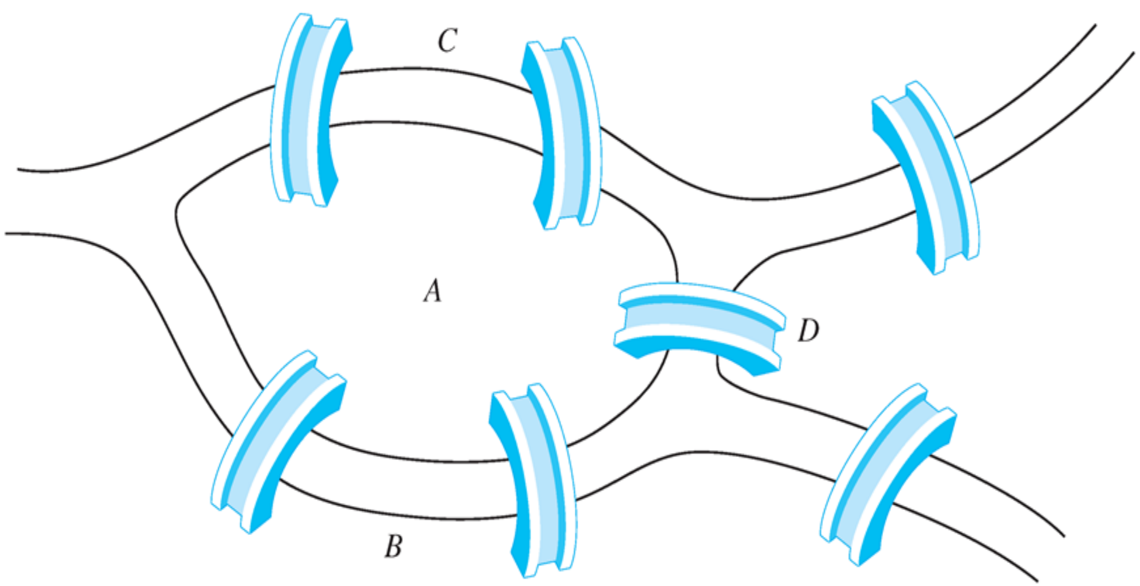
- **Königsberg seven-bridge problem**

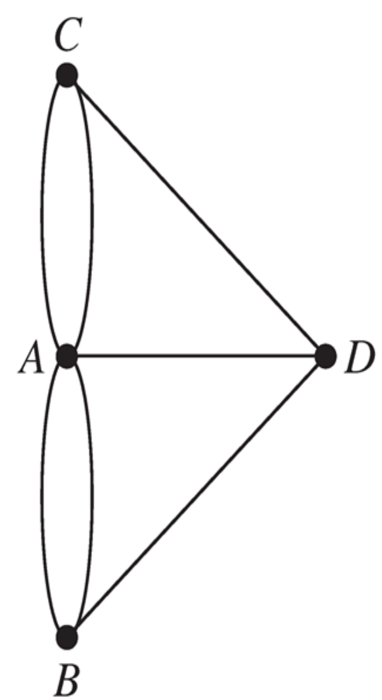
  People wondered whether it was possible to start at some location in the town, travel across all the bridges once without crossing any bridge twice, and return to the starting point.



- **Königsberg seven-bridge problem**

  People 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.



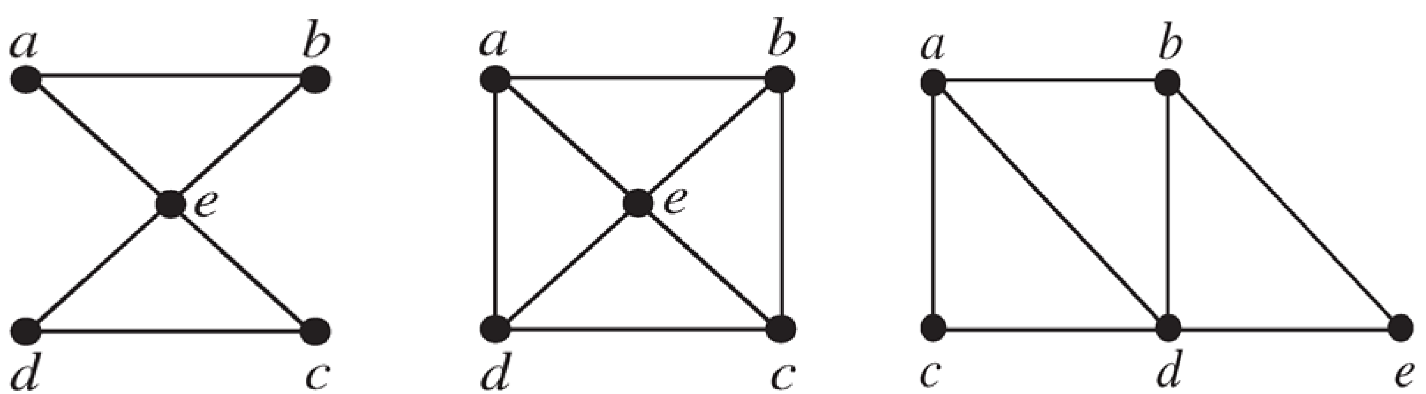


Euler Paths and Circuits

- **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.

- **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.

Example Which of the undirected graphs have an Euler circuit? Of those that do not, which have an Euler path?

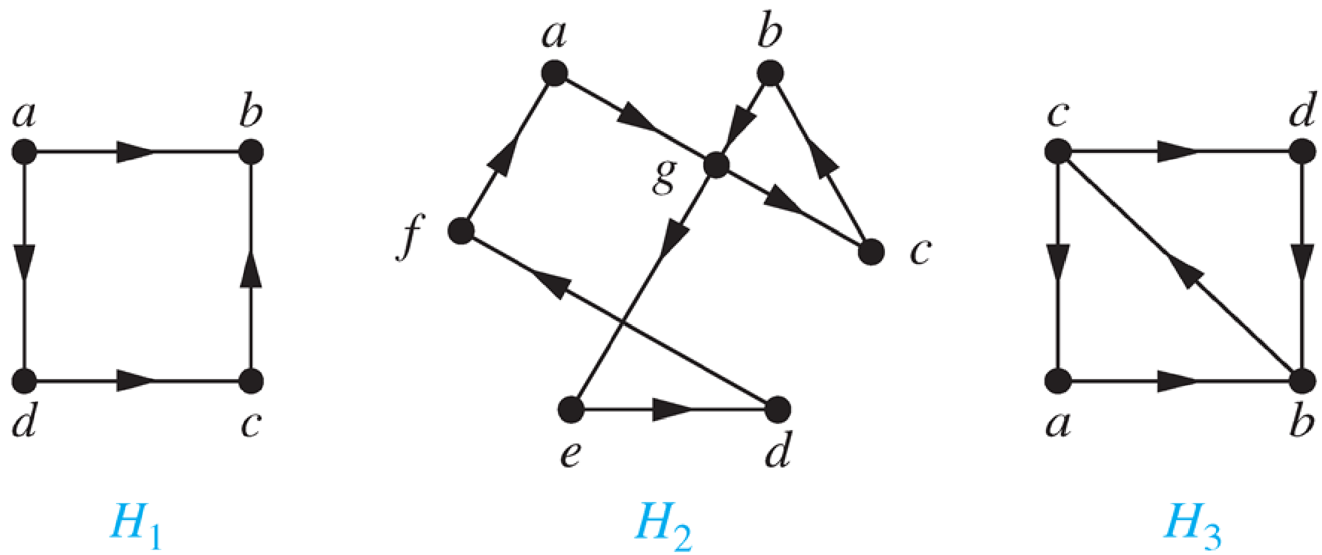


G_1 G_2 G_3

Euler Paths and Circuits

- **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.

Example Which of the undirected graphs have an Euler circuit? Of those that do not, which have an Euler path?



Necessary Conditions for Euler Circuits and Paths

- Euler Circuit $\Rightarrow$ The degree of every vertex must be even

Necessary Conditions for Euler Circuits and Paths

- Euler Circuit $\Rightarrow$ The degree of every vertex must be even
  - Each time the circuit passes through a vertex, it contributes two to the vertex's degree.

Necessary Conditions for Euler Circuits and Paths

- Euler Circuit $\Rightarrow$ The degree of every vertex must be even
  - Each time the circuit passes through a vertex, it contributes two to the vertex's degree.
  - The circuit starts with a vertex a and ends at a, then contributes two to $\deg(a)$.

Necessary Conditions for Euler Circuits and Paths

- Euler Circuit $\Rightarrow$ The degree of every vertex must be even
  - Each time the circuit passes through a vertex, it contributes two to the vertex's degree.
  - The circuit starts with a vertex a and ends at a, then contributes two to $\deg(a)$.

Euler Path $\Rightarrow$ The graph has exactly two vertices of odd degree

Necessary Conditions for Euler Circuits and Paths

- Euler Circuit $\Rightarrow$ The degree of every vertex must be even
  - Each time the circuit passes through a vertex, it contributes two to the vertex's degree.
  - The circuit starts with a vertex a and ends at a, then contributes two to $\deg(a)$.

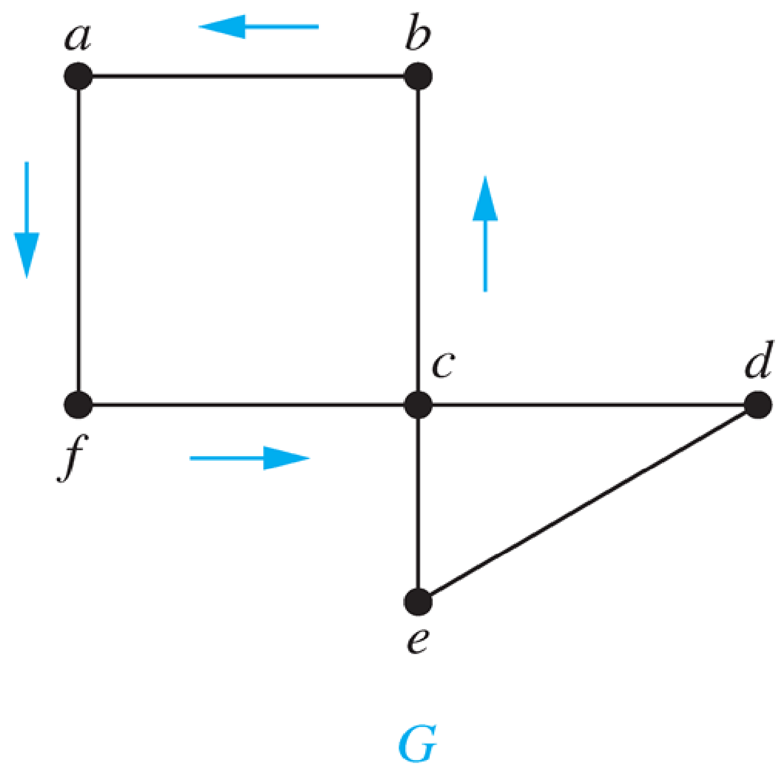
Euler Path $\Rightarrow$ The graph has exactly two vertices of odd degree

- The initial vertex and the final vertex of an Euler path have odd degree.

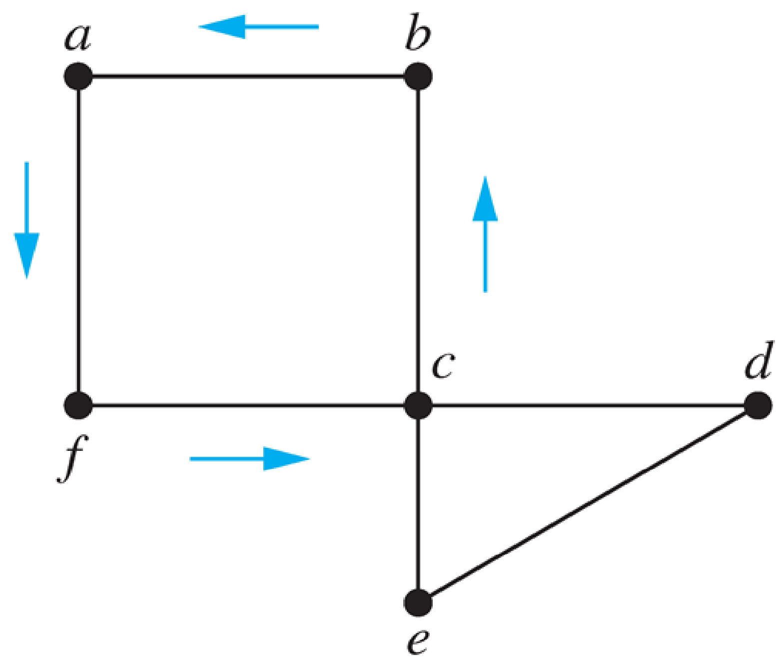
Sufficient Conditions for Euler Circuits and Paths

- Suppose that G is a connected multigraph with ≥ 2 vertices, all of even degree.

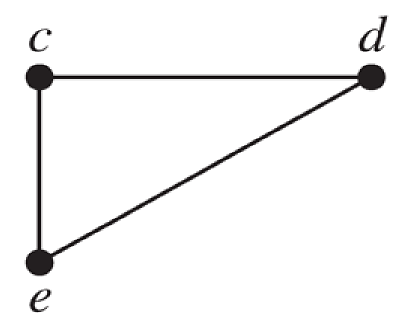
- Suppose that G is a connected multigraph with ≥ 2 vertices, all of even degree.



- Suppose that G is a connected multigraph with ≥ 2 vertices, all of even degree.



G



H

Algorithm for Constructing an Euler Circuit

- **procedure** *Euler*(G: connected multigraph with all vertices of even degree)
  circuit := a circuit in G beginning at an arbitrarily chosen vertex with edges
  successively added to form a path that returns to this vertex.
  H := G with the edges of this circuit removed
  while H has edges
  subciruit := a circuit in H beginning at a vertex in H that also is
  an endpoint of an edge in circuit.
  H := H with edges of *subciruit* and all isolated vertices removed
  circuit := *circuit* with *subcircuit* inserted at the appropriate vertex.
  return *circuit*{*circuit* is an Euler circuit}

Algorithm for Constructing an Euler Circuit

procedure *Euler*(G: connected multigraph with all vertices of even degree)
circuit := a circuit in G beginning at an arbitrarily chosen vertex with edges
successively added to form a path that returns to this vertex.
H := G with the edges of this circuit removed
while H has edges
subciruit := a circuit in H beginning at a vertex in H that also is
an endpoint of an edge in circuit.
H := H with edges of *subciruit* and all isolated vertices removed
circuit := *circuit* with *subcircuit* inserted at the appropriate vertex.
return *circuit*{*circuit* is an Euler circuit}

Algorithm for Constructing an Euler Circuit

- **procedure** *Euler*(G: connected multigraph with all vertices of even degree)
  circuit := a circuit in G beginning at an arbitrarily chosen vertex with edges successively added to form a path that returns to this vertex.
  $H := G$ with the edges of this circuit removed
  while H has edges
      subciruit := a circuit in H beginning at a vertex in H that also is an endpoint of an edge in circuit.
      $H := H$ with edges of *subciruit* and all isolated vertices removed
      circuit := *circuit* with *subcircuit* inserted at the appropriate vertex.
  return *circuit*{*circuit* is an Euler circuit}

Necessary and Sufficient Conditions

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

Necessary and Sufficient Conditions

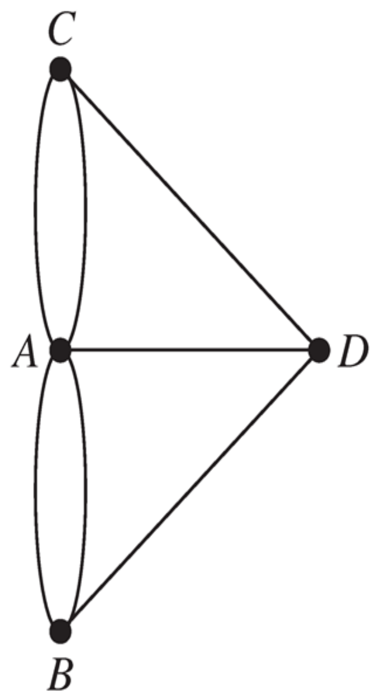
- **Theorem** A connected multigraph with at least two vertices has an *Euler circuit* if and only if each of its vertices has **even** degree.

  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.

Necessary and Sufficient Conditions

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

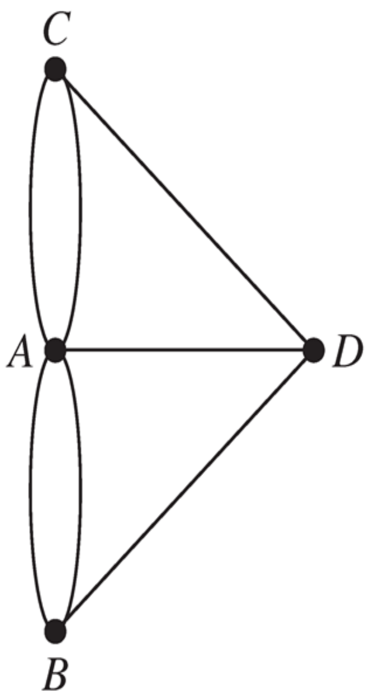
  Theorem A connected multigraph has an *Euler path* but not an *Euler circuit* if and only if it has exactly two vertices of odd degree.



Necessary and Sufficient Conditions

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

  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.



No Euler circuit

Euler Circuits and Paths

- **Example**

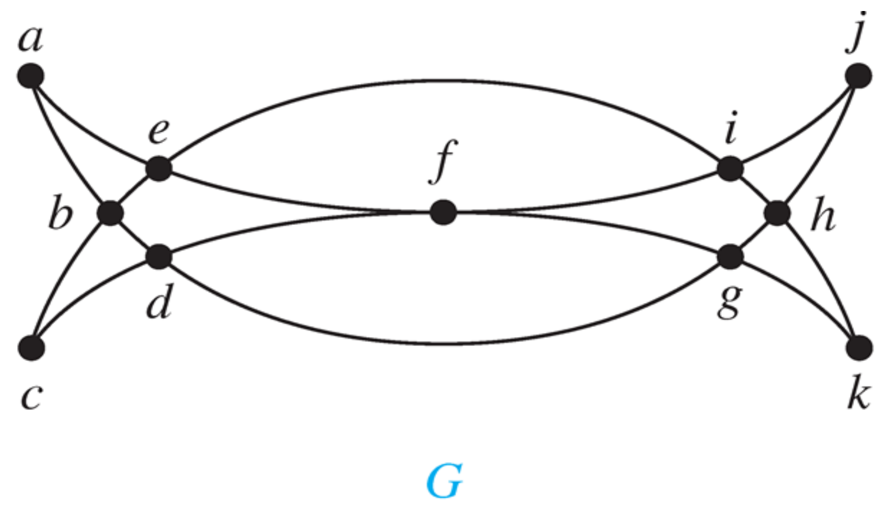
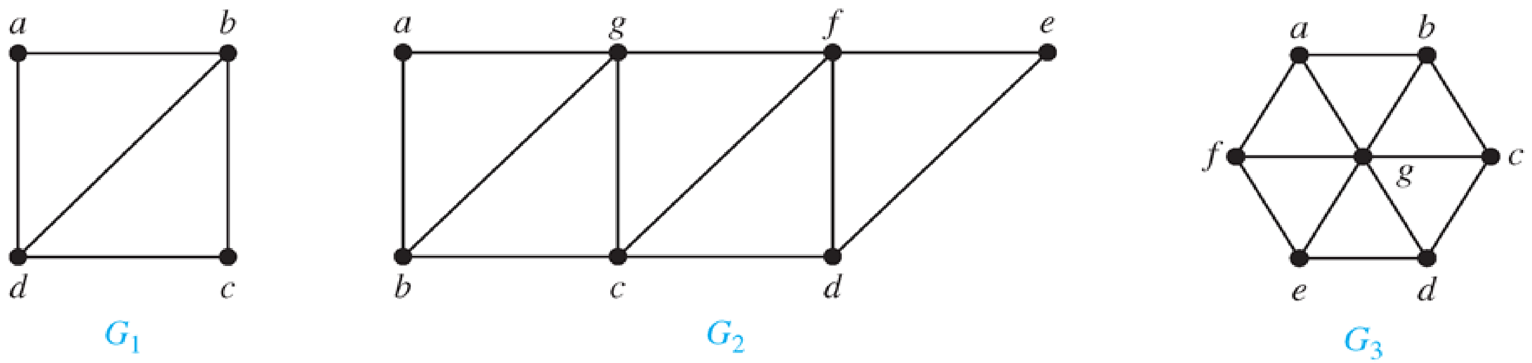


FIGURE 6 **Mohammed's Scimitars.**

- **Example**



Applications of Euler Paths and Circuits

- Finding a path or circuit that traverses each
  - ◇ street in a neighborhood
  - ◇ road in a transportation network
  - ◇ link in a communication network
  - ◇ ...

Applications of Euler Paths and Circuits

- Finding a path or circuit that traverses each
  - ◇ street in a neighborhood
  - ◇ road in a transportation network
  - ◇ link in a communication network
  - ◇ ...

Chinese Postman Problem Meigu Guan [60']

Applications of Euler Paths and Circuits

- Finding a path or circuit that traverses each
  - street in a neighborhood
  - road in a transportation network
  - link in a communication network
  - ...

Chinese Postman Problem Meigu Guan [60']

Given a graph $G = (V, E)$, for every $e \in E$, there is a nonnegative weight $w(e)$. Find a circuit W such that

$$\sum_{e \in W} w(e) = \min$$

Applications of Euler Paths and Circuits

- Finding a path or circuit that traverses each
  - ◇ street in a neighborhood
  - ◇ road in a transportation network
  - ◇ link in a communication network
  - ◇ ...

Chinese Postman Problem Meigu Guan [60']

Given a graph $G = (V, E)$, for every $e \in E$, there is a nonnegative weight $w(e)$. Find a circuit W such that

$$\sum_{e \in W} w(e) = \min$$

k-Postman Chinese Postman Problem (k-PCPP)

Applications of Euler Paths and Circuits

- Finding a path or circuit that traverses each
  - street in a neighborhood
  - road in a transportation network
  - link in a communication network
  - ...

Chinese Postman Problem Meigu Guan [60']

Given a graph $G = (V, E)$, for every $e \in E$, there is a nonnegative weight $w(e)$. Find a circuit W such that

$$\sum_{e \in W} w(e) = \min$$

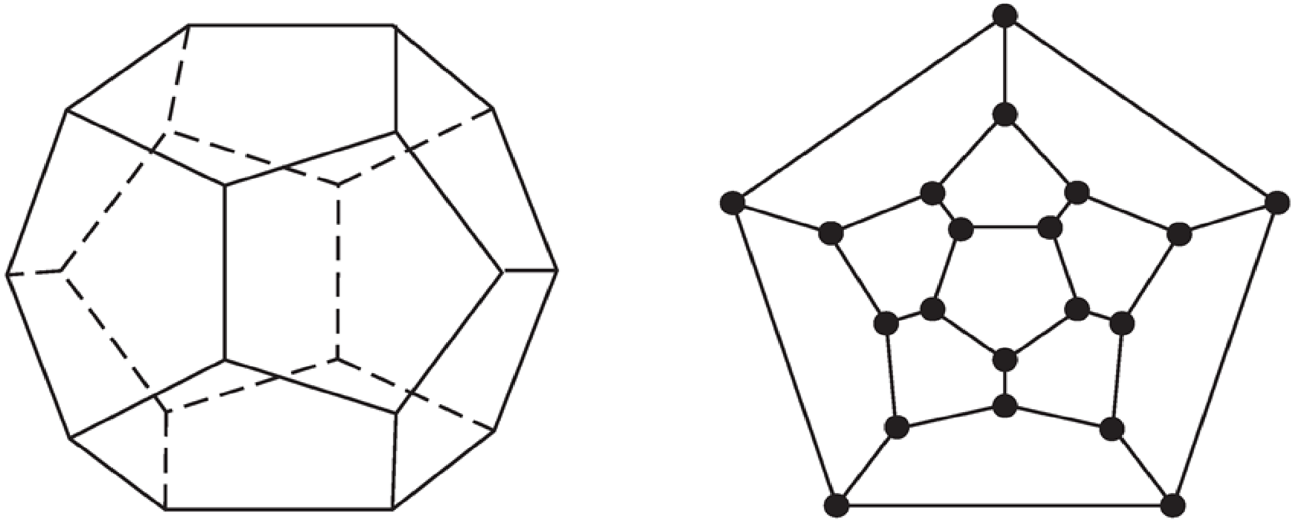
k-Postman Chinese Postman Problem (k-PCPP)
$\in$ NPC

Hamilton Paths and Circuits

- Euler paths and circuits contained every edge only once. What about containing every vertex exactly once?

Hamilton Paths and Circuits

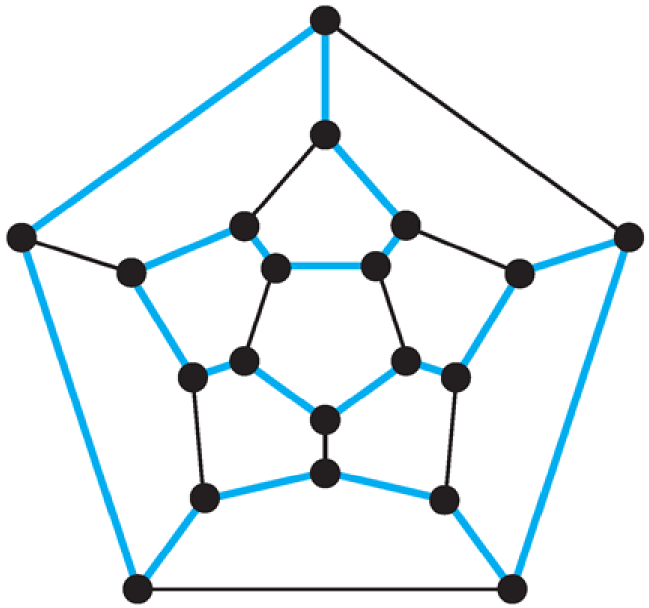
- Euler paths and circuits contained every edge only once. What about containing every vertex exactly once?



(a) (b)

Hamilton Paths and Circuits

- Euler paths and circuits contained every edge only once. What about containing every vertex exactly once?



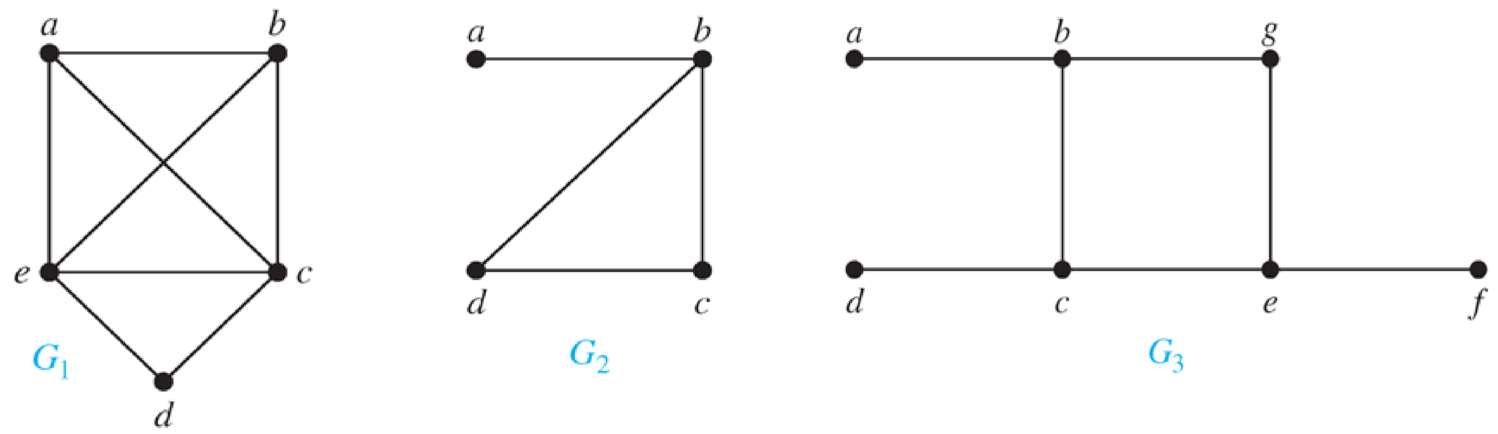
Hamilton Paths and Circuits

- **Definition**: 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*.

Hamilton Paths and Circuits

- **Definition**: 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*.

Example Which of these simple graphs has a Hamilton circuit or, if not, a Hamilton path?



Sufficient Conditions for Hamilton Circuits

- No simple necessary and sufficient conditions are known for the existence of a Hamilton circuit.

Sufficient Conditions for Hamilton Circuits

- No simple necessary and sufficient conditions are known for the existence of a Hamilton circuit.

  But, there are some useful sufficient conditions.

Sufficient Conditions for Hamilton Circuits

- No simple necessary and sufficient conditions are known for the existence of a Hamilton circuit.

  But, there are some useful sufficient conditions.

  Dirac's Theorem If G is a simple graph with $n \geq 3$ vertices such that the degree of every vertex in G is $\geq n/2$, then G has a Hamilton circuit.

Sufficient Conditions for Hamilton Circuits

- No simple necessary and sufficient conditions are known for the existence of a Hamilton circuit.

  But, there are some useful sufficient conditions.

  Dirac's Theorem If G is a simple graph with $n \geq 3$ vertices such that the degree of every vertex in G is $\geq n/2$, then G has a Hamilton circuit.

  Ore's Theorem If G is a simple graph with $n \geq 3$ vertices such that $\deg(u) + \deg(v) \geq n$ for every pair of nonadjacent vertices, then G has a Hamilton circuit.

Sufficient Conditions for Hamilton Circuits

- No simple necessary and sufficient conditions are known for the existence of a Hamilton circuit.

  But, there are some useful sufficient conditions.

  Dirac's Theorem If G is a simple graph with $n \geq 3$ vertices such that the degree of every vertex in G is $\geq n/2$, then G has a Hamilton circuit.

  Ore's Theorem If G is a simple graph with $n \geq 3$ vertices such that $\deg(u) + \deg(v) \geq n$ for every pair of nonadjacent vertices, then G has a Hamilton circuit.

  Hamilton path problem $\in$ NPC

Applications of Hamilton Paths and Circuits

- A path or a circuit that visits each city, or each node in a communication network exactly once, can be solved by finding a Hamilton path.

Applications of Hamilton Paths and Circuits

- A path or a circuit that visits each city, or each node in a communication network exactly once, can be solved by finding a Hamilton path.

  Traveling Salesperson Problem (TSP) asks for the shortest route a traveling salesperson should take to visit a set of cities.

Applications of Hamilton Paths and Circuits

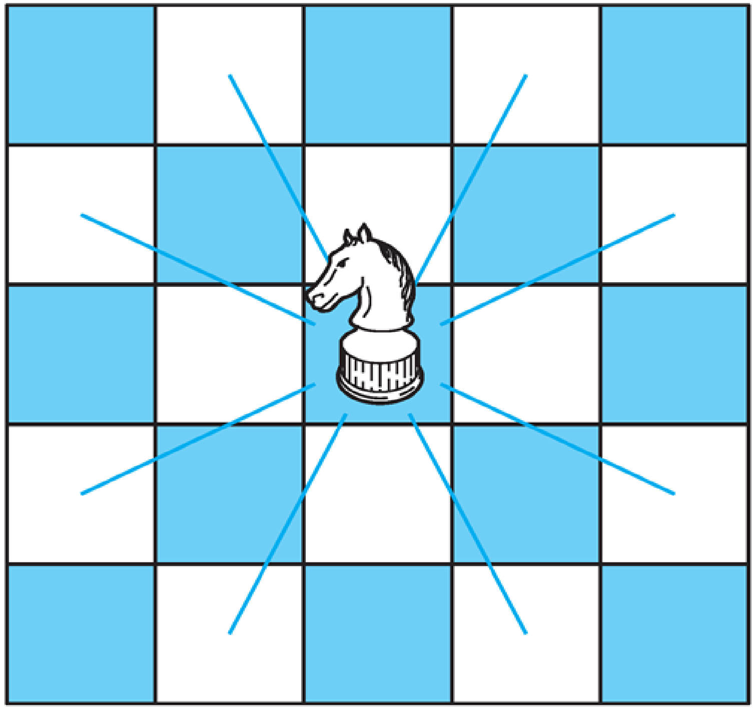
- A path or a circuit that visits each city, or each node in a communication network exactly once, can be solved by finding a Hamilton path.

  Traveling Salesperson Problem (TSP) asks for the shortest route a traveling salesperson should take to visit a set of cities.

  the decision version of the TSP $\in$ NPC

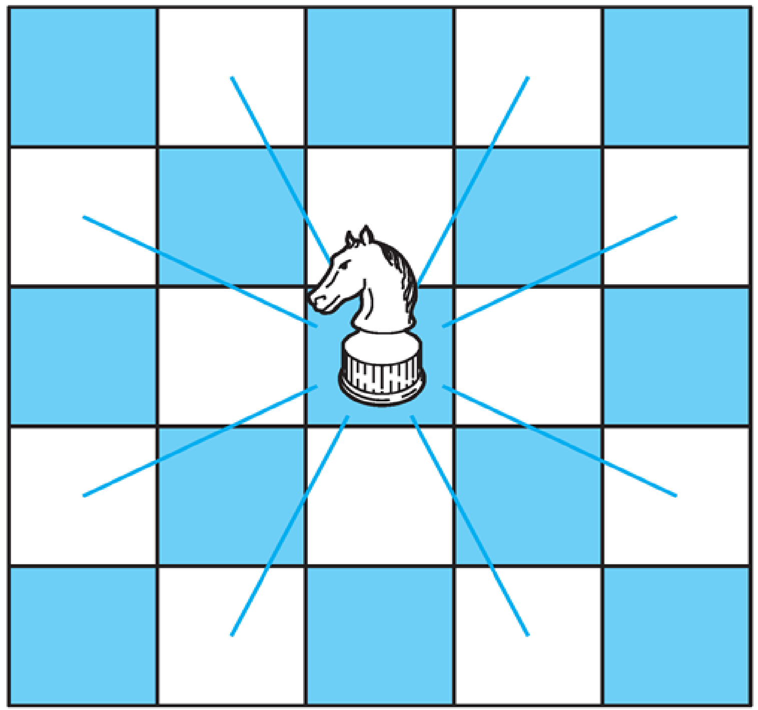
Applications of Hamilton Paths and Circuits

- Can we traverse every space (and come back) in the 5×5 chessboard?



Applications of Hamilton Paths and Circuits

- Can we traverse every space (and come back) in the 5×5 chessboard?



What about in 6×6 chessboard?

Next Lecture

- Graph theory III ...