

CS103
FALL 2025



Lecture 10: **Graph Theory**

Part 2 of 3

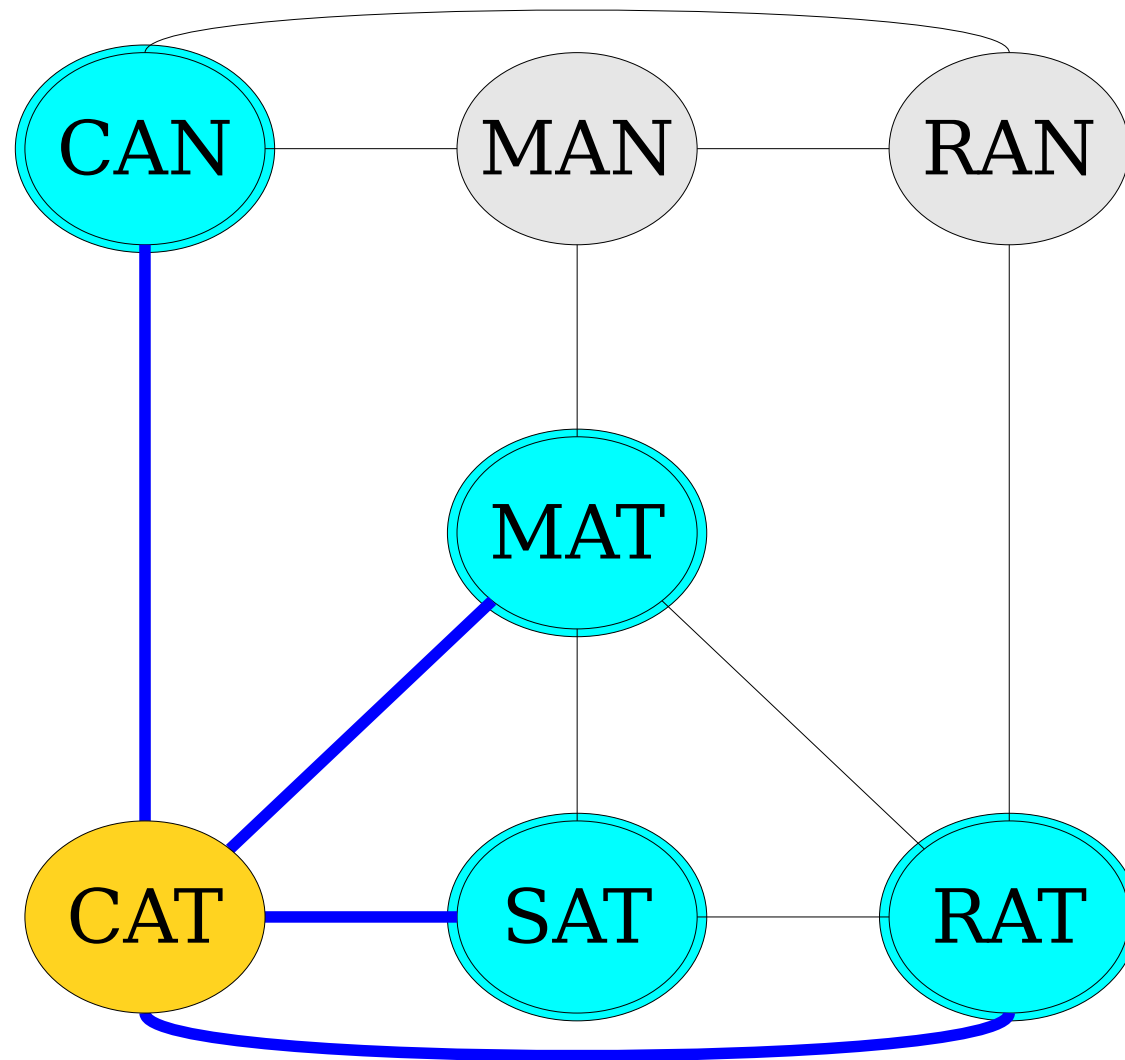
Outline for Today

- ***Walks, Paths, and Reachability***
 - Walking around a graph.
- ***Application: Local Area Networks***
 - Graphs meet computer networking.
- ***Trees***
 - A fundamental class of graphs.

Recap from Last Time

Graphs and Digraphs

- A **graph** is a pair $G = (V, E)$ of a set of nodes V and set of edges E .
 - Nodes can be anything.
 - Edges are **unordered pairs** of nodes. If $\{u, v\} \in E$, then there's an edge from u to v .
- A **digraph** is a pair $G = (V, E)$ of a set of nodes V and set of directed edges E .
 - Each edge is represented as the ordered pair (u, v) indicating an edge from u to v .



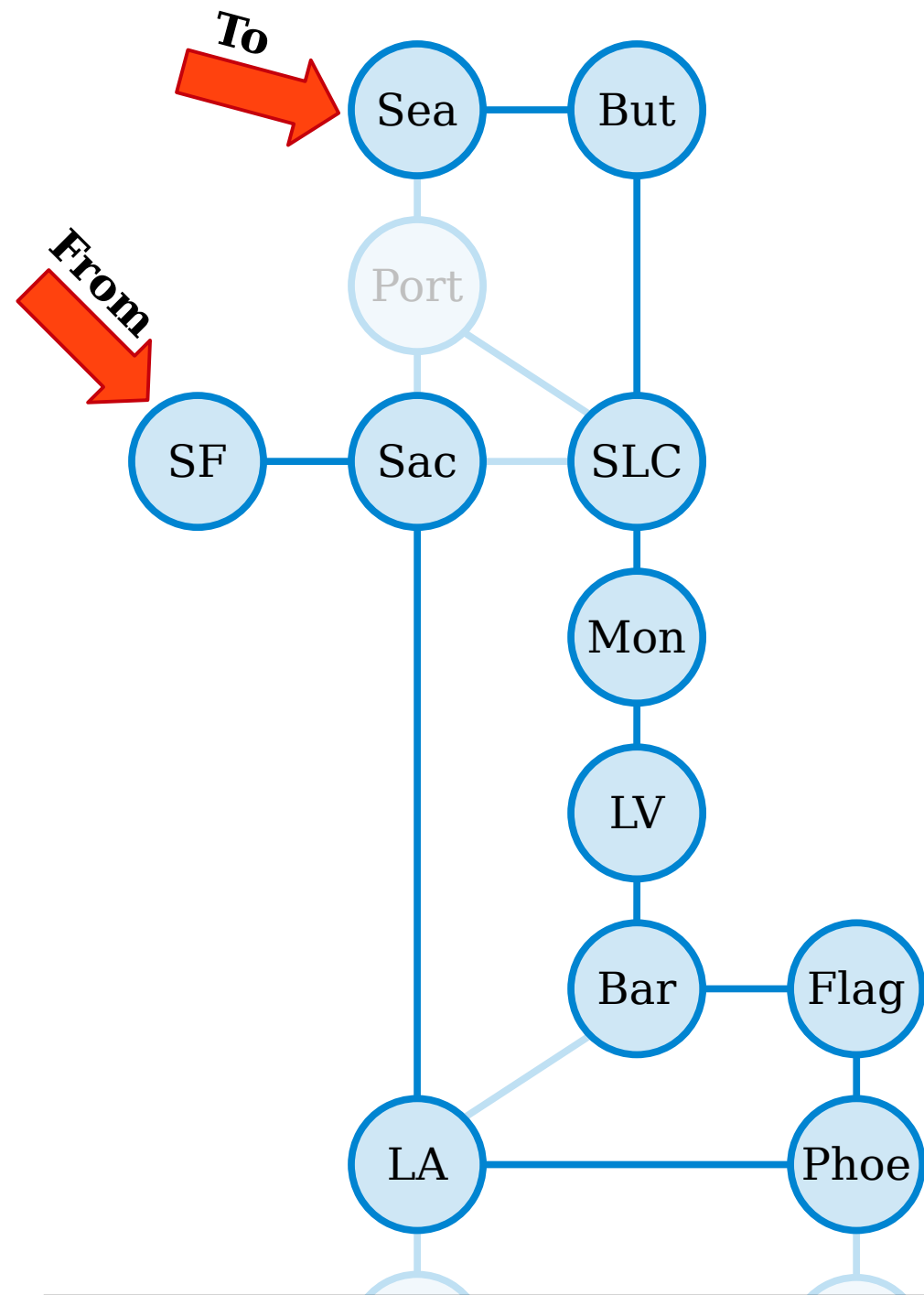
Two nodes in an undirected graph are called ***adjacent*** if there is an edge between them.

Using our Formalisms

- Let $G = (V, E)$ be an (undirected) graph.
- Intuitively, two nodes are adjacent if they're linked by an edge.
- Formally speaking, we say that two nodes $u, v \in V$ are **adjacent** if we have $\{u, v\} \in E$.
- There isn't an analogous notion for directed graphs. We usually just say “there's an edge from u to v ” as a way of reading $(u, v) \in E$ aloud.

New Stuff!

Walks, Paths, and Reachability

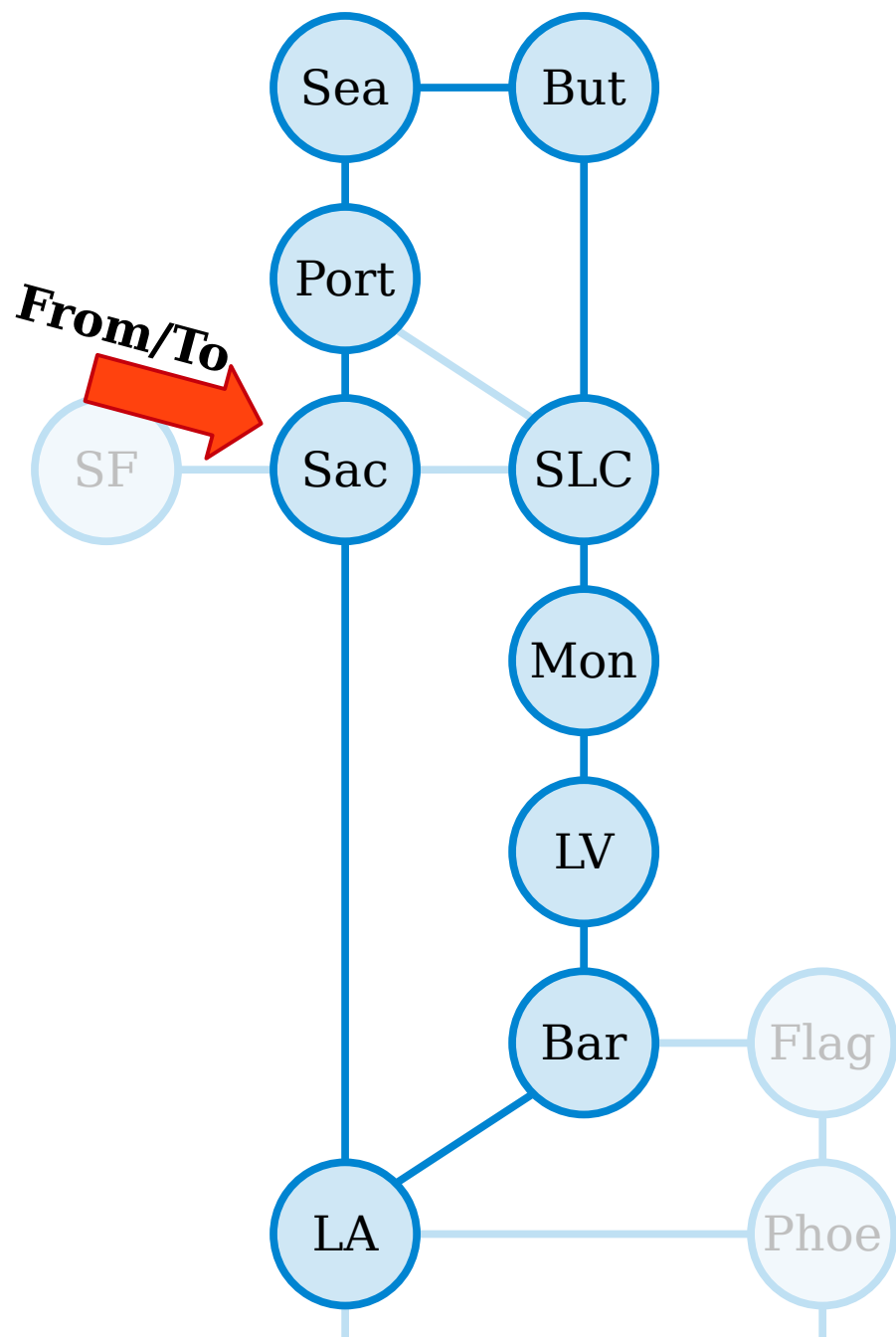


A **walk** in a graph $G = (V, E)$ is a sequence of one or more nodes $v_1, v_2, v_3, \dots, v_n$ such that any two consecutive nodes in the sequence are adjacent.

The **length** of the walk v_1, \dots, v_n is $n - 1$.

(This walk has length 10, but visits 11 cities.)

SF, Sac, LA, Phoe, Flag, Bar, LV, Mon, SLC, But, Sea



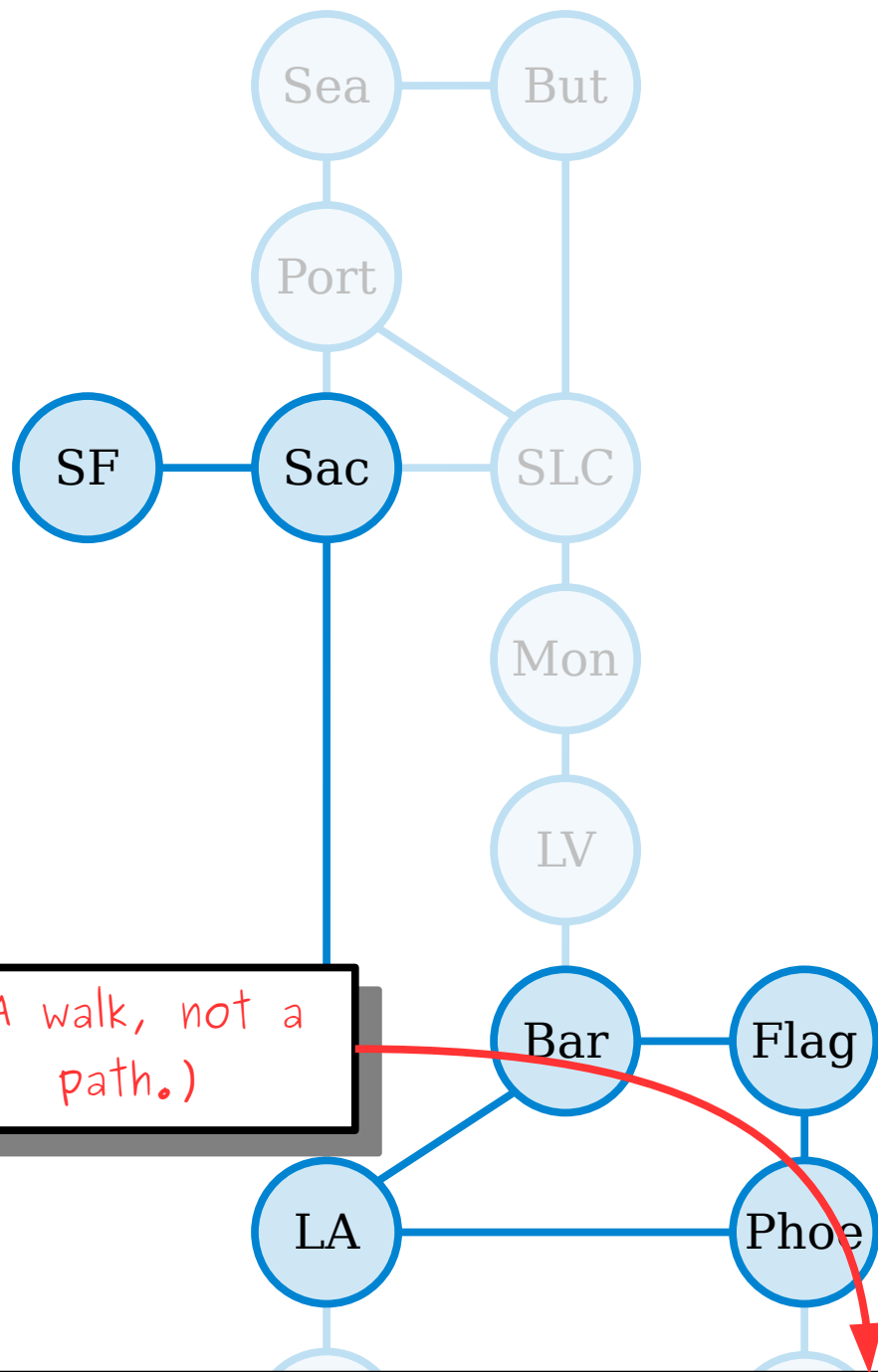
A **walk** in a graph $G = (V, E)$ is a sequence of one or more nodes $v_1, v_2, v_3, \dots, v_n$ such that any two consecutive nodes in the sequence are adjacent.

The **length** of the walk v_1, \dots, v_n is $n - 1$.

A **closed walk** in a graph is a walk from a node back to itself. (By convention, a closed walk cannot have length zero.)

(This closed walk has length nine and visits nine different cities.)

Sac, Port, Sea, But, SLC, Mon, LV, Bar, LA, Sac



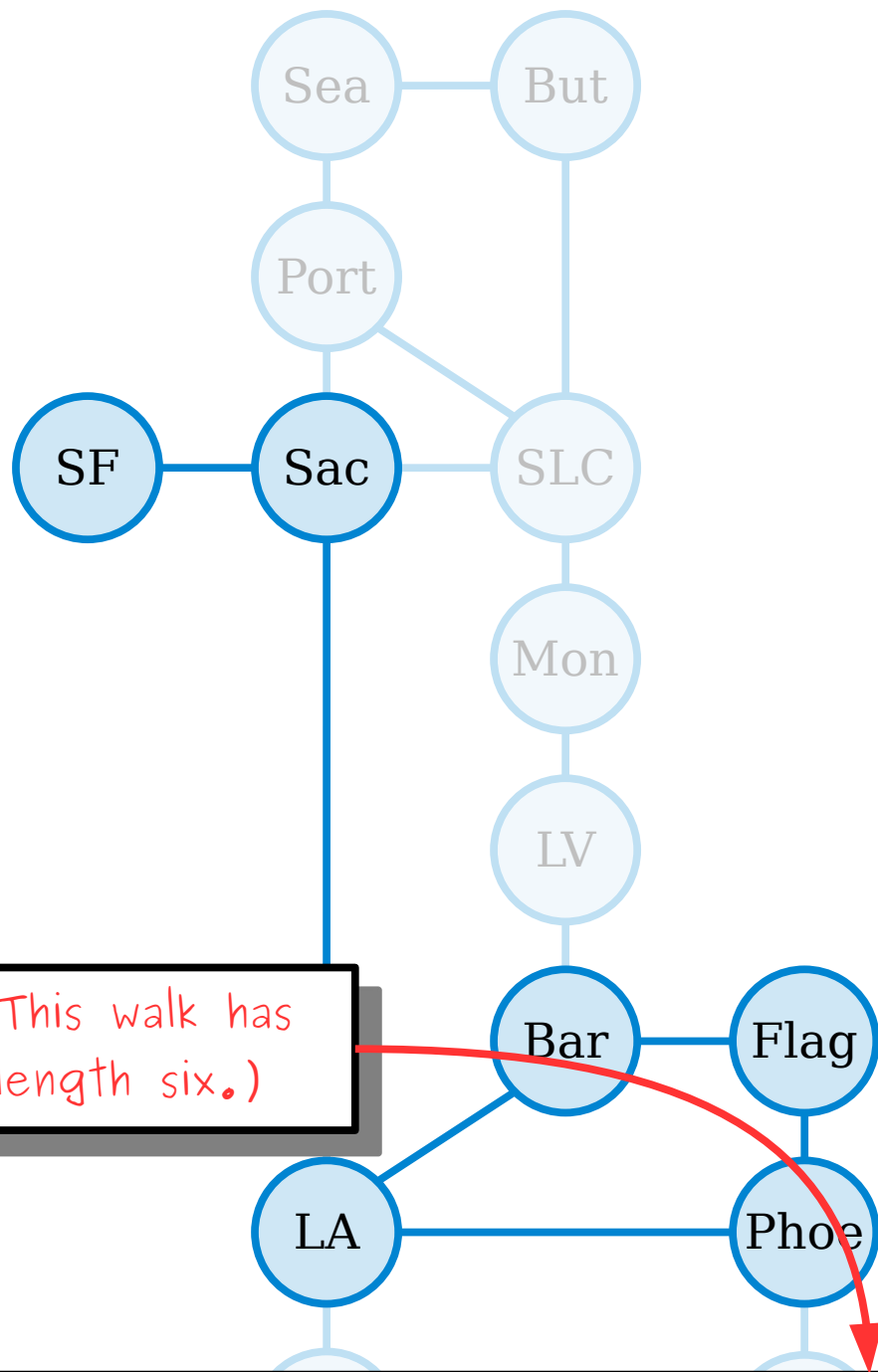
A **walk** in a graph $G = (V, E)$ is a sequence of one or more nodes $v_1, v_2, v_3, \dots, v_n$ such that any two consecutive nodes in the sequence are adjacent.

The **length** of the walk v_1, \dots, v_n is $n - 1$.

A **closed walk** in a graph is a walk from a node back to itself. (By convention, a closed walk cannot have length zero.)

A **path** in a graph is walk that does not repeat any nodes.

SF, Sac, LA, Phoe, Flag, Bar, LA



(This walk has length six.)

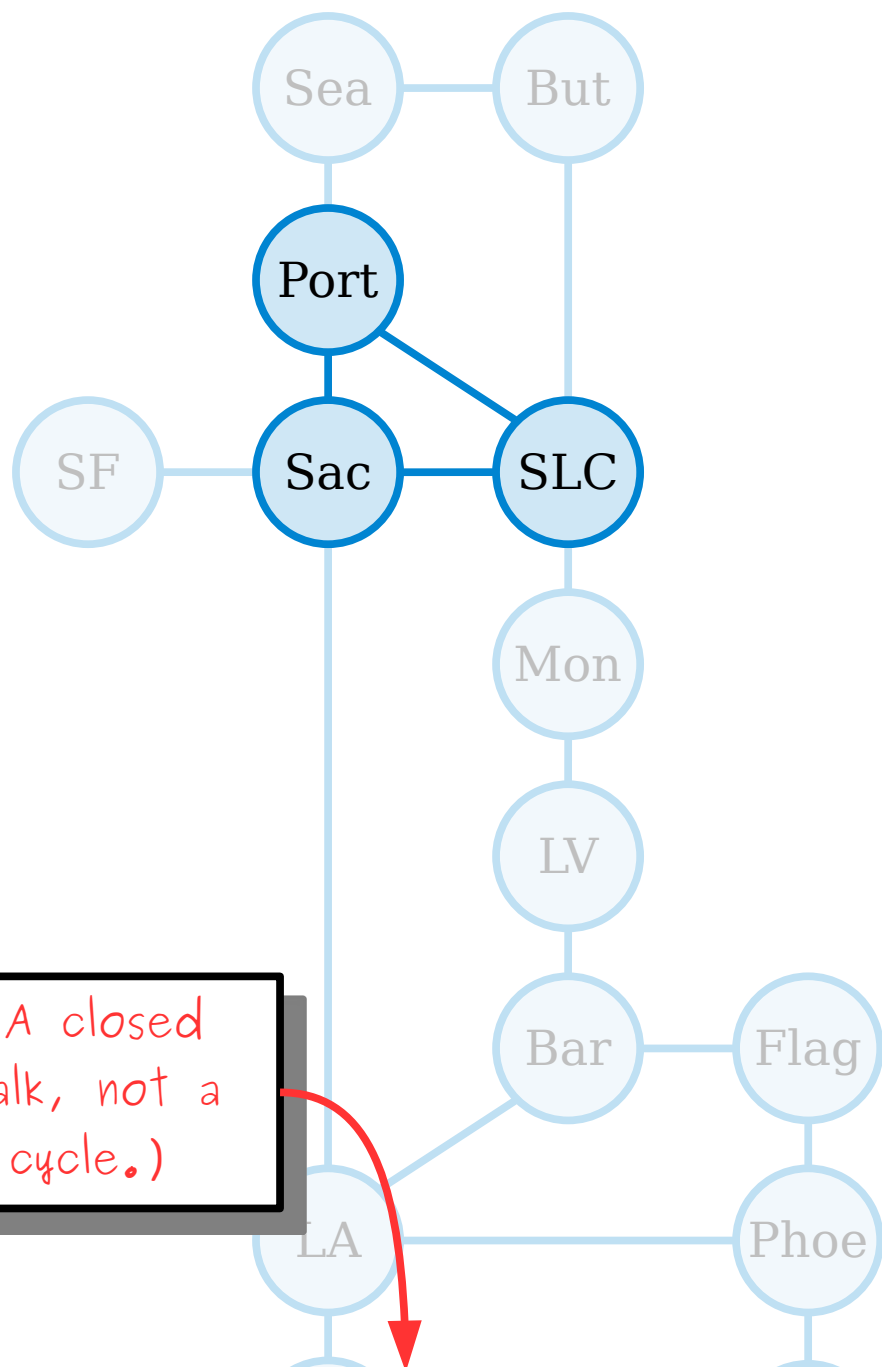
SF, Sac, LA, Phoe, Flag, Bar, LA

A **walk** in a graph $G = (V, E)$ is a sequence of one or more nodes $v_1, v_2, v_3, \dots, v_n$ such that any two consecutive nodes in the sequence are adjacent.

The **length** of the walk v_1, \dots, v_n is $n - 1$.

A **closed walk** in a graph is a walk from a node back to itself. (By convention, a closed walk cannot have length zero.)

A **path** in a graph is walk that does not repeat any nodes.



(A closed walk, not a cycle.)

Sac, SLC, Port, Sac, SLC, Port, Sac

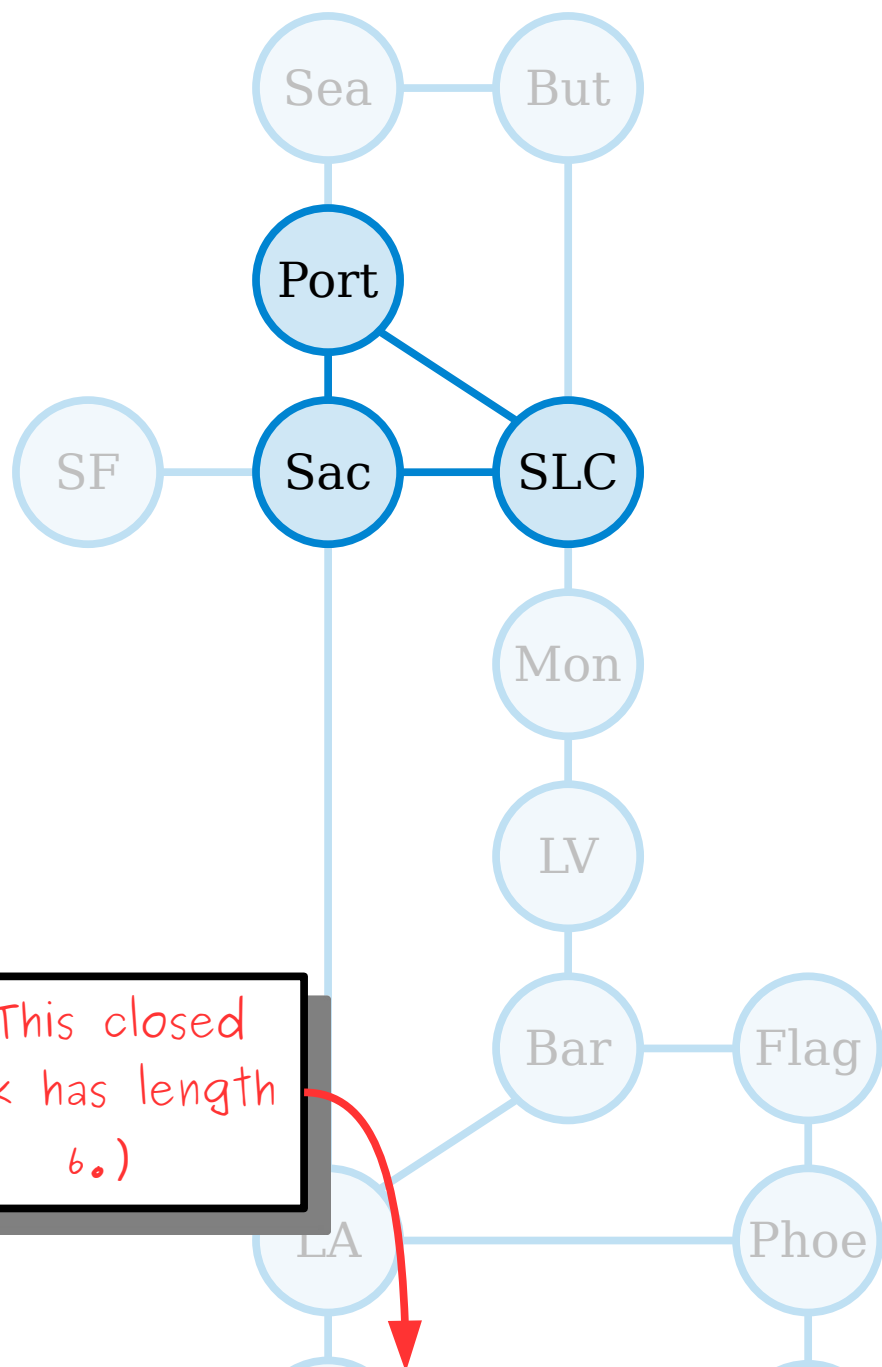
A **walk** in a graph $G = (V, E)$ is a sequence of one or more nodes $v_1, v_2, v_3, \dots, v_n$ such that any two consecutive nodes in the sequence are adjacent.

The **length** of the walk v_1, \dots, v_n is $n - 1$.

A **closed walk** in a graph is a walk from a node back to itself. (By convention, a closed walk cannot have length zero.)

A **path** in a graph is a walk that does not repeat any nodes.

A **cycle** in a graph is a closed walk that does not repeat any nodes or edges except the first/last node.



(This closed walk has length 6.)

Sac, SLC, Port, Sac, SLC, Port, Sac

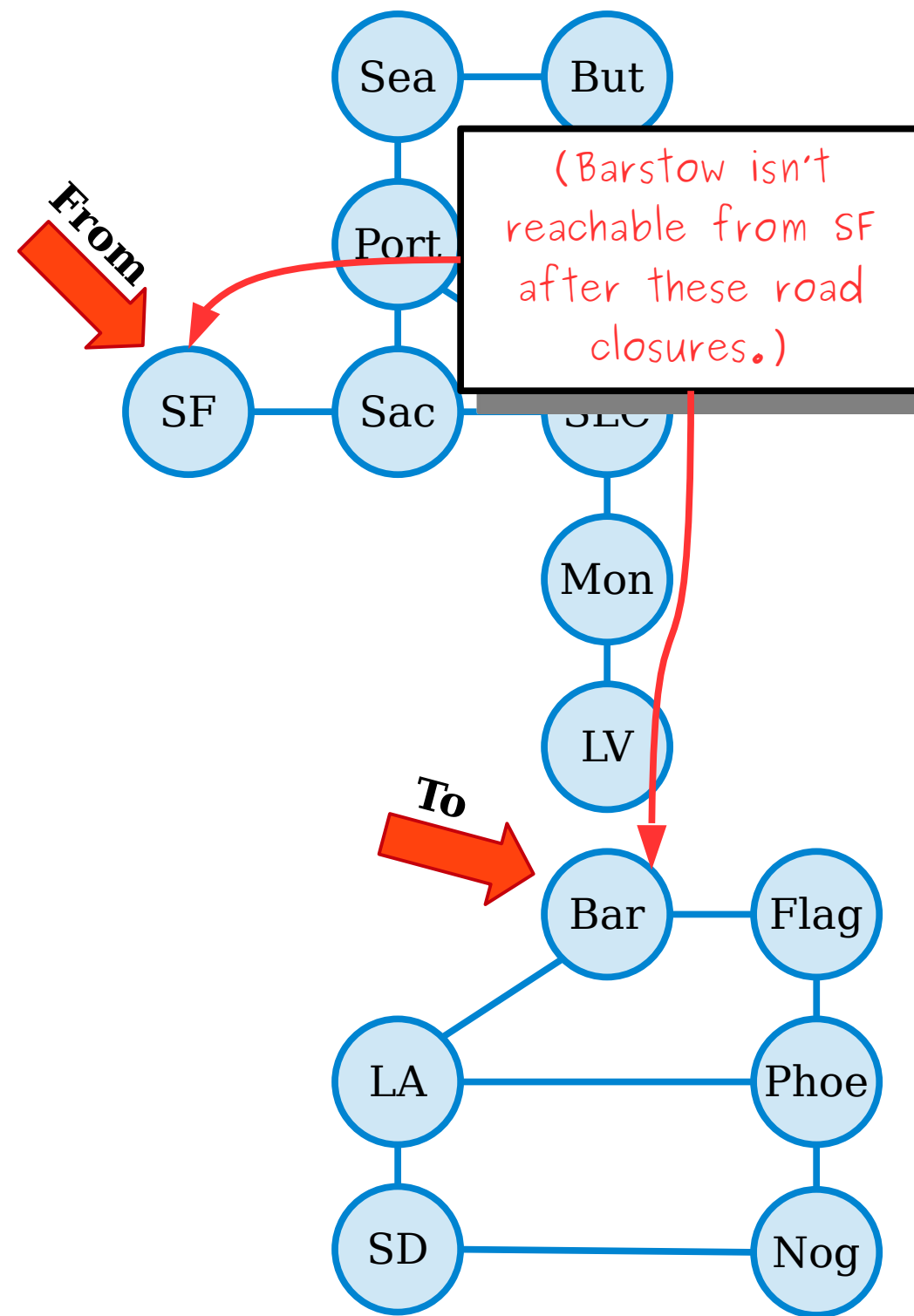
A **walk** in a graph $G = (V, E)$ is a sequence of one or more nodes $v_1, v_2, v_3, \dots, v_n$ such that any two consecutive nodes in the sequence are adjacent.

The **length** of the walk v_1, \dots, v_n is $n - 1$.

A **closed walk** in a graph is a walk from a node back to itself. (By convention, a closed walk cannot have length zero.)

A **path** in a graph is walk that does not repeat any nodes.

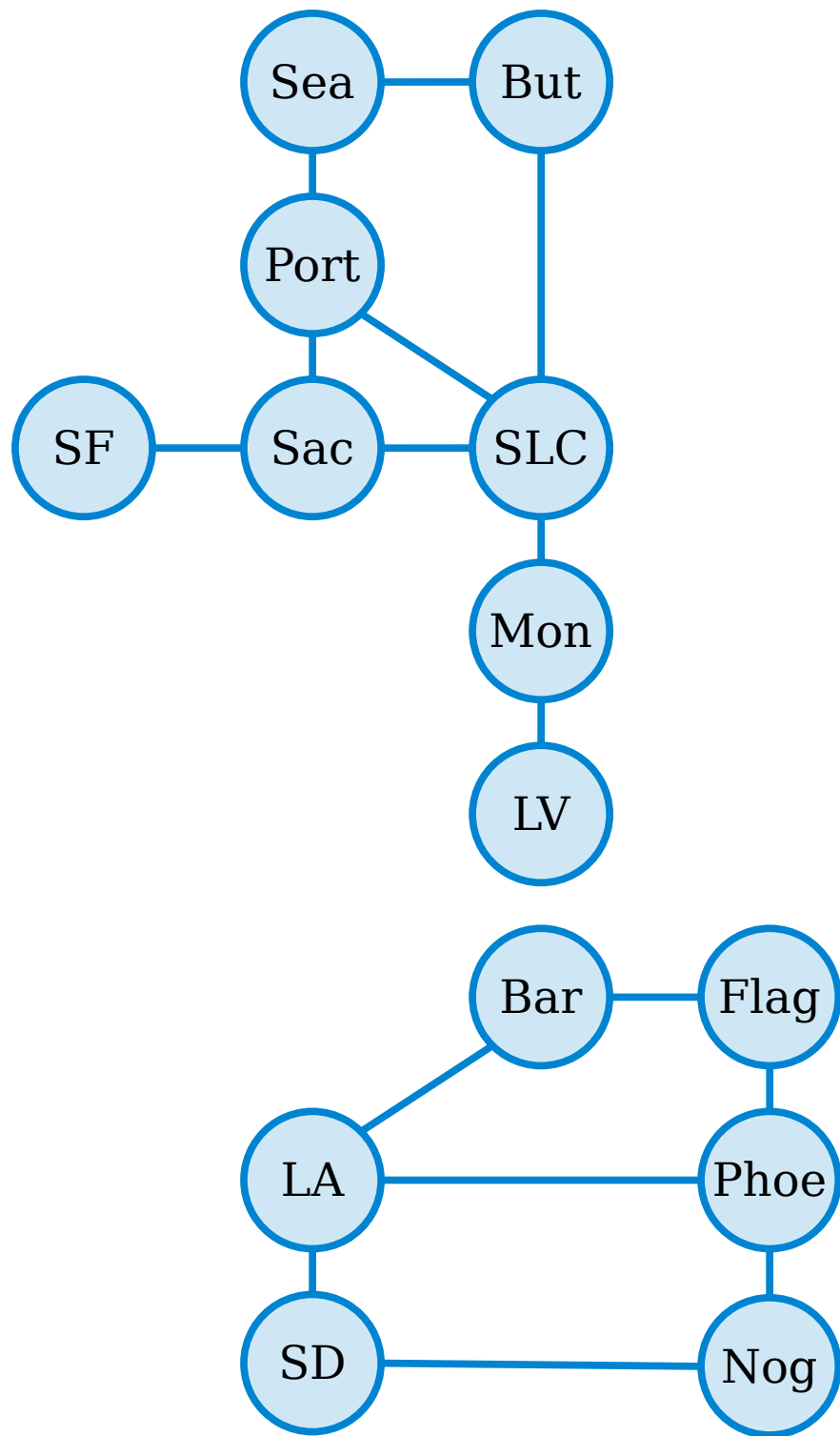
A **cycle** in a graph is a closed walk that does not repeat any nodes or edges except the first/last node.



A **walk** in a graph $G = (V, E)$ is a sequence of one or more nodes $v_1, v_2, v_3, \dots, v_n$ such that any two consecutive nodes in the sequence are adjacent.

A **path** in a graph is a walk that does not repeat any nodes.

A node v is **reachable** from a node u when there is a path from u to v .



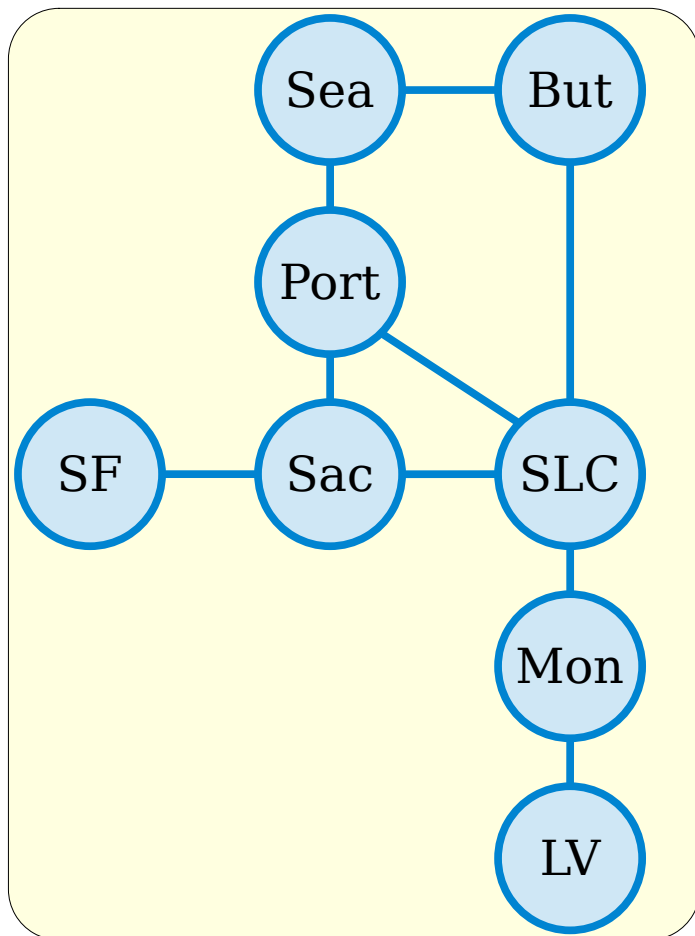
A **walk** in a graph $G = (V, E)$ is a sequence of one or more nodes $v_1, v_2, v_3, \dots, v_n$ such that any two consecutive nodes in the sequence are adjacent.

A **path** in a graph is walk that does not repeat any nodes.

A node v is **reachable** from a node u when there is a path from u to v .

A graph G is called **connected** when all pairs of distinct nodes in G are reachable.

(This graph is not connected.)



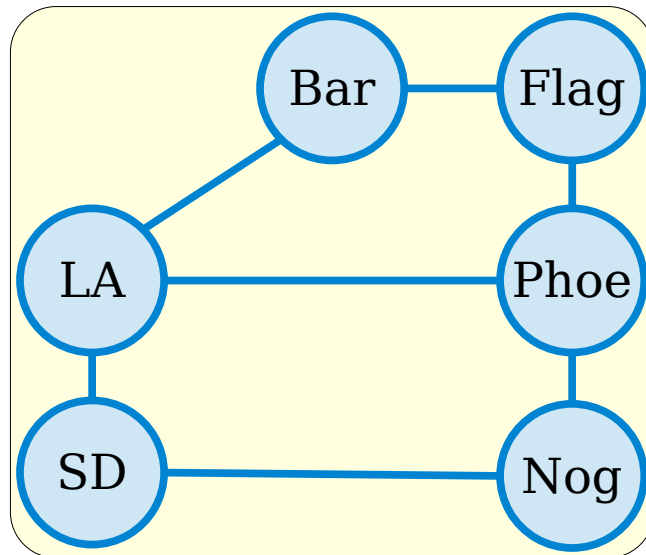
A **walk** in a graph $G = (V, E)$ is a sequence of one or more nodes $v_1, v_2, v_3, \dots, v_n$ such that any two consecutive nodes in the sequence are adjacent.

A **path** in a graph is walk that does not repeat any nodes.

A node v is **reachable** from a node u when there is a path from u to v .

A graph G is called **connected** when all pairs of distinct nodes in G are reachable.

A **connected component** (or **CC**) of G is a set consisting of a node and every node reachable from it.

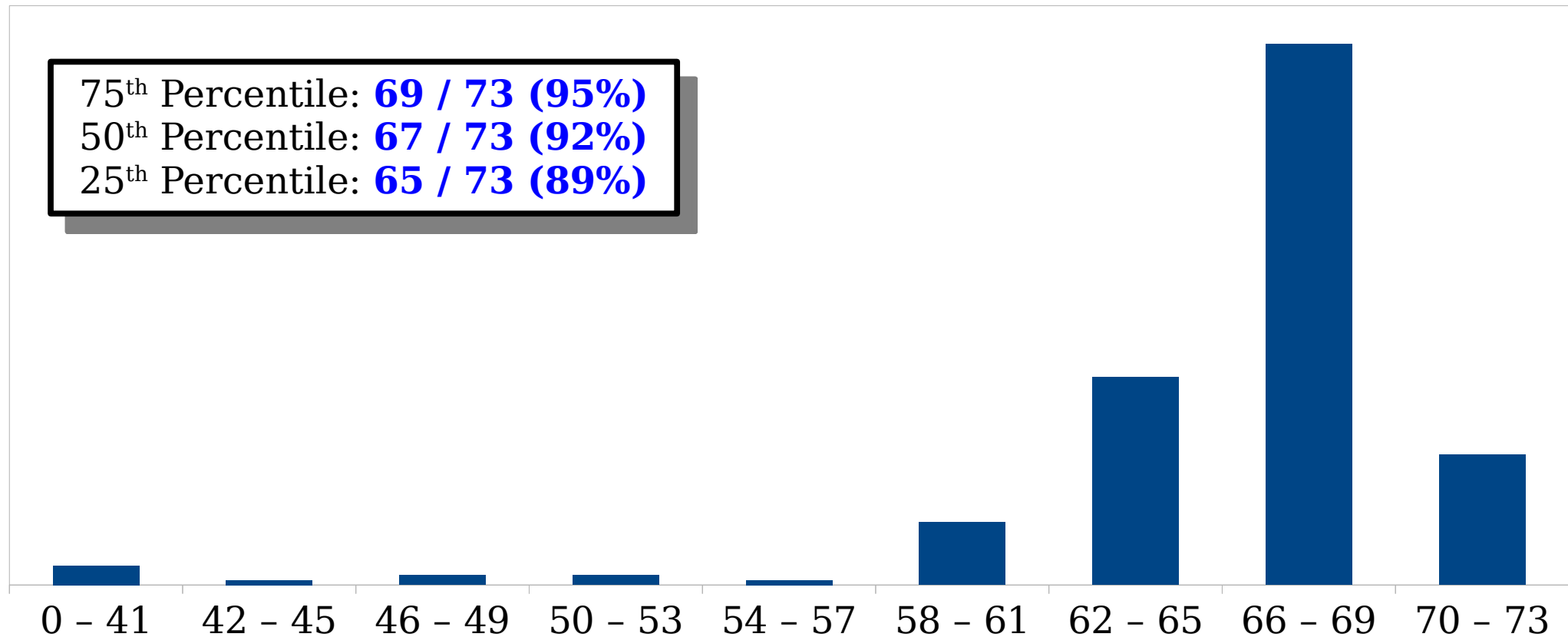


Fun Facts

- Here's a collection of useful facts about graphs that you can take as a given.
 - **Theorem:** If $G = (V, E)$ is a (directed or undirected) graph and $u, v \in V$, then there is a path from u to v if and only if there's a walk from u to v .
 - **Theorem:** If G is an undirected graph and C is a cycle in G , then C 's length is at least three and C contains at least three nodes.
 - **Theorem:** If $G = (V, E)$ is an undirected graph, then every node in V belongs to exactly one connected component of G .
 - **Theorem:** If $G = (V, E)$ is a (directed or undirected) graph and $u, y_0, y_1, \dots, y_m, v$ is a walk from u to v and $v, z_0, z_1, \dots, z_n, x$ is a walk from v to x , then $u, y_0, y_1, \dots, y_m, v, z_0, z_1, \dots, z_n, x$ is a walk from u to x .
- Looking for more practice working with formal definitions?
Prove these results!

Time-Out for Announcements!

Problem Set Two Graded



Midterm Exam Logistics

- Our first midterm exam is next **Monday, October 20th** from **7:00PM - 10:00PM**.
 - **Seating assignments are available**. Write your seat number down in case the WiFi cuts out before the exam.
- You're responsible for Lectures 00 – 05 and topics covered in PS1 – PS2.
 - Later lectures (functions forward) and problem sets (PS3 onward) won't be tested here.
 - Exam problems may build on the written or coding components from the problem sets.
- The exam is closed-book, closed-computer, and limited-note. You can bring a double-sided, 8.5" × 11" sheet of notes with you to the exam, decorated however you'd like.

Preparing for the Exam

- Your amazing CA Ari is holding a review session this Friday from 3PM – 4PM in CoDa E160.
- Make sure to ***review your feedback*** on PS1 and PS2.
 - “Make new mistakes.”
 - Come talk to us if you have questions!
- There’s a huge bank of practice problems up on the course website.
- Best of luck – ***you can do this!***

Participation Opt-Out

- By default, all on-campus students have 5% of their grade allocated from lecture attendance and participation.
- If you are an on-campus student and want to opt out, shifting that 5% onto your final exam, ***fill out the opt-out form on Ed by Friday at 11:59 PM.***

Back to CS103!

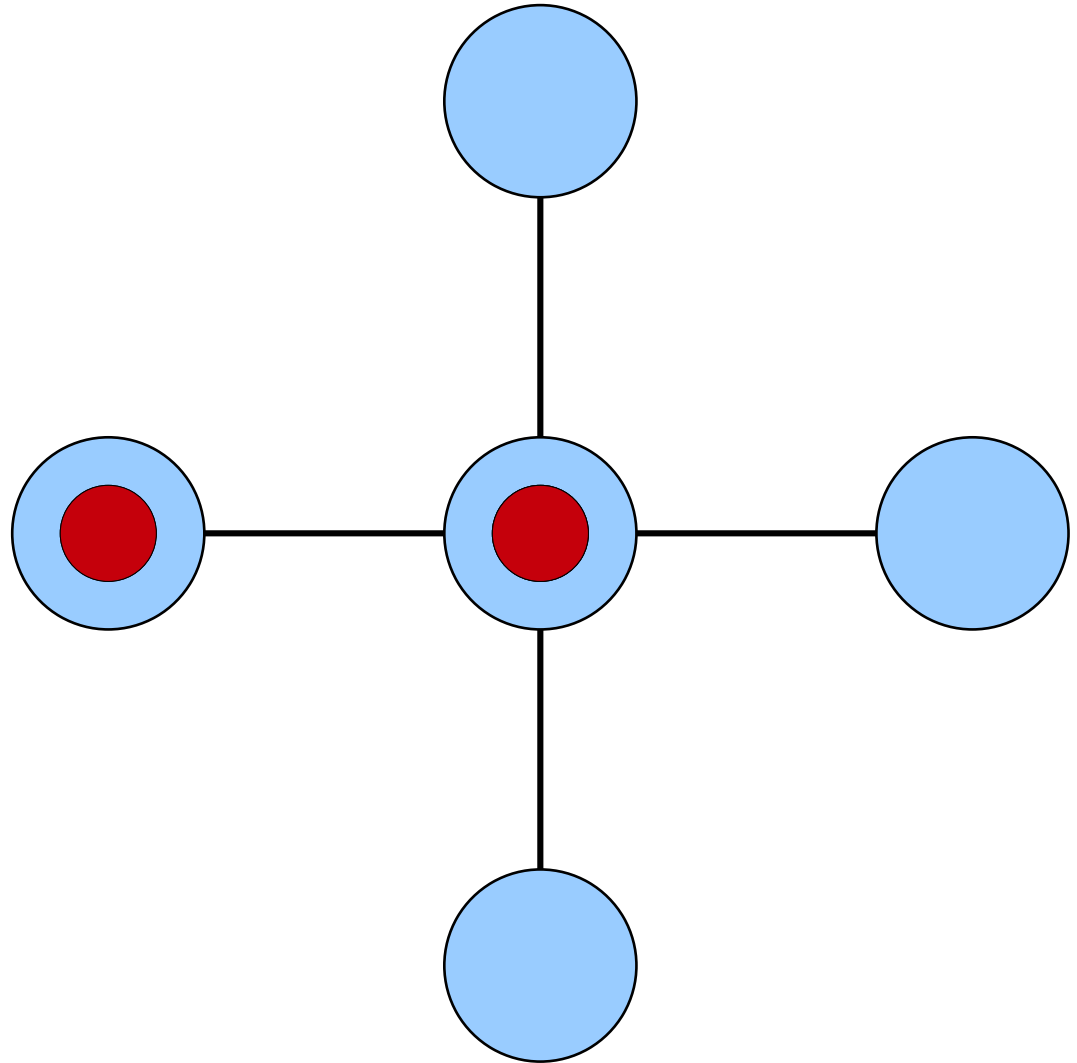
Application: ***Local Area Networks***

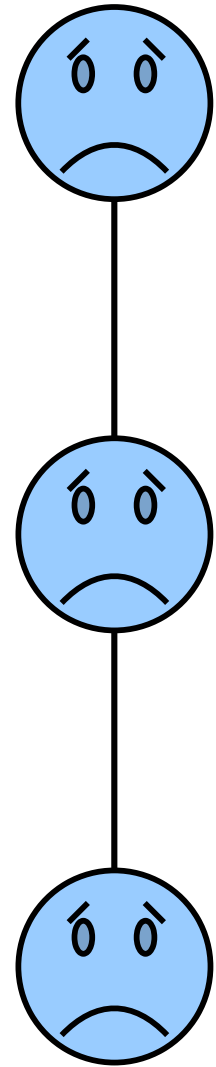
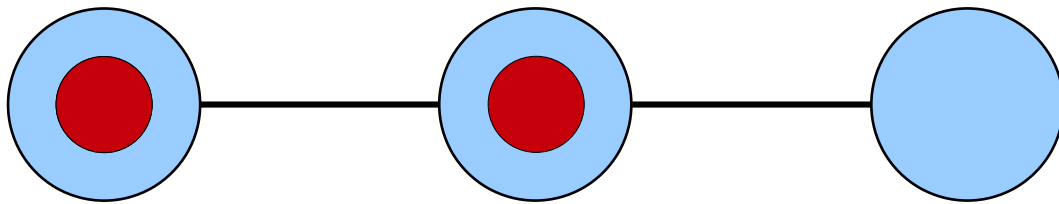
The Internet and LANs

- The internet consists of several separate **local area networks (LANs)** that are “internetworked” together.
- Local area networks cover small areas – a single hallway in a dorm, an office building, a college campus, etc.
- The internet then links those smaller LANs into one giant network where everyone can talk to everyone.
- **Focus for today:** How do messages flow through a LAN?

Message Movement

- When a computer receives a message, it repeats that message on all its links except the one it received the message on.
- The computers don't inspect the message contents or try to be clever – it's purely “came in on link X, goes out on all links but X.”



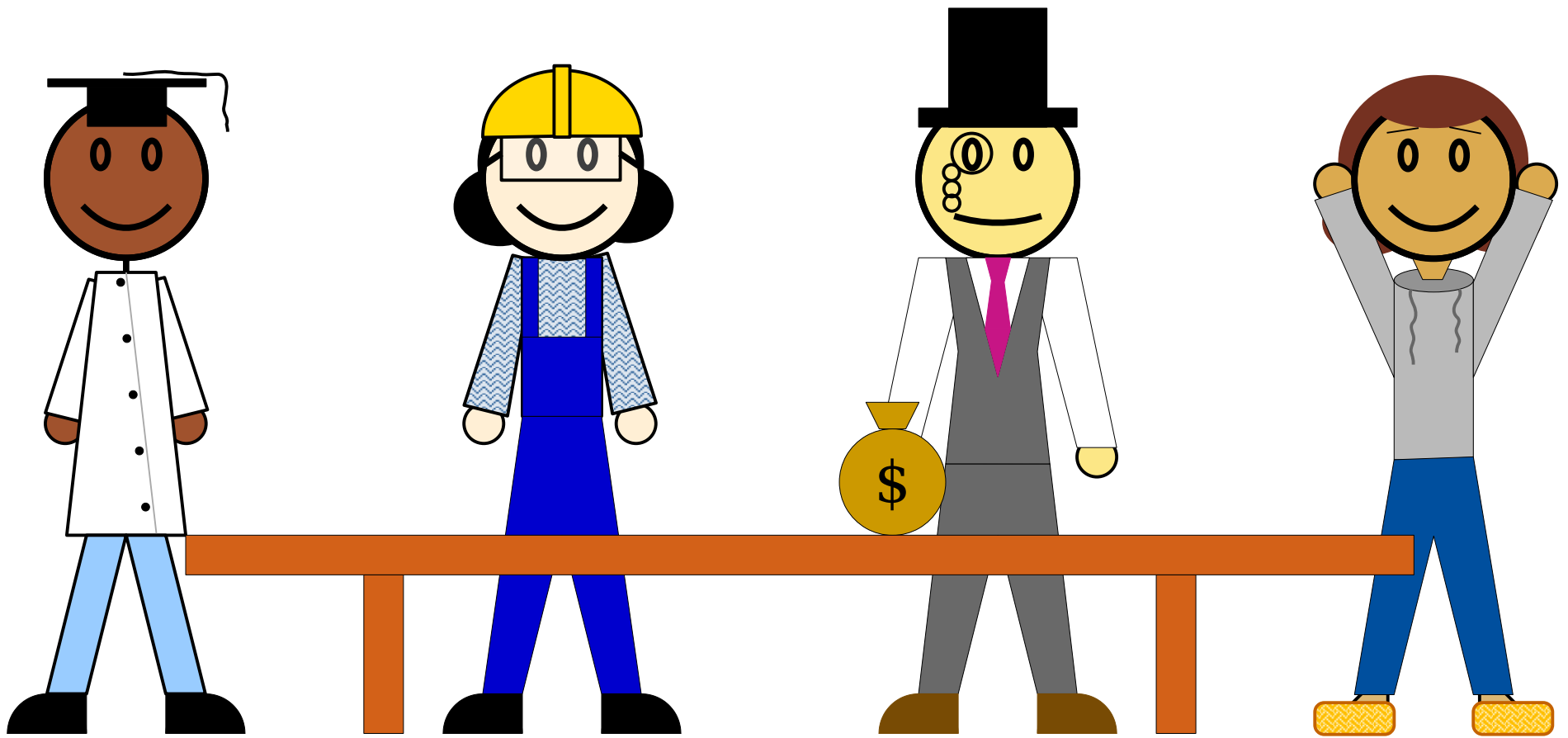


The network graph
must be **connected**.

Broadcast Storms

- A ***broadcast storm*** occurs when there's a cycle in the network graph.
- A single message can repeat forever, or exponentially amplify until the network fails.
- ***Solution:*** Don't let the network graph have any cycles.
- A graph $G = (V, E)$ is called ***acyclic*** if it has no cycles.

You have a collection of computers that need to be wired up into a LAN. How should you choose the shape of the network?



CTO

Connected,
No Cycles

COO

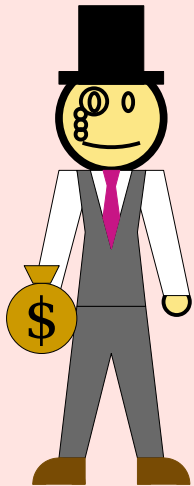
Most Links,
No Cycles

CFO

Fewest Links,
Connected

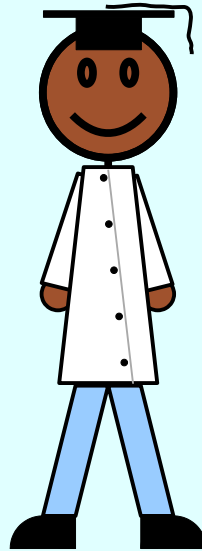
CEO

*Do all
three!*



Minimally Connected

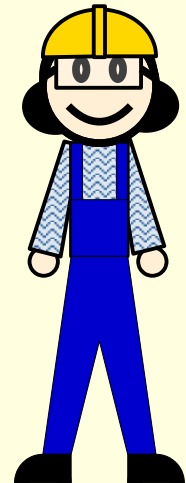
(Connected, but deleting any edge disconnects its endpoints.)



Connected, Acyclic

If *any* of these conditions hold, then *all* of these conditions hold.

A graph with any of these properties is called a ***tree***.



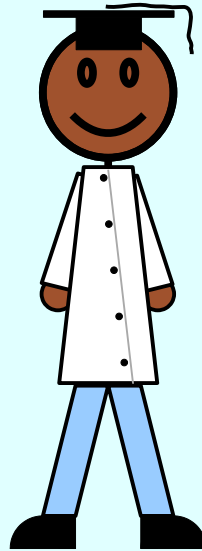
Maximally Acyclic

(Acyclic, but adding any missing edge creates a cycle.)

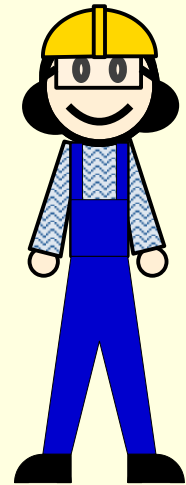


Minimally Connected

(Connected, but deleting any edge disconnects its endpoints.)

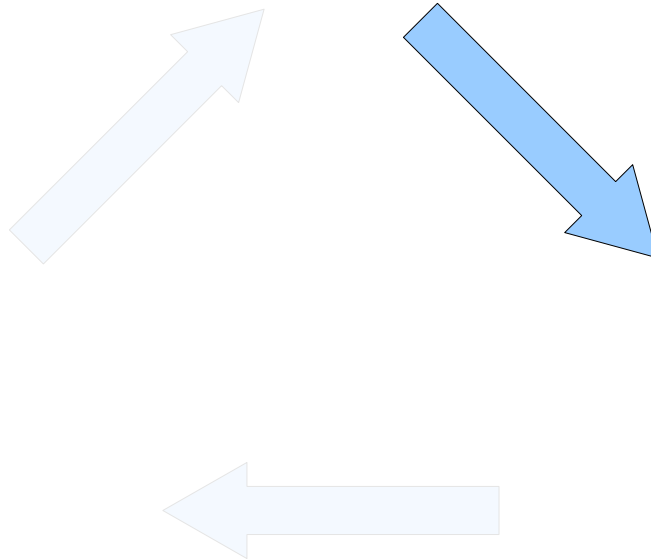


Connected, Acyclic



Maximally Acyclic

(Acyclic, but adding any missing edge creates a cycle.)



Theorem: Let $T = (V, E)$ be a graph. If T is connected and acyclic, then T is maximally acyclic.

Proof: Assume T is connected and has no cycles. We need to prove that T is maximally acyclic. We already know that T is acyclic. So choose distinct nodes $x, y \in V$ where $\{x, y\} \notin E$; we'll prove adding $\{x, y\}$ to E closes a cycle.

Because T is connected, there is a path x, \dots, y from x to y in T . Now add $\{x, y\}$ to E . Then we can form the closed walk x, \dots, y, x . We claim that this is a cycle. To see this, note the following:

- No node is repeated except the start/end node x : nodes x, \dots, y are all distinct because x, \dots, y is a path.
- No edge appears twice: none of the edges used in x, \dots, y are repeated (x, \dots, y is a path). Furthermore, the edge $\{x, y\}$ isn't repeated since the path x, \dots, y was formed before $\{x, y\}$ was added to E .

Thus adding $\{x, y\}$ to E closes a cycle, as required. ■

Check the appendix for the
other two steps of the proof.

More to Explore

- A tree kind of seems like a bad way to design a network. (Why?)
- Actual local area networks allow for cycles. They use something called the **spanning tree protocol (STP)** to selectively disable links to form a tree.
- Routing through the full internet – not just within a LAN – is a fascinating topic in its own right.
- Take CS144 (networking) for details!
- If we have time, we'll explore more on network routing later in the quarter.

Recap from Today

- ***Walks*** and ***closed walks*** represent ways of moving around a graph. ***Paths*** and ***cycles*** are “redundancy-free” walks and cycles.
- ***Trees*** are graphs that are connected and acyclic. They’re also minimally-connected graphs and maximally-acyclic graphs.
- Trees have applications throughout CS, including networking.

Next Time

- ***The Pigeonhole Principle***
 - A simple, powerful, versatile theorem.
- ***Graph Theory Party Tricks***
 - Applying math to graphs of people!
- ***A Little Movie Puzzle***
 - Who watched what?

Appendix

Theorem: Let $T = (V, E)$ be a graph. If T is minimally connected, then T is connected and acyclic.

Proof: Assume T is minimally connected. We need to show that T is connected and acyclic. Since T is minimally connected, it's connected, and so we just need to show that T is acyclic.

Suppose for the sake of contradiction that T contains a cycle x, \dots, y, x . Note in particular that this means x, \dots, y is a path in T and that this path does not use the edge $\{x, y\}$.

Since T is minimally connected, deleting the edge $\{x, y\}$ from T makes y not reachable from x . However, we said earlier that x, \dots, y is a path from x to y in T that does not use $\{x, y\}$, so x and y remain reachable after deleting $\{x, y\}$.

We have reached a contradiction, so our assumption was wrong and T is acyclic. ■

Theorem: Let $T = (V, E)$ be a graph. If T is maximally acyclic, then T is minimally connected.

Proof: Assume T is maximally acyclic. We need to prove that T is minimally connected. To do so, we first prove T is connected. Pick any $x, y \in V$ where $x \neq y$; we'll show there's a path from x to y . Consider two cases:

Case 1: $\{x, y\} \in E$. Then x, y is a path from x to y .

Case 2: $\{x, y\} \notin E$. Imagine adding $\{x, y\}$ to E . Since T is maximally acyclic, this closes a cycle x, \dots, y, x passing through $\{x, y\}$. Then x, \dots, y is a path in T from x to y .

In either case, we have a path from x to y , as needed.

Next, suppose for the sake of contradiction that there is an edge $\{x, y\} \in E$ where T remains connected after deleting $\{x, y\}$. This means that there is a path x, \dots, y in T after removing $\{x, y\}$. By adding $\{x, y\}$ to the end of the path, we form a cycle x, \dots, y, x in T . This is impossible because T is acyclic. We've reached a contradiction, so our assumption was wrong and T is minimally connected. ■