CS103
Winter 2026

Lecture 10:

Graph Theory (Part 2 of 3)

Graphs

Part 2

1. **Recap from Last Time**
2. Preliminary Definition: Adjacency
3. Walks, Paths, and Other Journeys
4. Announcements
5. Sending Messages through LANs
6. Shaping LANs (and a Proof on Graphs)
7. Spanning Tree Protocol (STP)
8. Recap and What’s Next?

Graphs and Digraphs

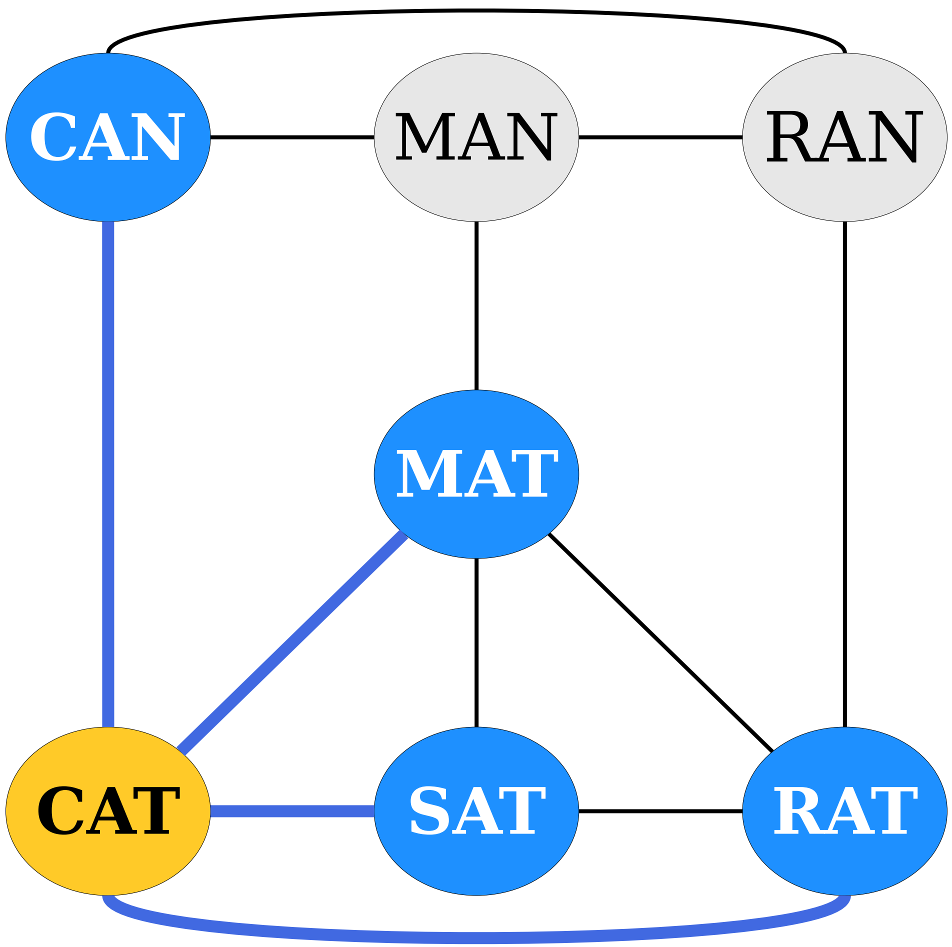
- A ***graph*** is a pair $G = (V, E)$ of a set of nodes V and set of edges E.
  - Nodes can be anything.
  - Edges are ***unordered pairs*** of nodes. If $\{u, v\} \in E$, then there's an edge from u to v.
- A ***digraph*** is a pair $G = (V, E)$ of a set of nodes V and set of directed edges E.
  -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

Graphs

Part 2

1. Recap from Last Time
2. **Preliminary Definition: Adjacency**
3. Walks, Paths, and Other Journeys
4. Announcements
5. Sending Messages through LANs
6. Shaping LANs (and a Proof on Graphs)
7. Spanning Tree Protocol (STP)
8. Recap and What’s Next?

Adjacency



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

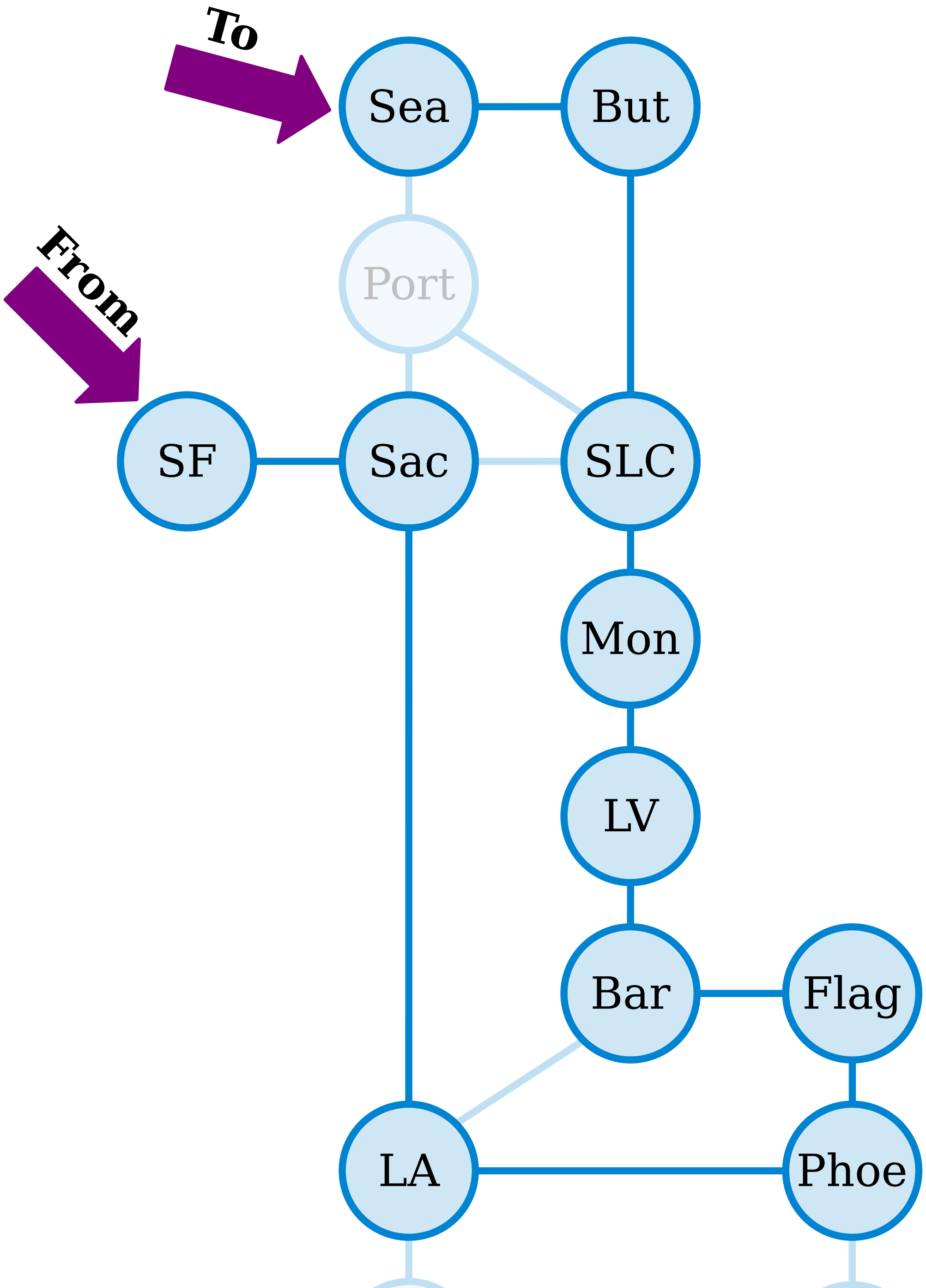
Adjacency

- 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.

Graphs

Part 2

1. Recap from Last Time
2. Preliminary Definition: Adjacency
3. **Walks, Paths, and Other Journeys**
4. Announcements
5. Sending Messages through LANs
6. Shaping LANs (and a Proof on Graphs)
7. Spanning Tree Protocol (STP)
8. Recap and What's Next?

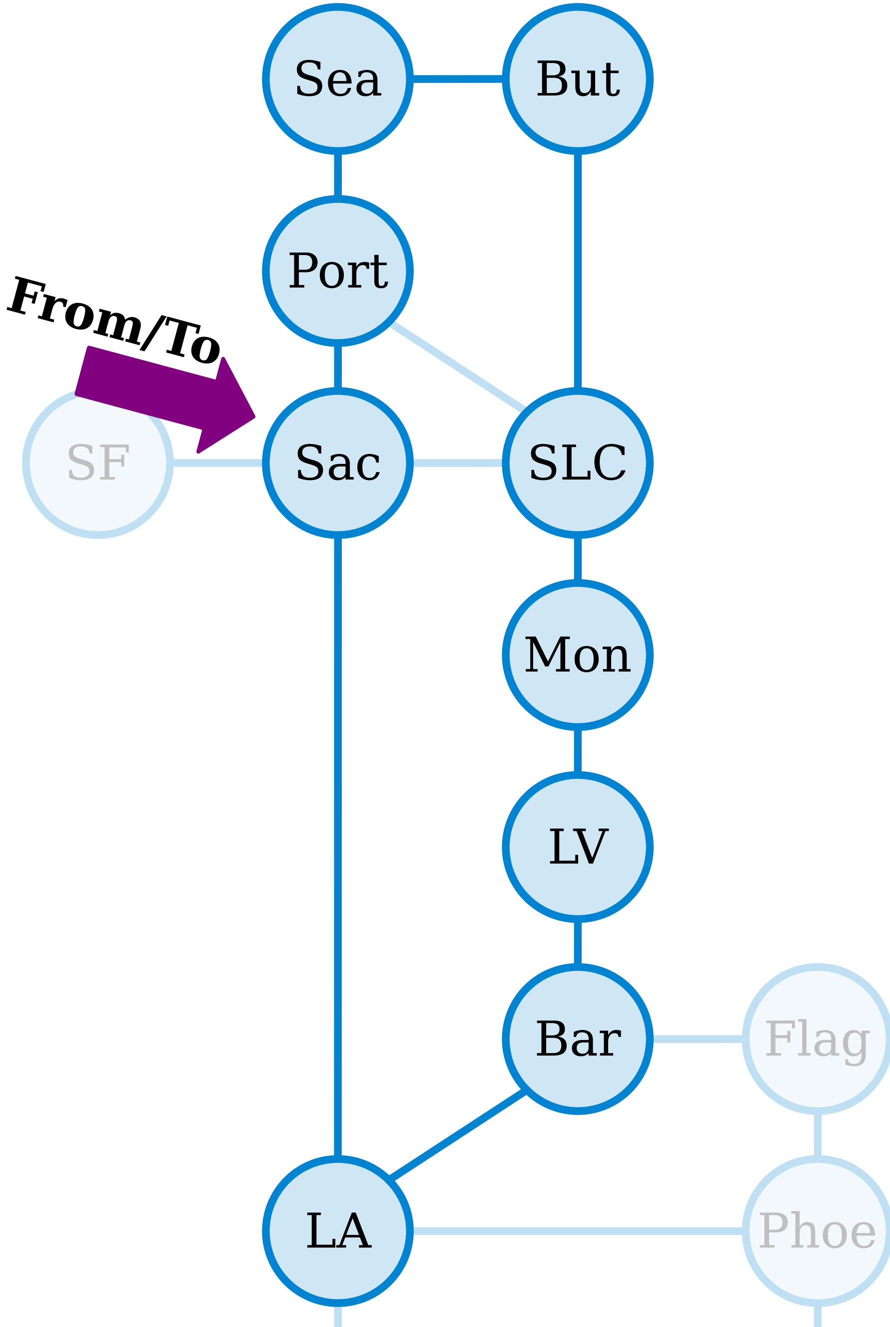


A ***walk*** in a graph $G = (V, E)$ is a sequence of one or more nodes v_1, v_2, v_3, …, v_n such that any two consecutive nodes in the sequence are adjacent.

The ***length*** of the walk v_1, …, 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



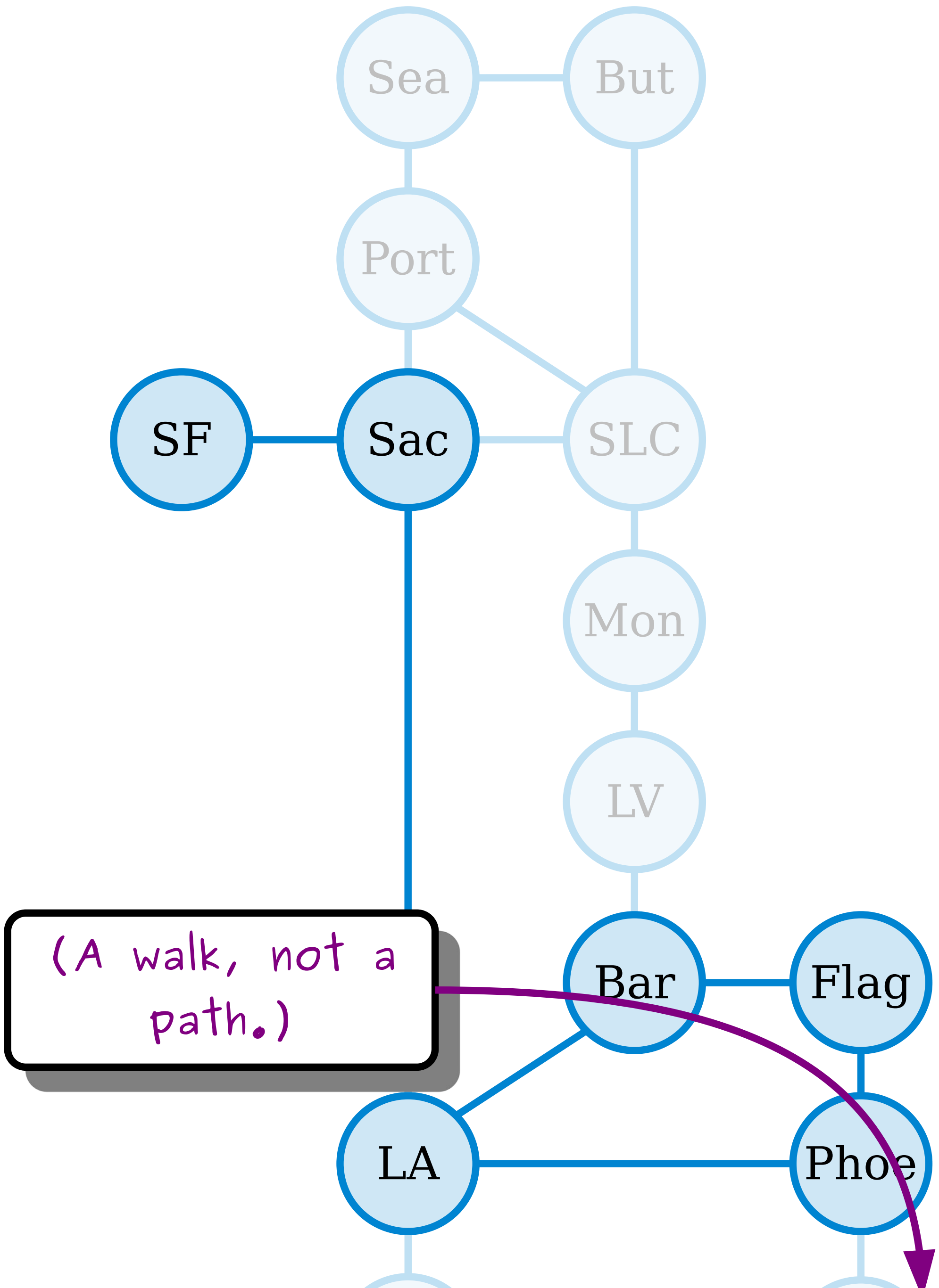
A ***walk*** in a graph $G = (V, E)$ is a sequence of one or more nodes v_1, v_2, v_3, …, v_n such that any two consecutive nodes in the sequence are adjacent.

The ***length*** of the walk v_1, …, 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



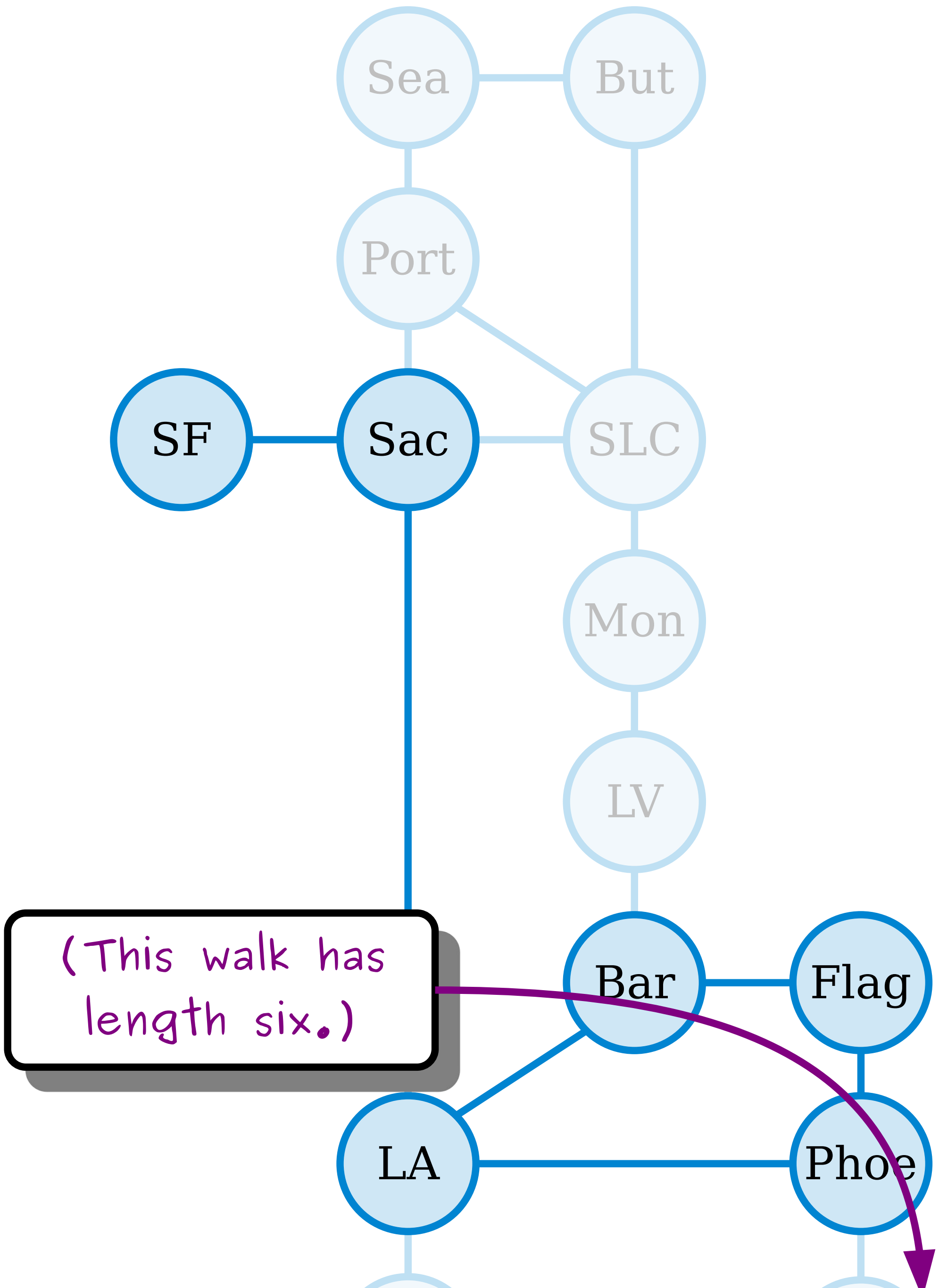
A ***walk*** in a graph $G = (V, E)$ is a sequence of one or more nodes v_1, v_2, v_3, …, v_n such that any two consecutive nodes in the sequence are adjacent.

The ***length*** of the walk v_1, …, 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



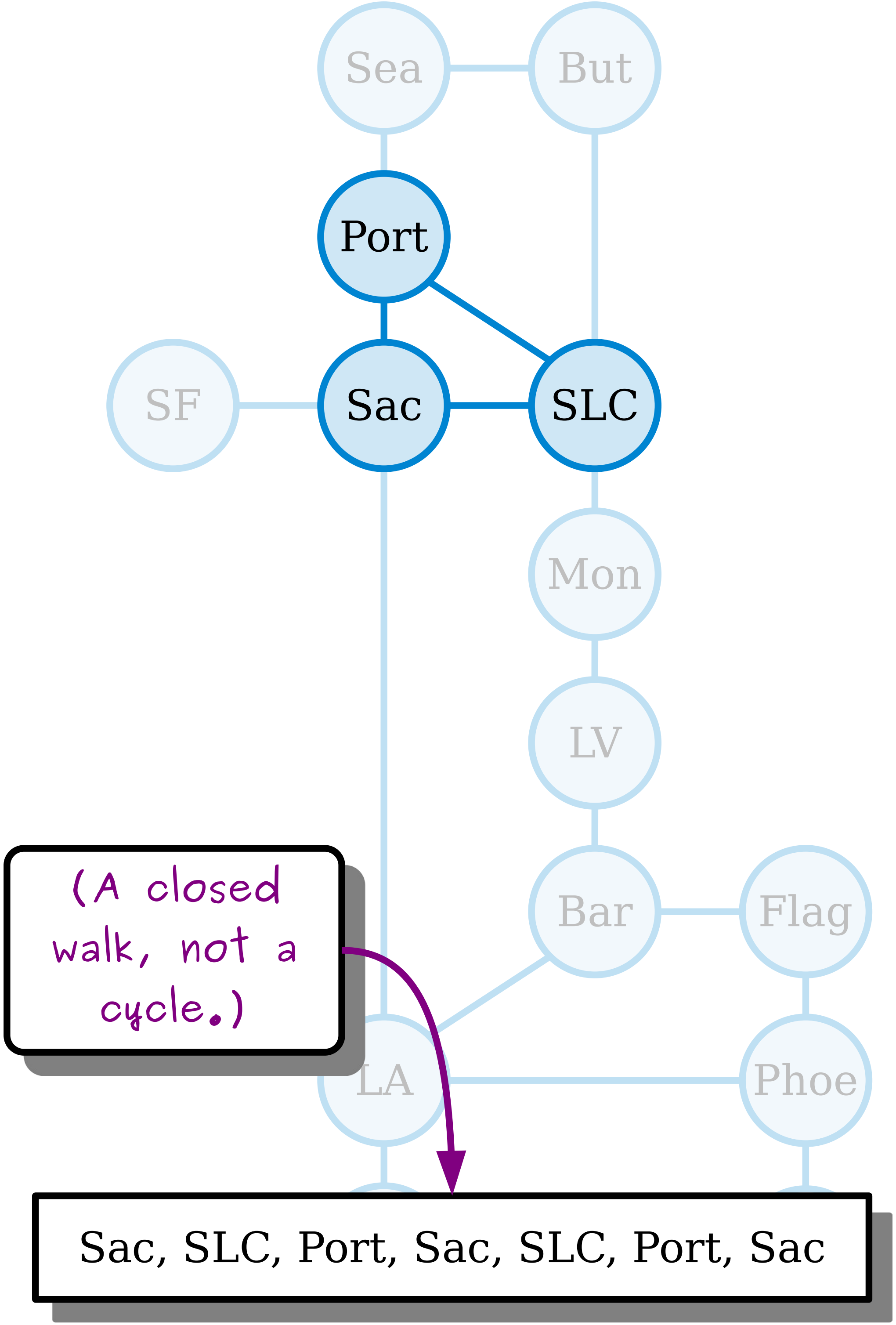
A ***walk*** in a graph $G = (V, E)$ is a sequence of one or more nodes v_1, v_2, v_3, …, v_n such that any two consecutive nodes in the sequence are adjacent.

The ***length*** of the walk v_1, …, 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



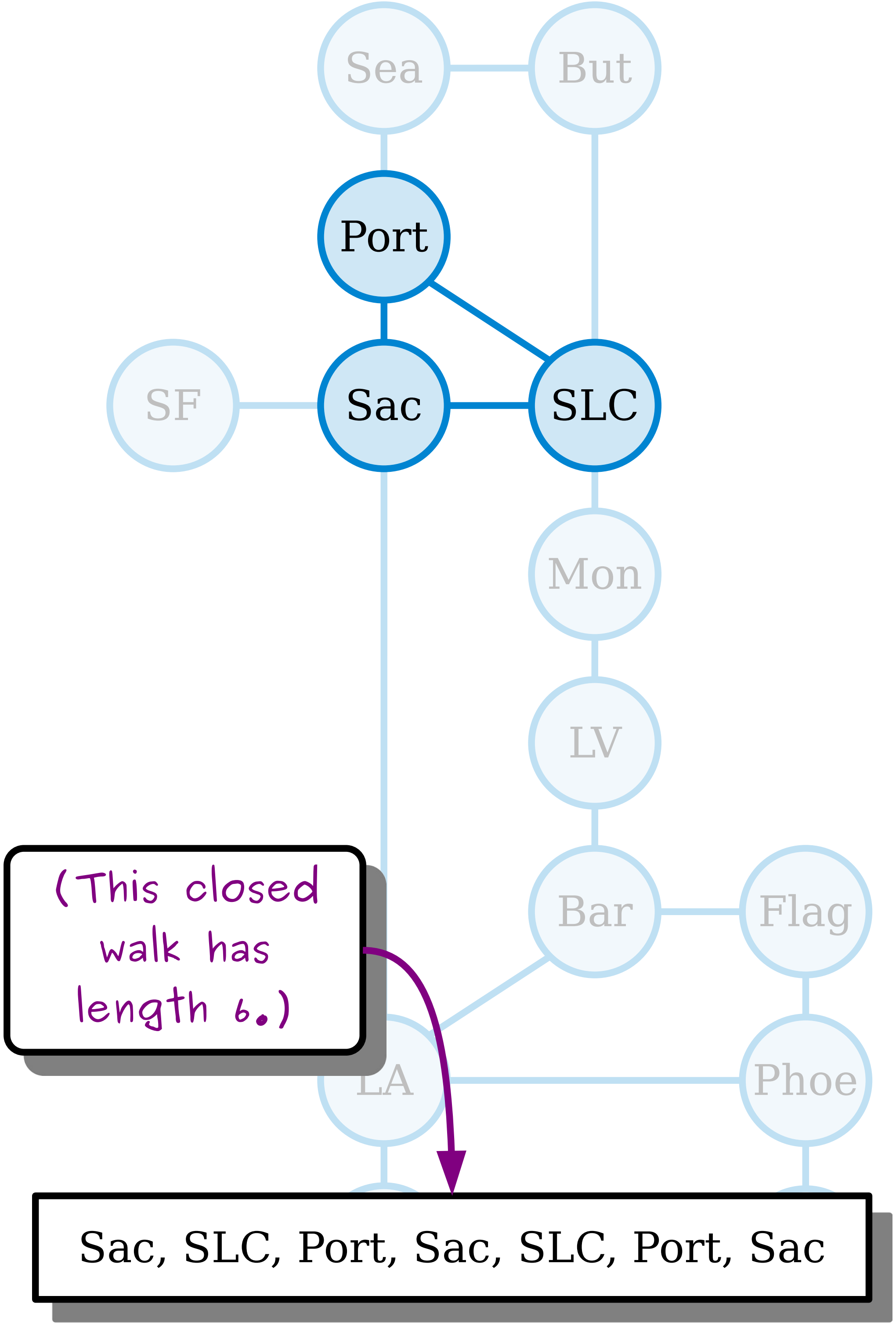
A ***walk*** in a graph $G = (V, E)$ is a sequence of one or more nodes v_1, v_2, v_3, …, v_n such that any two consecutive nodes in the sequence are adjacent.

The ***length*** of the walk v_1, …, 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.



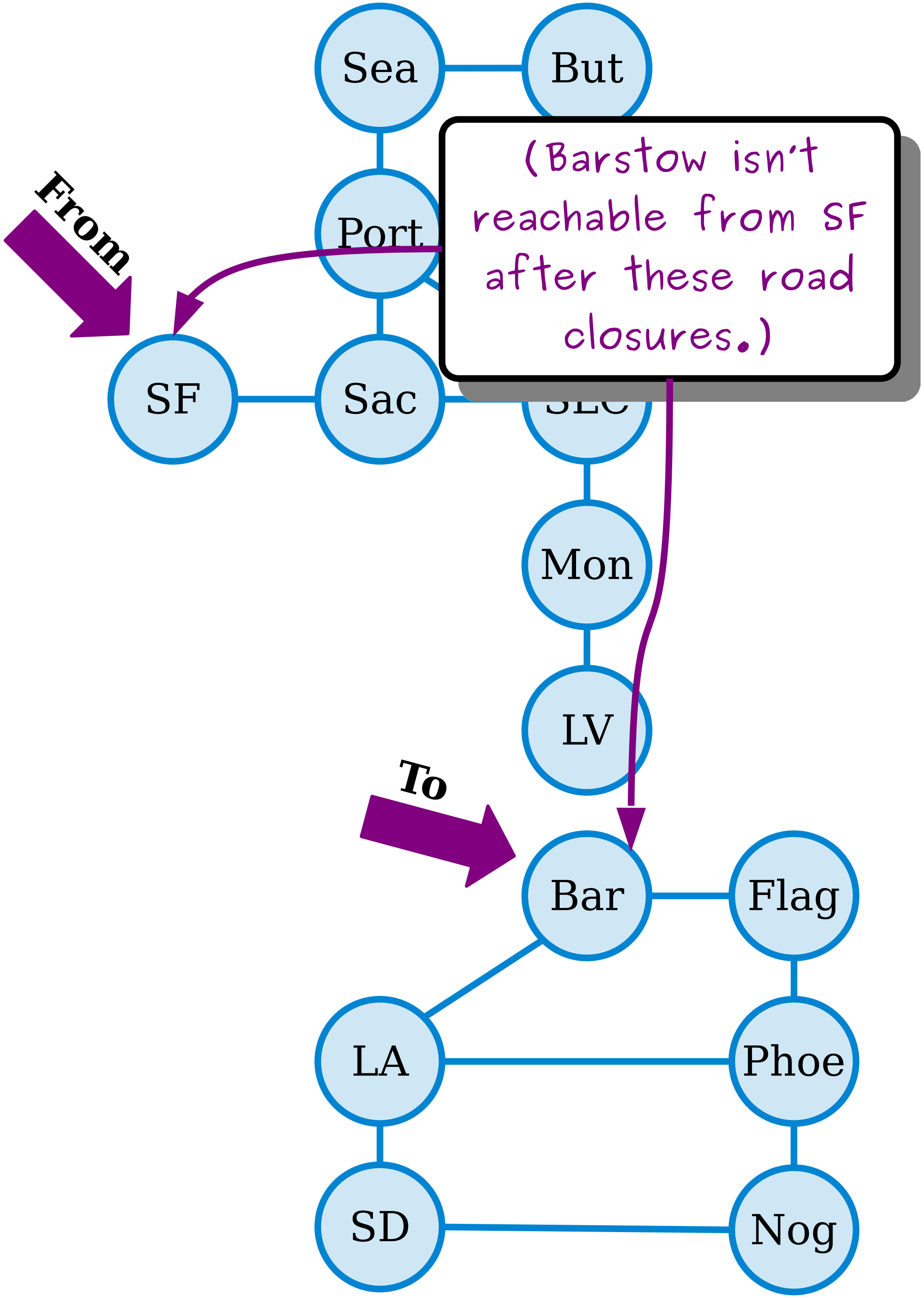
A ***walk*** in a graph $G = (V, E)$ is a sequence of one or more nodes v_1, v_2, v_3, …, v_n such that any two consecutive nodes in the sequence are adjacent.

The ***length*** of the walk v_1, …, 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.

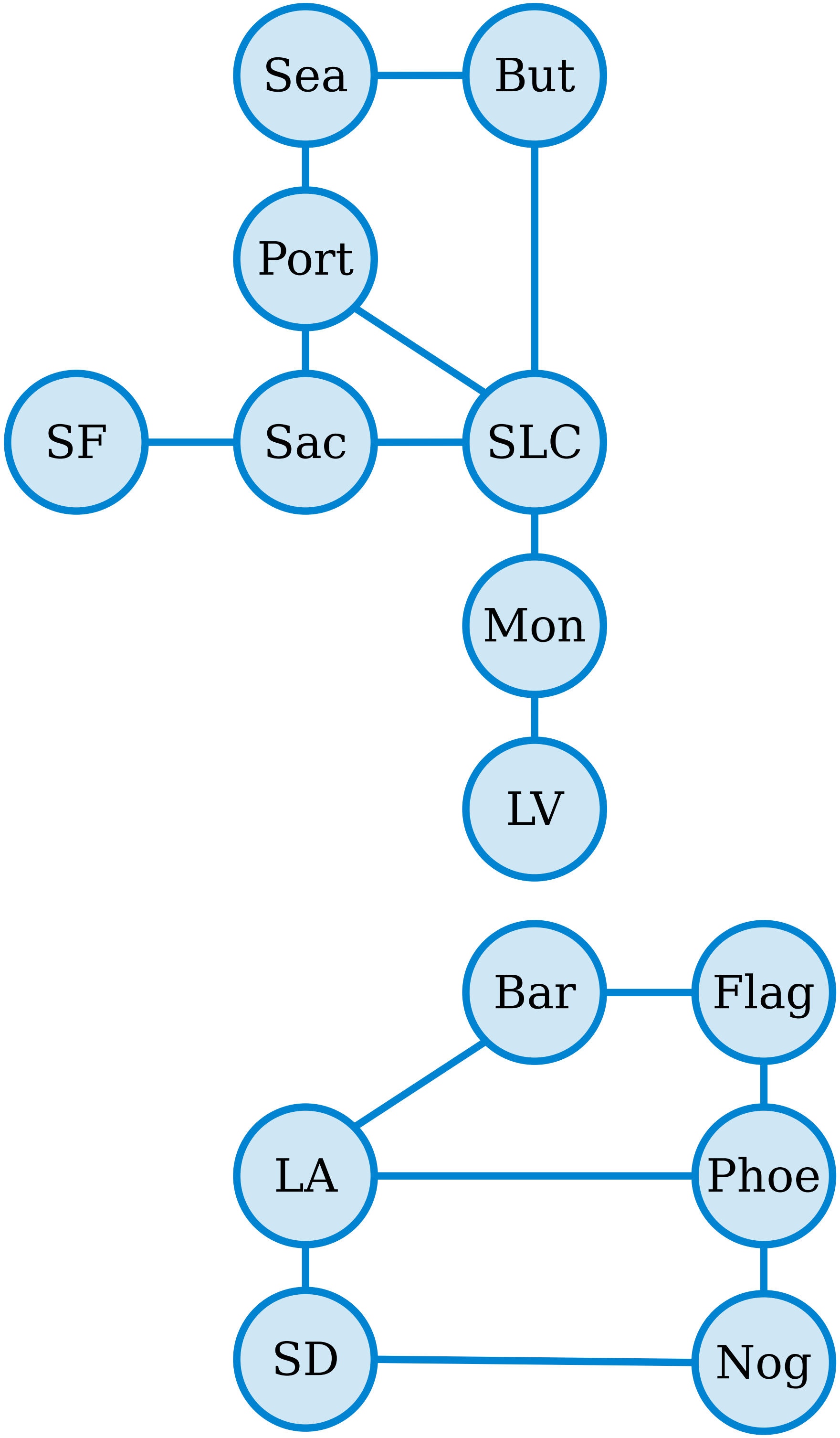
A ***cycle*** in a graph is a closed walk that does not repeat any nodes or edges except the first/last node.



A ***walk*** in a graph $G = (V, E)$ is a sequence of one or more nodes v_1, v_2, v_3, …, 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.



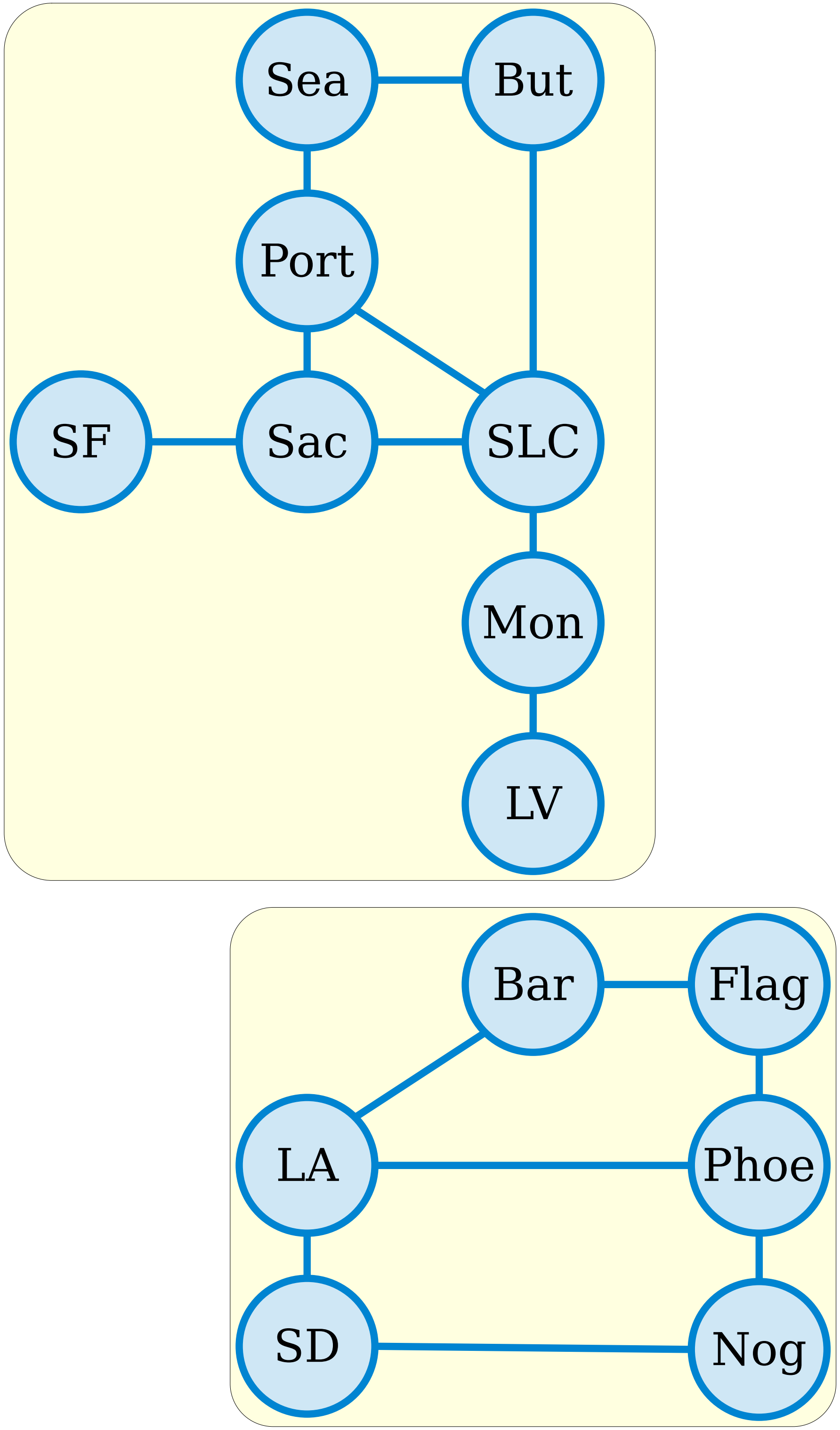
A ***walk*** in a graph $G = (V, E)$ is a sequence of one or more nodes v_1, v_2, v_3, …, v_n such that any two consecutive nodes in the sequence are adjacent.

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, …, 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 $u, y_0, y_1, \ldots, y_m, v$ is a walk from u to v and $v, z_0, z_1, \ldots, z_n, x$ is a walk from v to x, then $u, y_0, y_1, \ldots, y_m, v, z_0, z_1, \ldots, z_n, x$ is a walk from u to x.
- Looking for more practice working with formal definitions? Prove these results!

Graphs

Part 2

1. Recap from Last Time
2. Preliminary Definition: Adjacency
3. Walks, Paths, and Other Journeys
4. **Announcements**
5. Sending Messages through LANs
6. Shaping LANs (and a Proof on Graphs)
7. Spanning Tree Protocol (STP)
8. Recap and What's Next?

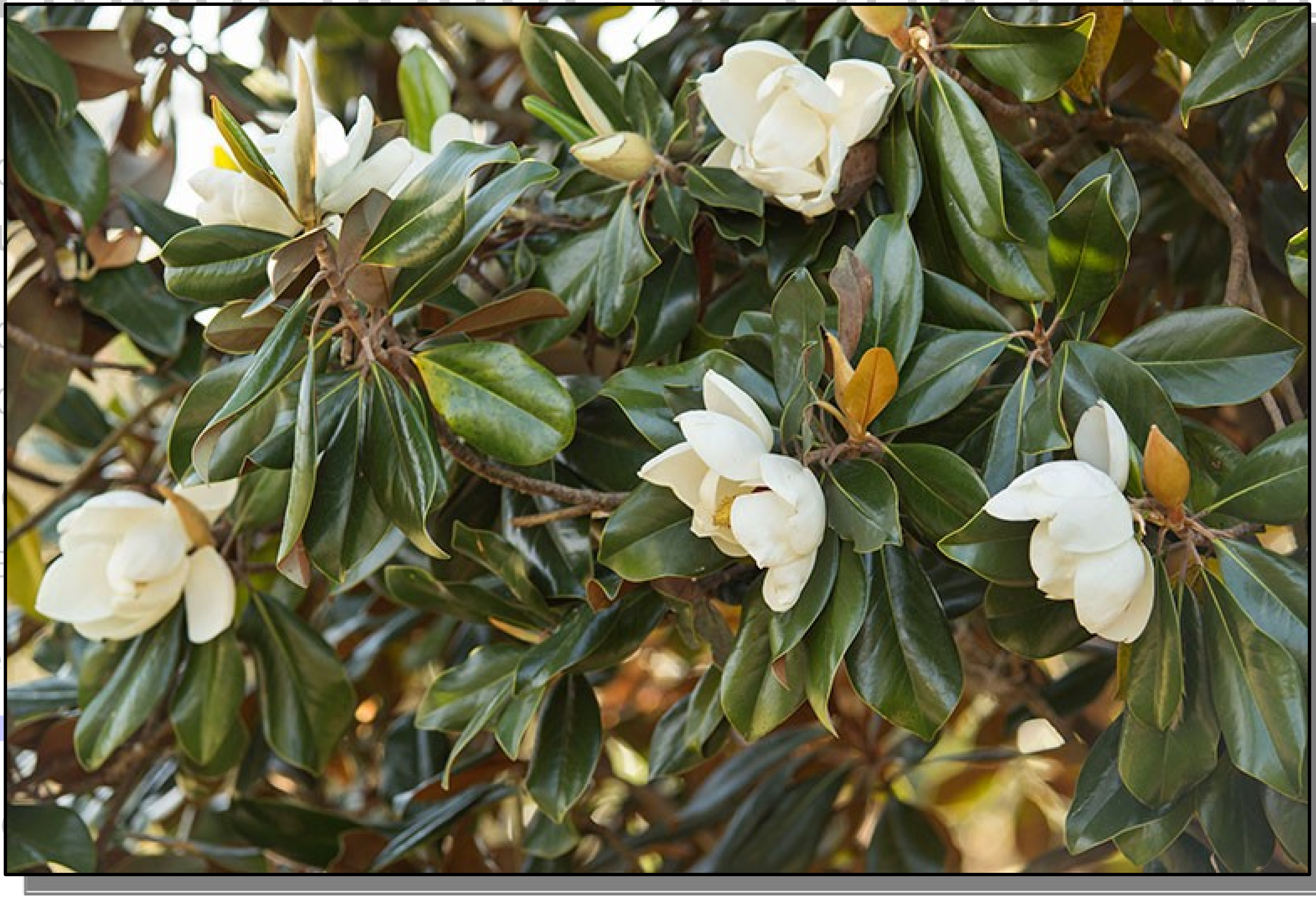
Midterm Exam

- Our midterm exam is next ***Tuesday, February 3rd,*** from ***7–10 PM***.
- Seating assignments are posted. You must take your exam in your assigned seat. Please write your seat number down in case the WiFi cuts out before the exam.
- 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.
- Reminder: No electronic devices may be out at any time during the exam.

A Hodgepodge of Other Things

- Problem Set 3 was due at 1:00 PM. Feel free to use one of your late days if needed.
- Problem Set 4 is posted! This one is shorter than usual to account for the midterm exam.
- The study group bonus due on Monday covers material from the 3-4 preceding lectures.
- Deadline for attendance/participation opt-out is ***tonight (Friday) at 11:59 PM***.
- The magnolia trees are starting to bloom!

A Hodgepodge of Other Things



There’s one by the bike rack at the southeastern corner of STLC, near the northeastern corner of CoDa. Consider stopping by to enjoy the scent of the magnolia blooms!

Graphs

Part 2

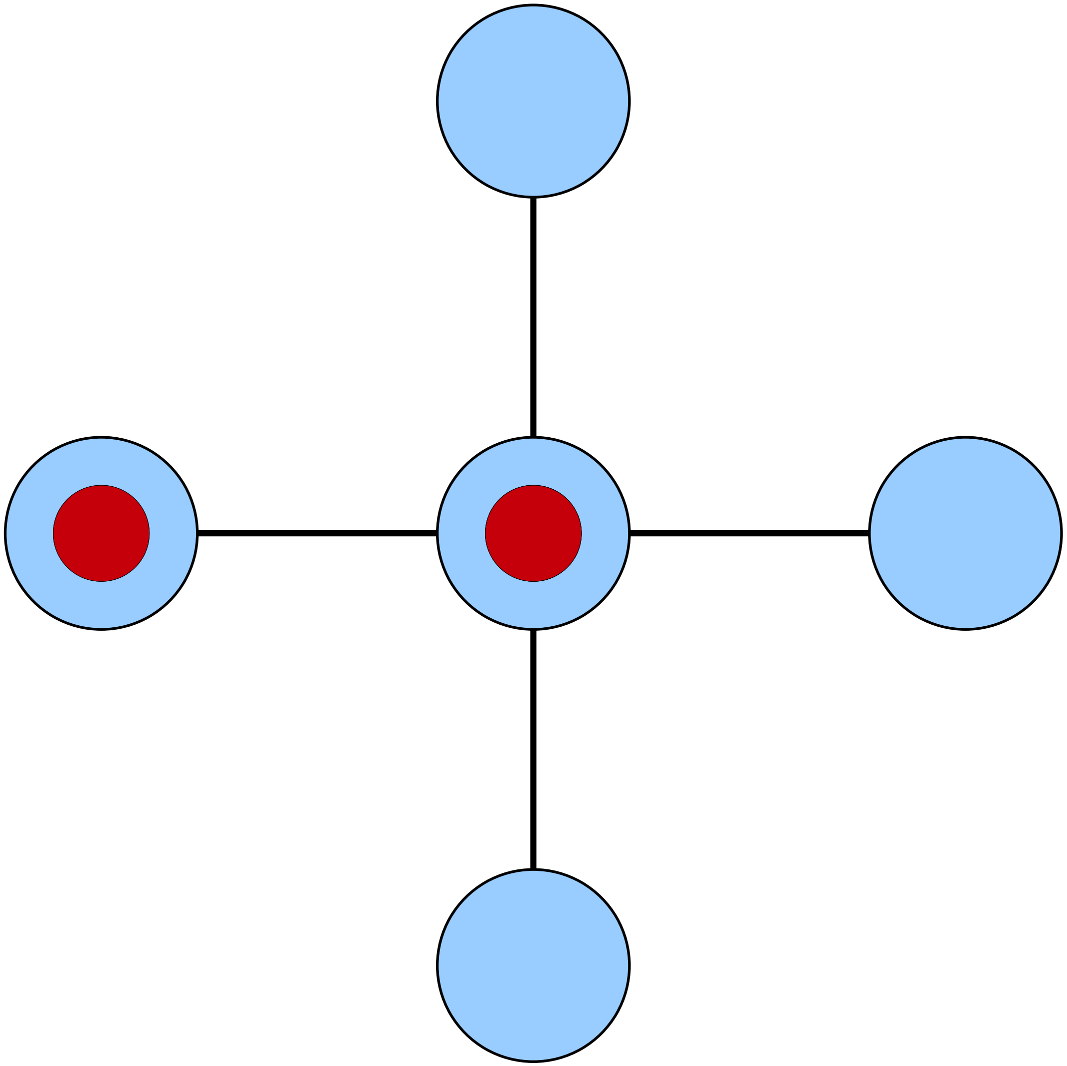
1. Recap from Last Time
2. Preliminary Definition: Adjacency
3. Walks, Paths, and Other Journeys
4. Announcements
5. **Sending Messages through LANs**
6. Shaping LANs (and a Proof on Graphs)
7. Spanning Tree Protocol (STP)
8. Recap and What's Next?

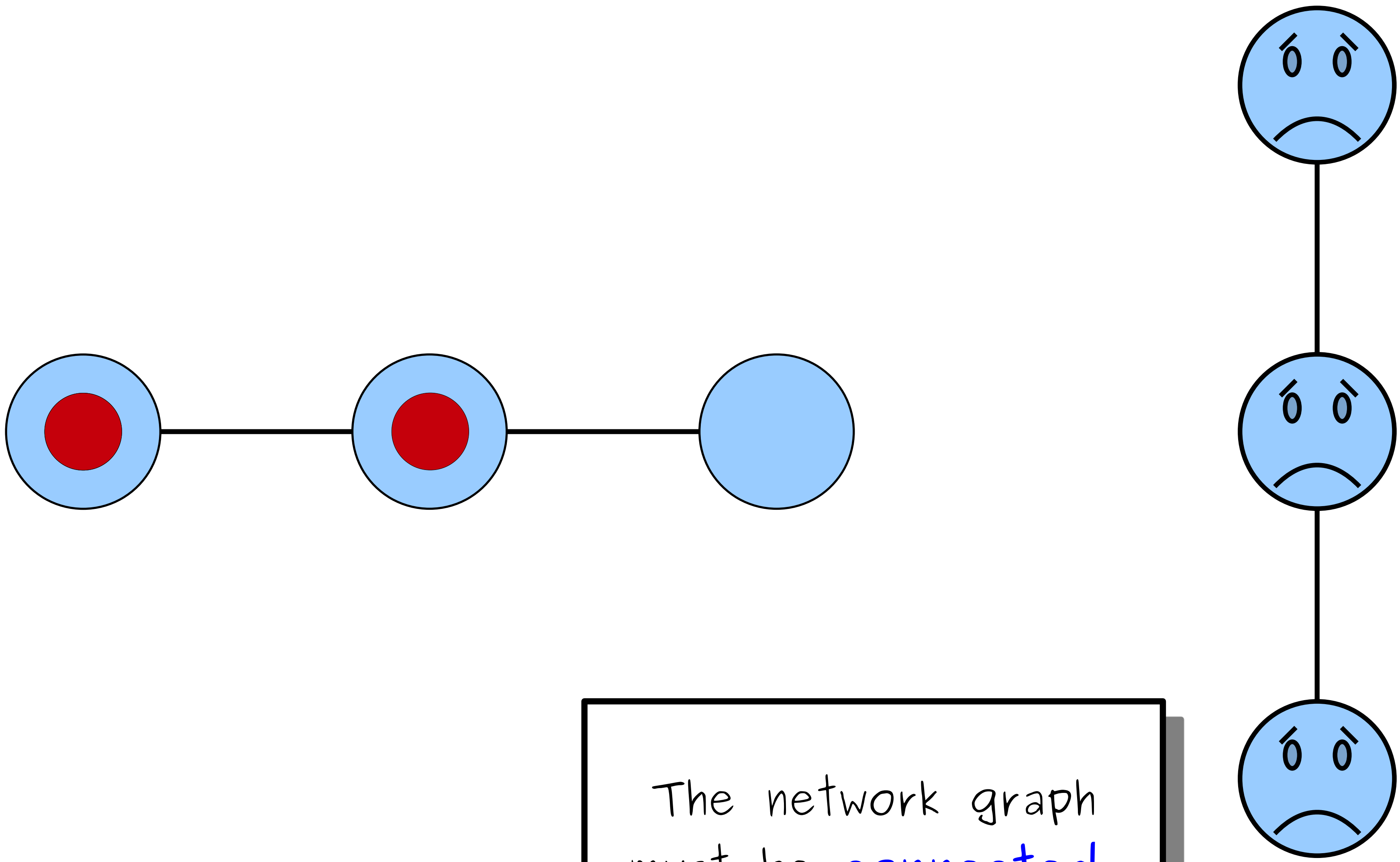
The Internet and LANs

- 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

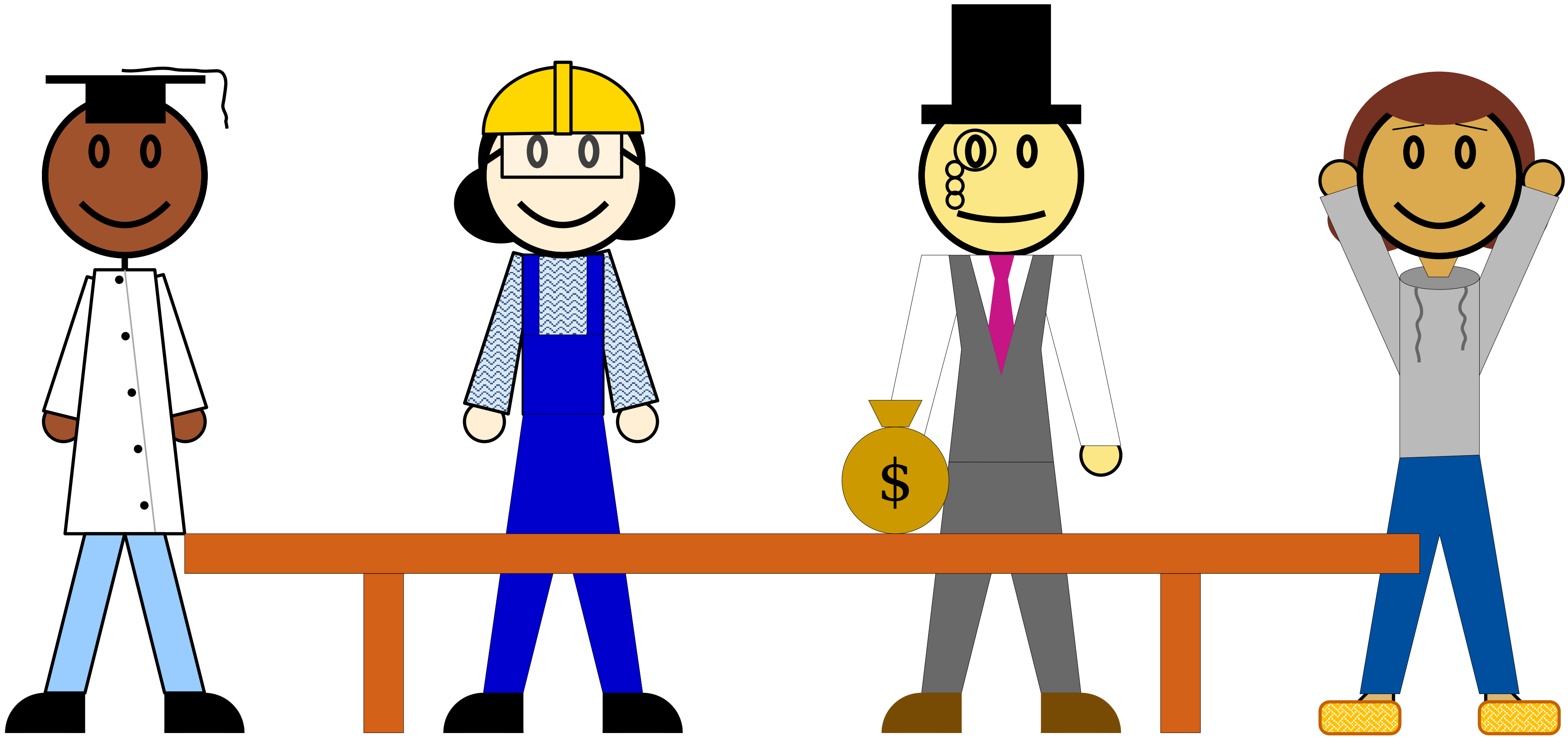
- A ***broadcast storm*** occurs when there's a cycle in the network graph.
- A single message can repeat forever, or exponentially amplify until the network fails.
- ***Solution:*** Don't let the network graph have any cycles.
- A graph $G = (V, E)$ is called ***acyclic*** if it has no cycles.

Graphs

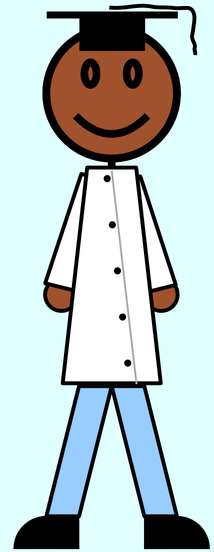
Part 2

1. Recap from Last Time
2. Preliminary Definition: Adjacency
3. Walks, Paths, and Other Journeys
4. Announcements
5. Sending Messages through LANs
6. **Shaping LANs (and a Proof on Graphs)**
7. Spanning Tree Protocol (STP)
8. Recap and What's Next?

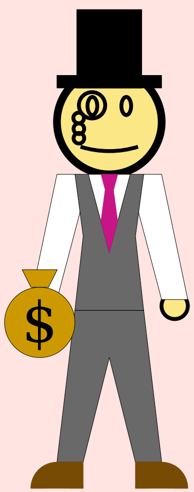
You have a collection of computers that need to be wired up into a LAN. How should you choose the shape of the network?



CTO	COO	CFO	CEO
Connected, No Cycles	Most Links, No Cycles	Fewest Links, Connected	*Do all three!*



Connected, Acyclic

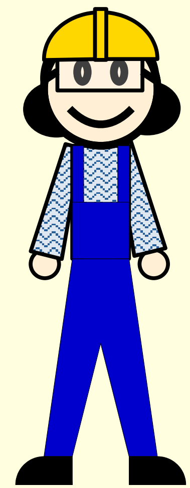


Minimally Connected

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

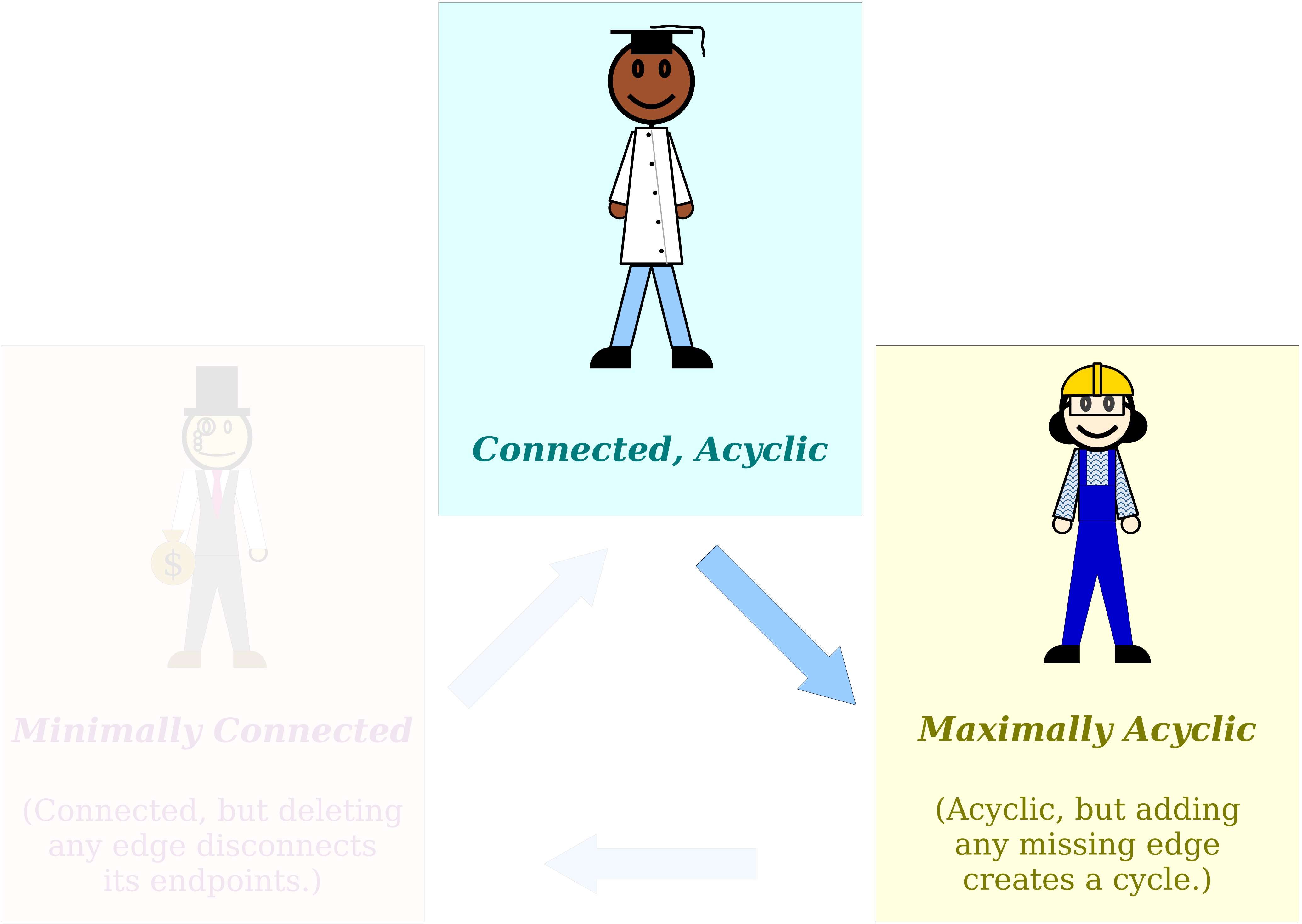
If *any* of these conditions hold, then *all* of these conditions hold.

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



Maximally Acyclic

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



Connected, Acyclic
Minimally Connected
(Connected, but deleting any edge disconnects its endpoints.)
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 $x, \ldots, y$ from x to y in T. Now add $\{x, y\}$ to E. Then we can form the closed walk $x, \ldots, y, x$. We claim that this is a cycle. To see this, note the following:

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

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

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

Graphs

Part 2

1. Recap from Last Time
2. Preliminary Definition: Adjacency
3. Walks, Paths, and Other Journeys
4. Announcements
5. Sending Messages through LANs
6. Shaping LANs (and a Proof on Graphs)
7. **Spanning Tree Protocol (STP)**
8. Recap and What's Next?

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.

Graphs

Part 2

1. Recap from Last Time
2. Preliminary Definition: Adjacency
3. Walks, Paths, and Other Journeys
4. Announcements
5. Sending Messages through LANs
6. Shaping LANs (and a Proof on Graphs)
7. Spanning Tree Protocol (STP)
8. **Recap and What's Next?**

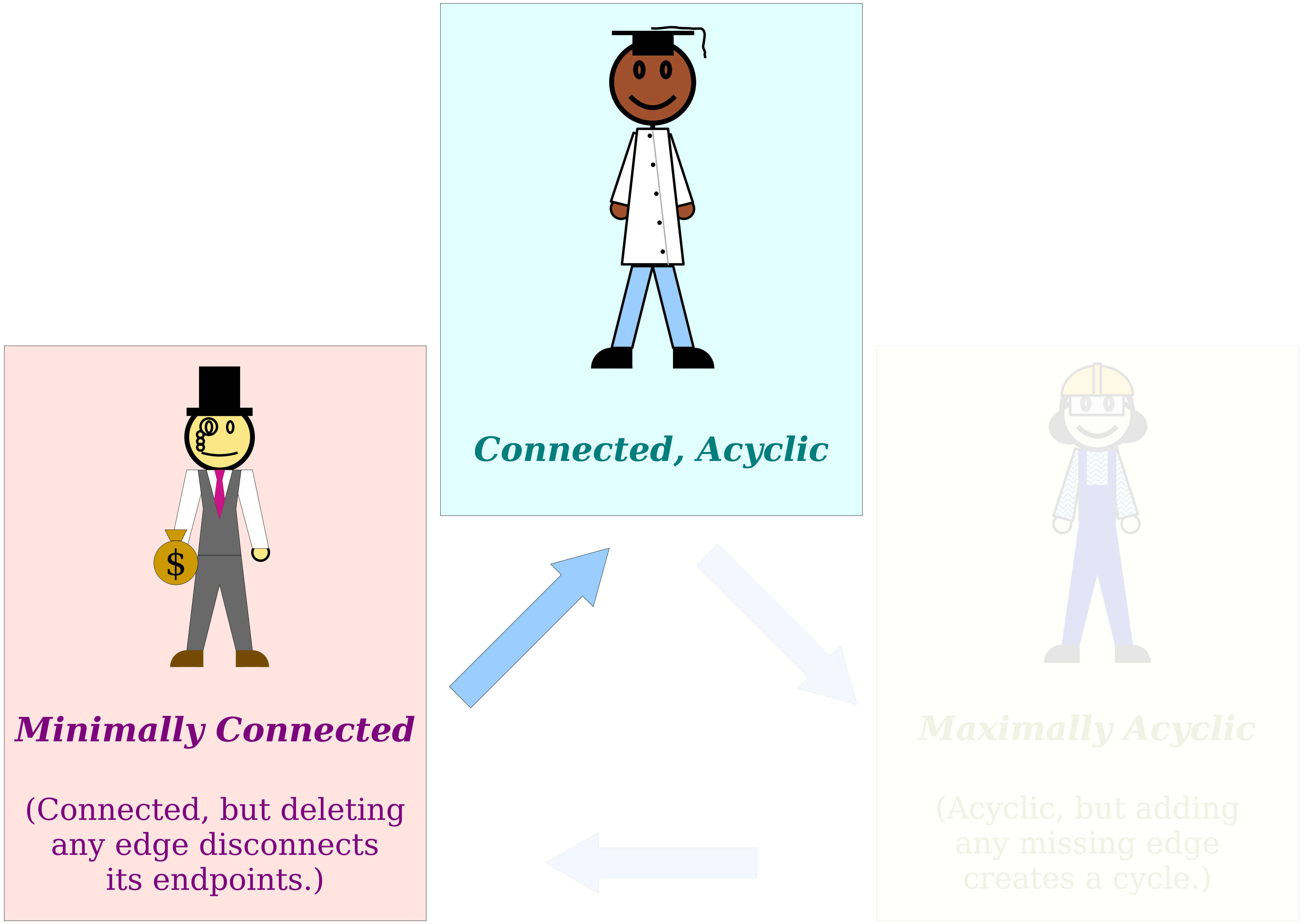
Recap from Today

- ***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


Connected, Acyclic
Minimally Connected
(Connected, but deleting any edge disconnects its endpoints.)
Maximally Acyclic
(Acyclic, but adding any missing edge creates a cycle.)

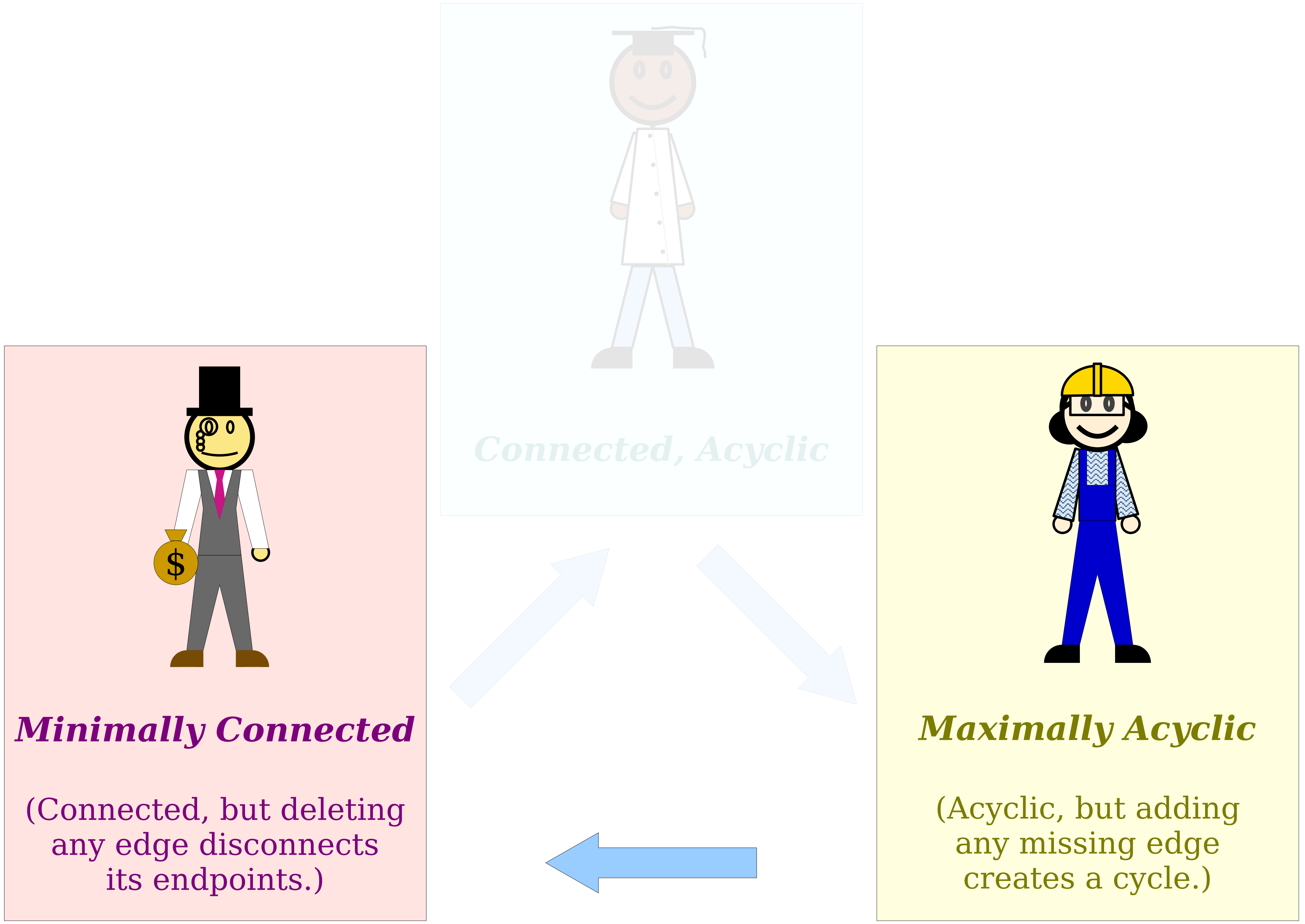
Theorem: Let $T = (V, E)$ be a graph. If T is minimally connected, then T is connected and acyclic.

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 $x, \ldots, y, x$. Note in particular that this means $x, \ldots, y$ 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 $x, \ldots, y$ 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. ■


Connected, Acyclic
Minimally Connected
(Connected, but deleting any edge disconnects its endpoints.)
Maximally Acyclic
(Acyclic, but adding any missing edge creates a cycle.)
$

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 $x, \ldots, y, x$ passing through $\{x, y\}$. Then $x, \ldots, y$ 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 $x, \ldots, y$ in T after removing $\{x, y\}$. By adding $\{x, y\}$ to the end of the path, we form a cycle $x, \ldots, y, x$ in T. This is impossible because T is acyclic. We've reached a contradiction, so our assumption was wrong and T is minimally connected. ■