COMP170
Discrete Mathematical Tools for Computer Science

Intro to Graphs

Version 2.0: Last updated, May 13, 2007

Discrete Math for Computer Science
K. Bogart, C. Stein and R.L. Drysdale
Section 6.1, pp. 309-320

Graphs

- Basic Definitions
- The Degree of a Vertex
- Connectivity
- Cycles
- Trees

Graphs

Graphs

Fundamental topic in discrete math and CS.

Graphs

Fundamental topic in discrete math and CS.

Important because it's used to model many common situations and to naturally describe many algorithms.

Graphs

Fundamental topic in discrete math and CS.

Important because it's used to model many common situations and to naturally describe many algorithms.

Example

Graphs

Fundamental topic in discrete math and CS.

Important because it's used to model many common situations and to naturally describe many algorithms.

Example

Map of some cities in eastern US.

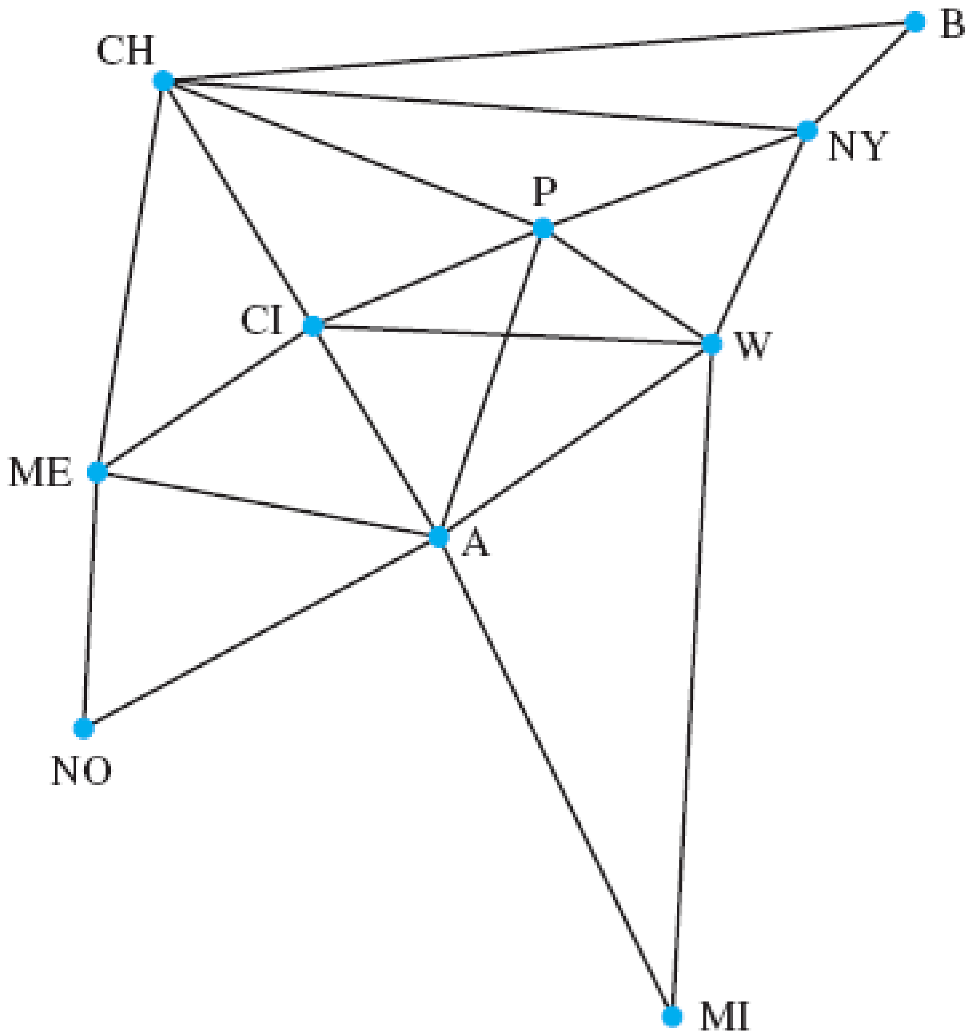
Graphs

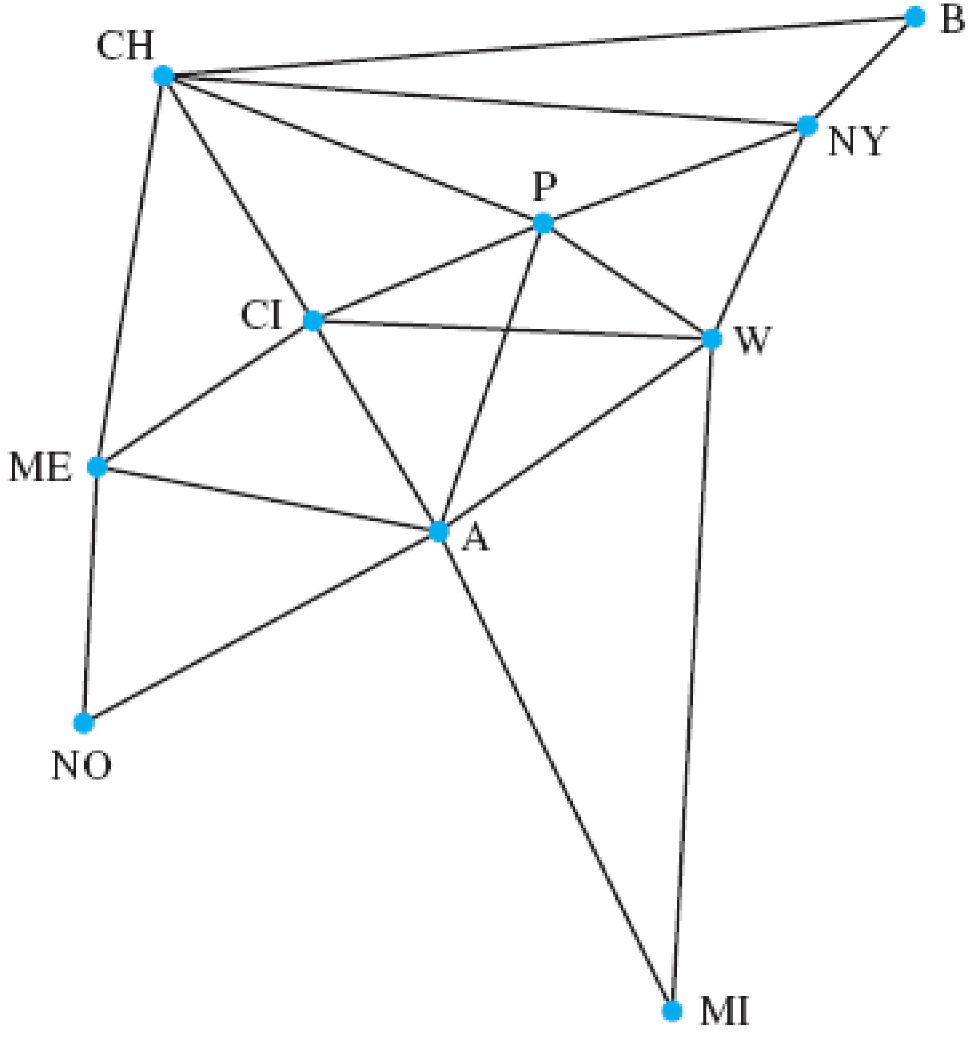
Fundamental topic in discrete math and CS.

Important because it's used to model many common situations and to naturally describe many algorithms.

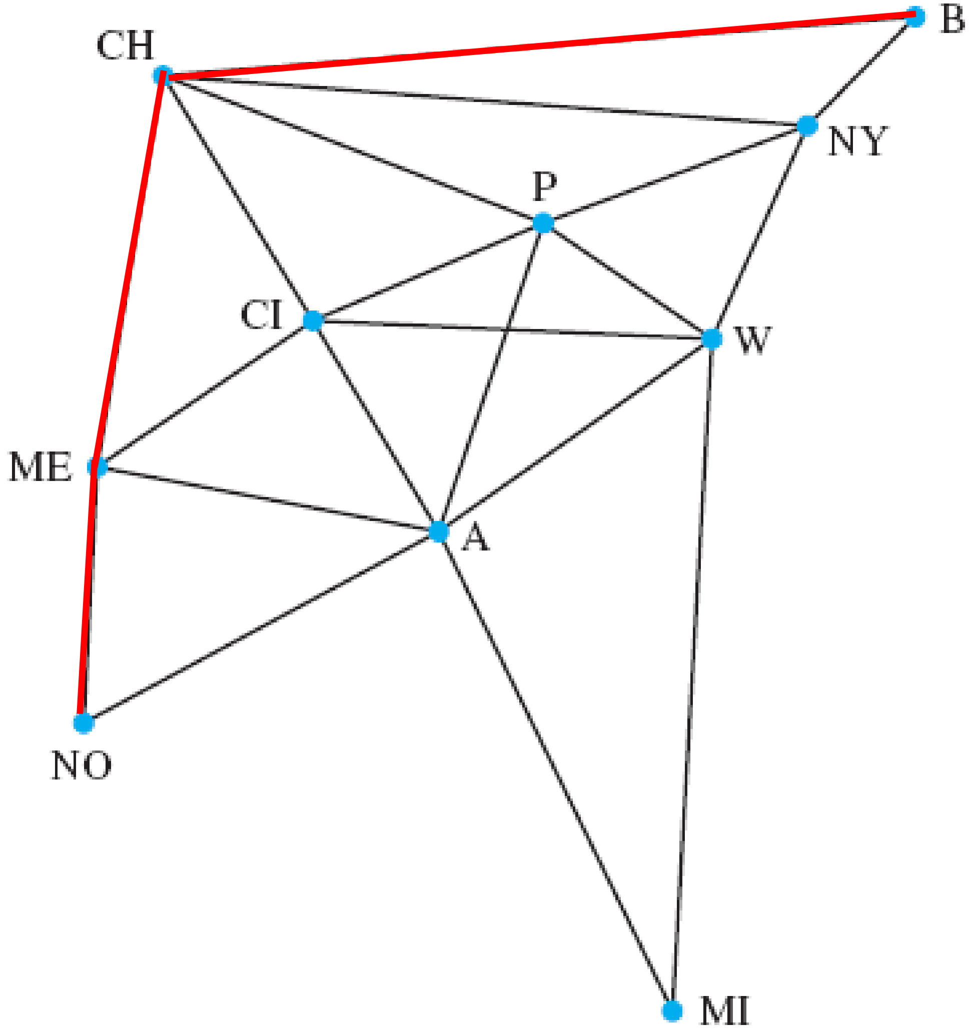
Example

Map of some cities in eastern US.
with communication lines existing
between certain pairs of these cities.

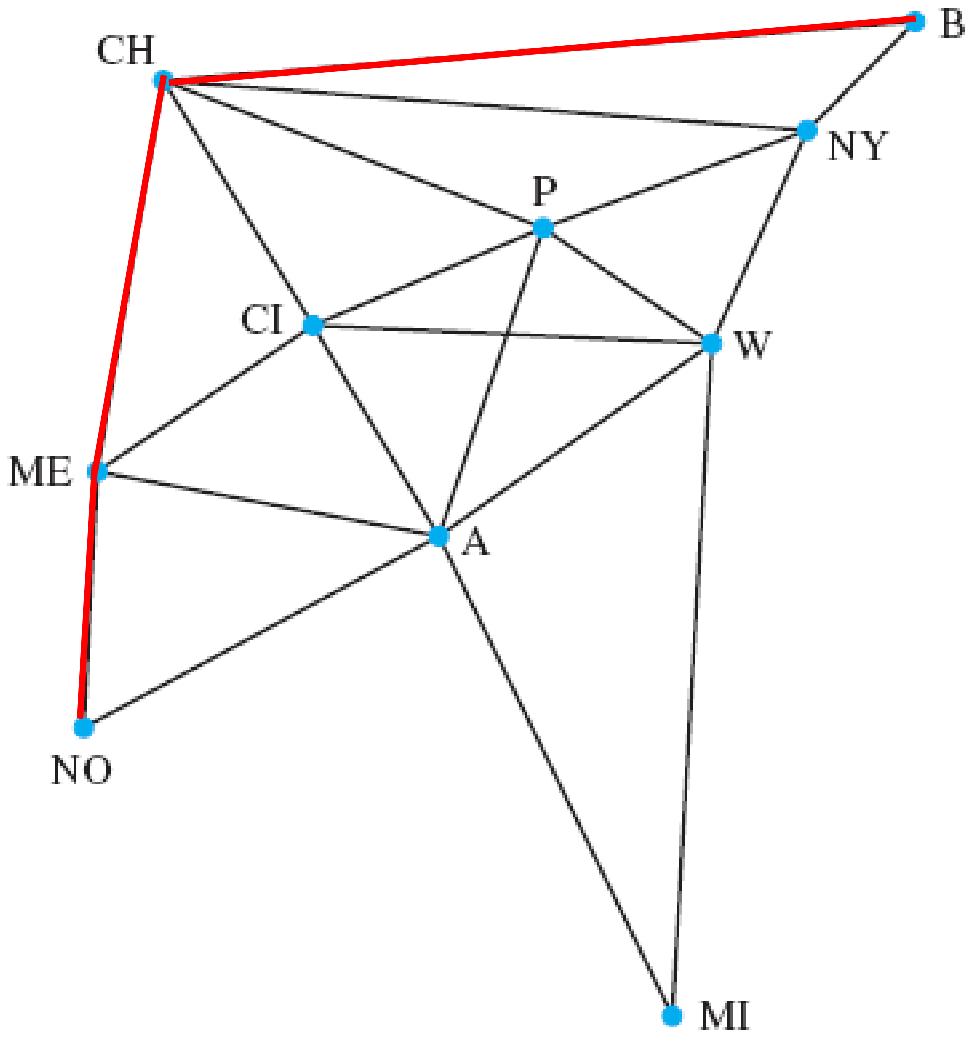

B
CH
NY
P
CI
W
ME
A
NO
MI



What is the minimum number of links needed to send a message from B to NO?

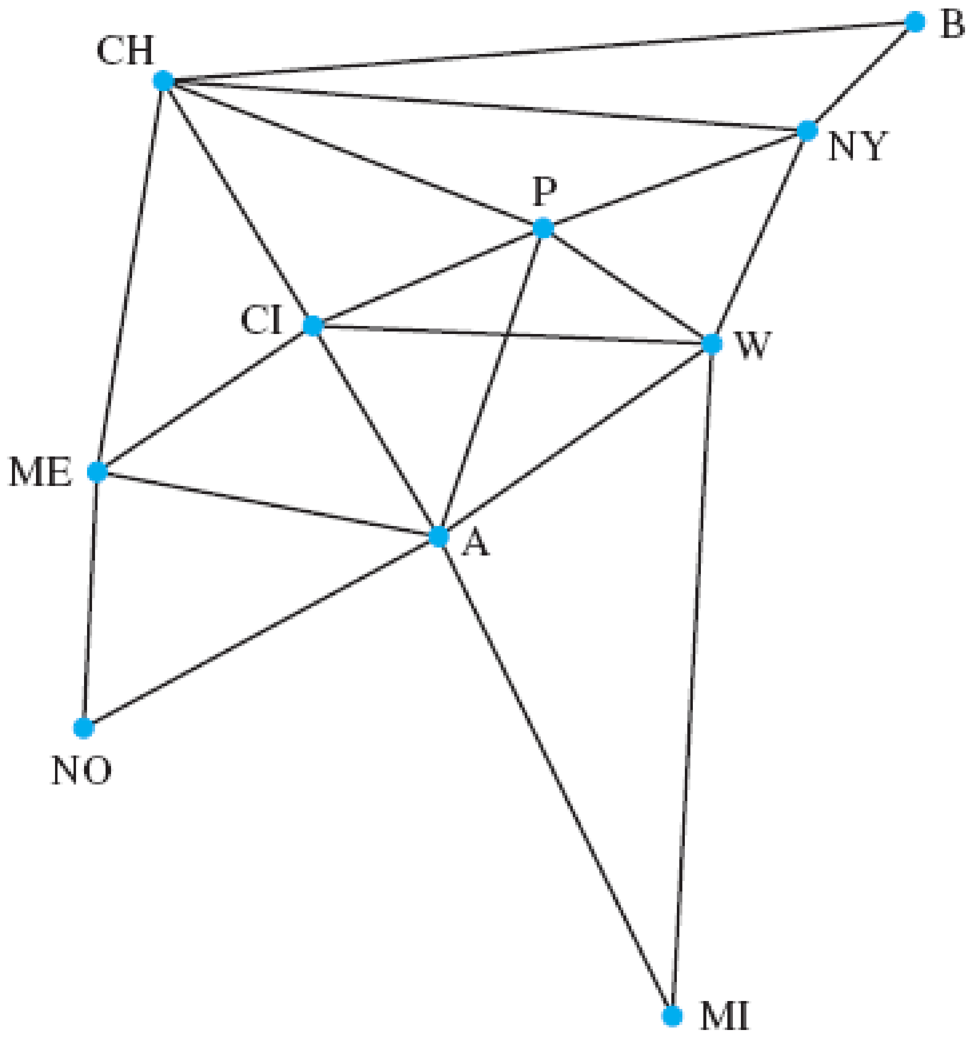


What is the minimum number of links needed to send a message from B to NO?



What is the minimum number of links needed to send a message from B to NO?

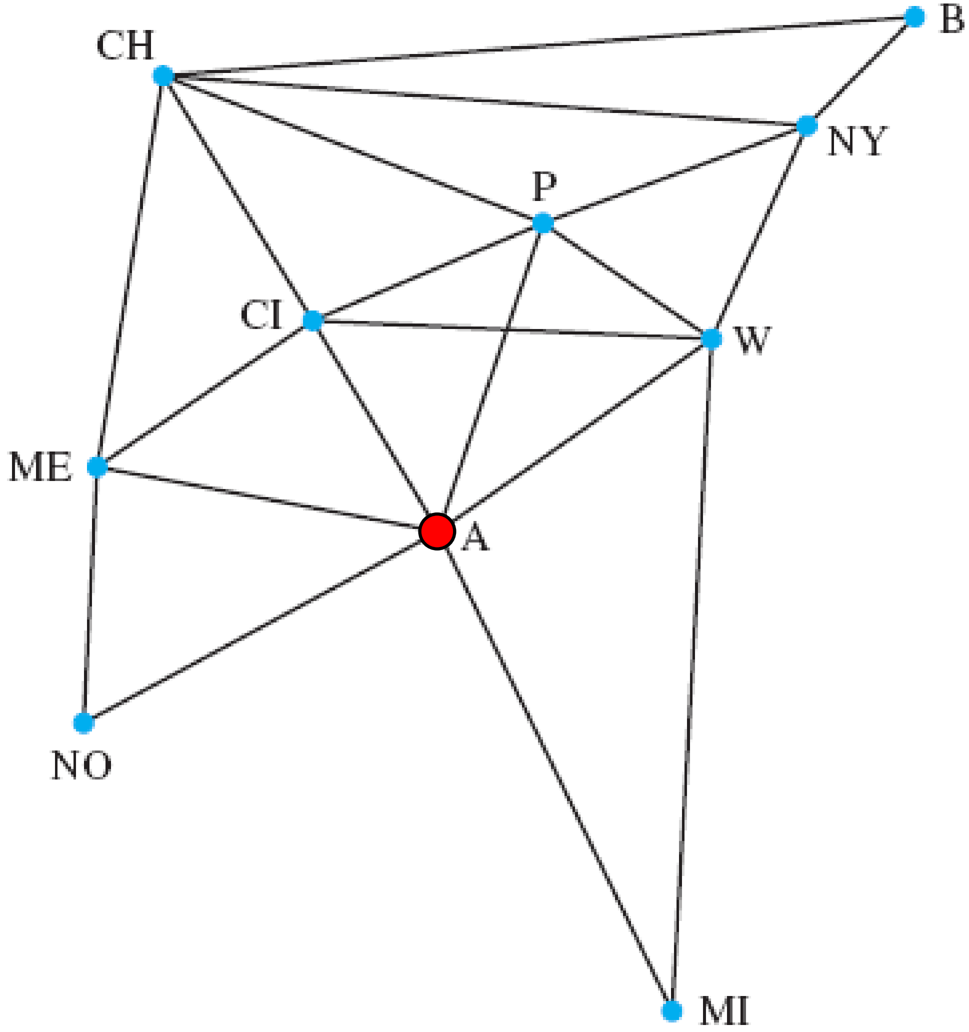
3: B – CH – ME – NO.



What is the minimum number of links needed to send a message from B to NO?

3: B – CH – ME – NO.

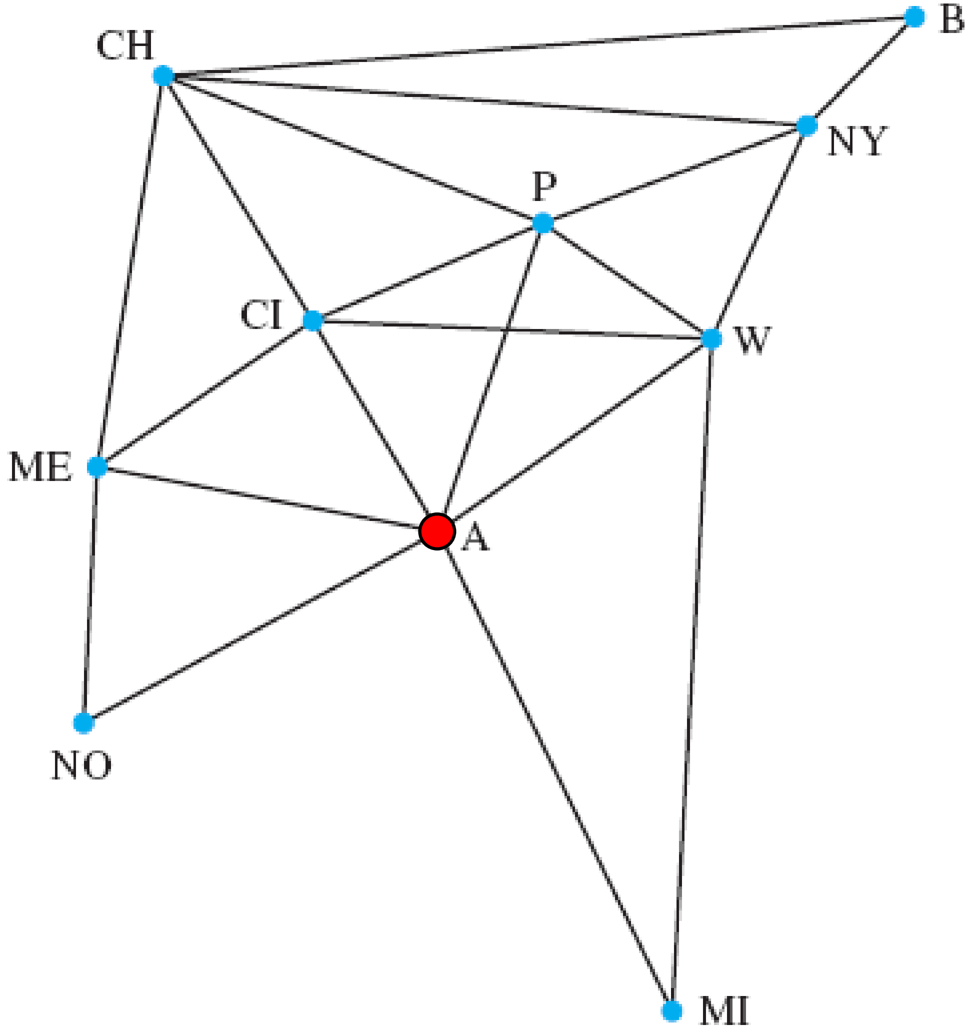
Which city/cities has/have the most communication links emanating from it/them?



What is the minimum number of links needed to send a message from B to NO?

3: B – CH – ME – NO.

Which city/cities has/have the most communication links emanating from it/them?

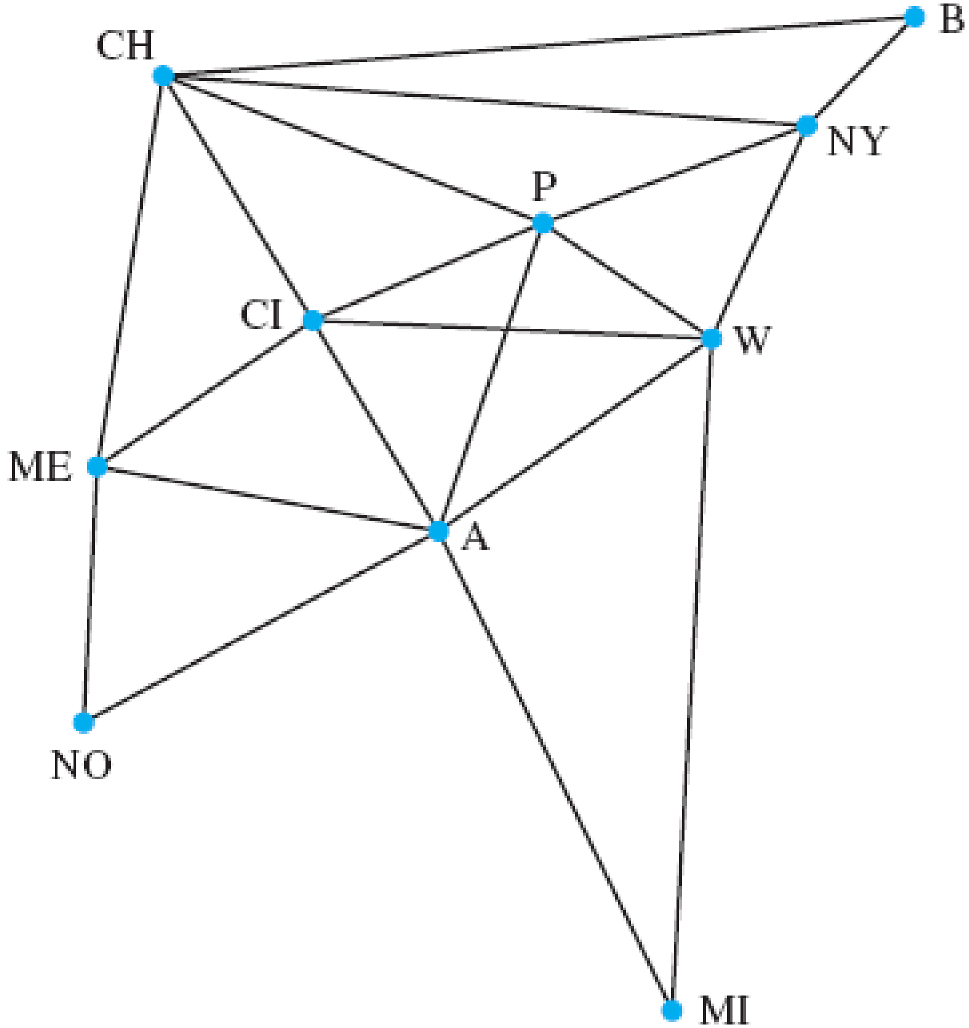


What is the minimum number of links needed to send a message from B to NO?

3: B – CH – ME – NO.

Which city/cities has/have the most communication links emanating from it/them?

A: 6 links.



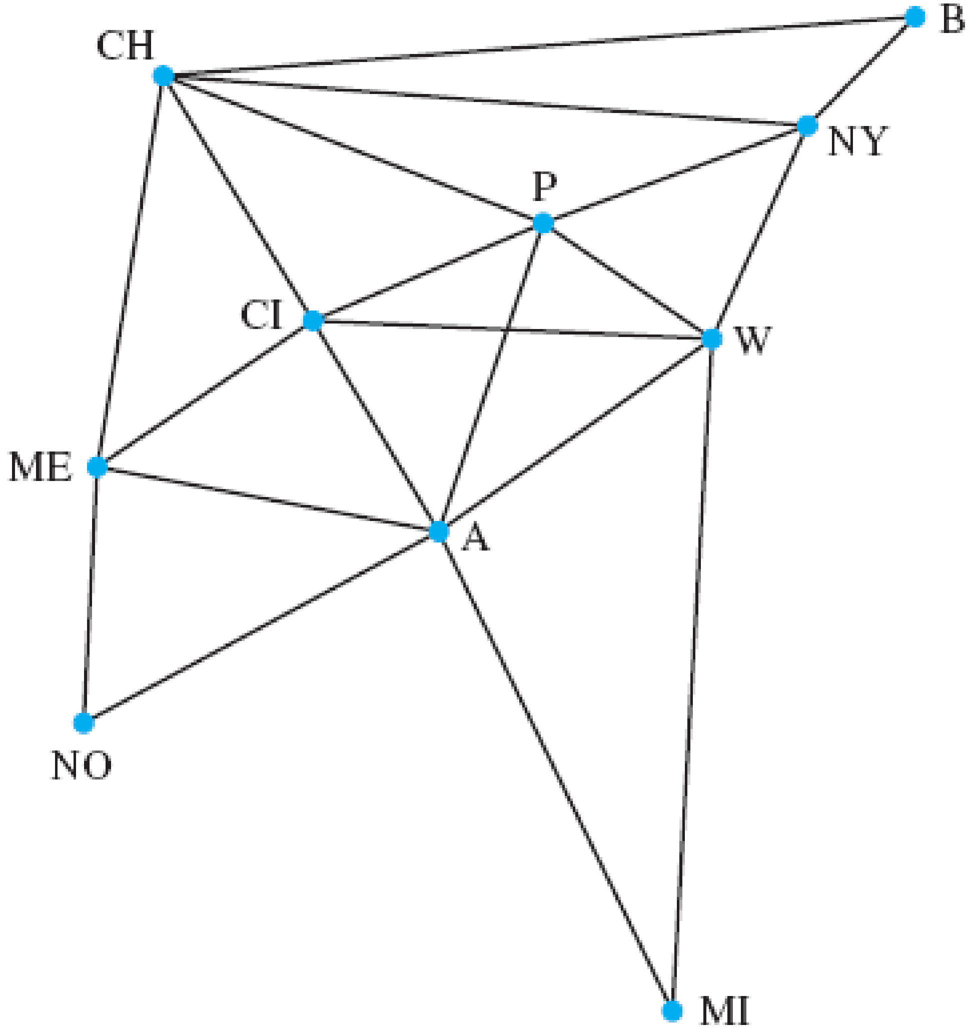
What is the minimum number of links needed to send a message from B to NO?

3: B – CH – ME – NO.

Which city/cities has/have the most communication links emanating from it/them?

A: 6 links.

What is the total number of communication links?



What is the minimum number of links needed to send a message from B to NO?

3: B – CH – ME – NO.

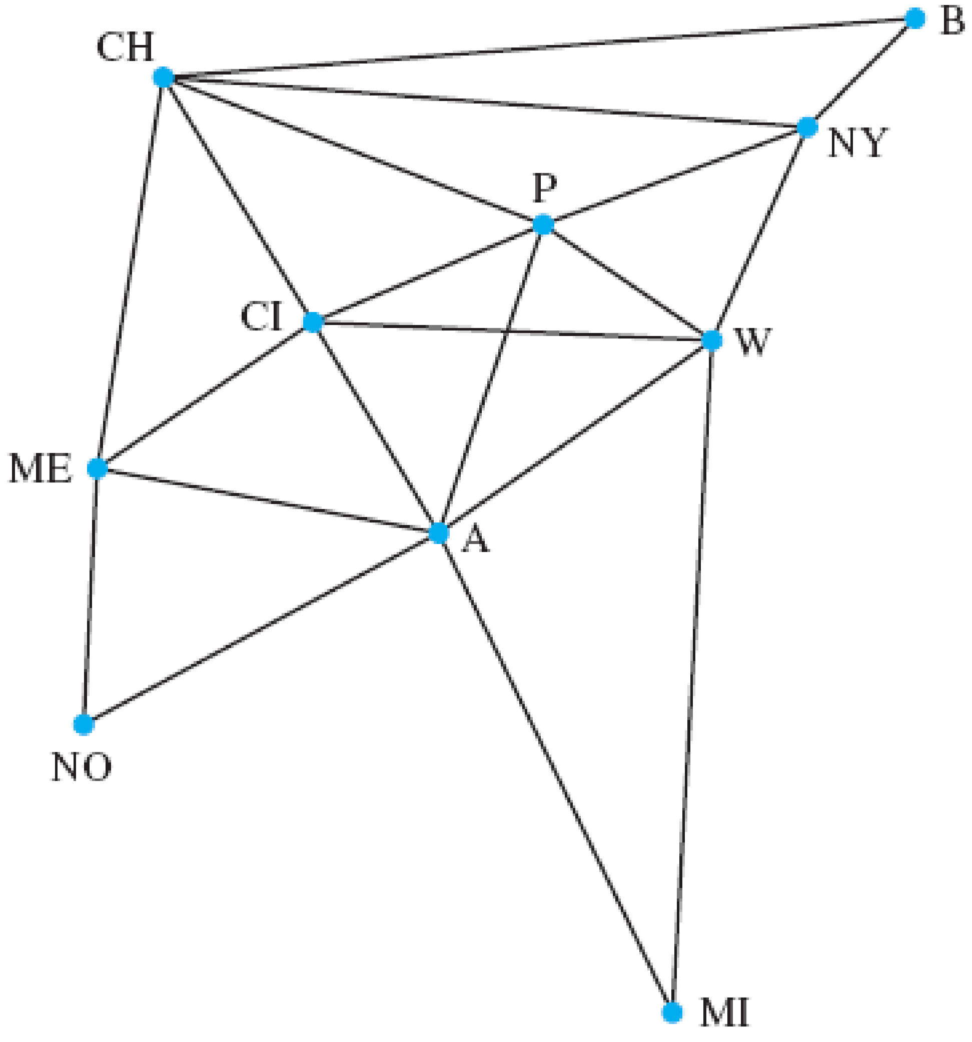
Which city/cities has/have the most communication links emanating from it/them?

A: 6 links.

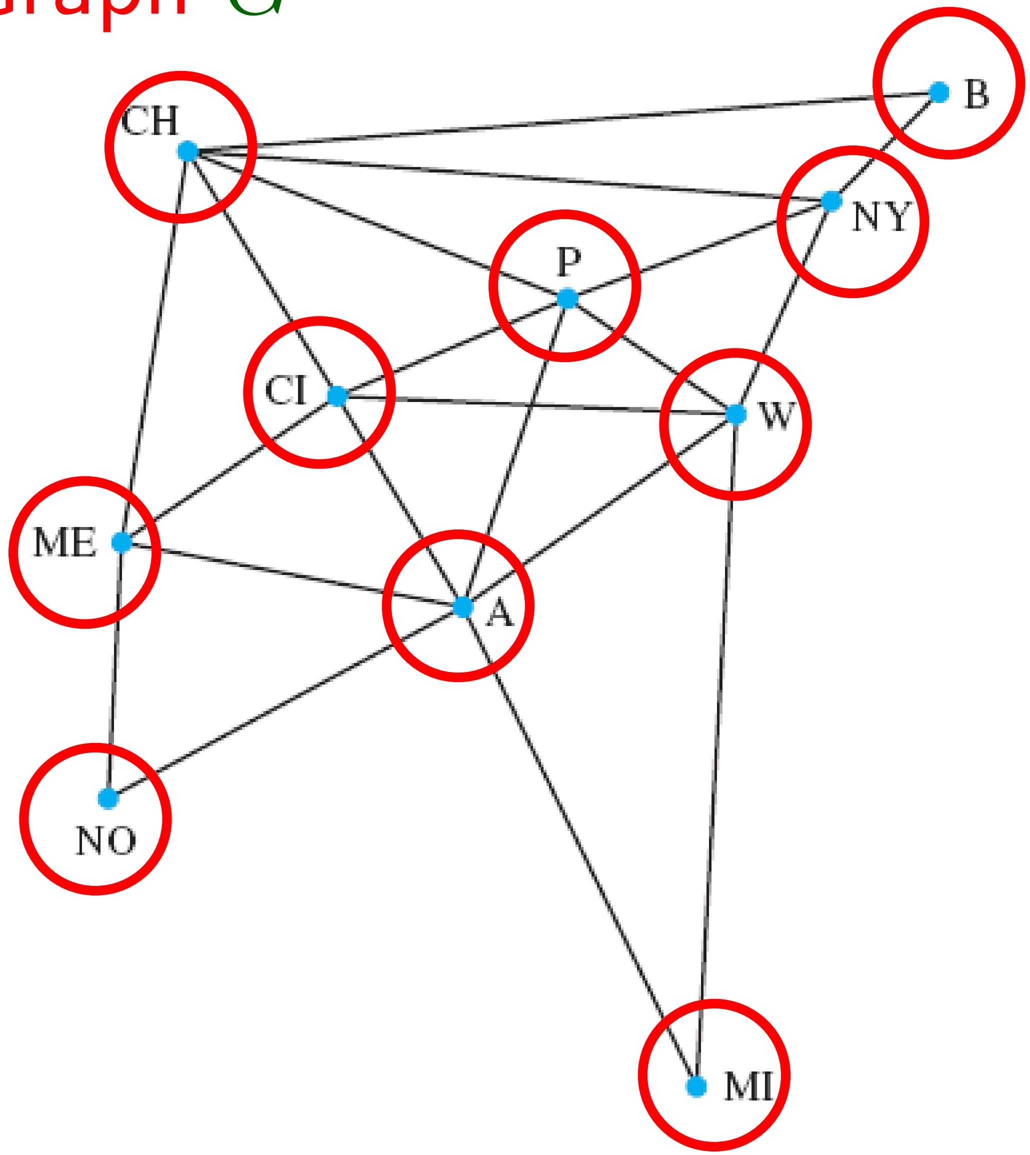
What is the total number of communication links?

20 links.

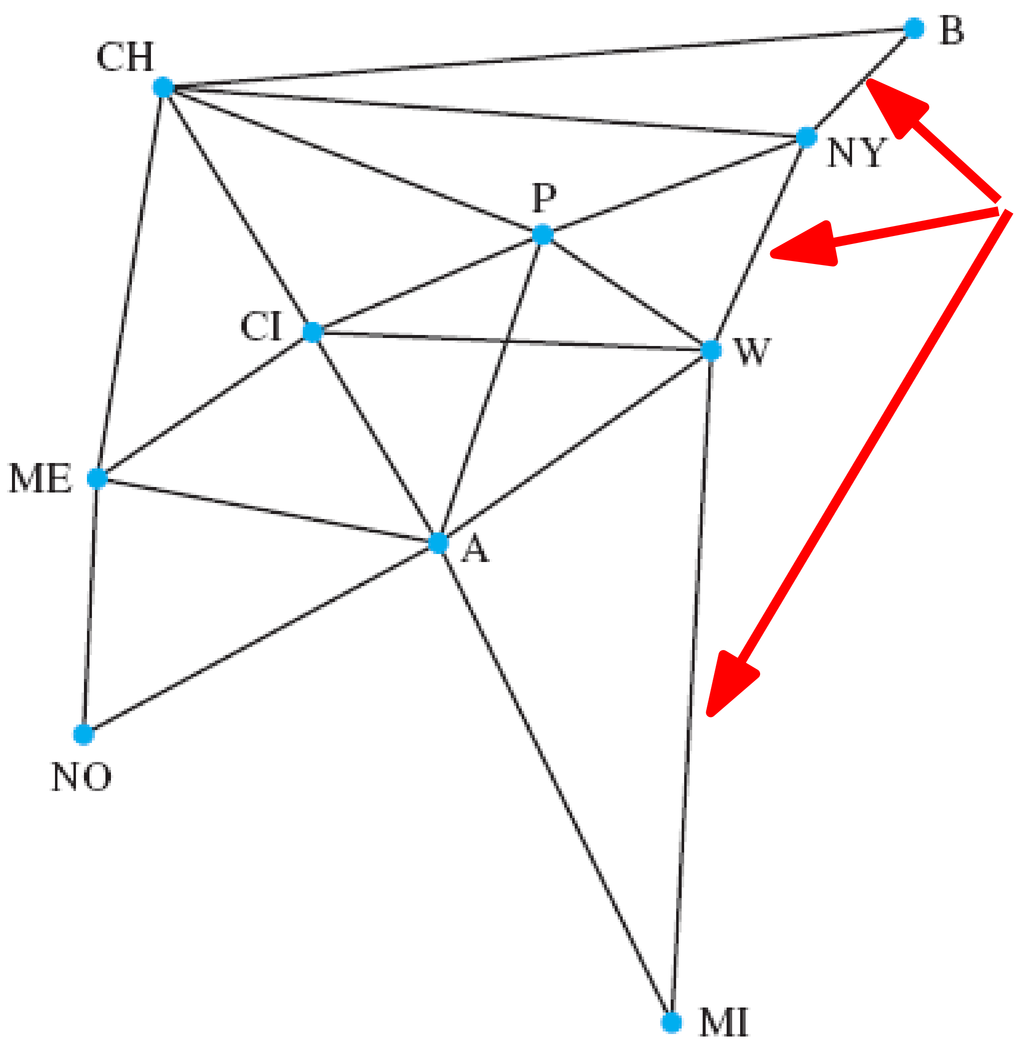
Graph G



Graph G

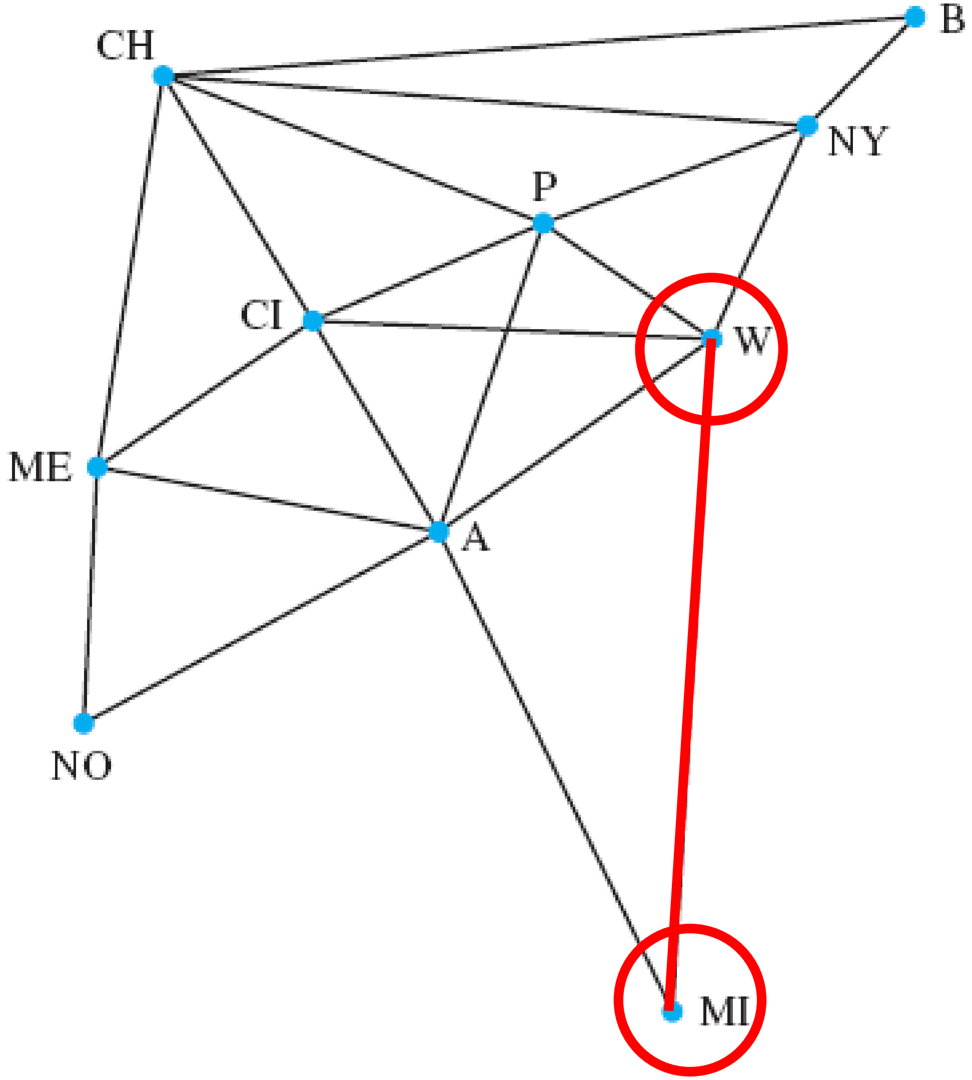


consists of a set of **vertices** V, $|V| = n$, Graph G



consists of a set of **vertices** V,
$|V| = n$,
and a set of **edges** E,
$|E| = m$.

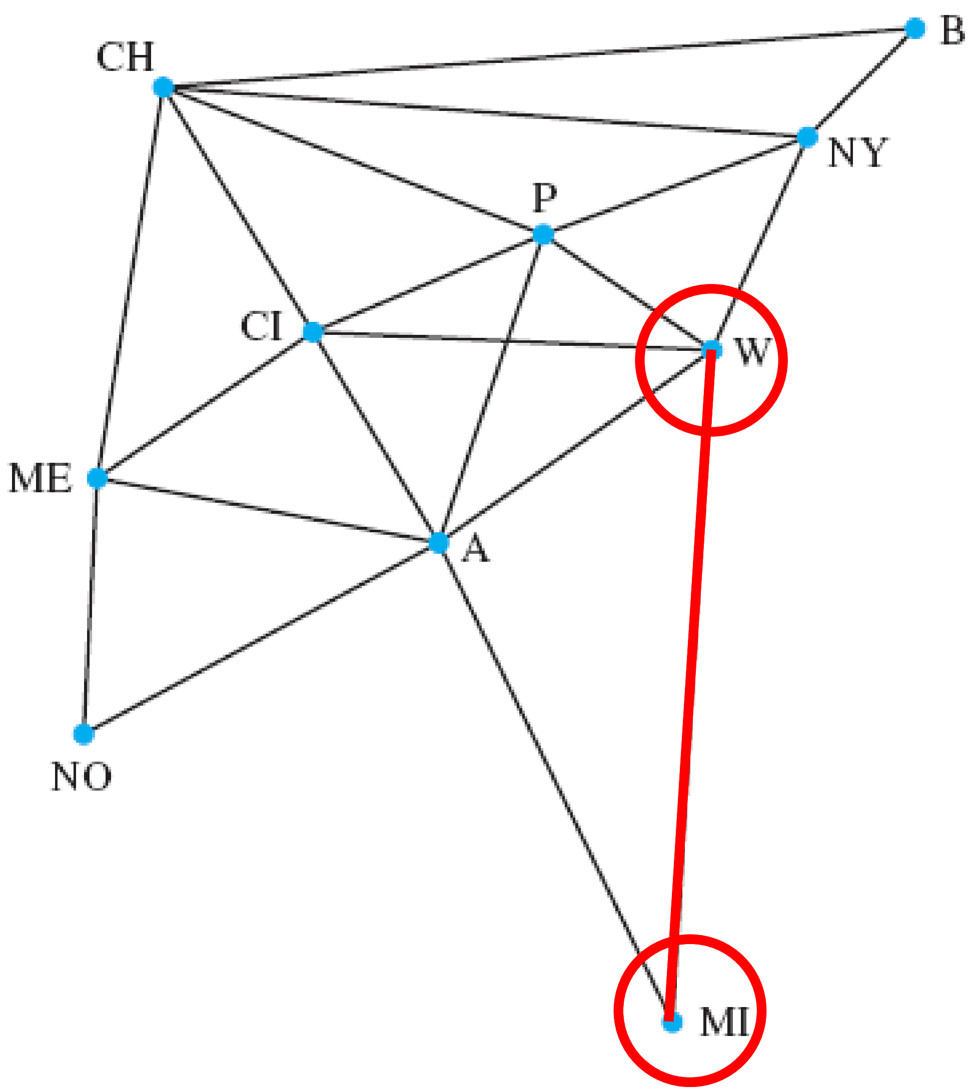
Graph G



consists of a set of **vertices** V,
$|V| = n$,
and a set of **edges** E,
$|E| = m$.

Each edge has two **endpoints**.

Graph G



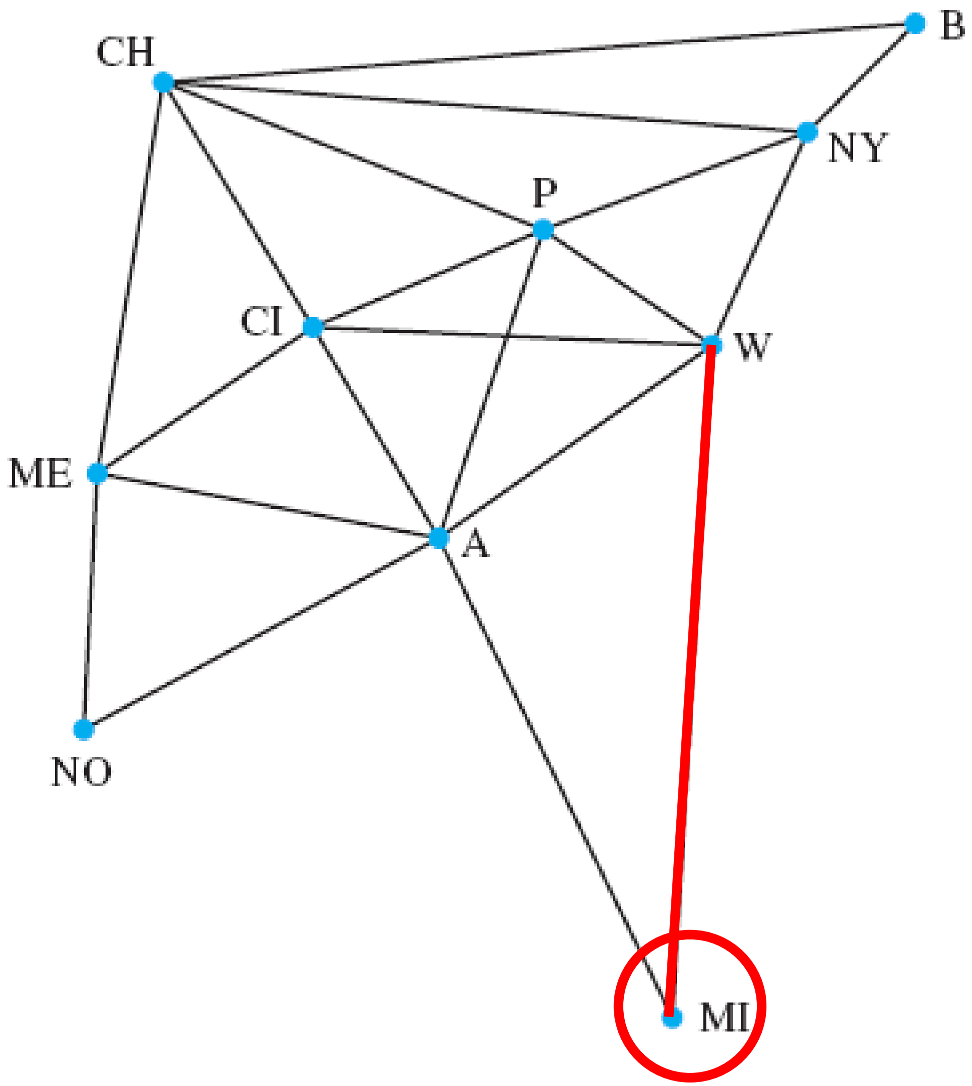
consists of a set of **vertices** V, $|V| = n$,

and a set of **edges** E, $|E| = m$.

Each edge has two **endpoints**.

An edge **joins** its endpoints, two endpoints are **adjacent** if they are joined by an edge.

Graph G



consists of a set of **vertices** V, $|V| = n$,

and a set of **edges** E, $|E| = m$.

Each edge has two **endpoints**.

An edge **joins** its endpoints, two endpoints are **adjacent** if they are joined by an edge.

When a vertex is an endpoint of an edge, we say that the edge and the vertex are **incident** to each other.

More Examples:

More Examples:

- Vertices: biological species
  Edges: species have a common ancestor

More Examples:

- Vertices: biological species
  Edges: species have a common ancestor

- Vertices: people
  Edges: people attend same school

More Examples:

- Vertices: biological species
  Edges: species have a common ancestor

- Vertices: people
  Edges: people attend same school

- Vertices: MTR stations
  Edges: direct connection

More Examples:

- Vertices: biological species
  Edges: species have a common ancestor

- Vertices: people
  Edges: people attend same school

- Vertices: MTR stations
  Edges: direct connection

- Vertices: Web sites
  Edges: A link from one site to another

More Examples:

- Vertices: biological species
  Edges: species have a common ancestor

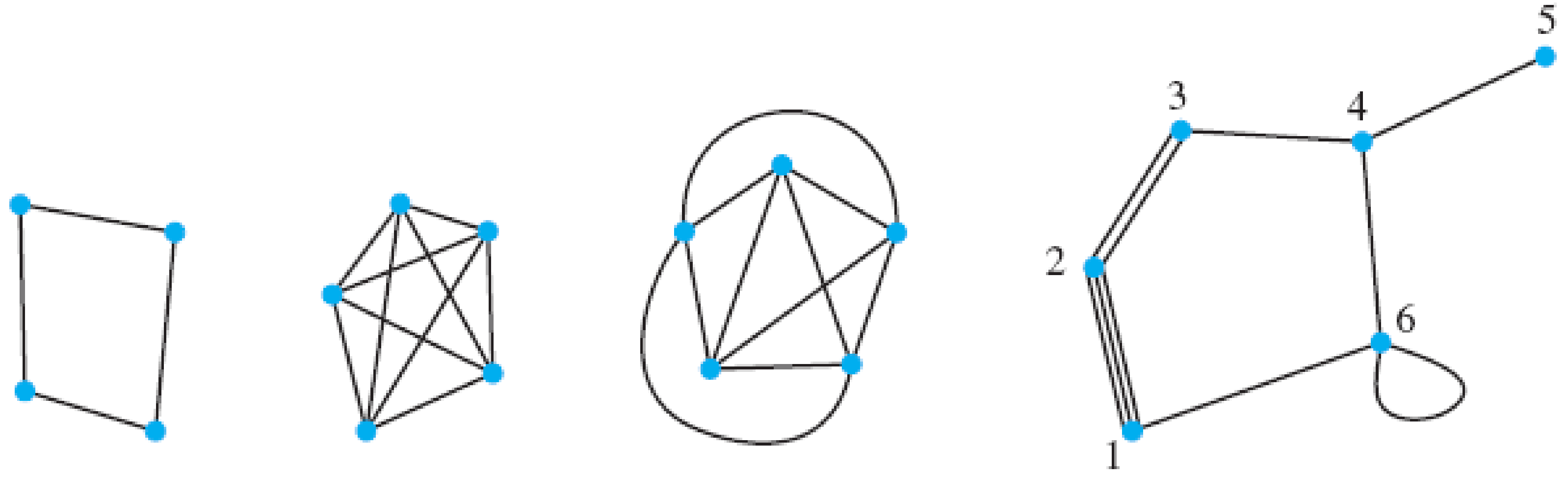
- Vertices: people
  Edges: people attend same school

- Vertices: MTR stations
  Edges: direct connection

- Vertices: Web sites
  Edges: A link from one site to another

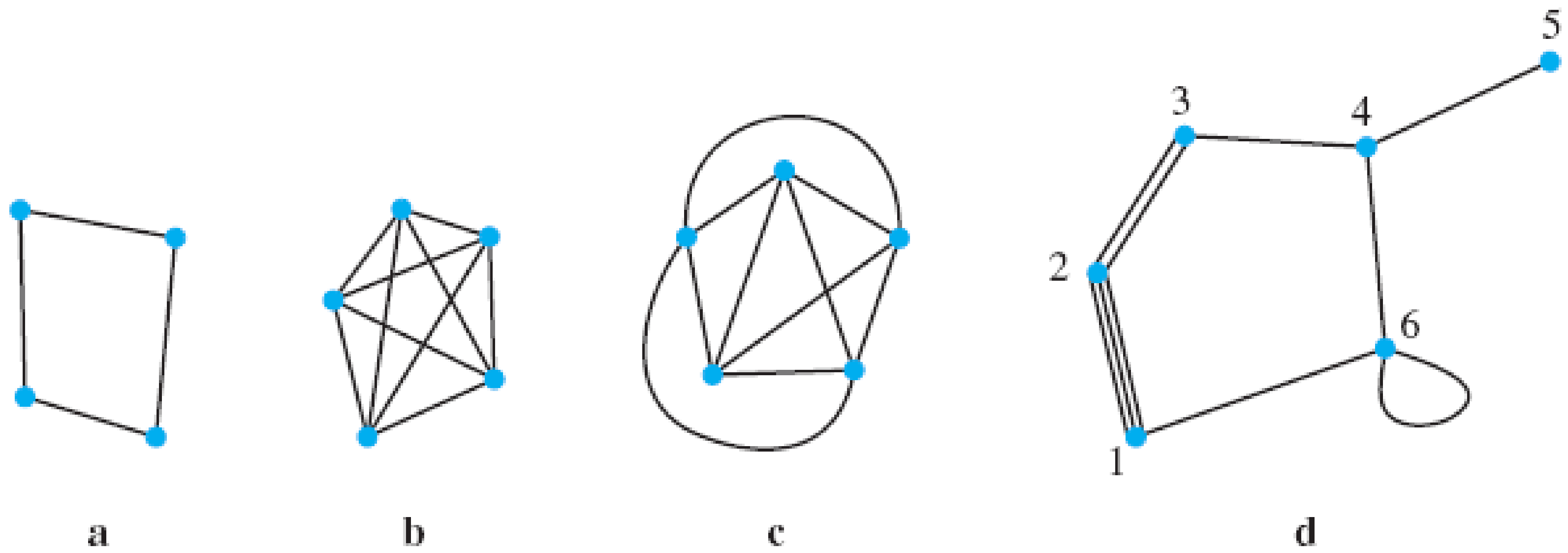
How Google models the Internet!

More Graphs:



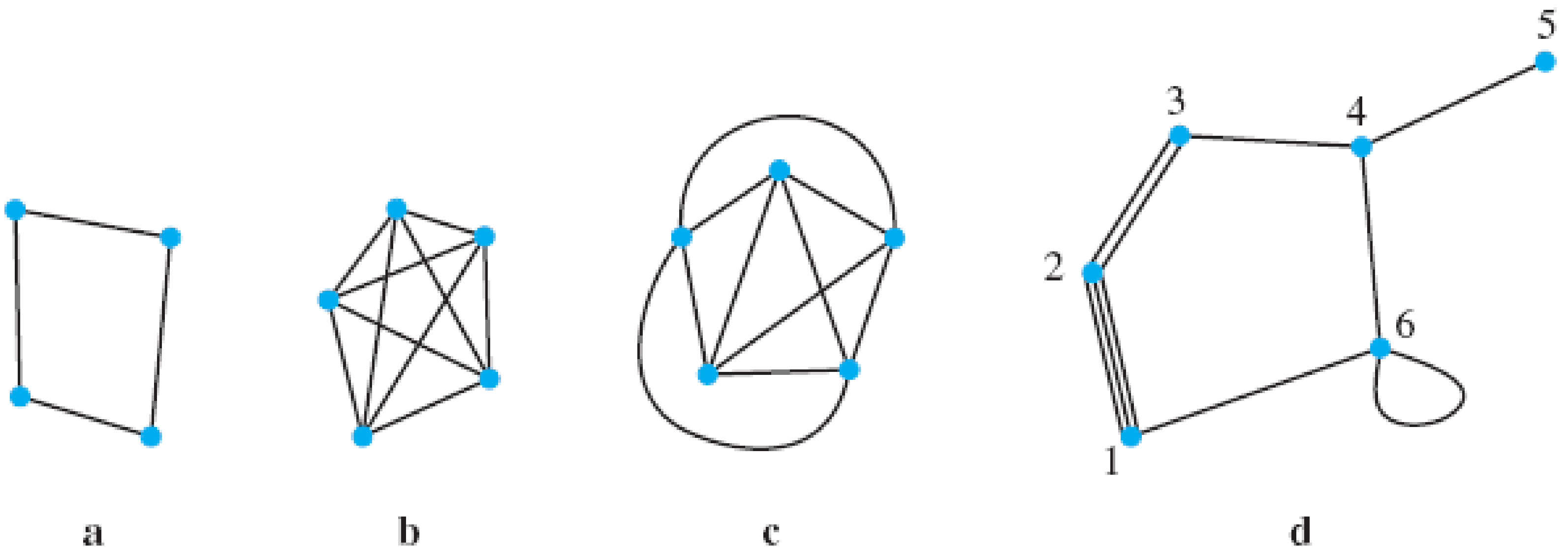
a b c d

More Graphs:



- **Simple Graph** (a, b, c): at most one edge joining each pair of distinct vertices (versus **multiple** edges (d)) and no edges joining a vertex to itself (= **loop**).

More Graphs:



- **Simple Graph** (a, b, c): at most one edge joining each pair of distinct vertices (versus **multiple** edges (d)) and no edges joining a vertex to itself ($=$ **loop**).
- **Complete Graph** K_n (b, c): graph with n vertices that has an edge between each pair of vertices.

A **path** in a graph is an alternating sequence of vertices and edges such that

A **path** in a graph is an alternating sequence of vertices and edges such that

- it starts and ends with a vertex,

A **path** in a graph is an alternating sequence of vertices and edges such that

- 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, and

A **path** in a graph is an alternating sequence of vertices and edges such that

- 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, and
- no vertex appears more than once in the sequence.

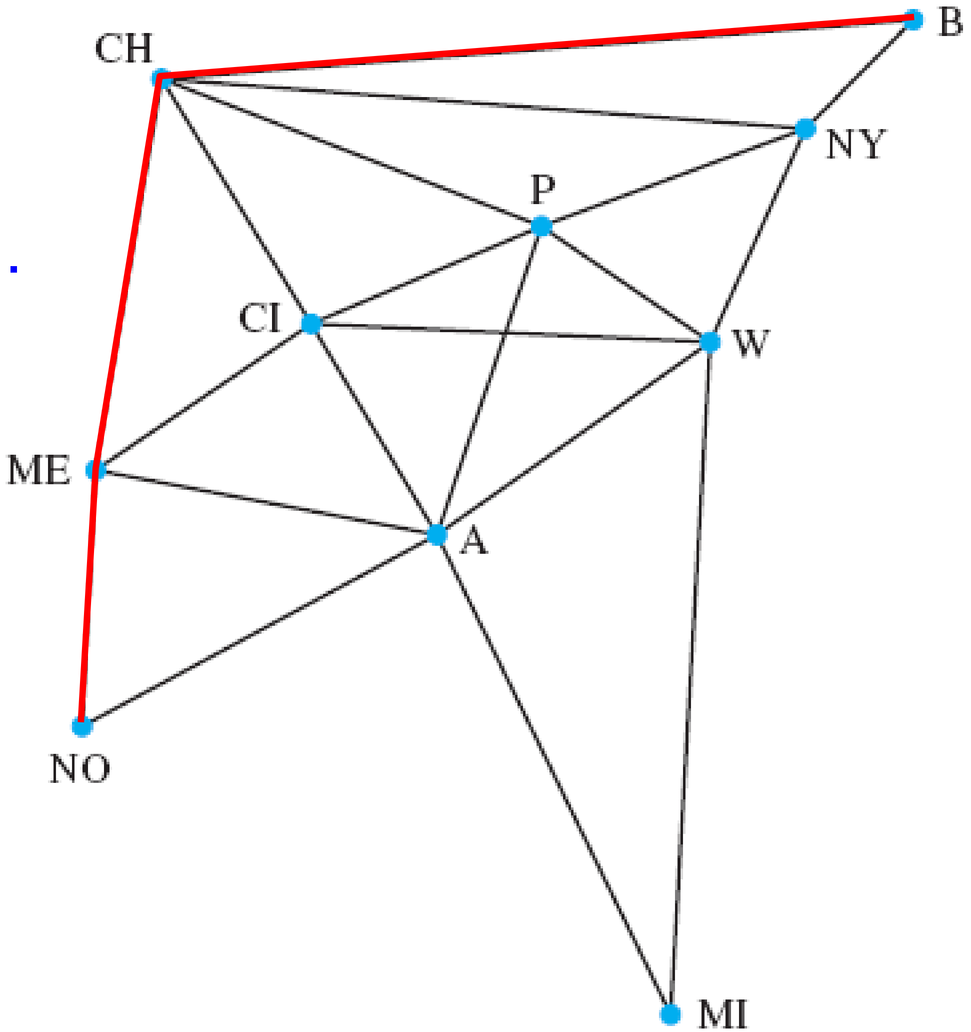
A **path** in a graph is an alternating sequence of vertices and edges such that

- 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, and
- no vertex appears more than once in the sequence.

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

Example

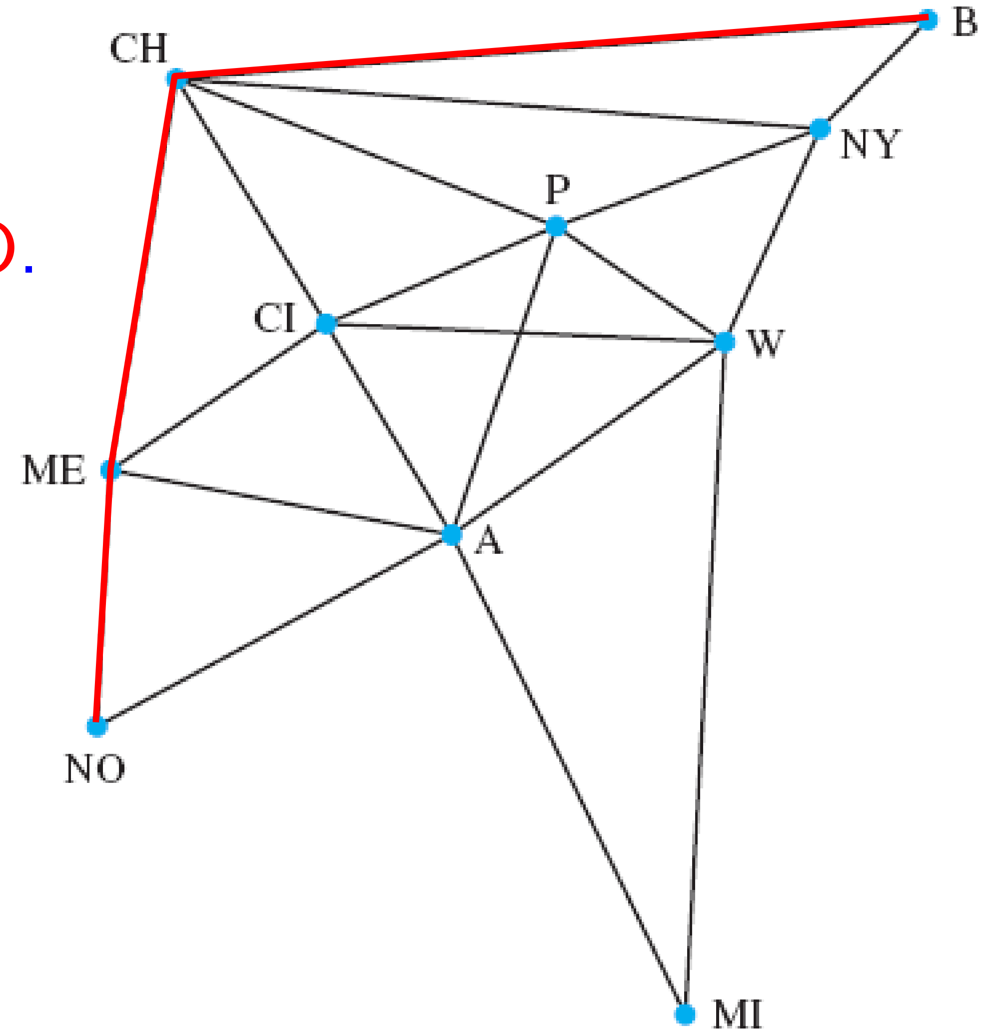
Path from Boston to New Orleans is
B{B,CH}CH{CH,ME}ME{ME,NO}NO.



Example

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

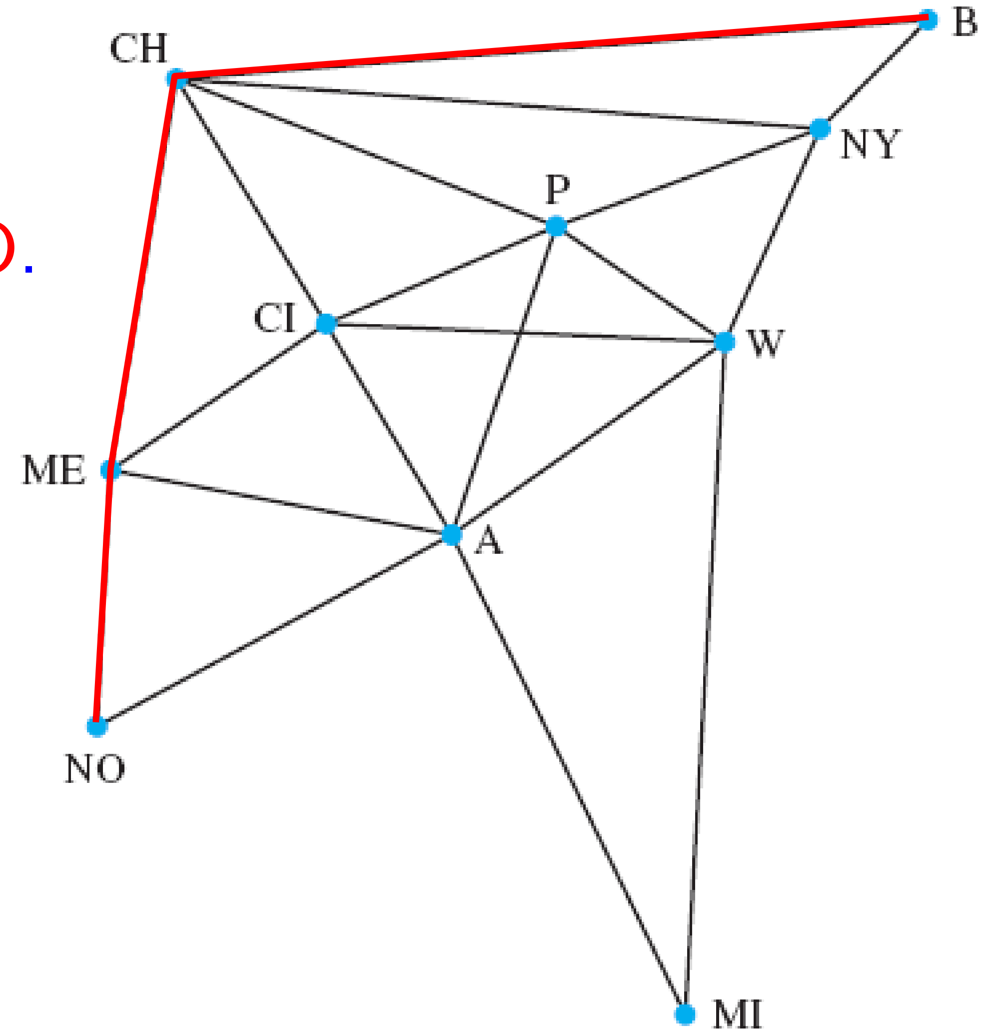
Since the 2^{nd} endpoint of an edge is the 1^{st} endpoint of the following edge, we usually just write the successive endpoints, e.g., B,CH,ME,NO.



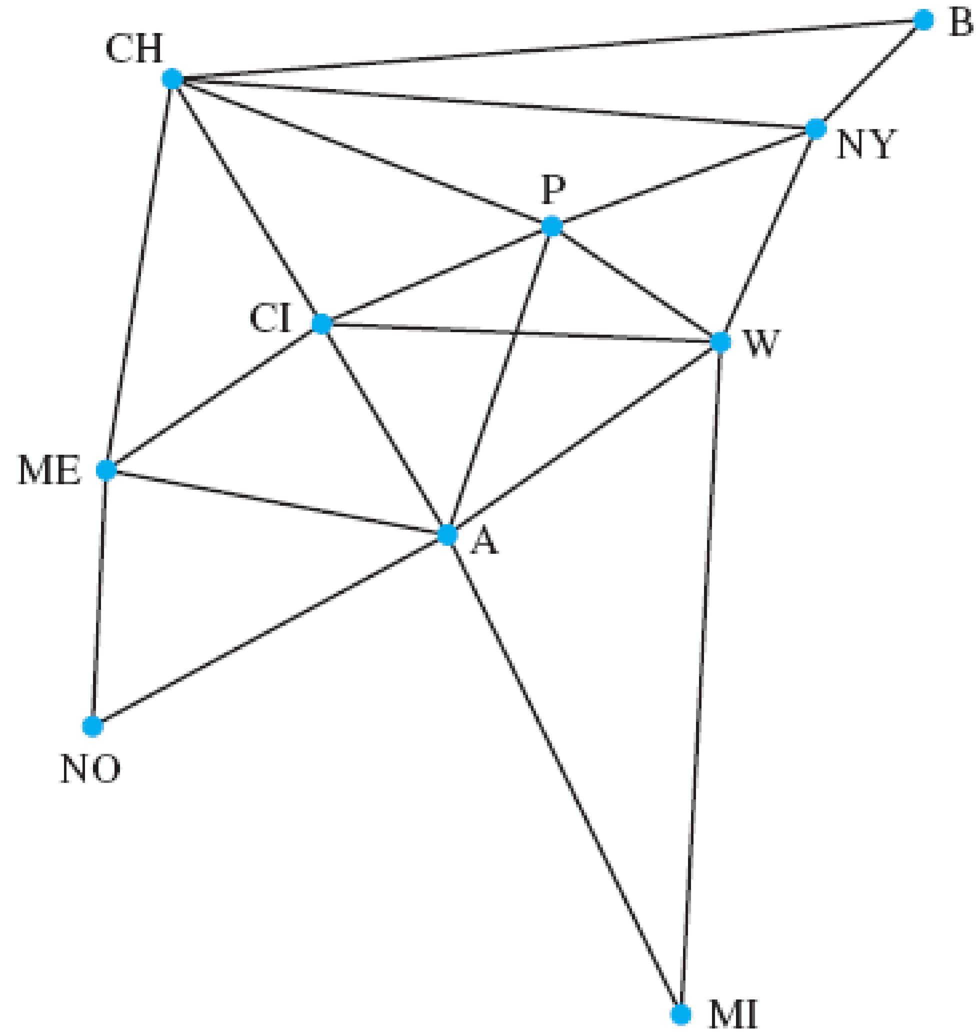
Example

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

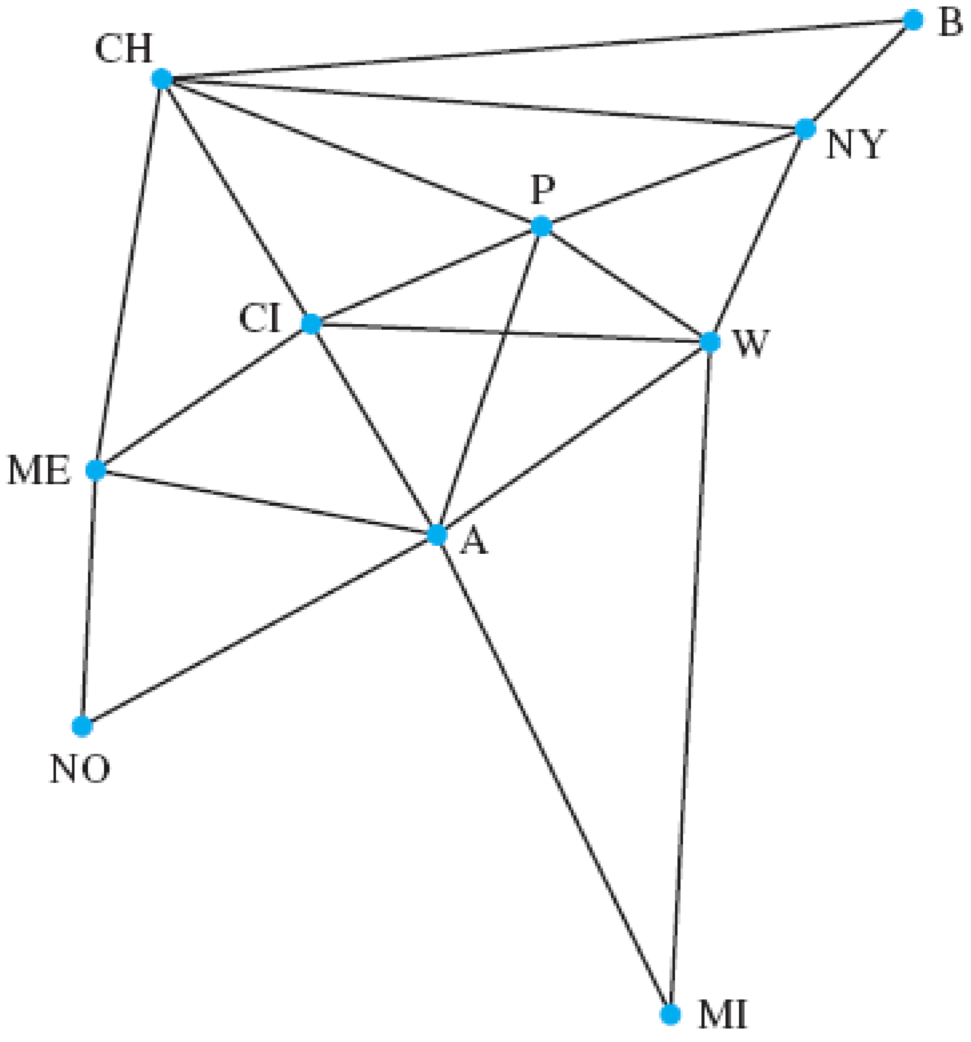
Since the 2^{nd} endpoint of an edge is the 1^{st} endpoint of the following edge, we usually just write the successive endpoints, e.g., B,CH,ME,NO.



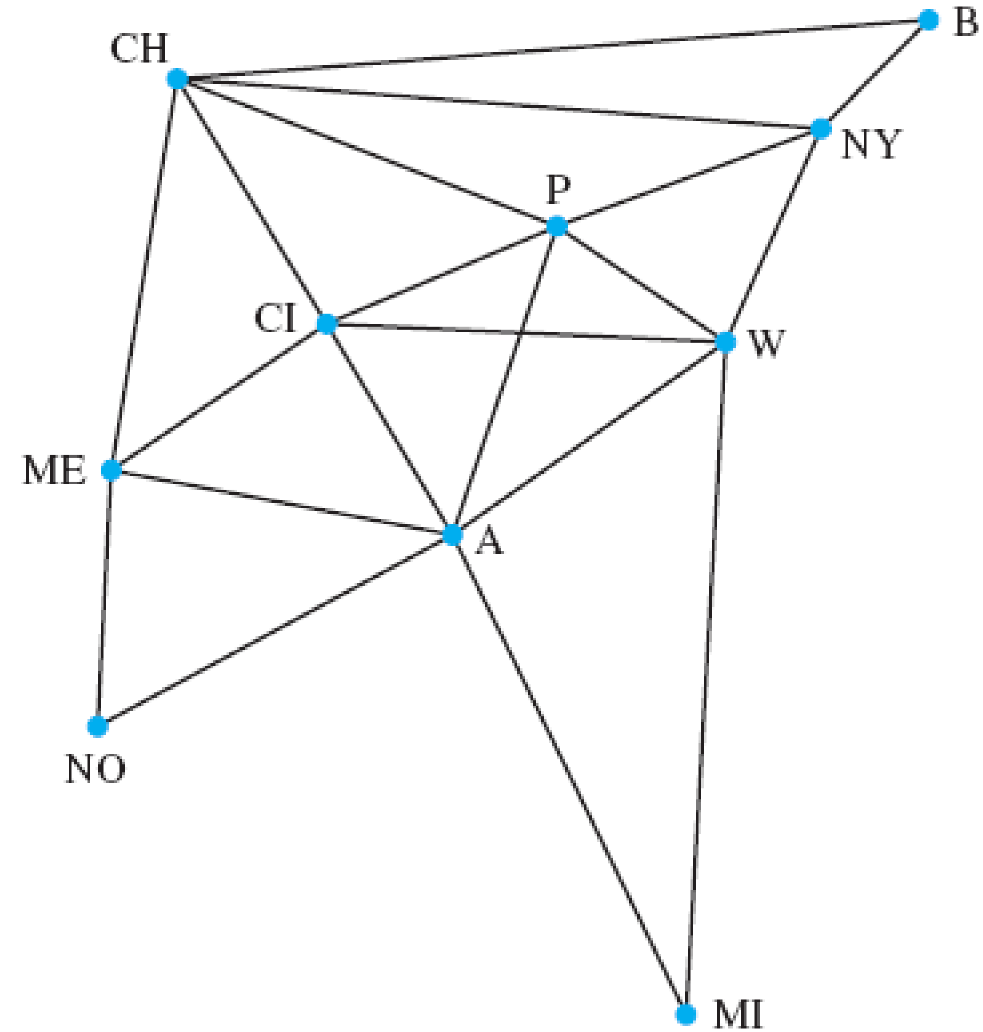
This path has length 3.


CH
B
NY
P
CI
W
ME
A
NO
MI

The distance between two vertices is the length of the shortest path between them.



The distance between two vertices is the length of the shortest path between them.



Examples:

$$\begin{aligned} \text{dist(CI, W)} &= 1 \\ \text{dist(CI, B)} &= 2 \\ \text{dist(CI, NO)} &= 2 \end{aligned}$$

A **walk** is like a path except that it may *repeat* vertices (many times)

A **walk** is like a path except that it may *repeat* vertices (many times)

Lemma 6.1 If there is a walk between two distinct vertices x and y of a graph G, then there is a path between x and y in G.

A **walk** is like a path except that it may *repeat* vertices (many times)

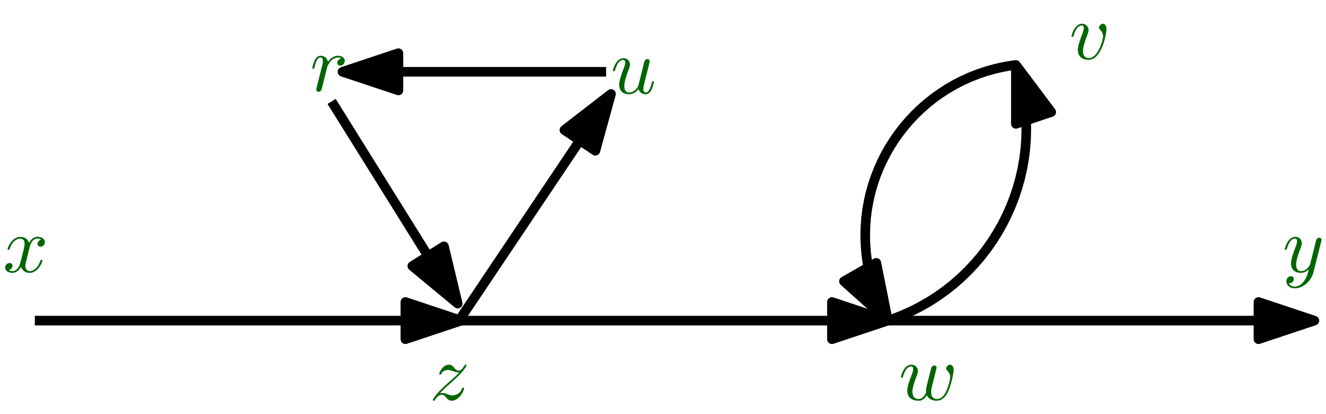
Lemma 6.1 If there is a walk between two distinct vertices x and y of a graph G, then there is a path between x and y in G.

Proof sketch: Just delete cycles (loops).

A **walk** is like a path except that it may *repeat* vertices (many times)

Lemma 6.1 If there is a walk between two distinct vertices x and y of a graph G, then there is a path between x and y in G.

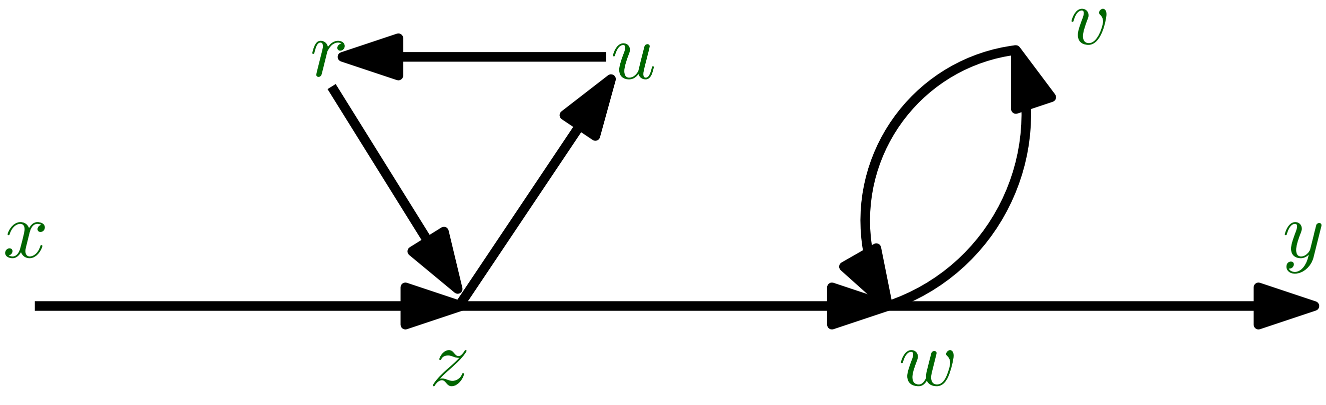
Proof sketch: Just delete cycles (loops).



A **walk** is like a path except that it may *repeat* vertices (many times)

Lemma 6.1 If there is a walk between two distinct vertices x and y of a graph G, then there is a path between x and y in G.

Proof sketch: Just delete cycles (loops).



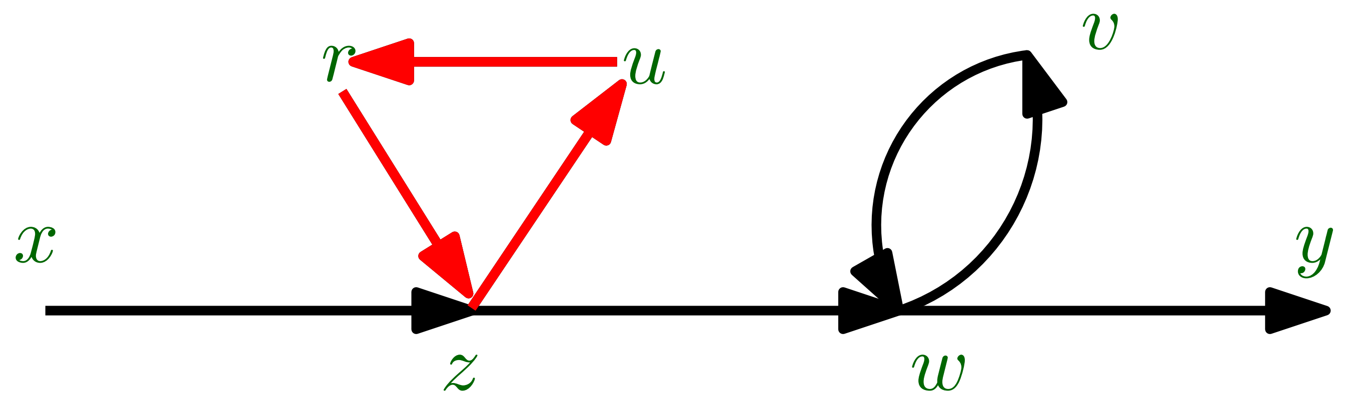
Walk from x to y

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

A **walk** is like a path except that it may *repeat* vertices (many times)

Lemma 6.1 If there is a walk between two distinct vertices x and y of a graph G, then there is a path between x and y in G.

Proof sketch: Just delete cycles (loops).



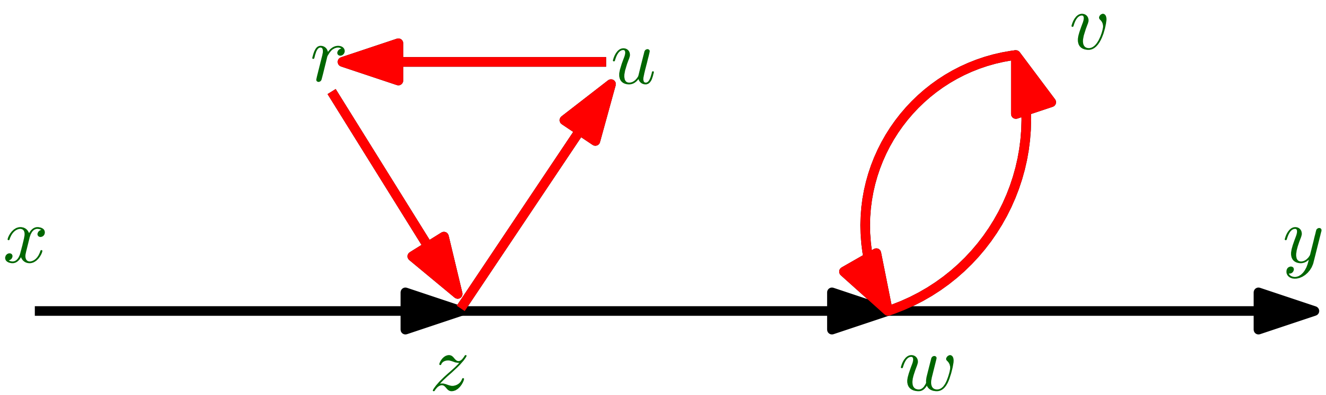
Walk from x to y
$x, z, u, r, z, w, v, w, y$.

Walk from x to y
x, z, w, v, w, y.

A **walk** is like a path except that it may *repeat* vertices (many times)

Lemma 6.1 If there is a walk between two distinct vertices x and y of a graph G, then there is a path between x and y in G.

Proof sketch: Just delete cycles (loops).



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

Graphs

- Basic Definitions
- The Degree of a Vertex
- Connectivity
- Cycles
- Trees

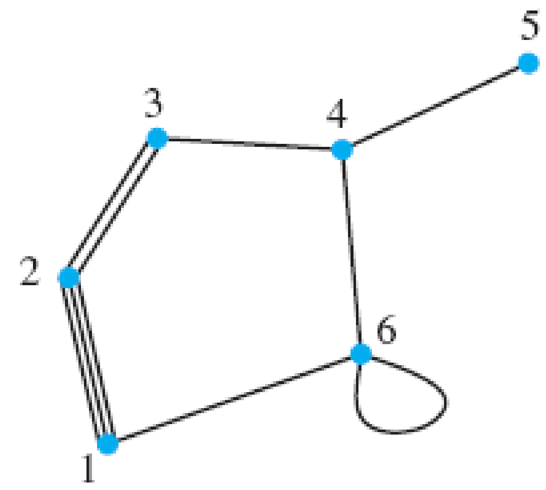
The Degree of a Vertex

The Degree of a Vertex

The **degree** of a vertex in a graph is the number of times it is incident with edges of the graph; that is, the degree of a vertex x is the number of edges between x and other vertices plus twice the number of loops at vertex x.

The Degree of a Vertex

The **degree** of a vertex in a graph is the number of times it is incident with edges of the graph; that is, the degree of a vertex x is the number of edges between x and other vertices plus twice the number of loops at vertex x.



Example: Vertex 2 has degree 5, vertex 6 has degree 4 and vertex 4 has degree 3.

Theorem 6.2 (Hand-Shaking Lemma)
Suppose a graph has a finite number of edges. Then
sum of degrees of vertices is twice number of edges.

Theorem 6.2 (Hand-Shaking Lemma)
Suppose a graph has a finite number of edges. Then
sum of degrees of vertices is twice number of edges.

Proof (by induction):

Theorem 6.2 (Hand-Shaking Lemma)
Suppose a graph has a finite number of edges. Then
sum of degrees of vertices is twice number of edges.

Proof (by induction):

Base case:

If a graph has no edges, then each vertex has degree 0 and the sum of the degrees is 0, which is twice the number of edges.

Theorem 6.2 (Hand-Shaking Lemma)
Suppose a graph has a finite number of edges. Then
sum of degrees of vertices is twice number of edges.

Proof (by induction):

Base case:
If a graph has no edges, then each vertex has degree 0 and the sum of the degrees is 0, which is twice the number of edges.

Let m be the number of edges of G.

Theorem 6.2 (Hand-Shaking Lemma)
Suppose a graph has a finite number of edges. Then
sum of degrees of vertices is twice number of edges.

Proof (by induction):

Base case:

If a graph has no edges, then each vertex has degree 0 and the sum of the degrees is 0, which is twice the number of edges.

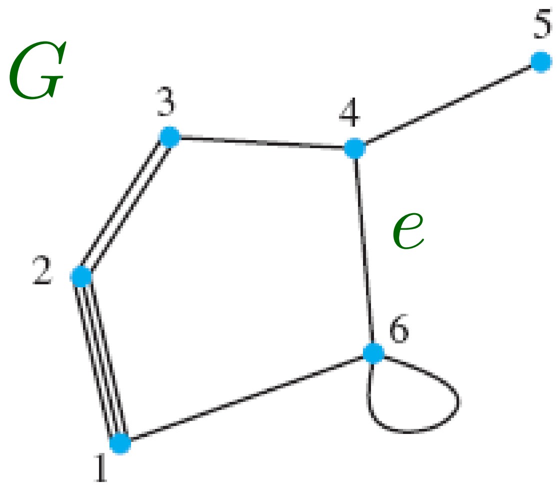
Let m be the number of edges of G.

Inductive Hypothesis:

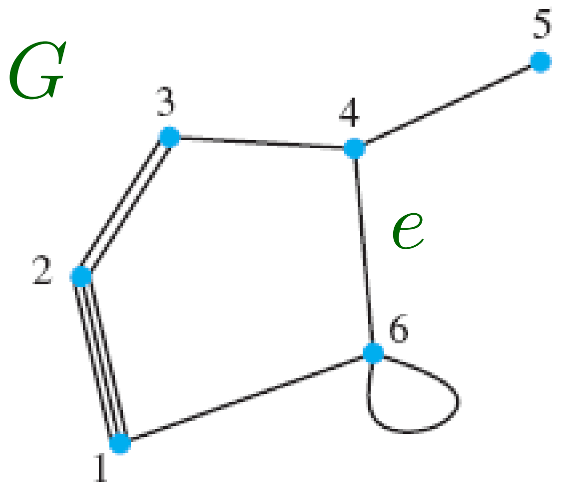
Suppose that $m > 0$ and that the theorem is true whenever a graph has fewer than m edges.

Let G be a graph with m edges and let e be an edge of G.

Let G be a graph with m edges and let e be an edge of G.

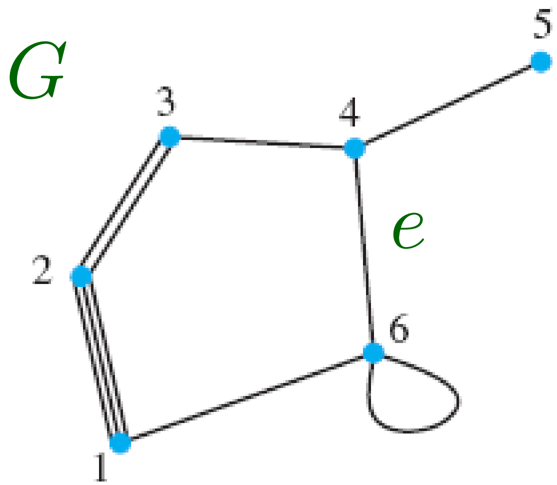


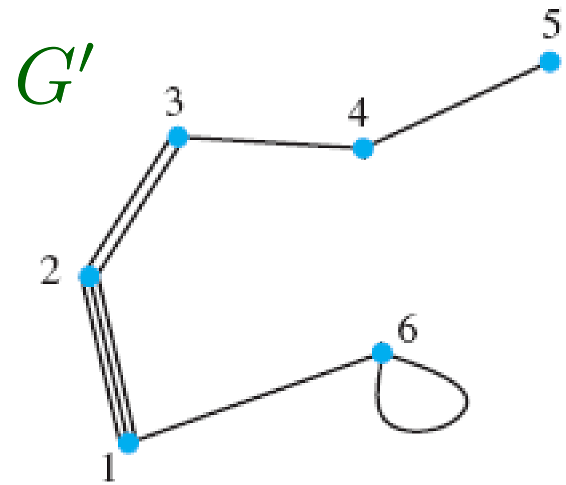
Let G be a graph with m edges and let e be an edge of G.



Let G' be a graph (on the same vertex set as G) that we get by deleting e from the edge set E of G.

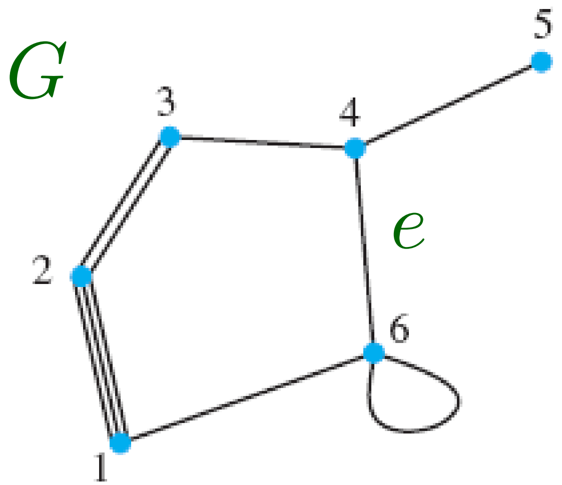
Let G be a graph with m edges and let e be an edge of G.

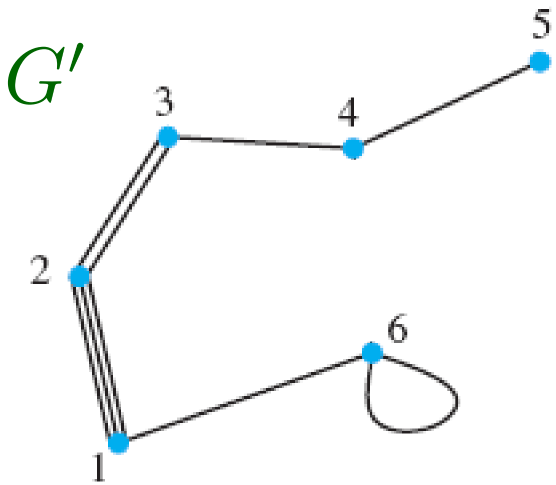




Let G' be a graph (on the same vertex set as G) that we get by deleting e from the edge set E of G.

Let G be a graph with m edges and let e be an edge of G.

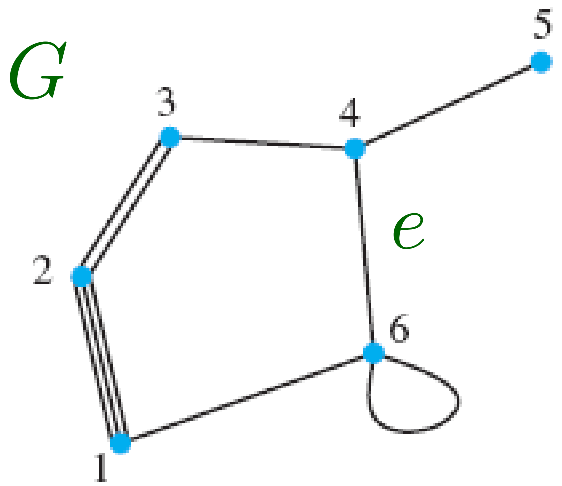


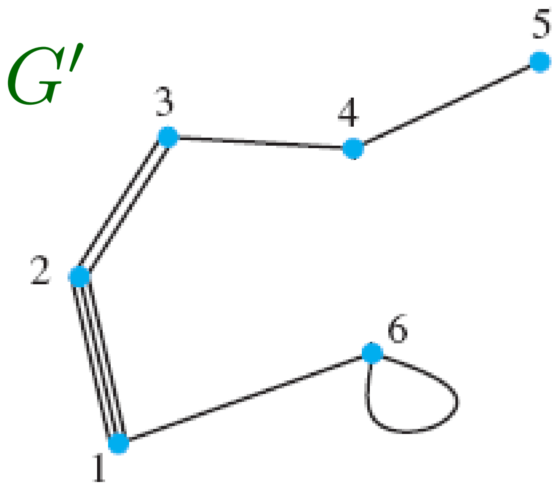


Let G' be a graph (on the same vertex set as G) that we get by deleting e from the edge set E of G.

Then G' has $m-1$ edges, and so, by our inductive hypothesis, sum of degrees of vertices of G' is $2(m-1)$.

Let G be a graph with m edges and let e be an edge of G.



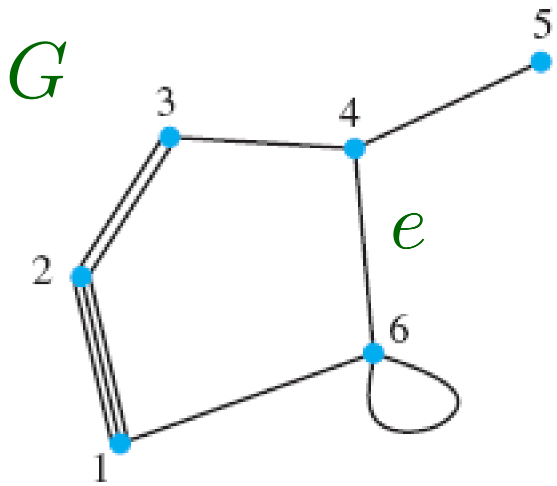


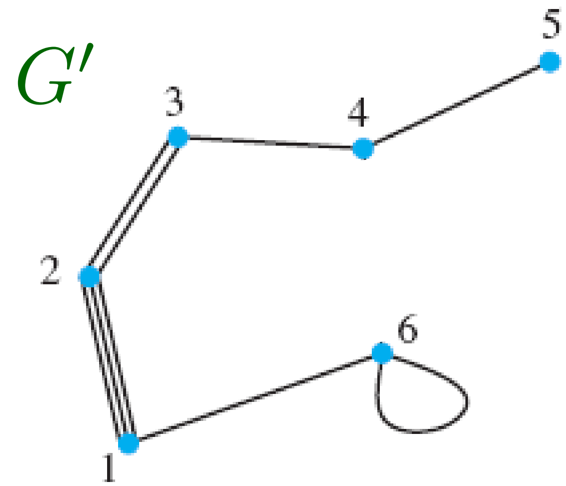
Let G' be a graph (on the same vertex set as G) that we get by deleting e from the edge set E of G.

Then G' has $m-1$ edges, and so, by our inductive hypothesis, sum of degrees of vertices of G' is $2(m-1)$.

Two possible cases:

Let G be a graph with m edges and let e be an edge of G.





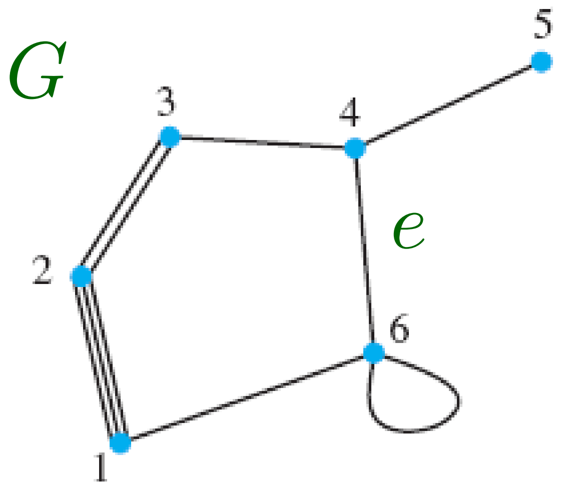
Let G' be a graph (on the same vertex set as G) that we get by deleting e from the edge set E of G.

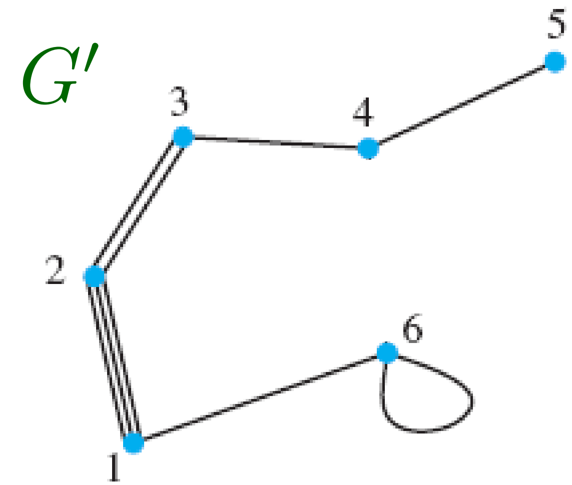
Then G' has $m-1$ edges, and so, by our inductive hypothesis, sum of degrees of vertices of G' is $2(m-1)$.

Two possible cases:

- e is a loop

Let G be a graph with m edges and let e be an edge of G.





Let G' be a graph (on the same vertex set as G) that we get by deleting e from the edge set E of G.

Then G' has $m-1$ edges, and so, by our inductive hypothesis, sum of degrees of vertices of G' is $2(m-1)$.

Two possible cases:

- e is a loop
- e has two distinct endpoints

Two possible cases:

- e is a loop, so one vertex of G' has degree two less in G' than it has in G

Two possible cases:

- e is a loop, so one vertex of G' has degree two less in G' than it has in G
- e has two distinct endpoints, so exactly two vertices of G' have degree one less than their degree in G.

Two possible cases:

- e is a loop, so one vertex of G' has degree two less in G' than it has in G
- e has two distinct endpoints, so exactly two vertices of G' have degree one less than their degree in G.

$\Rightarrow$ in both cases, sum of degrees of vertices in G' is two less than sum of degrees of vertices in G.

Two possible cases:

- e is a loop, so one vertex of G' has degree two less in G' than it has in G
- e has two distinct endpoints, so exactly two vertices of G' have degree one less than their degree in G.

$\Rightarrow$ in both cases, sum of degrees of vertices in G' is two less than sum of degrees of vertices in G.

Therefore, sum of degrees of vertices in G is $(2m-2)+2 = 2m$.

Two possible cases:

- e is a loop, so one vertex of G' has degree two less in G' than it has in G
- e has two distinct endpoints, so exactly two vertices of G' have degree one less than their degree in G.

$\Rightarrow$ in both cases, sum of degrees of vertices in G' is two less than sum of degrees of vertices in G.

Therefore, sum of degrees of vertices in G is $(2m-2)+2 = 2m$.

$\Rightarrow$ truth of theorem for graphs with $m-1$ edges implies truth of theorem for graphs with m edges.

Two possible cases:

- e is a loop, so one vertex of G' has degree two less in G' than it has in G
- e has two distinct endpoints, so exactly two vertices of G' have degree one less than their degree in G.

$\Rightarrow$ in both cases, sum of degrees of vertices in G' is two less than sum of degrees of vertices in G.

Therefore, sum of degrees of vertices in G is $(2m-2)+2 = 2m$.

$\Rightarrow$ truth of theorem for graphs with $m-1$ edges implies truth of theorem for graphs with m edges.

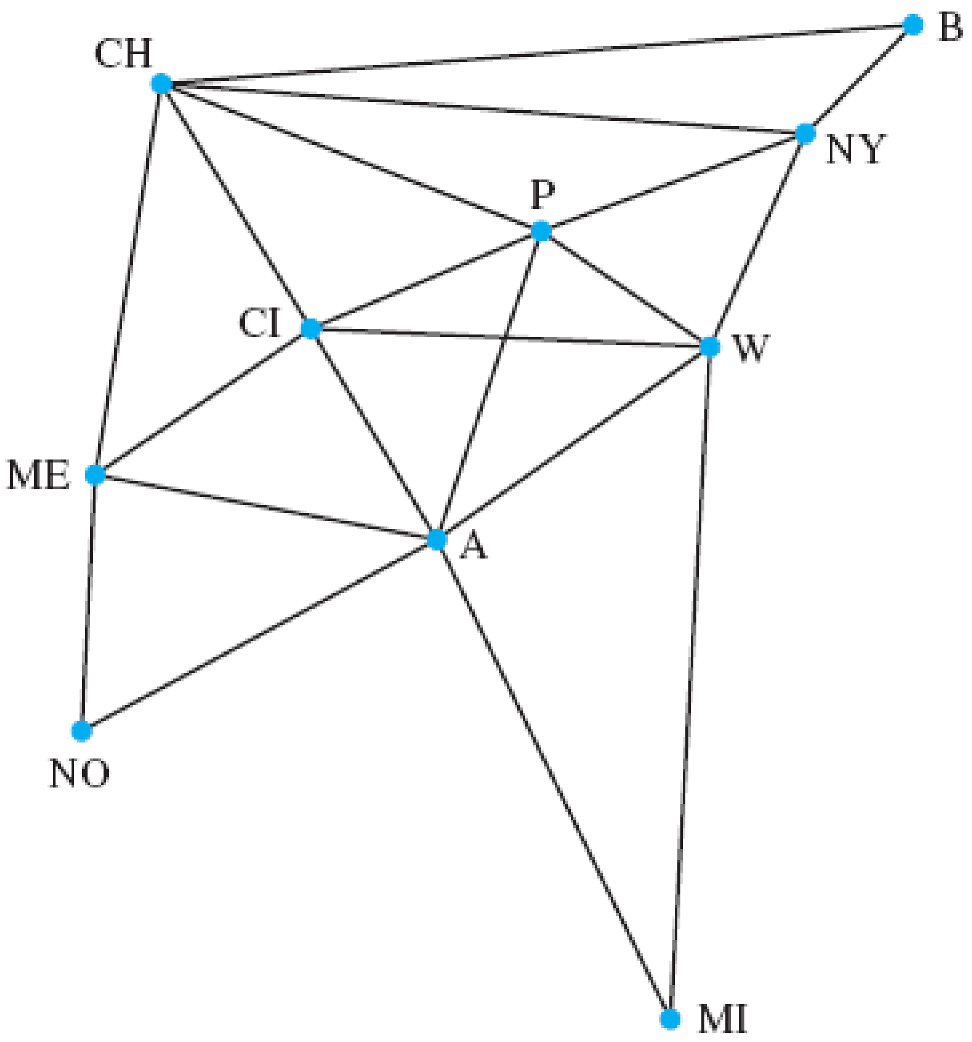
Therefore, by principle of mathematical induction, theorem is true for a graph with any finite number of edges.

Graphs

- Basic Definitions
- The Degree of a Vertex
- Connectivity
- Cycles
- Trees

Connectivity

Connectivity

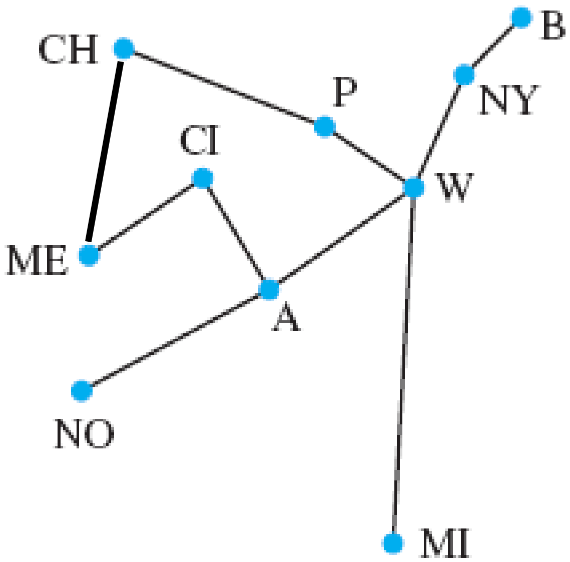


Company decides to lease only minimum number of communication lines it needs to be able to send a message from any city to any other city by using any number of intermediate cities.

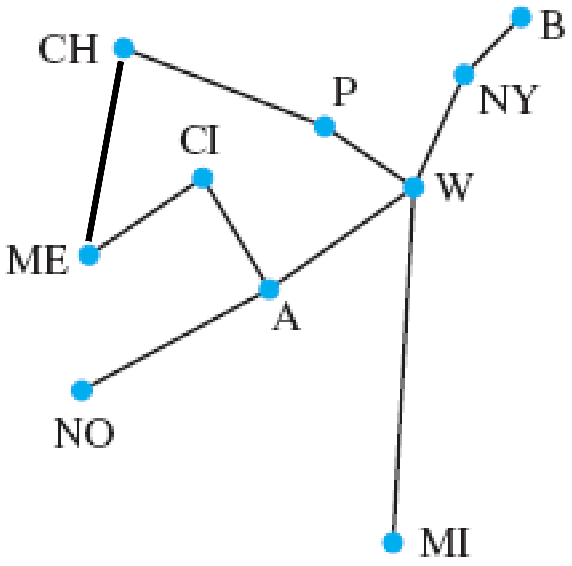
What is **minimum** number of lines it needs to lease?

Choosing 10 edges?

Choosing 10 edges?



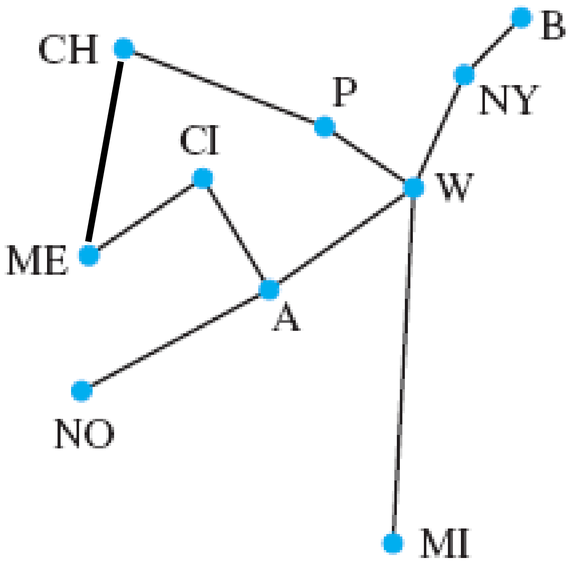
Choosing 10 edges?



Too many.

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

Choosing 10 edges?

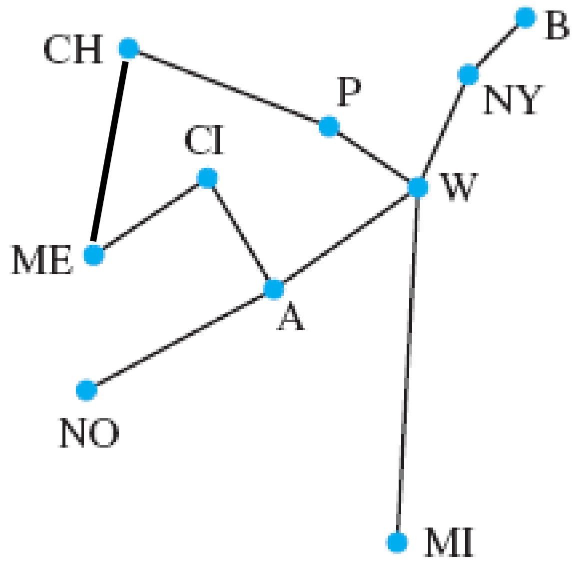


Too many.

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

Choosing 8 edges?

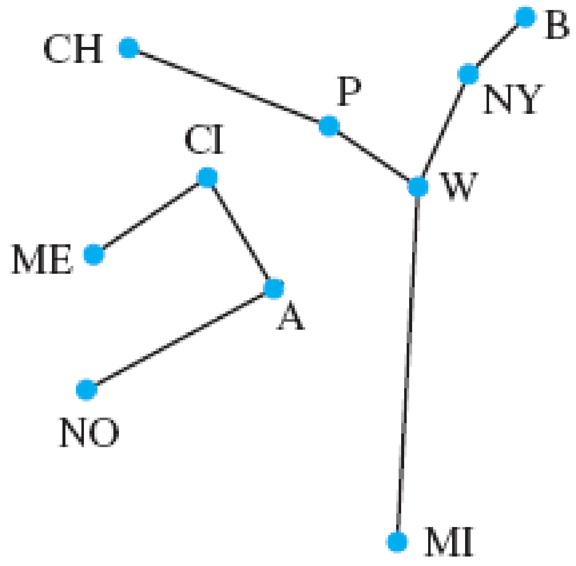
Choosing 10 edges?



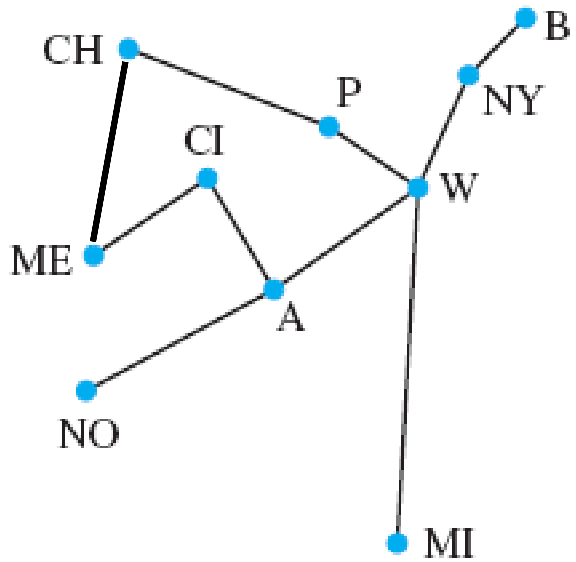
Too many.

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

Choosing 8 edges?



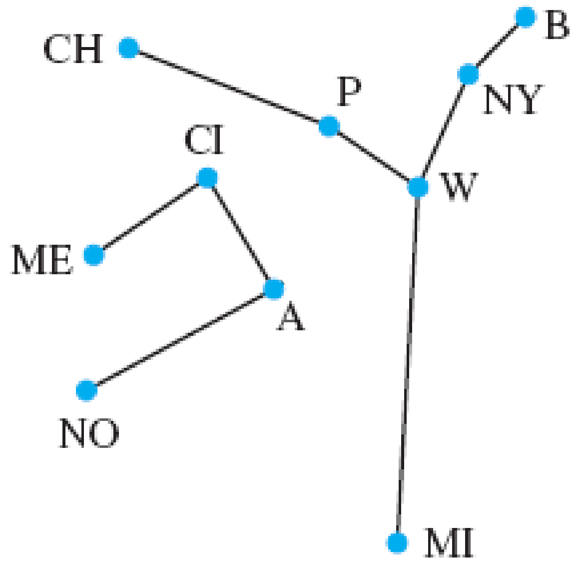
Choosing 10 edges?



Too many.

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

Choosing 8 edges?

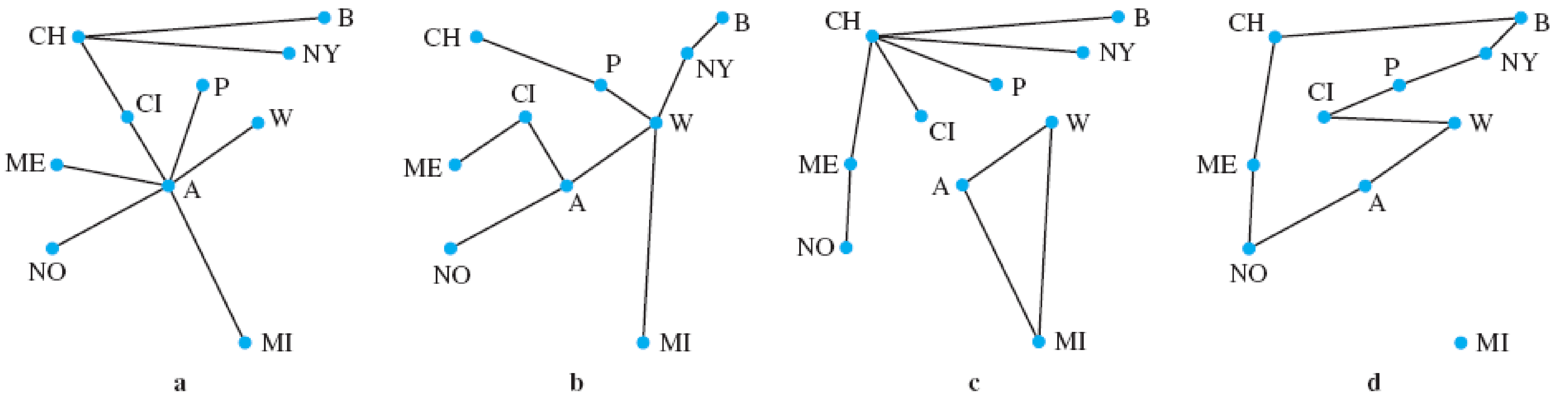


Not enough.

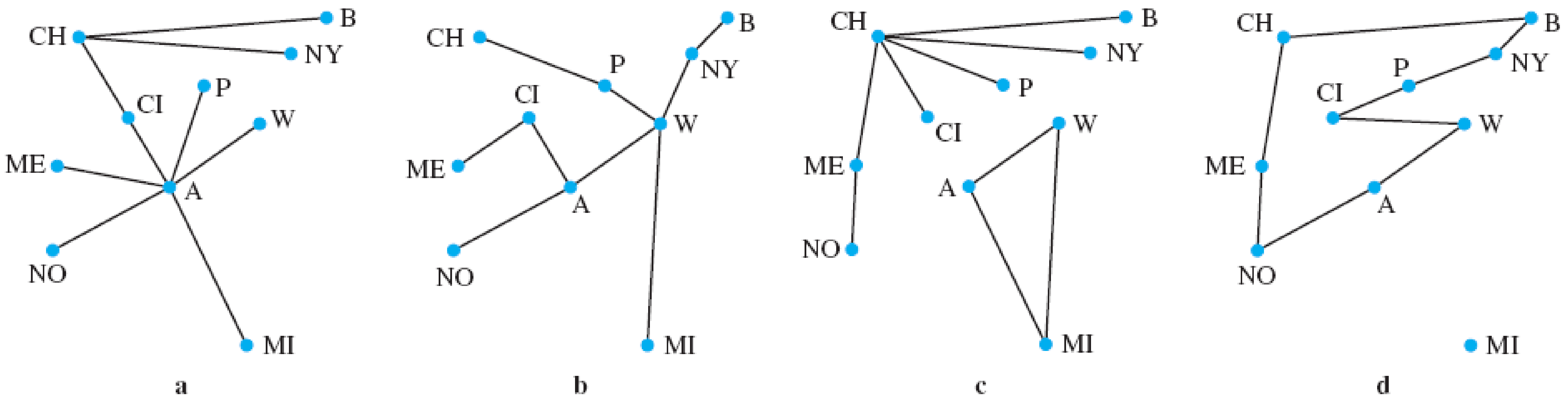
There is no path from, e.g., NO to B.

Choosing 9 edges:

Choosing 9 edges:



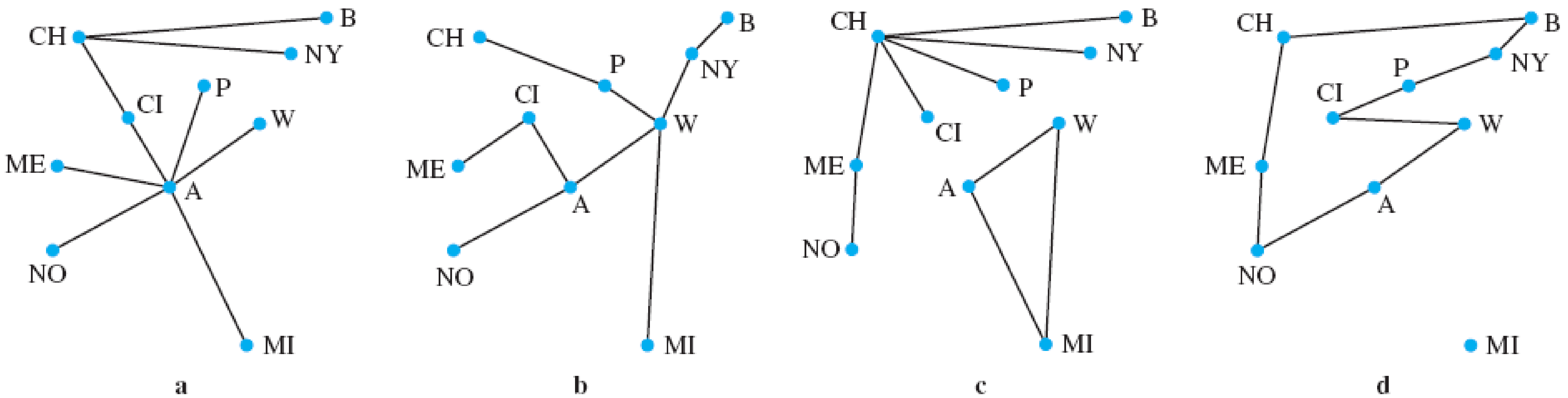
Choosing 9 edges:



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

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

Choosing 9 edges:

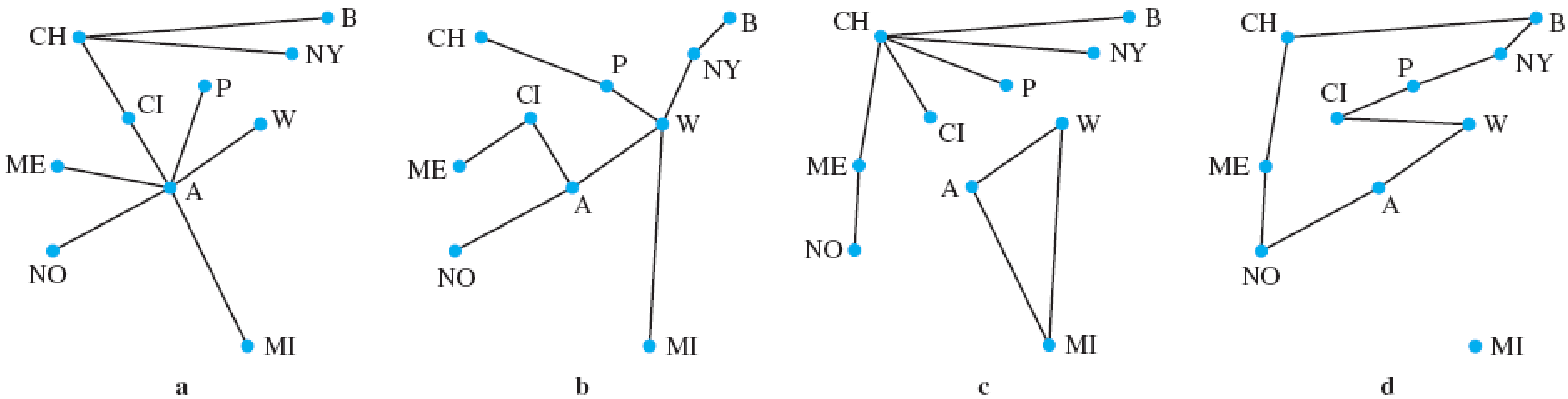


Two vertices are **connected** if there is a path between them.

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

Graph is **connected** when every pair of vertices is connected.

Choosing 9 edges:



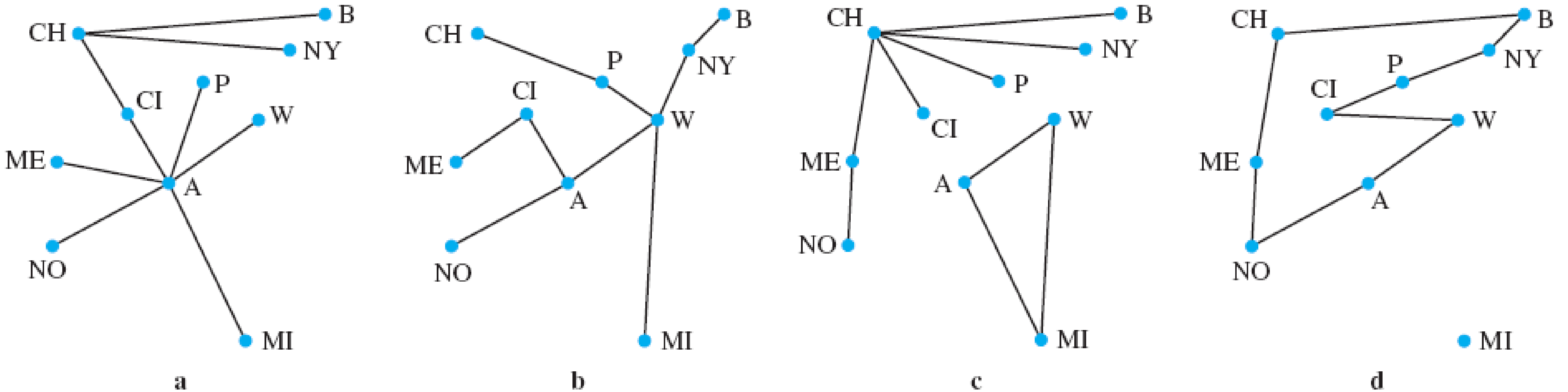
Two vertices are **connected** if there is a path between them.

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

Graph is **connected** when every pair of vertices is connected.

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

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).

Graph is **connected** when every pair of vertices is connected.

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

In (d), we say that M(iami) is an **isolated vertex**.

Relationship of "being connected to" is called the **connectivity relation**.

Relationship of "being connected to" is called the **connectivity relation**.

The blocks into which this relationship partitions the graph are called **connectivity classes**.

Relationship of "being connected to" is called the **connectivity relation**.

The blocks into which this relationship partitions the graph are called **connectivity classes**.

The graph consisting of a connectivity class C, together with the edges $E(C)$ (the edges connecting the items in the class), is called a **connected component (cc)** of our original graph.

Relationship of "being connected to" is called the **connectivity relation**.

The blocks into which this relationship partitions the graph are called **connectivity classes**.

The graph consisting of a connectivity class C, together with the edges $E(C)$ (the edges connecting the items in the class), is called a **connected component (cc)** of our original graph.

Example:
(c) and (d) each have two connected components.

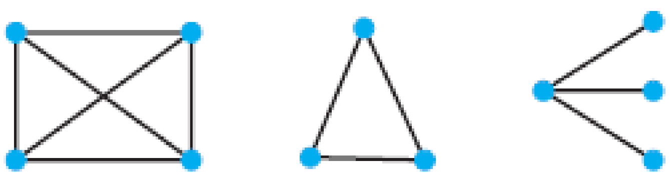
Relationship of "being connected to" is called the **connectivity relation**.

The blocks into which this relationship partitions the graph are called **connectivity classes**.

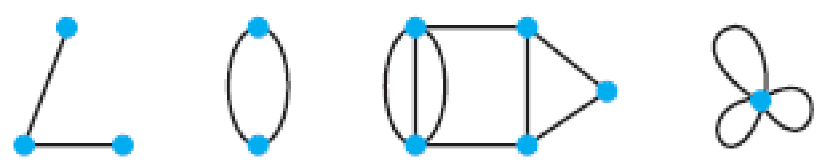
The graph consisting of a connectivity class C, together with the edges $E(C)$ (the edges connecting the items in the class), is called a **connected component (cc)** of our original graph.

Example:
(c) and (d) each have two connected components.

More Examples:



G_1 3 cc's



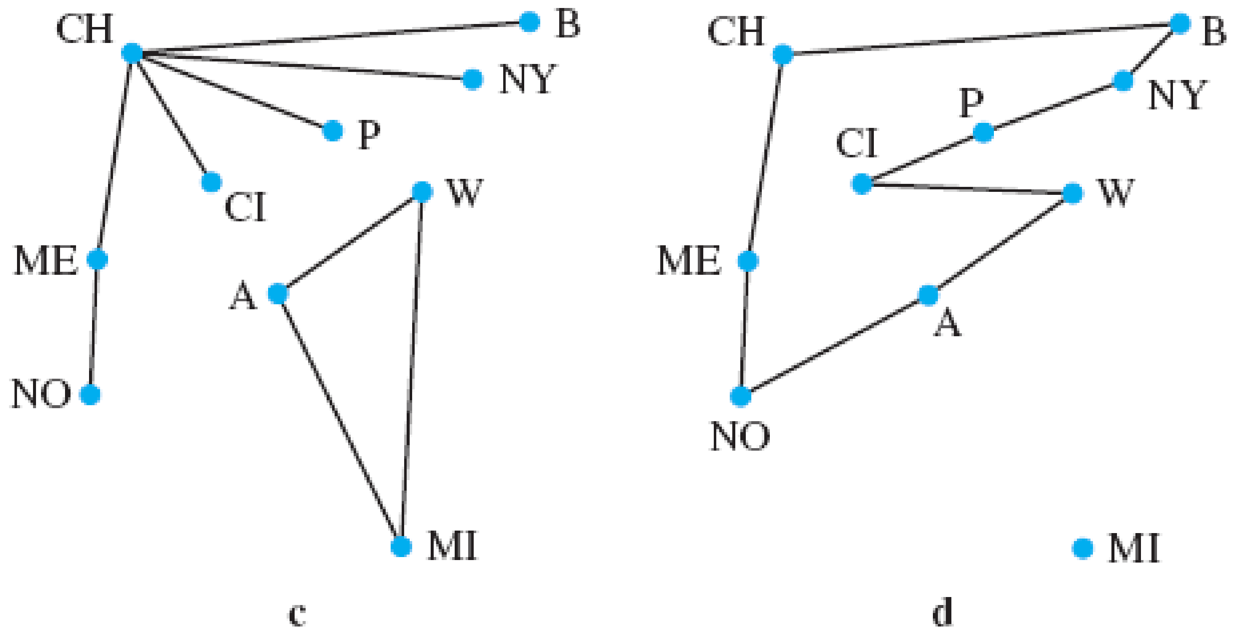
G_2 4 cc's

Graphs

- Basic Definitions
- The Degree of a Vertex
- Connectivity
- Cycles
- Trees

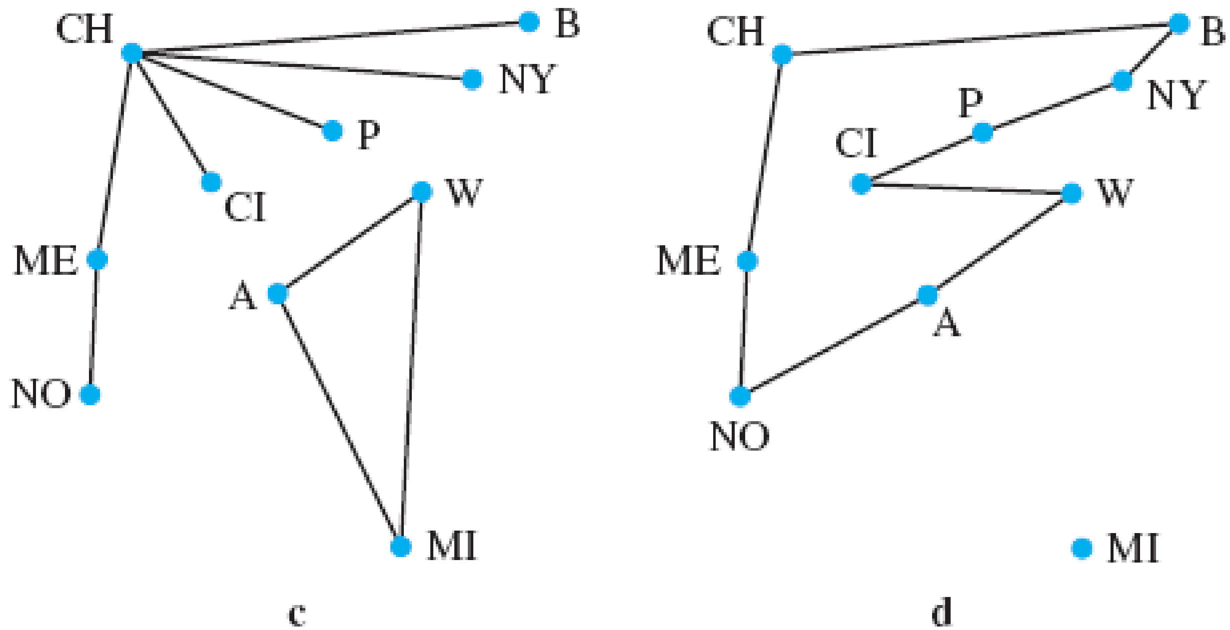
Cycles

Cycles



A walk that "starts" and "ends" with the same vertex is called a **closed walk**.

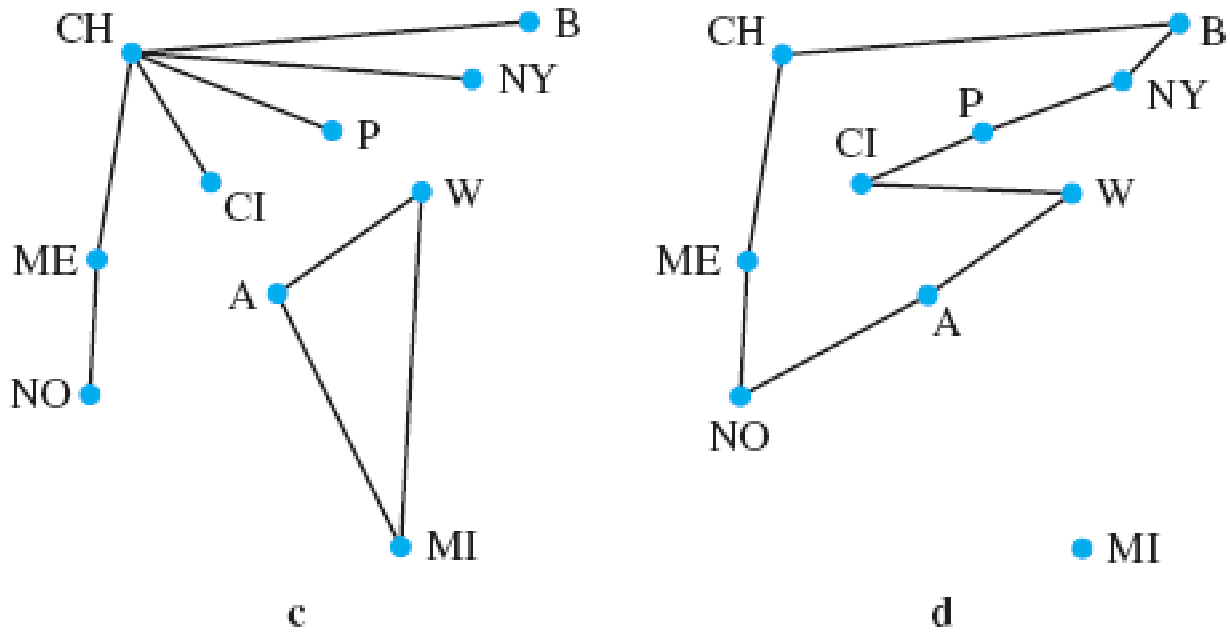
Cycles



A walk that "starts" and "ends" with the same vertex is called a **closed walk**.

A closed walk that does not "repeat" any vertex is called a **cycle**.

Cycles



A walk that "starts" and "ends" with the same vertex is called a **closed walk**.

A closed walk that does not "repeat" any vertex is called a **cycle**.

Example: The closed walks in (c) and (d) are, respectively, cycles A, W, M, A and NO, ME, CH, B, NY, P, CI, W, A, NO.

Graph H is a **subgraph** of graph G if
all vertices and edges of H are vertices and edges of G.

Graph H is a **subgraph** of graph G if
all vertices and edges of H are vertices and edges of G.

In other words, $H = (V', E')$ is a subgraph of $G = (V, E)$ if
$V' \subseteq V$ and $E' \subseteq E$.

Graph H is a **subgraph** of graph G if
all vertices and edges of H are vertices and edges of G.

In other words, $H = (V', E')$ is a subgraph of $G = (V, E)$ if
$V' \subseteq V$ and $E' \subseteq E$.

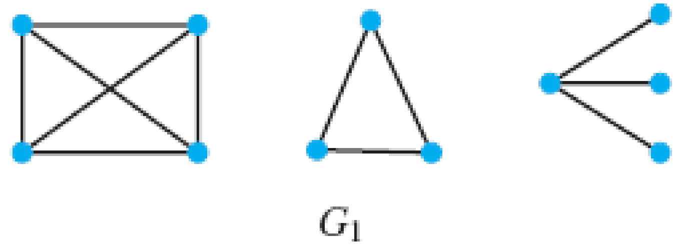
Graph H is an induced subgraph of G if
H is a subgraph of G and
every edge of G connecting vertices of H is an edge of H.

Graph H is a **subgraph** of graph G if
all vertices and edges of H are vertices and edges of G.

In other words, $H = (V', E')$ is a subgraph of $G = (V, E)$ if
$V' \subseteq V$ and $E' \subseteq E$.

Graph H is an induced subgraph of G if
H is a subgraph of G and
every edge of G connecting vertices of H is an edge of H.

Example:

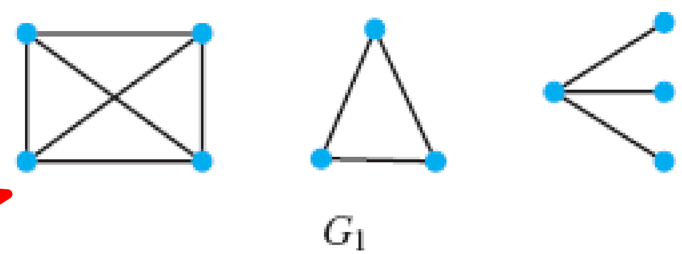


Graph H is a **subgraph** of graph G if
all vertices and edges of H are vertices and edges of G.

In other words, $H = (V', E')$ is a subgraph of $G = (V, E)$ if
$V' \subseteq V$ and $E' \subseteq E$.

Graph H is an induced subgraph of G if
H is a subgraph of G and
every edge of G connecting vertices of H is an edge of H.

Example:



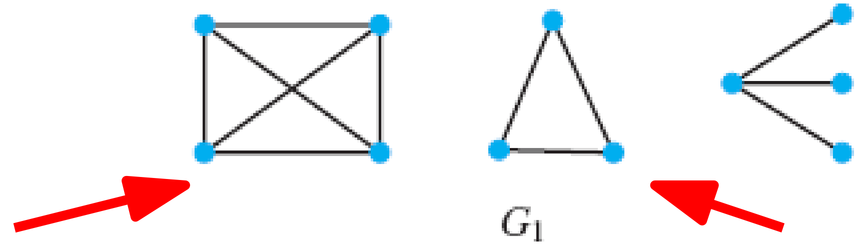
induced K_4

Graph H is a **subgraph** of graph G if
all vertices and edges of H are vertices and edges of G.

In other words, $H = (V', E')$ is a subgraph of $G = (V, E)$ if
$V' \subseteq V$ and $E' \subseteq E$.

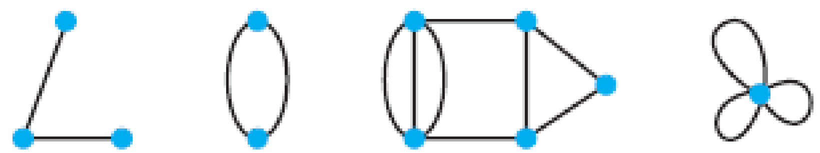
Graph H is an induced subgraph of G if
H is a subgraph of G and
every edge of G connecting vertices of H is an edge of H.

Example:

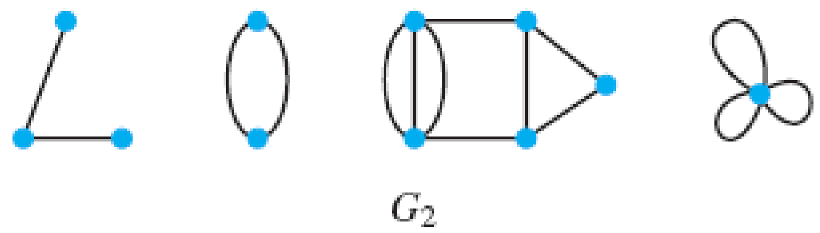


induced K_4

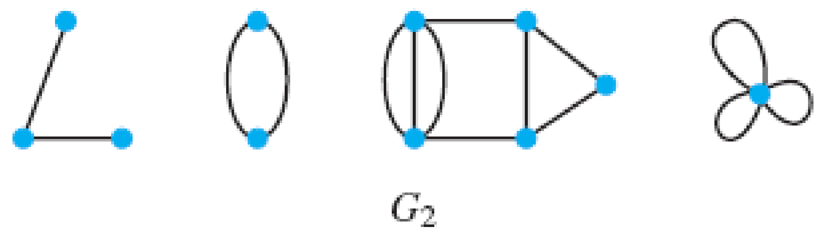
induced K_3



G_2

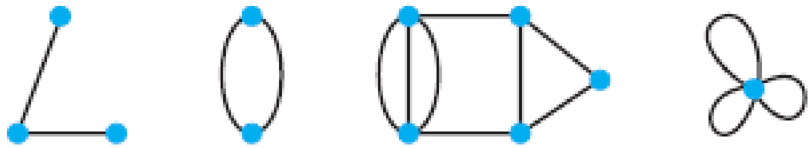


A graph is a **cycle on n vertices**, or an n-cycle, denoted by C_n, if its vertex set is the vertex set of a cycle and its edge set is the edge set of that cycle.



A graph is a **cycle on n vertices**, or an n-cycle, denoted by C_n, if its vertex set is the vertex set of a cycle and its edge set is the edge set of that cycle.

A graph is a **path on n vertices**, denoted by P_n, if its vertex set is the vertex set of a path and its edge set is the edge set of that path.



A graph is a **cycle on n vertices**, or an n-cycle, denoted by C_n, if its vertex set is the vertex set of a cycle and its edge set is the edge set of that cycle.

A graph is a **path on n vertices**, denoted by P_n, if its vertex set is the vertex set of a path and its edge set is the edge set of that path.

Examples:
Graph G_2 has an induced P_3 and an induced C_2 as subgraphs.

Graphs

- Basic Definitions
- The Degree of a Vertex
- Connectivity
- Cycles
- Trees

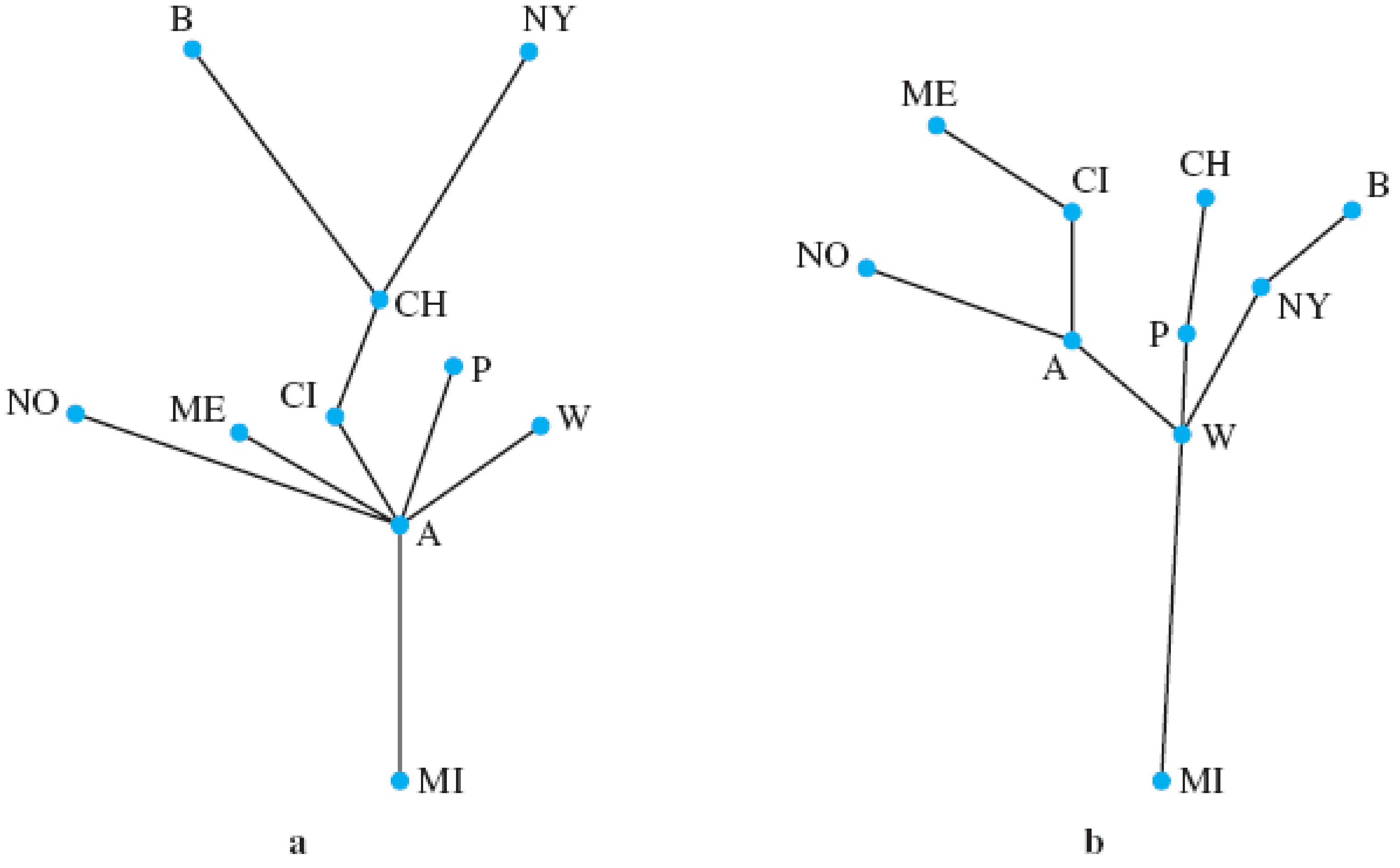
Trees

Trees

A connected graph with no cycles is called a **tree**.

Trees

A connected graph with no cycles is called a **tree**.



Properties of Trees

Properties of Trees

Given any two vertices in a tree, how many distinct paths are there between these two vertices?

Properties of Trees

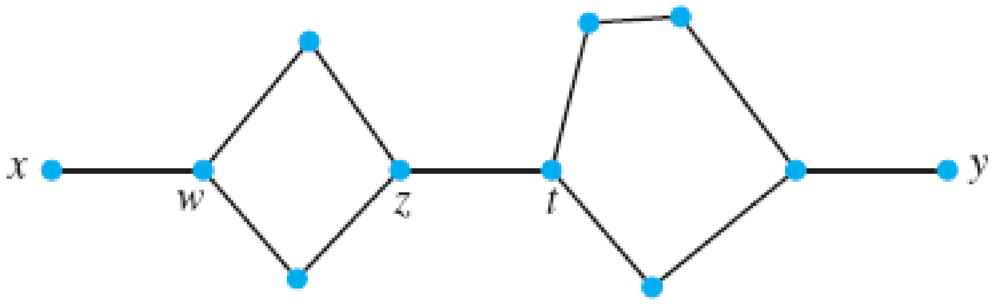
Given any two vertices in a tree, how many distinct paths are there between these two vertices?

Suppose we have two distinct paths from vertex x to vertex y.

Properties of Trees

Given any two vertices in a tree, how many distinct paths are there between these two vertices?

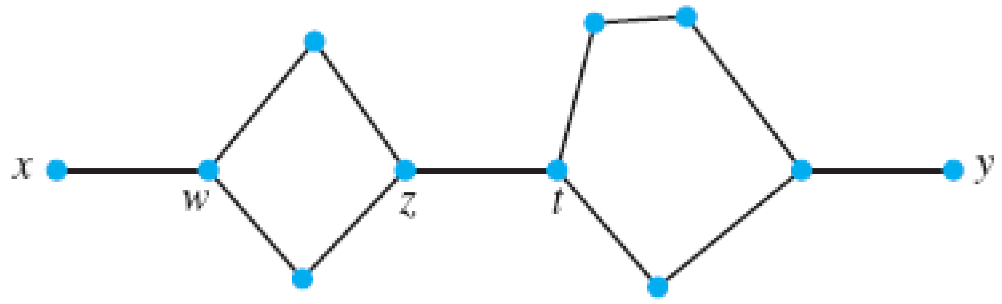
Suppose we have two distinct paths from vertex x to vertex y.



Properties of Trees

Given any two vertices in a tree, how many distinct paths are there between these two vertices?

Suppose we have two distinct paths from vertex x to vertex y.

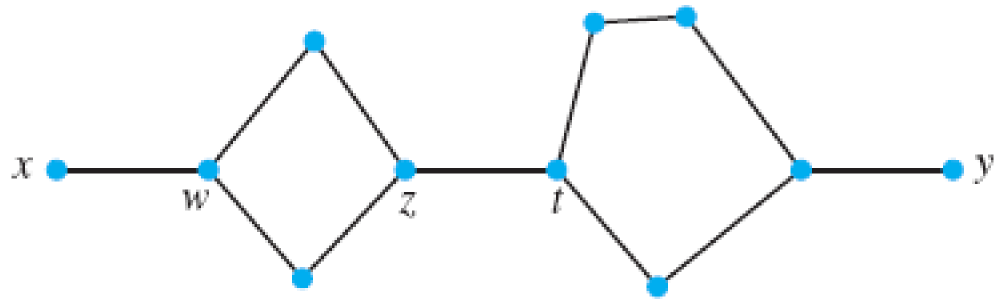


The paths begin with the same vertex x
(and might have some more edges in common).

Properties of Trees

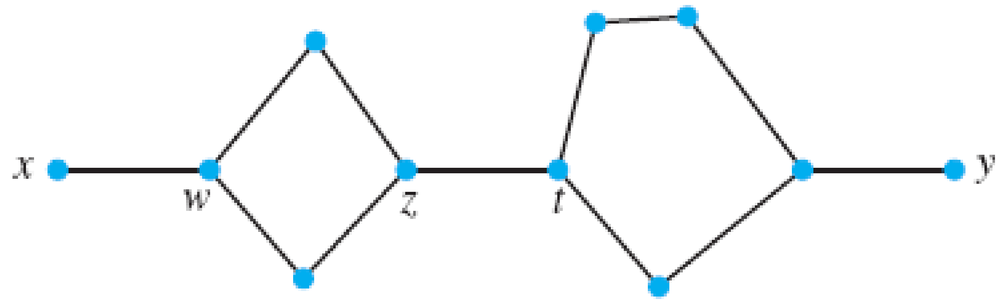
Given any two vertices in a tree, how many distinct paths are there between these two vertices?

Suppose we have two distinct paths from vertex x to vertex y.

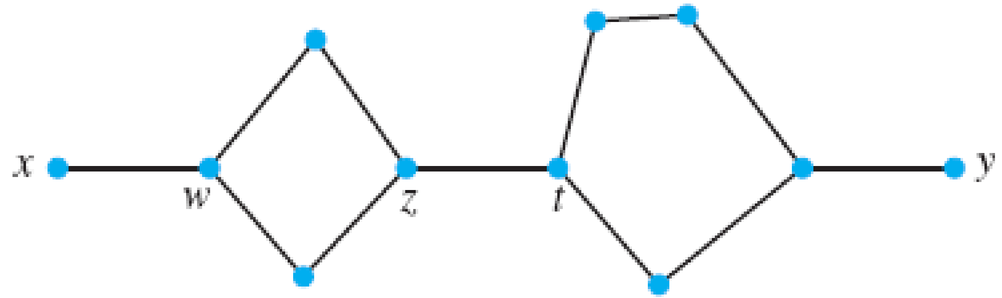


The paths begin with the same vertex x
(and might have some more edges in common).

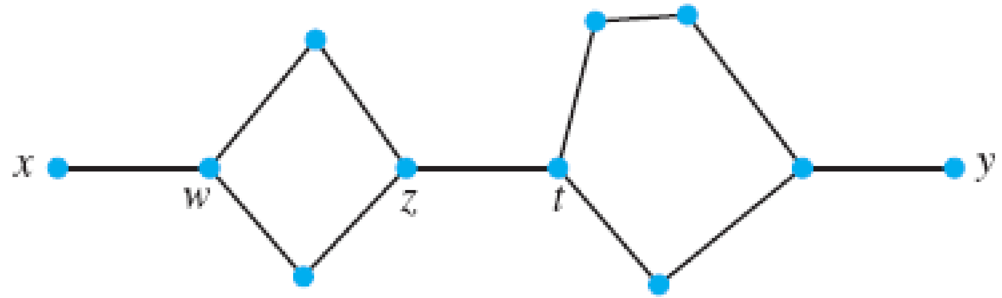
Let w be the last vertex after (or including) x that the paths share before they contain their first different edge.



x
w
z
t
y

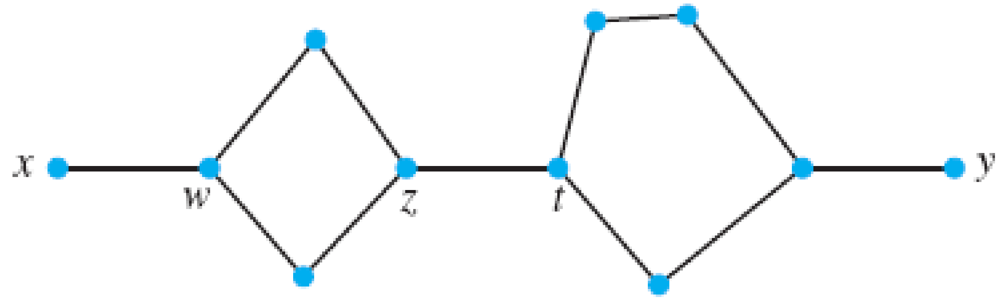


We now examine the two paths starting at w; they must join together again at y, though they might have joined at some earlier vertex.



We now examine the two paths starting at w; they must join together again at y, though they might have joined at some earlier vertex.

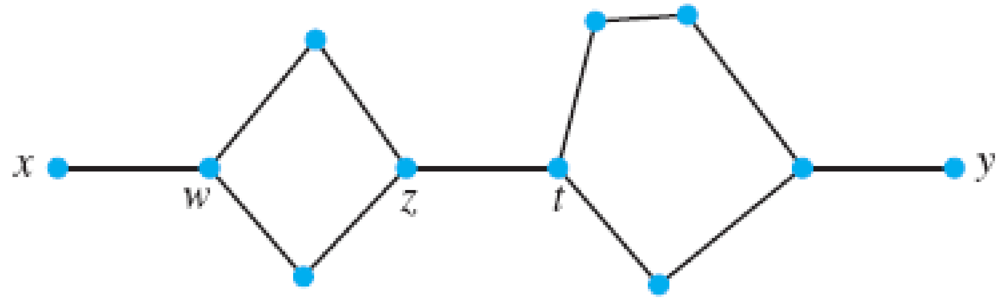
Let z be the first vertex the paths have in common after w.



We now examine the two paths starting at w; they must join together again at y, though they might have joined at some earlier vertex.

Let z be the first vertex the paths have in common after w.

Then there are two paths from w to z
that have only w and z in common.

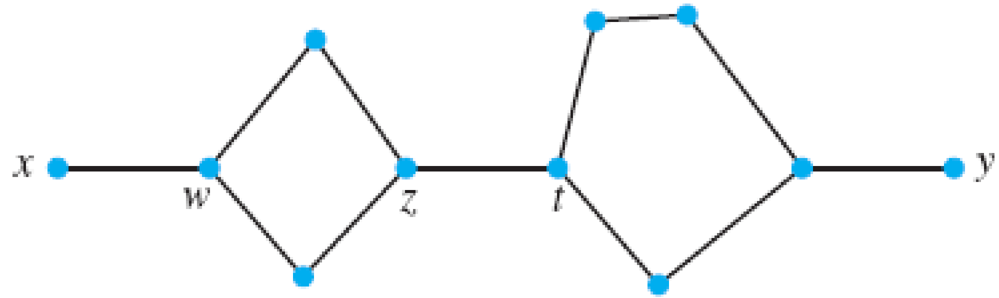


We now examine the two paths starting at w; they must join together again at y, though they might have joined at some earlier vertex.

Let z be the first vertex the paths have in common after w.

Then there are two paths from w to z
that have only w and z in common.

Travelling one of these paths from w to z and then travelling the other from z to w gives us cycle, and so the graph is not a tree.



We now examine the two paths starting at w; they must join together again at y, though they might have joined at some earlier vertex.

Let z be the first vertex the paths have in common after w.

Then there are two paths from w to z
that have only w and z in common.

Travelling one of these paths from w to z and then travelling the other from z to w gives us cycle, and so the graph is not a tree.

We have shown that if a graph has two distinct paths from x to y, then it is not a tree.

By contrapositive inference, then, if a graph is a tree, it does not have two distinct paths between two vertices x and y.

By contrapositive inference, then, if a graph is a tree, it does not have two distinct paths between two vertices x and y.

Theorem 6.3
There is exactly one path between each pair of vertices in a tree.

By contrapositive inference, then, if a graph is a tree, it does not have two distinct paths between two vertices x and y.

Theorem 6.3
There is exactly one path between each pair of vertices in a tree.

Proof:

By contrapositive inference, then, if a graph is a tree, it does not have two distinct paths between two vertices x and y.

Theorem 6.3
There is exactly one path between each pair of vertices in a tree.

Proof:

By the definition of a tree, there is at least one path between each pair of vertices.

By contrapositive inference, then, if a graph is a tree, it does not have two distinct paths between two vertices x and y.

Theorem 6.3
There is exactly one path between each pair of vertices in a tree.

Proof:

By the definition of a tree, there is at least one path between each pair of vertices.

By our argument above, there is at most one path between each pair of vertices.

By contrapositive inference, then, if a graph is a tree, it does not have two distinct paths between two vertices x and y.

Theorem 6.3
There is exactly one path between each pair of vertices in a tree.

Proof:

By the definition of a tree, there is at least one path between each pair of vertices.

By our argument above, there is at most one path between each pair of vertices.

Thus, there is exactly one path.

What happens when we delete an edge from a tree?

What happens when we delete an edge from a tree?

Suppose edge e connects x to y,
$\Rightarrow x, e, y$ is the unique path from x to y in the tree.

What happens when we delete an edge from a tree?

Suppose edge e connects x to y,
$\Rightarrow x, e, y$ is the unique path from x to y in the tree.

Now, suppose we delete e from the edge set of the tree.

What happens when we delete an edge from a tree?

Suppose edge e connects x to y,
$\Rightarrow x, e, y$ is the unique path from x to y in the tree.

Now, suppose we delete e from the edge set of the tree.

If there were still a path from x to y in the resulting graph, then it would also be a path from x to y in the tree, which would contradict Theorem 6.3.

What happens when we delete an edge from a tree?

Suppose edge e connects x to y,
$\Rightarrow x, e, y$ is the unique path from x to y in the tree.

Now, suppose we delete e from the edge set of the tree.

If there were still a path from x to y in the resulting graph, then it would also be a path from x to y in the tree, which would contradict Theorem 6.3.

Thus, the only possibility is that
there is no path between x and y in the resulting graph.

What happens when we delete an edge from a tree?

Suppose edge e connects x to y,
$\Rightarrow x, e, y$ is the unique path from x to y in the tree.

Now, suppose we delete e from the edge set of the tree.

If there were still a path from x to y in the resulting graph, then it would also be a path from x to y in the tree, which would contradict Theorem 6.3.

Thus, the only possibility is that
there is no path between x and y in the resulting graph.

Thus, it is not connected and is therefore not a tree.

If $G = (V, E)$ is a graph, and we add an edge that joins vertices of V, what can happen to the number of connected components?

If $G = (V, E)$ is a graph, and we add an edge that joins vertices of V, what can happen to the number of connected components?

- If the endpoints are in the same connected component then the number of cc's will not change.

If $G = (V, E)$ is a graph, and we add an edge that joins vertices of V, what can happen to the number of connected components?

- If the endpoints are in the same connected component then the number of cc's will not change.
- If the endpoints of the edge are in different cc's, then the number of cc's will decrease by one.

Lemma 6.4
Removing one edge from the edge set of a tree gives a graph with two connected components, each of which is a tree.

Lemma 6.4
Removing one edge from the edge set of a tree gives a graph with two connected components, each of which is a tree.

Proof:

Lemma 6.4
Removing one edge from the edge set of a tree gives a graph with two connected components, each of which is a tree.

Proof:

Suppose that e is an edge from x to y in a tree.

Lemma 6.4
Removing one edge from the edge set of a tree gives a graph with two connected components, each of which is a tree.

Proof:

Suppose that e is an edge from x to y in a tree.

We have seen that the graph that we get by deleting e from the edge set of the tree is not connected.

Lemma 6.4
Removing one edge from the edge set of a tree gives a graph with two connected components, each of which is a tree.

Proof:

Suppose that e is an edge from x to y in a tree.

We have seen that the graph that we get by deleting e from the edge set of the tree is not connected.

So the graph has at least two connected components.

Lemma 6.4
Removing one edge from the edge set of a tree gives a graph with two connected components, each of which is a tree.

Proof:

Suppose that e is an edge from x to y in a tree.

We have seen that the graph that we get by deleting e from the edge set of the tree is not connected.

So the graph has at least two connected components.

But adding the edge back in can only reduce the number of connected components by one.

Lemma 6.4
Removing one edge from the edge set of a tree gives a graph with two connected components, each of which is a tree.

Proof:

Suppose that e is an edge from x to y in a tree.

We have seen that the graph that we get by deleting e from the edge set of the tree is not connected.

So the graph has at least two connected components.

But adding the edge back in can only reduce the number of connected components by one.

Therefore, the graph has exactly two connected components.

Lemma 6.4
Removing one edge from the edge set of a tree gives a graph with two connected components, each of which is a tree.

Proof:

Suppose that e is an edge from x to y in a tree.

We have seen that the graph that we get by deleting e from the edge set of the tree is not connected.

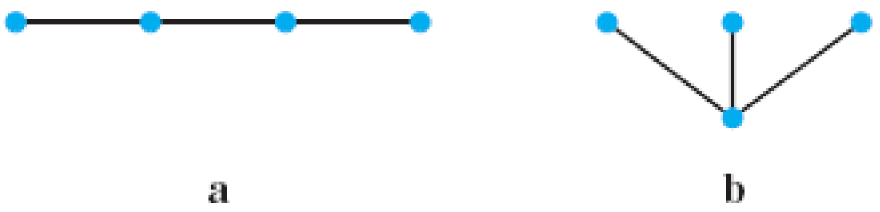
So the graph has at least two connected components.

But adding the edge back in can only reduce the number of connected components by one.

Therefore, the graph has exactly two connected components.

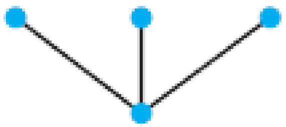
Because neither has any cycles (why?), both are trees.

Two trees on four vertices



Two trees on four vertices

a



b

Theorem 6.5

For all integers $n \geq 1$, a tree with n vertices has $n - 1$ edges.

Two trees on four vertices

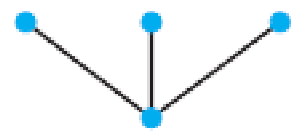
Theorem 6.5

For all integers $n \geq 1$, a tree with n vertices has $n-1$ edges.

Proof:

Two trees on four vertices

a



b

Theorem 6.5
For all integers $n \geq 1$, a tree with n vertices has $n-1$ edges.

Proof:

If a tree has 1 vertex, it can have no edges,
since any edge would have to connect that vertex to itself giving a cycle.

Two trees on four vertices

Theorem 6.5

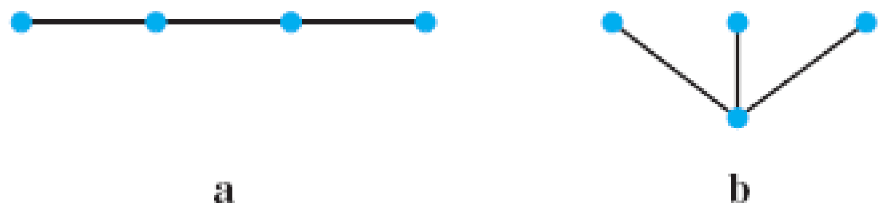
For all integers $n \geq 1$, a tree with n vertices has $n-1$ edges.

Proof:

If a tree has 1 vertex, it can have no edges, since any edge would have to connect that vertex to itself giving a cycle.

A tree with two or more vertices must, in order to be connected, contain at least one edge.

Two trees on four vertices



Theorem 6.5
For all integers $n \geq 1$, a tree with n vertices has $n-1$ edges.

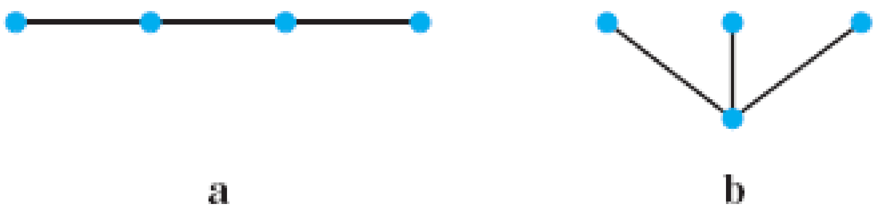
Proof:

If a tree has 1 vertex, it can have no edges, since any edge would have to connect that vertex to itself giving a cycle.

A tree with two or more vertices must, in order to be connected, contain at least one edge.

We can use the deletion of an edge $+$ Lemma 6.4 to complete an inductive proof that a tree with n vertices has $n-1$ edges.

Two trees on four vertices



Theorem 6.5

For all integers $n \geq 1$, a tree with n vertices has $n-1$ edges.

Proof:

If a tree has 1 vertex, it can have no edges, since any edge would have to connect that vertex to itself giving a cycle.

A tree with two or more vertices must, in order to be connected, contain at least one edge.

We can use the deletion of an edge $+$ Lemma 6.4 to complete an inductive proof that a tree with n vertices has $n-1$ edges.

Therefore, for all $n \geq 1$, a tree with n vertices has $n-1$ edges.

Corollary 6.6

A finite tree with more than one vertex has at least one vertex of degree 1.

Corollary 6.6
A finite tree with more than one vertex has at least one vertex of degree 1.

Proof:

We give a contrapositive argument to show that a finite tree with more than one vertex has a vertex of degree 1.

Corollary 6.6
A finite tree with more than one vertex has at least one vertex of degree 1.

Proof:

We give a contrapositive argument to show that a finite tree with more than one vertex has a vertex of degree 1.

Suppose that G is a connected graph with n vertices and that all vertices of G have degree 2 or more.

Corollary 6.6
A finite tree with more than one vertex has at least one vertex of degree 1.

Proof:

We give a contrapositive argument to show that a finite tree with more than one vertex has a vertex of degree 1.

Suppose that G is a connected graph with n vertices and that all vertices of G have degree 2 or more.

$\Rightarrow$ sum of degrees of vertices is at least $2n$.

Corollary 6.6
A finite tree with more than one vertex has at least one vertex of degree 1.

Proof:

We give a contrapositive argument to show that a finite tree with more than one vertex has a vertex of degree 1.

Suppose that G is a connected graph with n vertices and that all vertices of G have degree 2 or more.

$\Rightarrow$ sum of degrees of vertices is at least $2n$.

$\Rightarrow$ by Theorem 6.2, number of edges is at least n.

Corollary 6.6
A finite tree with more than one vertex has at least one vertex of degree 1.

Proof:

We give a contrapositive argument to show that a finite tree with more than one vertex has a vertex of degree 1.

Suppose that G is a connected graph with n vertices and that all vertices of G have degree 2 or more.

$\Rightarrow$ sum of degrees of vertices is at least $2n$.

$\Rightarrow$ by Theorem 6.2, number of edges is at least n.

$\Rightarrow$ by Theorem 6.5, G is not a tree.

Corollary 6.6
A finite tree with more than one vertex has at least one vertex of degree 1.

Proof:

We give a contrapositive argument to show that a finite tree with more than one vertex has a vertex of degree 1.

Suppose that G is a connected graph with n vertices and that all vertices of G have degree 2 or more.

$\Rightarrow$ sum of degrees of vertices is at least $2n$.

$\Rightarrow$ by Theorem 6.2, number of edges is at least n.

$\Rightarrow$ by Theorem 6.5, G is not a tree.

$\Rightarrow$ by contrapositive inference,
if T is a tree, then T must have at least one vertex of degree 1.