Extremely useful tool in modeling problems
Graphs

- Extremely useful tool in modeling problems

- Consist of:
  - Vertices
  - Edges

**Vertices** can be considered as “sites” or locations.

**Edges** represent connections.
Each vertex represents a city
Each edge represents a direct flight between two cities
A query on direct flight = a query on whether an edge exists
A query on how to get to a location = does a path exist from A to B
We can even associate costs to edges (weighted graphs), then ask "what is the cheapest path from A to B"
Each vertex represents a city
Graph Application

- Each vertex represents a city
- Each edge represents a direct flight between two cities
Each vertex represents a city
Each edge represents a direct flight between two cities
A query on direct flight = a query on whether an edge exists
Each vertex represents a city
Each edge represents a direct flight between two cities
A query on direct flight = a query on whether an edge exists
A query on how to get to a location = does a path exist from A to B
- Each vertex represents a city
- Each edge represents a direct flight between two cities
- A query on direct flight = a query on whether an edge exists
- A query on how to get to a location = does a path exist from A to B
- We can even associate costs to edges (weighted graphs), then ask “what is the cheapest path from A to B”
Graph Application

- Each vertex represents a city
- Each edge represents a direct flight between two cities
- A query on direct flight = a query on whether an edge exists
- A query on how to get to a location = does a path exist from A to B
- We can even associate costs/time to edges (weighted graphs), then ask “what is the cheapest/fastest path from A to B”
Graphs are a ubiquitous data structure in computer science.
Graphs are a ubiquitous data structure in computer science

- Networks: LAN, the Internet, wireless networks
Graphs are a ubiquitous data structure in computer science
- Networks: LAN, the Internet, wireless networks
- Logistics: transportation, supply chain management
Graphs are a ubiquitous data structure in computer science

- Networks: LAN, the Internet, wireless networks
- Logistics: transportation, supply chain management
- Relationship between objects: online dating, social networks (Facebook!)
Why Graph Algorithms?

- Graphs are a ubiquitous data structure in computer science
  - Networks: LAN, the Internet, wireless networks
  - Logistics: transportation, supply chain management
  - Relationship between objects: online dating, social networks (Facebook!)

- Hundreds of interesting computational problems defined on graphs
Why Graph Algorithms?

- Graphs are a ubiquitous data structure in computer science
  - Networks: LAN, the Internet, wireless networks
  - Logistics: transportation, supply chain management
  - Relationship between objects: online dating, social networks (Facebook!)

- Hundreds of interesting computational problems defined on graphs

- We will sample a few basic ones
A graph $G = (V, E)$ consists of
Definition

- A graph $G = (V, E)$ consists of
  - a set of vertices $V$, $|V| = n$, and
A graph $G = (V, E)$ consists of
- a set of vertices $V$, $|V| = n$, and
- a set of edges $E$, $|E| = m$
A graph $G = (V, E)$ consists of

- a set of vertices $V$, $|V| = n$, and
- a set of edges $E$, $|E| = m$

Each edge is a pair of $(u, v)$, where $u, v$ belongs to $V$
Definition

- A graph \( G = (V, E) \) consists of
  - a set of vertices \( V \), \(|V| = n\), and
  - a set of edges \( E \), \(|E| = m\)
- Each edge is a pair of \((u, v)\), where \( u, v \) belongs to \( V \)
A graph \( G = (V, E) \) consists of
- a set of vertices \( V \), \(|V| = n\), and
- a set of edges \( E \), \(|E| = m\)

Each edge is a pair of \((u, v)\), where \(u, v\) belongs to \(V\)

\[ V = \{A, B, C, H, L, M, N, P, S, T\} \]
A graph \( G = (V, E) \) consists of
- a set of vertices \( V \), \(|V| = n\), and
- a set of edges \( E \), \(|E| = m\)

Each edge is a pair of \((u, v)\), where \( u, v \) belongs to \( V \)

\[
V = \{A, B, C, H, L, M, N, P, S, T\} \\
E = \{(A, C), (A, H), \ldots, (H, P), \ldots\}
\]
A graph $G = (V, E)$ consists of
- a set of vertices $V$, $|V| = n$, and
- a set of edges $E$, $|E| = m$

Each edge is a pair of $(u, v)$, where $u, v$ belongs to $V$

$$V = \{A, B, C, H, L, M, N, P, S, T\}$$
$$E = \{(A, C), (A, H), \ldots, (H, P), \ldots\}$$

For directed graph, we distinguish between edge $(u, v)$ and edge $(v, u)$; for undirected graph, no such distinction is made.
Each edge has two endpoints.
Each edge has two endpoints. 
- $H$ and $B$ are the endpoints of $(H, B)$. 

![Diagram of a graph with vertices A, B, C, M, H, S, P, T, L, N, and edges connecting them.](image-url)
- Each edge has two endpoints
  - $H$ and $B$ are the endpoints of $(H, B)$
- An edge joins its endpoints
Each edge has two endpoints
- $H$ and $B$ are the endpoints of $(H, B)$

An edge joins its endpoints
- $(H, B)$ joins $H$ and $B$
Each edge has two endpoints
  - $H$ and $B$ are the endpoints of $(H, B)$

An edge joins its endpoints
  - $(H, B)$ joins $H$ and $B$

Two vertices are adjacent (or neighbors) if they are joined by an edge.
Each edge has two endpoints
- \( H \) and \( B \) are the endpoints of \((H, B)\)

An edge joins its endpoints
- \((H, B)\) joins \(H\) and \(B\)

Two vertices are adjacent (or neighbors) if they are joined by an edge.
- \(H\) and \(B\) are adjacent

Terminology
Each edge has two endpoints
- $H$ and $B$ are the endpoints of $(H, B)$

An edge joins its endpoints
- $(H, B)$ joins $H$ and $B$

Two vertices are adjacent (or neighbors) if they are joined by an edge.
- $H$ and $B$ are adjacent
- $H$ is a neighbor of $B$
- Each edge has two endpoints
  - $H$ and $B$ are the endpoints of $(H, B)$
- An edge joins its endpoints
  - $(H, B)$ joins $H$ and $B$
- Two vertices are adjacent (or neighbors) if they are joined by an edge.
  - $H$ and $B$ are adjacent
  - $H$ is a neighbor of $B$
- If vertex $v$ is an endpoint of edge $e$, then the edge $e$ is said to be incident on $v$. Also, the vertex $v$ is said to be incident on $e$
Each edge has two endpoints
- $H$ and $B$ are the endpoints of $(H, B)$

An edge joins its endpoints
- $(H, B)$ joins $H$ and $B$

Two vertices are adjacent (or neighbors) if they are joined by an edge.
- $H$ and $B$ are adjacent
- $H$ is a neighbor of $B$

If vertex $v$ is an endpoint of edge $e$, then the edge $e$ is said to be incident on $v$. Also, the vertex $v$ is said to be incident on $e$
- $(H, B)$ is incident on $H$ and $B$
- Each edge has two endpoints
  - $H$ and $B$ are the endpoints of $(H, B)$
- An edge joins its endpoints
  - $(H, B)$ joins $H$ and $B$
- Two vertices are adjacent (or neighbors) if they are joined by an edge.
  - $H$ and $B$ are adjacent
  - $H$ is a neighbor of $B$
- If vertex $v$ is an endpoint of edge $e$, then the edge $e$ is said to be incident on $v$. Also, the vertex $v$ is said to be incident on $e$
  - $(H, B)$ is incident on $H$ and $B$
  - $H$ and $B$ are incident on $(H, B)$
The degree of a vertex \( v \) \((\text{degree}(v))\) in a graph is the number of edges incident on it.
The degree of a vertex $v$ (degree($v$)) in a graph is the number of edges incident on it.
The degree of a vertex $v$ (degree($v$)) in a graph is the number of edges incident on it.

- Vertex $H$ has degree 5
The degree of a vertex $v$ (degree$(v)$) in a graph is the number of edges incident on it.

- Vertex $H$ has degree 5
- Vertex $N$ has degree 1

\[
\sum_{v \in V} \text{degree}(v) = 2|E|.
\]

**Proof.** An edge $e = (u, v)$ in a graph contributes one to degree$(u)$ and contributes one to degree$(v)$.
The degree of a vertex $v$ (degree$(v)$) in a graph is the number of edges incident on it.

- Vertex $H$ has degree 5
- Vertex $N$ has degree 1
- Vertex $X$ has degree 0
The degree of a vertex \( v \) (\( \text{degree}(v) \)) in a graph is the number of edges incident on it.

- Vertex \( H \) has degree 5
- Vertex \( N \) has degree 1
- Vertex \( X \) has degree 0

(It is called an isolated vertex)
The degree of a vertex \( v \) (\( \text{degree}(v) \)) in a graph is the number of edges incident on it.

- Vertex \( H \) has degree 5
- Vertex \( N \) has degree 1
- Vertex \( X \) has degree 0

(It is called an isolated vertex)

**Lemma**

\[
\sum_{v \in V} \text{degree}(v) = 2|E|.
\]
The degree of a vertex $v$ (\text{degree}(v)) in a graph is the number of edges incident on it.

- Vertex $H$ has degree 5
- Vertex $N$ has degree 1
- Vertex $X$ has degree 0
  (It is called an \textit{isolated} vertex)

**Lemma**

$$\sum_{v \in V} \text{degree}(v) = 2|E|.$$ 

**Proof.**

An edge $e = (u, v)$ in a graph contributes one to \text{degree}(u) and contributes one to \text{degree}(v).
A path in a graph is a sequence \( \langle v_0, v_1, v_2, \ldots, v_k \rangle \) of vertices such that \((v_{i-1}, v_i) \in E\) for \(i = 1, 2, \ldots, k\).
A path in a graph is a sequence $\langle v_0, v_1, v_2, \ldots, v_k \rangle$ of vertices such that $(v_{i-1}, v_i) \in E$ for $i = 1, 2, \ldots, k$

- There is a path from $v_0$ to $v_k$
- **Length** of a path = # of edges on the path
- **Path contains** the vertices $v_0, v_1, \ldots, v_k$ and the edges $(v_0, v_1), (v_1, v_2), \ldots, (v_{k-1}, v_k)$
- For any $0 \leq i \leq j \leq k$, $\langle v_i, v_{i+1}, \ldots, v_j \rangle$ is its subpath
- If there is a path $p$ from $u$ to $v$, $v$ is said to be **reachable** from $u$
- A path is **simple** if all vertices in the path are distinct
A path $\langle v_0, v_1, v_2, \ldots, v_k \rangle$ forms a **cycle** if $v_0 = v_k$ and all edges on the path are distinct.
A path $\langle v_0, v_1, v_2, \ldots, v_k \rangle$ forms a cycle if $v_0 = v_k$ and all edges on the path are distinct.

- A cycle is simple if $v_1, v_2, \ldots, v_k$ are distinct.
A path $\langle v_0, v_1, v_2, \ldots, v_k \rangle$ forms a cycle if $v_0 = v_k$ and all edges on the path are distinct

- A cycle is simple if $v_1, v_2, \ldots, v_k$ are distinct
- A graph with no cycles is acyclic
A path \( \langle v_0, v_1, v_2, \ldots, v_k \rangle \) forms a cycle if \( v_0 = v_k \) and all edges on the path are distinct.

- A cycle is simple if \( v_1, v_2, \ldots, v_k \) are distinct.
- A graph with no cycles is acyclic.
A path \( \langle v_0, v_1, v_2, \ldots, v_k \rangle \) forms a cycle if \( v_0 = v_k \) and all edges on the path are distinct.

- A cycle is simple if \( v_1, v_2, \ldots, v_k \) are distinct.
- A graph with no cycles is acyclic.

\[ \langle T, S, H, T \rangle \] is a simple cycle.
A path \( \langle v_0, v_1, v_2, \ldots, v_k \rangle \) forms a cycle if \( v_0 = v_k \) and all edges on the path are distinct.

- A cycle is simple if \( v_1, v_2, \ldots, v_k \) are distinct.
- A graph with no cycles is acyclic.

\[ \langle T, S, H, T \rangle \] is a simple cycle.

\[ \langle A, C, L, P, H, A \rangle \] is a simple cycle.
Connectivity

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

- Two vertices are **connected** if there is a path between them.
- A graph is **connected** if every pair of vertices is connected; otherwise, the graph is **disconnected**.
Connectivity

- Two vertices are **connected** if there is a path between them.
- A graph is **connected** if every pair of vertices is connected; otherwise, the graph is **disconnected**.
- The **connected components** of a graph are the equivalence classes of vertices under the “is reachable from” relation.
Connectivity

- Two vertices are **connected** if there is a path between them.
- A graph is **connected** if every pair of vertices is connected; otherwise, the graph is **disconnected**.
- The **connected components** of a graph are the equivalence classes of vertices under the “is reachable from” relation.

- **Connected graph**
  - one connected component
    \{A, B, C, H, L, M, N, P, S, T\}

- **Disconnected graph**
  - 3 connected components
    - \{1, 2, 5\}
    - \{3, 6\}
    - \{4\}
Graph $G' = (V', E')$ is a subgraph of $G = (V, E)$ if $V' \subseteq V$ and $E' \subseteq E$. 

A subgraph is an induced subgraph if every edge of $G$ connecting vertices of $G'$ is also an edge of $G'$. 

Version of October 11, 2014

Introduction to Graph Algorithms
Graph $G' = (V', E')$ is a subgraph of $G = (V, E)$ if $V' \subseteq V$ and $E' \subseteq E$

$G'$ is an induced subgraph of $G$ if $G'$ is a subgraph of $G$ and every edge of $G$ connecting vertices of $G'$ is an edge of $G'$. 
Graph $G' = (V', E')$ is a subgraph of $G = (V, E)$ if $V' \subseteq V$ and $E' \subseteq E$.

$G'$ is an induced subgraph of $G$ if $G'$ is a subgraph of $G$ and every edge of $G$ connecting vertices of $G'$ is an edge of $G'$. 
A tree is a connected, acyclic, undirected graph
- A **tree** is a connected, acyclic, undirected graph.

- If an undirected graph is acyclic but possibly disconnected, it is a **forest**.
• A **tree** is a connected, acyclic, undirected graph

• If an undirected graph is acyclic but possibly disconnected, it is a **forest**
Properties of Trees

Let $G = (V, E)$ be an undirected graph. The following statements are equivalent.

1. $G$ is a tree
2. Any two vertices in $G$ are connected by a unique simple path
3. $G$ is connected, but if any edge is removed from $E$, the resulting graph is disconnected
4. $G$ is connected, and $|E| = |V| - 1$
5. $G$ is acyclic, and $|E| = |V| - 1$
6. $G$ is acyclic, but if any edge is added to $E$, the resulting graph contains a cycle
Let $G = (V, E)$ be an undirected graph. The following statements are equivalent.

1. $G$ is a tree
Let $G = (V, E)$ be an undirected graph. The following statements are equivalent.

1. $G$ is a tree
2. Any two vertices in $G$ are connected by a unique simple path
Let $G = (V, E)$ be an undirected graph. The following statements are equivalent.

1. $G$ is a tree
2. Any two vertices in $G$ are connected by a unique simple path
3. $G$ is connected, but if any edge is removed from $E$, the resulting graph is disconnected
Let $G = (V, E)$ be an undirected graph. The following statements are equivalent.

1. $G$ is a tree
2. Any two vertices in $G$ are connected by a unique simple path
3. $G$ is connected, but if any edge is removed from $E$, the resulting graph is disconnected
4. $G$ is connected, and $|E| = |V| - 1$
Properties of Trees

Let $G = (V, E)$ be an undirected graph. The following statements are equivalent.

1. $G$ is a tree
2. Any two vertices in $G$ are connected by a unique simple path
3. $G$ is connected, but if any edge is removed from $E$, the resulting graph is disconnected
4. $G$ is connected, and $|E| = |V| - 1$
5. $G$ is acyclic, and $|E| = |V| - 1$
Let $G = (V, E)$ be an undirected graph. The following statements are equivalent.

1. $G$ is a tree
2. Any two vertices in $G$ are connected by a unique simple path
3. $G$ is connected, but if any edge is removed from $E$, the resulting graph is disconnected
4. $G$ is connected, and $|E| = |V| - 1$
5. $G$ is acyclic, and $|E| = |V| - 1$
6. $G$ is acyclic, but if any edge is added to $E$, the resulting graph contains a cycle
Proof

(1) $G$ is a tree
⇒ (2) Any two vertices in $G$ are connected by a unique simple path
Proof

(1) $G$ is a tree

$\Rightarrow$ (2) Any two vertices in $G$ are connected by a unique simple path

- Proof by contradiction

- Suppose that vertices $u$ and $v$ are connected by two distinct simple paths $p_1$ and $p_2$, as shown in the above figure.
Proof

(1) \( G \) is a tree
\[ \Rightarrow \quad (2) \text{Any two vertices in } G \text{ are connected by a unique simple path} \]

- Proof by contradiction
- Suppose that vertices \( u \) and \( v \) are connected by two distinct simple paths \( p_1 \) and \( p_2 \), as shown in the above figure
  - \( p_1 \) and \( p_2 \) first diverge at vertex \( w \)
  - \( p_1 \) and \( p_2 \) first reconverge at vertex \( z \)
Proof

(1) $G$ is a tree

$\Rightarrow$ (2) Any two vertices in $G$ are connected by a unique simple path

Proof by contradiction

Suppose that vertices $u$ and $v$ are connected by two distinct simple paths $p_1$ and $p_2$, as shown in the above figure

- $p_1$ and $p_2$ first diverge at vertex $w$
- $p_1$ and $p_2$ first reconverge at vertex $z$
- $p'$ is the subpath of $p_1$ from $w$ through $x$ to $z$
- $p''$ is the subpath of $p_2$ from $w$ through $y$ to $z$
(1) $G$ is a tree
\[ \Rightarrow (2) \text{Any two vertices in } G \text{ are connected by a unique simple path} \]

Proof by contradiction

Suppose that vertices $u$ and $v$ are connected by two distinct simple paths $p_1$ and $p_2$, as shown in the above figure:

- $p_1$ and $p_2$ first diverge at vertex $w$
- $p_1$ and $p_2$ first reconverge at vertex $z$
- $p'$ is the subpath of $p_1$ from $w$ through $x$ to $z$
- $p''$ is the subpath of $p_2$ from $w$ through $y$ to $z$
- The path obtained by concatenating $p'$ and the reverse of $p''$ is a cycle, which yields the contradiction!
Proof

(2) Any two vertices in \( G \) are connected by a unique simple path
\( \Rightarrow \) (3) \( G \) is connected, but if any edge is removed from \( E \), the resulting graph is disconnected
Proof

(2) Any two vertices in $G$ are connected by a unique simple path
⇒ (3) $G$ is connected, but if any edge is removed from $E$, the resulting graph is disconnected

- If any two vertices in $G$ are connected by a unique simple path, then $G$ is connected
Proof

(2) Any two vertices in $G$ are connected by a unique simple path

⇒ (3) $G$ is connected, but if any edge is removed from $E$, the resulting graph is disconnected

• If any two vertices in $G$ are connected by a unique simple path, then $G$ is connected

• Let $(u, v)$ be any edge in $E$
Proof

(2) Any two vertices in $G$ are connected by a unique simple path

$\Rightarrow$ (3) $G$ is connected, but if any edge is removed from $E$, the resulting graph is disconnected

- If any two vertices in $G$ are connected by a unique simple path, then $G$ is connected
- Let $(u, v)$ be any edge in $E$
- This edge is a path from $u$ to $v$, and so it must be the unique path from $u$ to $v$
Proof

(2) Any two vertices in $G$ are connected by a unique simple path
⇒ (3) $G$ is connected, but if any edge is removed from $E$, the resulting graph is disconnected

- If any two vertices in $G$ are connected by a unique simple path, then $G$ is connected
- Let $(u, v)$ be any edge in $E$
- This edge is a path from $u$ to $v$, and so it must be the unique path from $u$ to $v$
- If $(u, v)$ is deleted from $G$, there is no path from $u$ to $v$, and hence its removal disconnects $G$
Proof

(3) $G$ is connected, but if any edge is removed from $E$, the resulting graph is disconnected

$\Rightarrow$ (4) $G$ is connected, and $|E| = |V| - 1$
Proof

(3) $G$ is connected, but if any edge is removed from $E$, the resulting graph is disconnected

$\Rightarrow$ (4) $G$ is connected, and $|E| = |V| - 1$

- By assumption, the graph $G$ is connected
Proof

(3) $G$ is connected, but if any edge is removed from $E$, the resulting graph is disconnected

$\Rightarrow$ (4) $G$ is connected, and $|E| = |V| - 1$

- By assumption, the graph $G$ is connected
- Prove $|E| = |V| - 1$ by induction
Proof

(3) $G$ is connected, but if any edge is removed from $E$, the resulting graph is disconnected

$\Rightarrow$ (4) $G$ is connected, and $|E| = |V| - 1$

- By assumption, the graph $G$ is connected
- Prove $|E| = |V| - 1$ by induction
  - Base ($n = 1$): A connected graph with one vertex has zero edge
Proof

(3) \( G \) is connected, but if any edge is removed from \( E \), the resulting graph is disconnected
\[ \Rightarrow (4) \text{ } G \text{ is connected, and } |E| = |V| - 1 \]

- By assumption, the graph \( G \) is connected
- Prove \( |E| = |V| - 1 \) by induction
  - Base \( (n = 1) \): A connected graph with one vertex has zero edge
  - Suppose that \( G \) has \( n \geq 2 \) vertices and that all graphs satisfying (3) with fewer than \( n \) vertices also satisfy \( |E| = |V| - 1 \)
Proof

(3) \( G \) is connected, but if any edge is removed from \( E \), the resulting graph is disconnected

\[ \Rightarrow (4) \; G \text{ is connected, and } |E| = |V| - 1 \]

- By assumption, the graph \( G \) is connected
- Prove \( |E| = |V| - 1 \) by induction
  - Base \( (n = 1) \): A connected graph with one vertex has zero edge
  - Suppose that \( G \) has \( n \geq 2 \) vertices and that all graphs satisfying (3) with fewer than \( n \) vertices also satisfy \( |E| = |V| - 1 \)
  - Removing an arbitrary edge from \( G \) separates the graph into 2 connected components
Proof

(3) $G$ is connected, but if any edge is removed from $E$, the resulting graph is disconnected
$\Rightarrow$ (4) $G$ is connected, and $|E| = |V| - 1$

- By assumption, the graph $G$ is connected
- Prove $|E| = |V| - 1$ by induction
  - Base ($n = 1$): A connected graph with one vertex has zero edge
  - Suppose that $G$ has $n \geq 2$ vertices and that all graphs satisfying (3) with fewer than $n$ vertices also satisfy $|E| = |V| - 1$
  - Removing an arbitrary edge from $G$ separates the graph into 2 connected components
  - Each component satisfies (3), or else $G$ would not satisfy (3)
Proof

(3) $G$ is connected, but if any edge is removed from $E$, the resulting graph is disconnected

$\Rightarrow$ (4) $G$ is connected, and $|E| = |V| - 1$

- By assumption, the graph $G$ is connected
- Prove $|E| = |V| - 1$ by induction
  - Base ($n = 1$): A connected graph with one vertex has zero edge
  - Suppose that $G$ has $n \geq 2$ vertices and that all graphs satisfying (3) with fewer than $n$ vertices also satisfy $|E| = |V| - 1$
  - Removing an arbitrary edge from $G$ separates the graph into 2 connected components
  - Each component satisfies (3), or else $G$ would not satisfy (3)
  - Thus, by induction, the number of edges in 2 components combined is $|V| - 2$
Proof

(3) $G$ is connected, but if any edge is removed from $E$, the resulting graph is disconnected

$\Rightarrow$ (4) $G$ is connected, and $|E| = |V| - 1$

- By assumption, the graph $G$ is connected
- Prove $|E| = |V| - 1$ by induction
  - Base ($n = 1$): A connected graph with one vertex has zero edge
  - Suppose that $G$ has $n \geq 2$ vertices and that all graphs satisfying (3) with fewer than $n$ vertices also satisfy $|E| = |V| - 1$
  - Removing an arbitrary edge from $G$ separates the graph into 2 connected components
  - Each component satisfies (3), or else $G$ would not satisfy (3)
  - Thus, by induction, the number of edges in 2 components combined is $|V| - 2$
  - Adding in the removed edge yields $|E| = |V| - 1$
Proof

(4) $G$ is connected, and $|E| = |V| - 1$
⇒ (5) $G$ is acyclic, and $|E| = |V| - 1$
Proof

(4) $G$ is connected, and $|E| = |V| - 1$

$\Rightarrow$ (5) $G$ is acyclic, and $|E| = |V| - 1$

- Suppose that $G$ is connected and that $|E| = |V| - 1$
- Suppose that $G$ has a cycle containing $k$ vertices $v_1, v_2, \ldots, v_k$, and without loss of generality assume that this cycle is simple
Proof

(4) $G$ is connected, and $|E| = |V| - 1$

$\Rightarrow$ (5) $G$ is acyclic, and $|E| = |V| - 1$

- Suppose that $G$ is connected and that $|E| = |V| - 1$
- Suppose that $G$ has a cycle containing $k$ vertices $v_1, v_2, \ldots, v_k$, and without loss of generality assume that this cycle is simple
- Let $G_k = (V_k, E_k)$ be the subgraph of $G$ consisting of the cycle
- Note that $|V_k| = |E_k| = k$
Proof

(4) $G$ is connected, and $|E| = |V| - 1$

$\Rightarrow$ (5) $G$ is acyclic, and $|E| = |V| - 1$

- Suppose that $G$ is connected and that $|E| = |V| - 1$
- Suppose that $G$ has a cycle containing $k$ vertices $v_1, v_2, \ldots, v_k$, and without loss of generality assume that this cycle is simple
- Let $G_k = (V_k, E_k)$ be the subgraph of $G$ consisting of the cycle
- Note that $|V_k| = |E_k| = k$
- If $k < |V|$, there must be a vertex $v_{k+1} \in V - V_k$ that is adjacent to some vertex $v_i \in V_k$, since $G$ is connected
(4) $G$ is connected, and $|E| = |V| - 1$

$\Rightarrow$ (5) $G$ is acyclic, and $|E| = |V| - 1$

- Suppose that $G$ is connected and that $|E| = |V| - 1$
- Suppose that $G$ has a cycle containing $k$ vertices $v_1, v_2, \ldots, v_k$, and without loss of generality assume that this cycle is simple
- Let $G_k = (V_k, E_k)$ be the subgraph of $G$ consisting of the cycle
- Note that $|V_k| = |E_k| = k$
- If $k < |V|$, there must be a vertex $v_{k+1} \in V - V_k$ that is adjacent to some vertex $v_i \in V_k$, since $G$ is connected
- Define $G_{k+1} = (V_{k+1}, E_{k+1})$ to be the subgraph of $G$ with $V_{k+1} = V_k \cup \{v_{k+1}\}$ and $E_{k+1} = E_k \cup \{(v_i, v_{k+1})\}$
Proof

(4) \( G \) is connected, and \( |E| = |V| - 1 \)

\[ \Rightarrow (5) \ G \text{ is acyclic, and } |E| = |V| - 1 \]

- Suppose that \( G \) is connected and that \( |E| = |V| - 1 \)
- Suppose that \( G \) has a cycle containing \( k \) vertices \( v_1, v_2, \ldots, v_k \), and without loss of generality assume that this cycle is simple
- Let \( G_k = (V_k, E_k) \) be the subgraph of \( G \) consisting of the cycle
- Note that \( |V_k| = |E_k| = k \)
- If \( k < |V| \), there must be a vertex \( v_{k+1} \in V - V_k \) that is adjacent to some vertex \( v_i \in V_k \), since \( G \) is connected
- Define \( G_{k+1} = (V_{k+1}, E_{k+1}) \) to be the subgraph of \( G \) with \( V_{k+1} = V_k \cup \{v_{k+1}\} \) and \( E_{k+1} = E_k \cup \{(v_i, v_{k+1})\} \)
- Note that \( |V_{k+1}| = |E_{k+1}| = k + 1 \)
Proof

(4) $G$ is connected, and $|E| = |V| - 1$

$\Rightarrow$ (5) $G$ is acyclic, and $|E| = |V| - 1$

- Suppose that $G$ is connected and that $|E| = |V| - 1$
- Suppose that $G$ has a cycle containing $k$ vertices $v_1, v_2, \ldots, v_k$, and without loss of generality assume that this cycle is simple
- Let $G_k = (V_k, E_k)$ be the subgraph of $G$ consisting of the cycle
- Note that $|V_k| = |E_k| = k$
- If $k < |V|$, there must be a vertex $v_{k+1} \in V - V_k$ that is adjacent to some vertex $v_i \in V_k$, since $G$ is connected
- Define $G_{k+1} = (V_{k+1}, E_{k+1})$ to be the subgraph of $G$ with $V_{k+1} = V_k \cup \{v_{k+1}\}$ and $E_{k+1} = E_k \cup \{(v_i, v_{k+1})\}$
- Note that $|V_{k+1}| = |E_{k+1}| = k + 1$
- If $k + 1 < |V|$, we can continue, defining $G_{k+2}$ in the same manner, and so forth, until we obtain $G_n = (V_n, E_n)$, where $n = |V|, V_n = V$, and $|E_n| = |V_n| = |V|$
(4) $G$ is connected, and $|E| = |V| - 1$
$\Rightarrow$ (5) $G$ is acyclic, and $|E| = |V| - 1$

- Suppose that $G$ is connected and that $|E| = |V| - 1$
- Suppose that $G$ has a cycle containing $k$ vertices $v_1, v_2, \ldots, v_k$, and without loss of generality assume that this cycle is simple.
- Let $G_k = (V_k, E_k)$ be the subgraph of $G$ consisting of the cycle.
- Note that $|V_k| = |E_k| = k$
- If $k < |V|$, there must be a vertex $v_{k+1} \in V - V_k$ that is adjacent to some vertex $v_i \in V_k$, since $G$ is connected.
- Define $G_{k+1} = (V_{k+1}, E_{k+1})$ to be the subgraph of $G$ with $V_{k+1} = V_k \cup \{v_{k+1}\}$ and $E_{k+1} = E_k \cup \{(v_i, v_{k+1})\}$
- Note that $|V_{k+1}| = |E_{k+1}| = k + 1$
- If $k + 1 < |V|$, we can continue, defining $G_{k+2}$ in the same manner, and so forth, until we obtain $G_n = (V_n, E_n)$, where $n = |V|$, $V_n = V$, and $|E_n| = |V_n| = |V|$
- Since $G_n$ is a subgraph of $G$, we have $E_n \subseteq E$, and hence $|E| \geq |V|$, which contradicts the assumption that $|E| = |V| - 1$
Proof

(5) $G$ is acyclic, and $|E| = |V| - 1$

$\Rightarrow$ (6) $G$ is acyclic, but if any edge is added to $E$, the resulting graph contains a cycle
Proof

(5) \( G \) is acyclic, and \(|E| = |V| - 1\)

\( \Rightarrow \) (6) \( G \) is acyclic, but if any edge is added to \( E \), the resulting graph contains a cycle

- Suppose that \( G \) is acyclic and that \(|E| = |V| - 1\)
Proof

(5) $G$ is acyclic, and $|E| = |V| - 1$

$\Rightarrow$ (6) $G$ is acyclic, but if any edge is added to $E$, the resulting graph contains a cycle

- Suppose that $G$ is acyclic and that $|E| = |V| - 1$
- Let $k$ be the number of connected components of $G$
Proof

(5) $G$ is acyclic, and $|E| = |V| - 1$

$\Rightarrow$ (6) $G$ is acyclic, but if any edge is added to $E$, the resulting graph contains a cycle

- Suppose that $G$ is acyclic and that $|E| = |V| - 1$
- Let $k$ be the number of connected components of $G$
- Each connected component is a free tree by definition, and since (1) implies (5), the sum of all edges in all connected components of $G$ is $|V| - k$
(5) $G$ is acyclic, and $|E| = |V| - 1$

$\Rightarrow$ (6) $G$ is acyclic, but if any edge is added to $E$, the resulting graph contains a cycle

- Suppose that $G$ is acyclic and that $|E| = |V| - 1$
- Let $k$ be the number of connected components of $G$
- Each connected component is a free tree by definition, and since (1) implies (5), the sum of all edges in all connected components of $G$ is $|V| - k$
- Consequently, we must have $k = 1$, and $G$ is in fact a tree (That is, (1) holds)
Proof

(5) \( G \) is acyclic, and \( |E| = |V| - 1 \)
⇒ (6) \( G \) is acyclic, but if any edge is added to \( E \), the resulting graph contains a cycle

- Suppose that \( G \) is acyclic and that \( |E| = |V| - 1 \)
- Let \( k \) be the number of connected components of \( G \)
- Each connected component is a free tree by definition, and since (1) implies (5), the sum of all edges in all connected components of \( G \) is \( |V| - k \)
- Consequently, we must have \( k = 1 \), and \( G \) is in fact a tree (That is, (1) holds)
- Since (1) implies (2), any two vertices in \( G \) are connected by a unique simple path
(5) \( G \) is acyclic, and \(|E| = |V| - 1\)
\[\Rightarrow (6) \text{ } G \text{ is acyclic, but if any edge is added to } E, \text{ the resulting graph contains a cycle} \]

- Suppose that \( G \) is acyclic and that \(|E| = |V| - 1\)
- Let \( k \) be the number of connected components of \( G \)
- Each connected component is a free tree by definition, and since (1) implies (5), the sum of all edges in all connected components of \( G \) is \(|V| - k\)
- Consequently, we must have \( k = 1 \), and \( G \) is in fact a tree (That is, (1) holds)
- Since (1) implies (2), any two vertices in \( G \) are connected by a unique simple path
- Thus, adding any edge to \( G \) creates a cycle
Proof

(6) \( G \) is acyclic, but if any edge is added to \( E \), the resulting graph contains a cycle
\[ \Rightarrow (1) \ G \text{ is a tree} \]
(6) $G$ is acyclic, but if any edge is added to $E$, the resulting graph contains a cycle

$\Rightarrow$ (1) $G$ is a tree

- Suppose that $G$ is acyclic but that if any edge is added to $E$, a cycle is created
(6) $G$ is acyclic, but if any edge is added to $E$, the resulting graph contains a cycle

$\Rightarrow$ (1) $G$ is a tree

- Suppose that $G$ is acyclic but that if any edge is added to $E$, a cycle is created
- We must show that $G$ is connected
(6) $G$ is acyclic, but if any edge is added to $E$, the resulting graph contains a cycle
\[ \Rightarrow \] (1) $G$ is a tree

- Suppose that $G$ is acyclic but that if any edge is added to $E$, a cycle is created
- We must show that $G$ is connected
- Let $u$ and $v$ be arbitrary vertices in $G$
Proof

(6) $G$ is acyclic, but if any edge is added to $E$, the resulting graph contains a cycle
$\Rightarrow$ (1) $G$ is a tree

- Suppose that $G$ is acyclic but that if any edge is added to $E$, a cycle is created
- We must show that $G$ is connected
- Let $u$ and $v$ be arbitrary vertices in $G$
- If $u$ and $v$ are not already adjacent, adding the edge $(u, v)$ creates a cycle in which all edges but $(u, v)$ belong to $G$
(6) $G$ is acyclic, but if any edge is added to $E$, the resulting graph contains a cycle

$\Rightarrow$ (1) $G$ is a tree

- Suppose that $G$ is acyclic but that if any edge is added to $E$, a cycle is created
- We must show that $G$ is connected
- Let $u$ and $v$ be arbitrary vertices in $G$
- If $u$ and $v$ are not already adjacent, adding the edge $(u, v)$ creates a cycle in which all edges but $(u, v)$ belong to $G$
- Thus, there is a path from $u$ to $v$, and since $u$ and $v$ were chosen arbitrarily, $G$ is connected