

Graph Theory

Part Two

Outline for Today

- ***Walks, Paths, and Reachability***
 - Walking around a graph.
- ***Graph Complements***
 - Flipping what's in a graph.
- ***The Pigeonhole Principle***
 - Everyone finding a place.

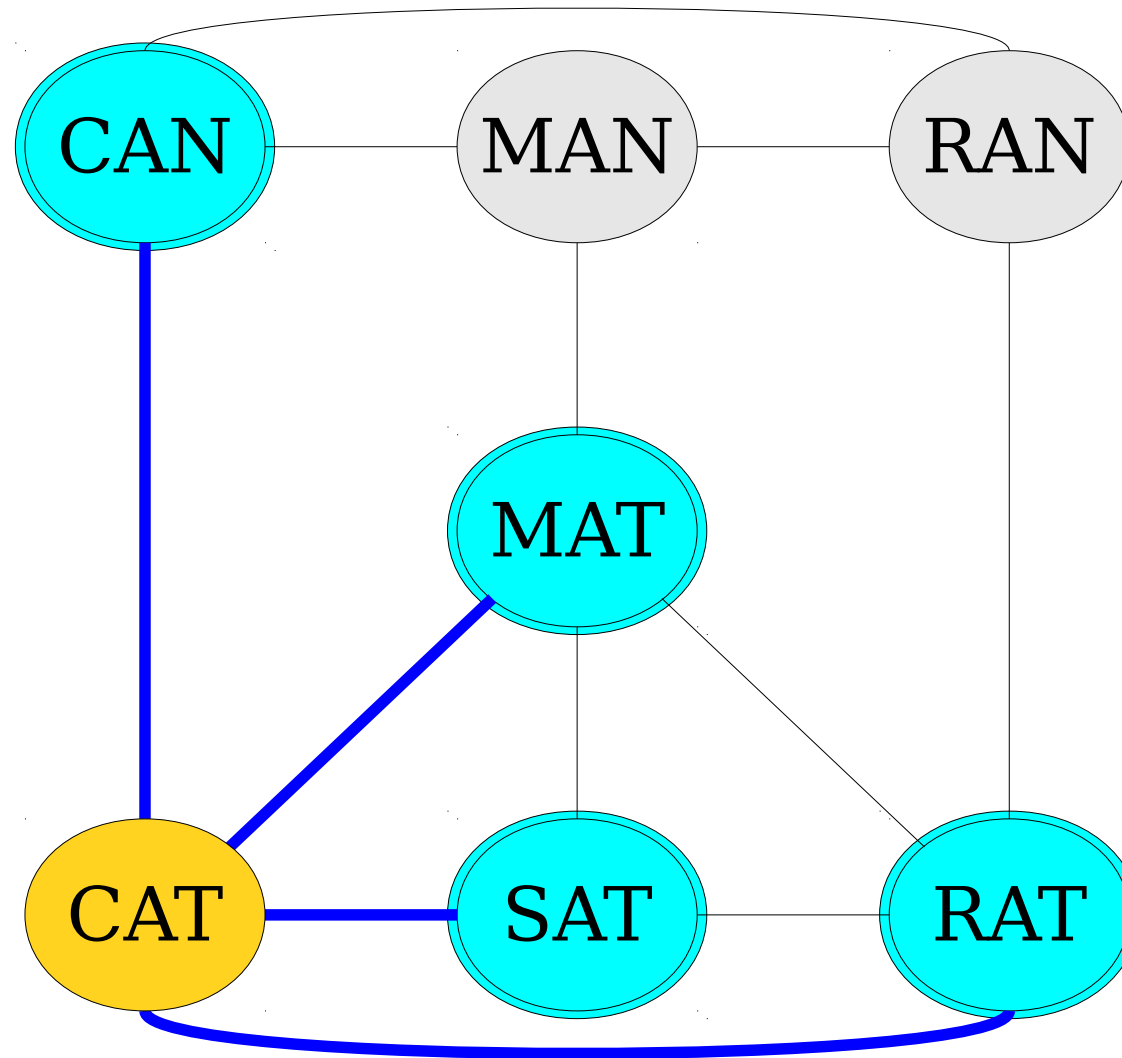
Recap from Last Time

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** (i.e., sets with cardinality 2) of nodes. If $\{u, v\} \in E$, then there's an edge from u to v .

New Stuff!

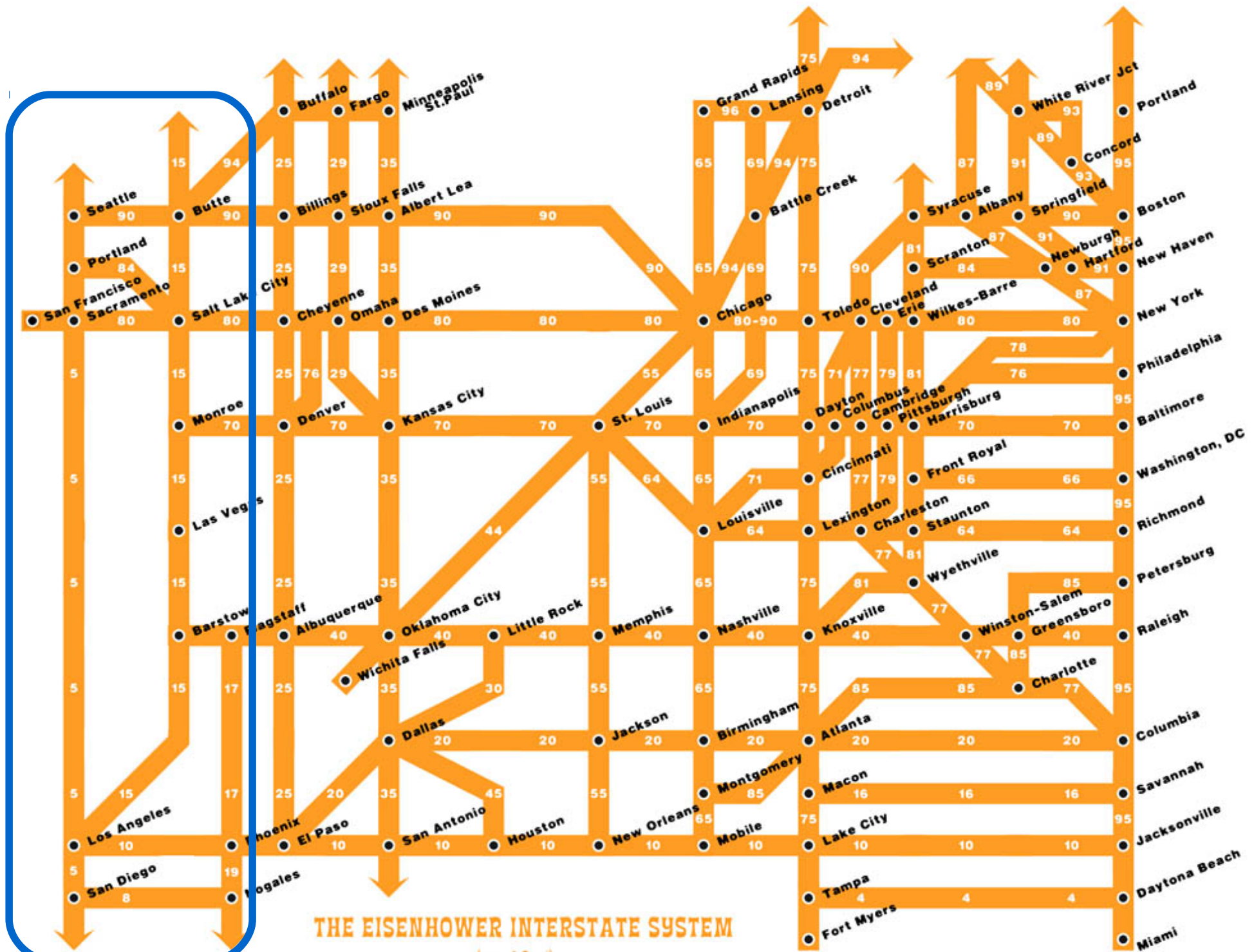
Walks, Paths, and Reachability

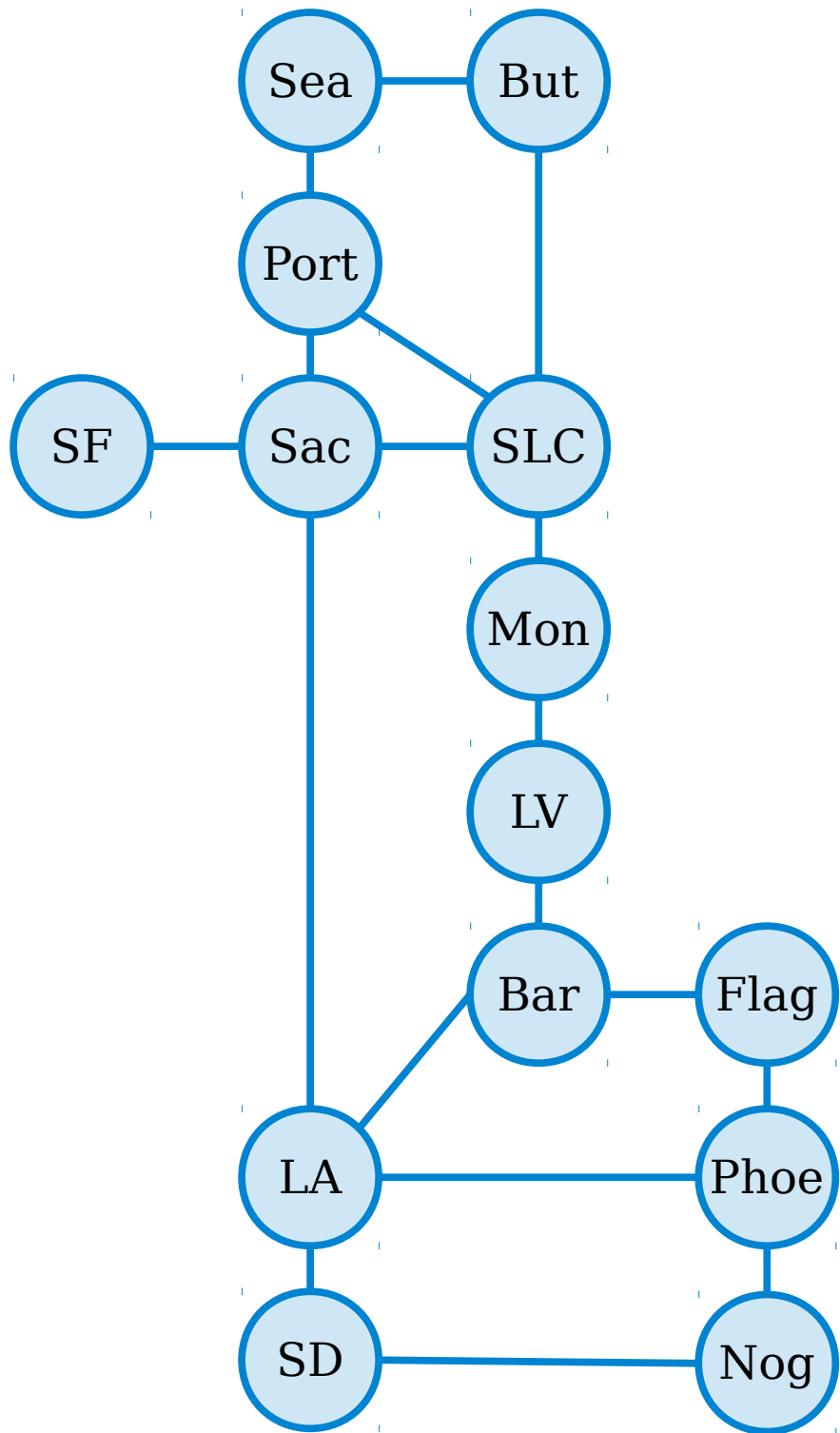


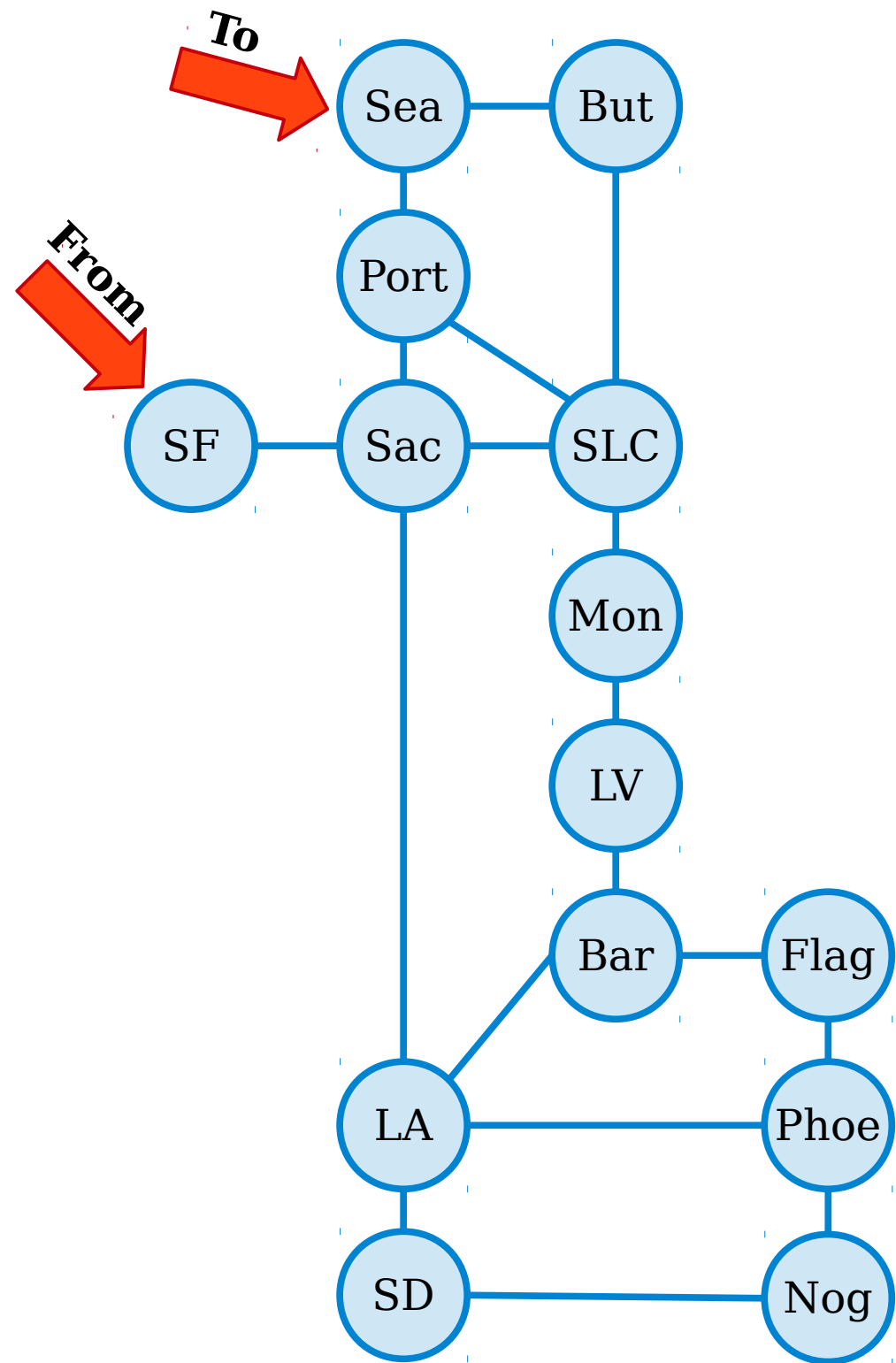
Two nodes are called *adjacent* if there is an edge between them.

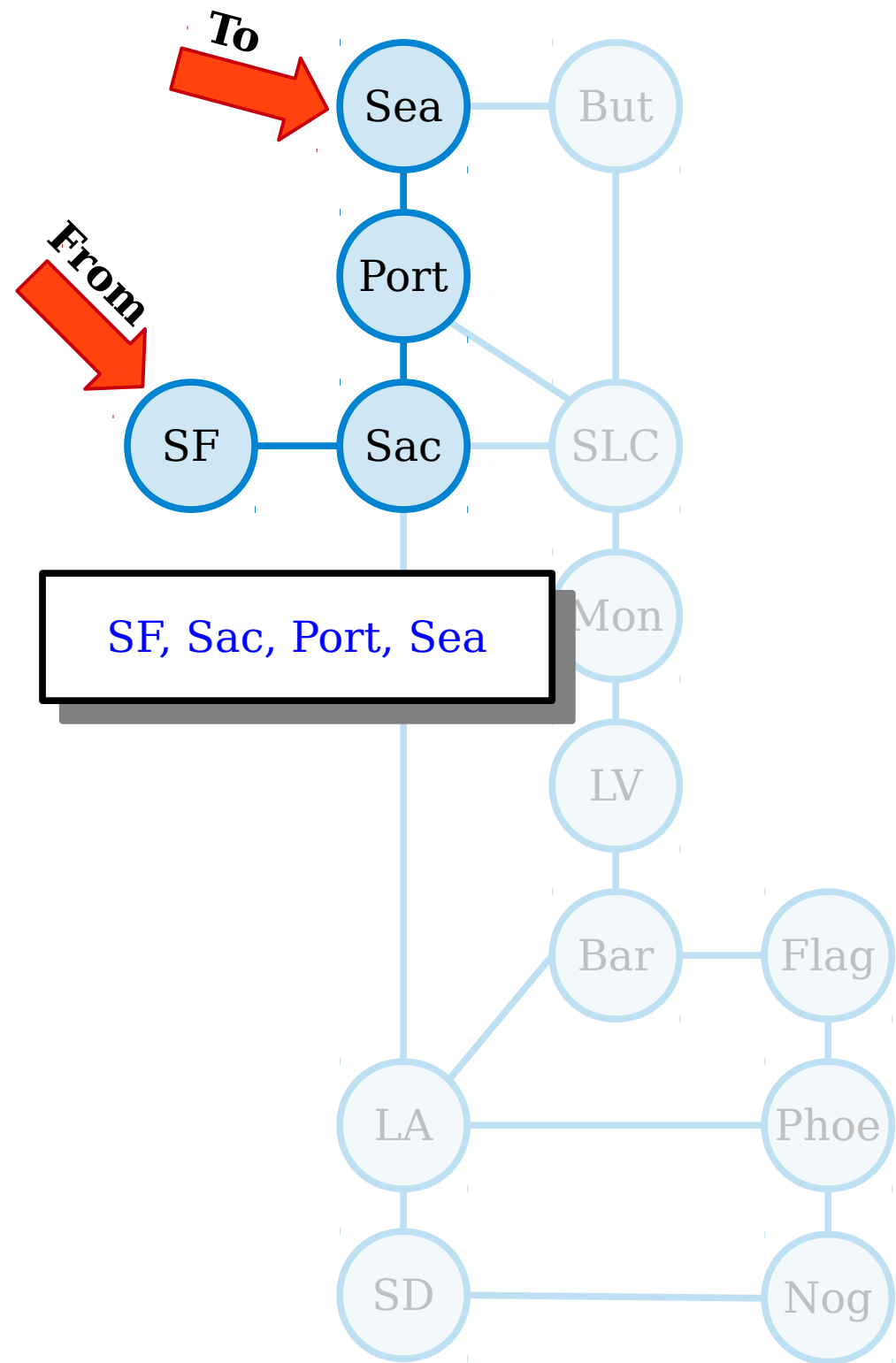
Using our Formalisms

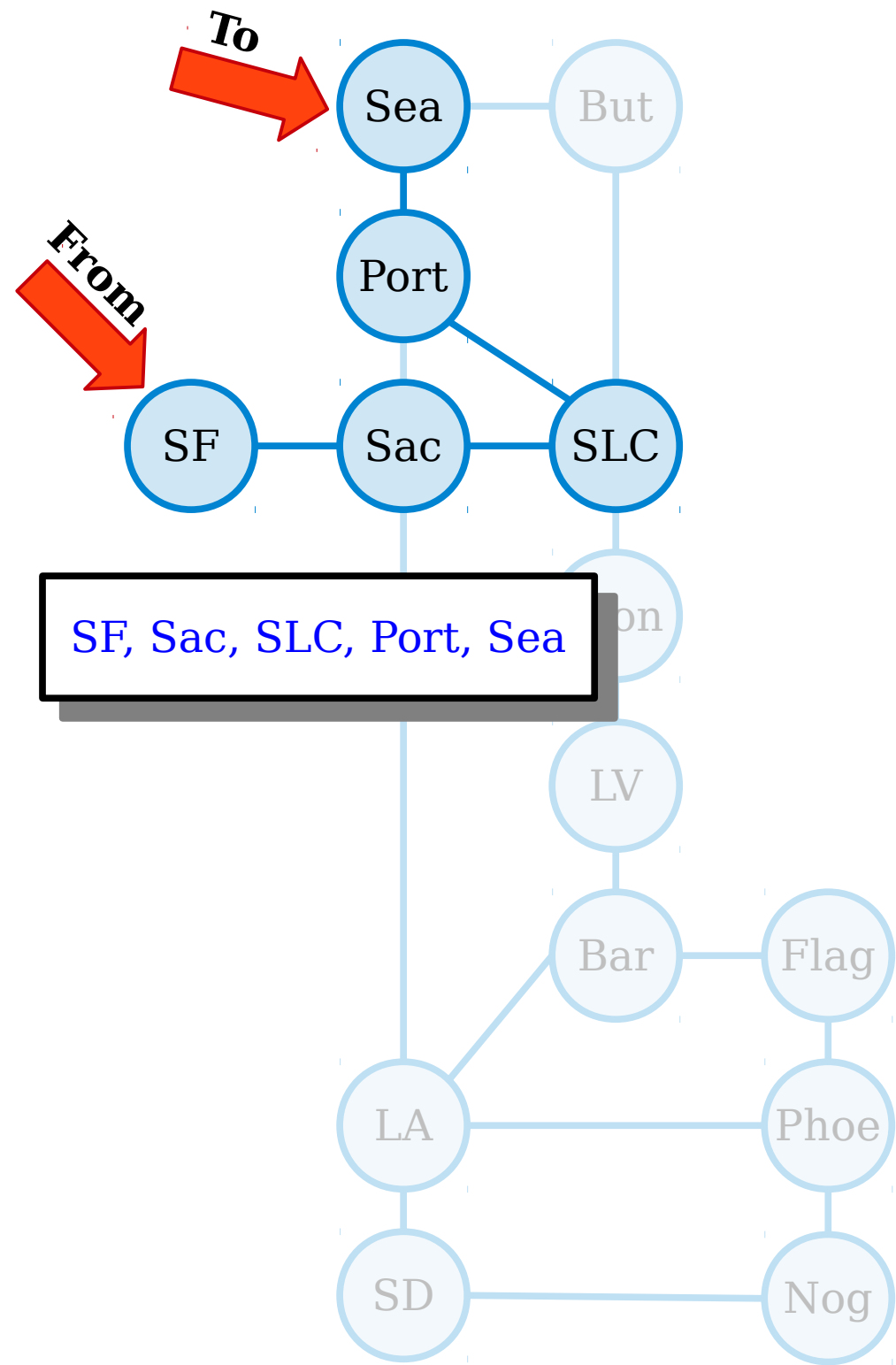
- Let $G = (V, E)$ be a graph.
- Intuitively, two nodes are adjacent if they're linked by an edge.
- Formally speaking, we say that two nodes $u, v \in V$ are ***adjacent*** if we have $\{u, v\} \in E$.

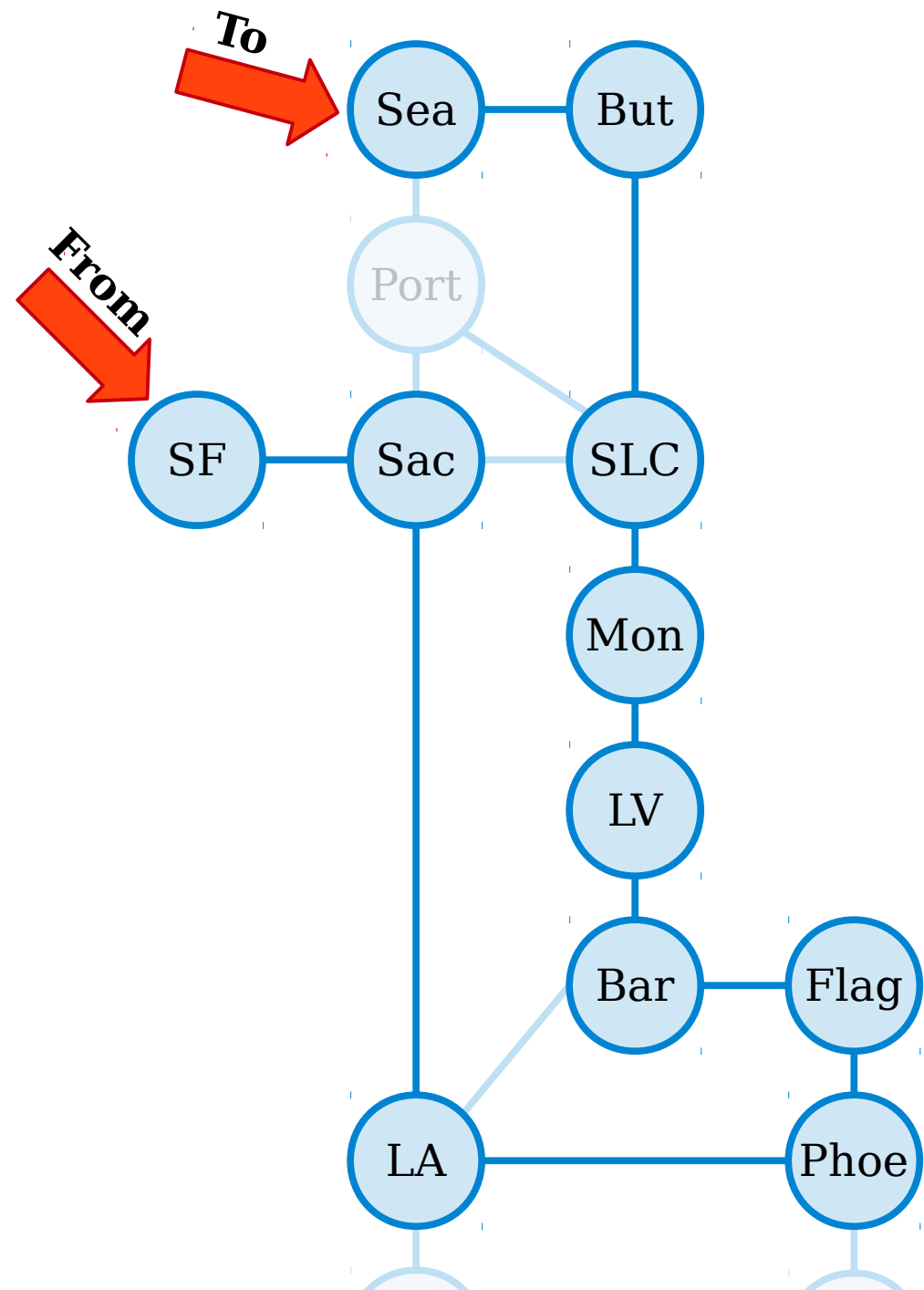






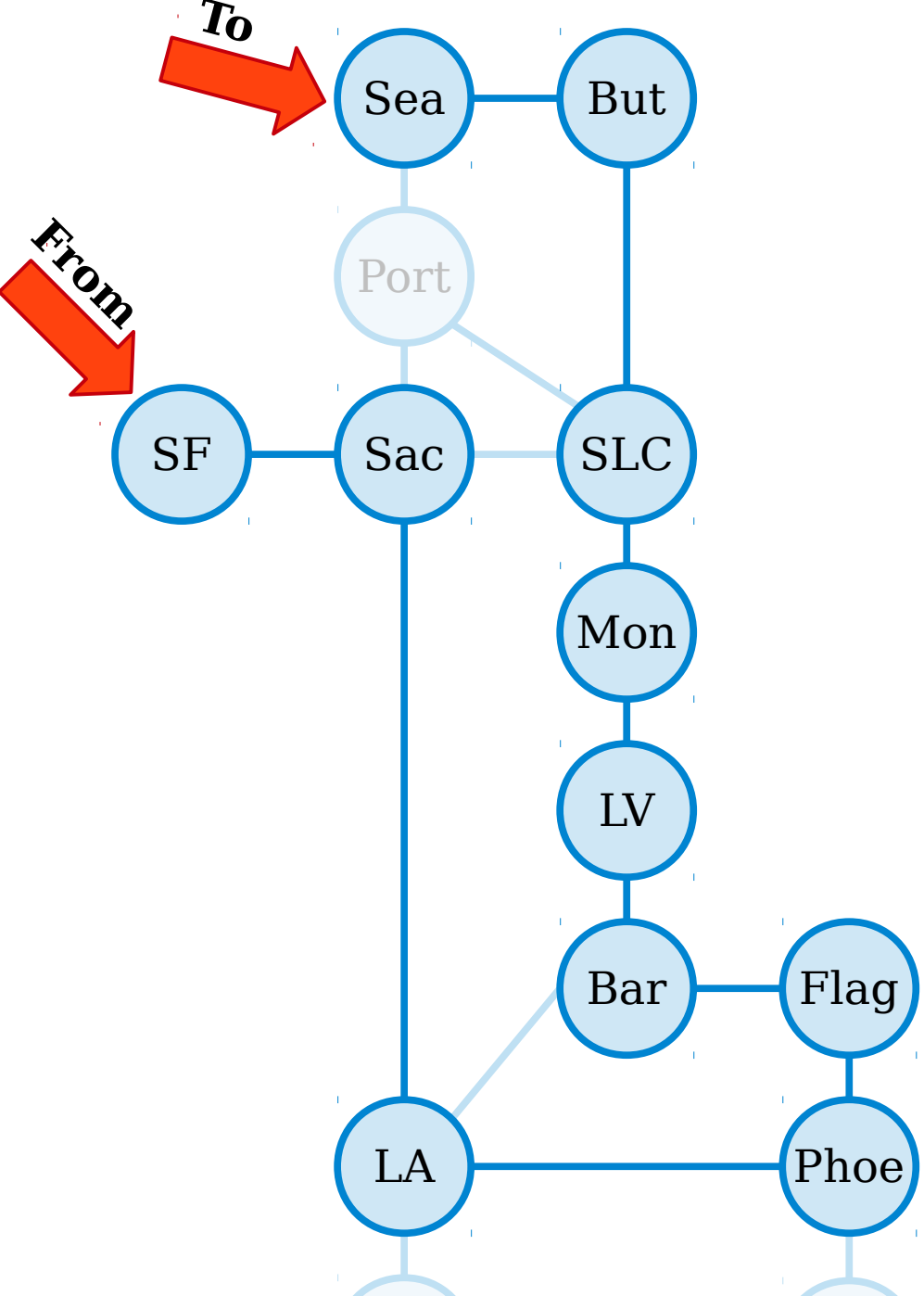




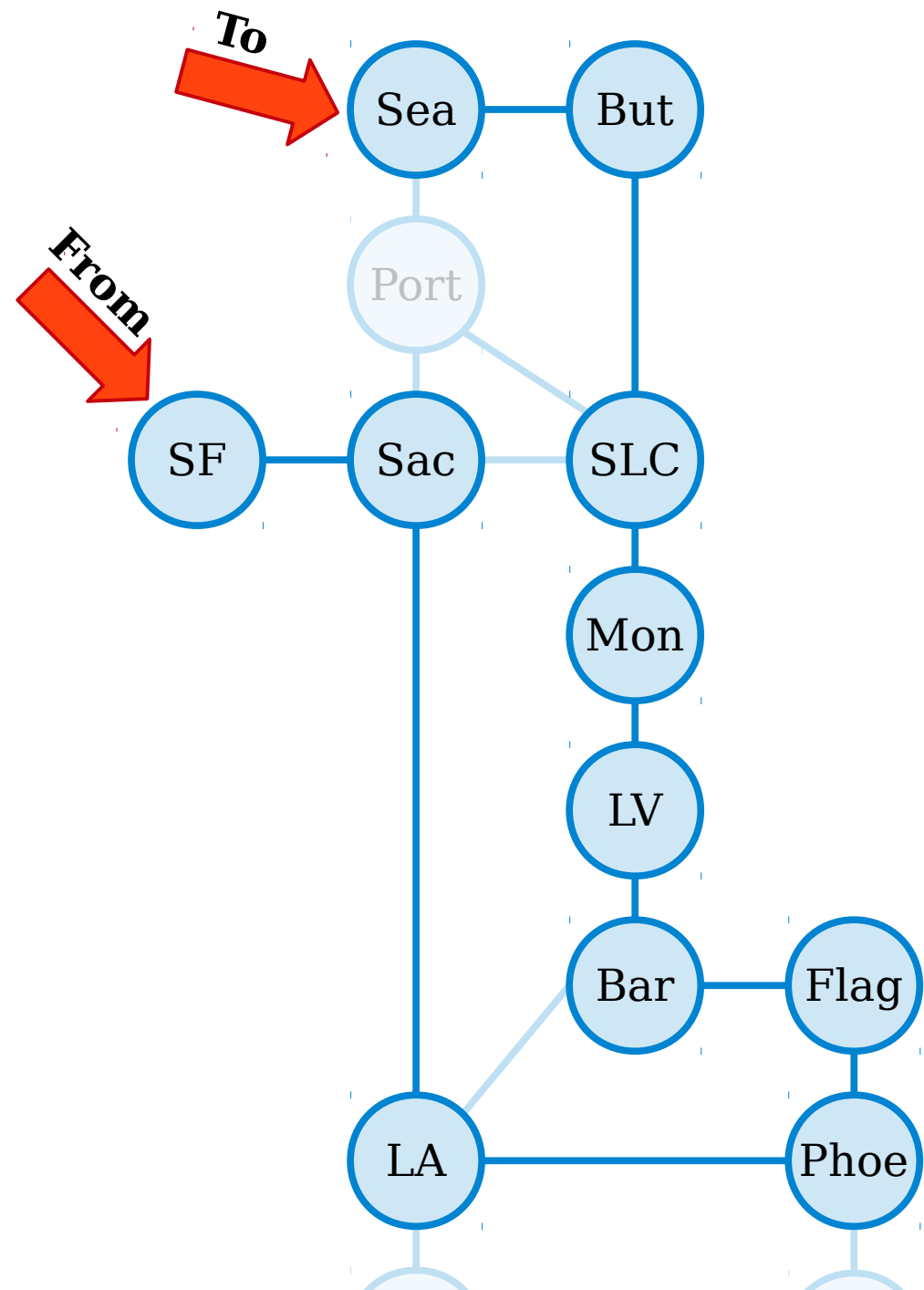


SF, Sac, LA, Phoe, Flag, Bar, LV, Mon, SLC, But, Sea

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.



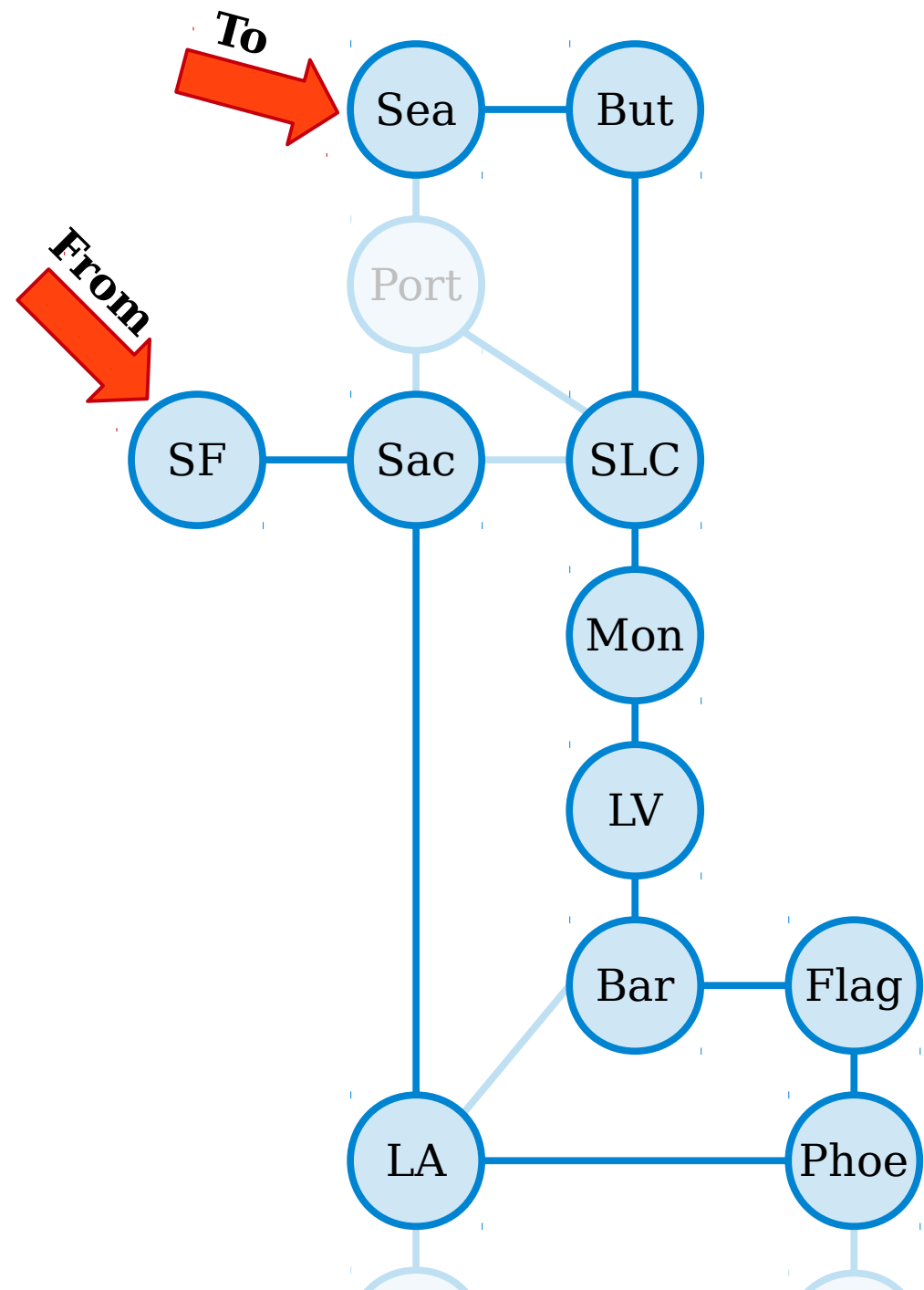
SF, Sac, LA, Phoe, Flag, Bar, LV, Mon, SLC, But, Sea



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 $n - 1$.

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

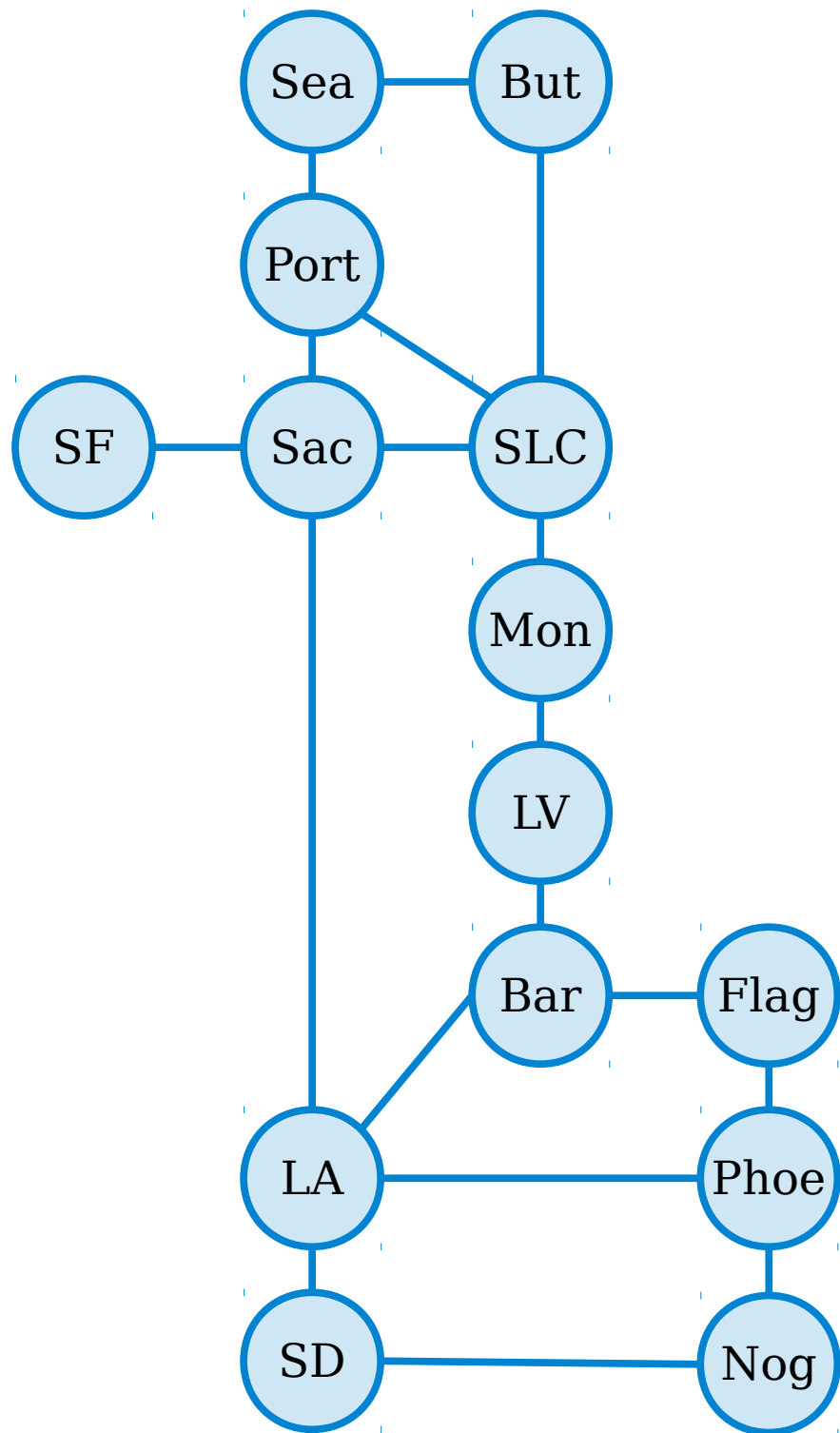


A **walk** in a graph $G = (V, E)$ is a sequence of one or more nodes $v_1, v_2, v_3, \dots, v_n$ such that any two consecutive nodes in the sequence are adjacent.

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

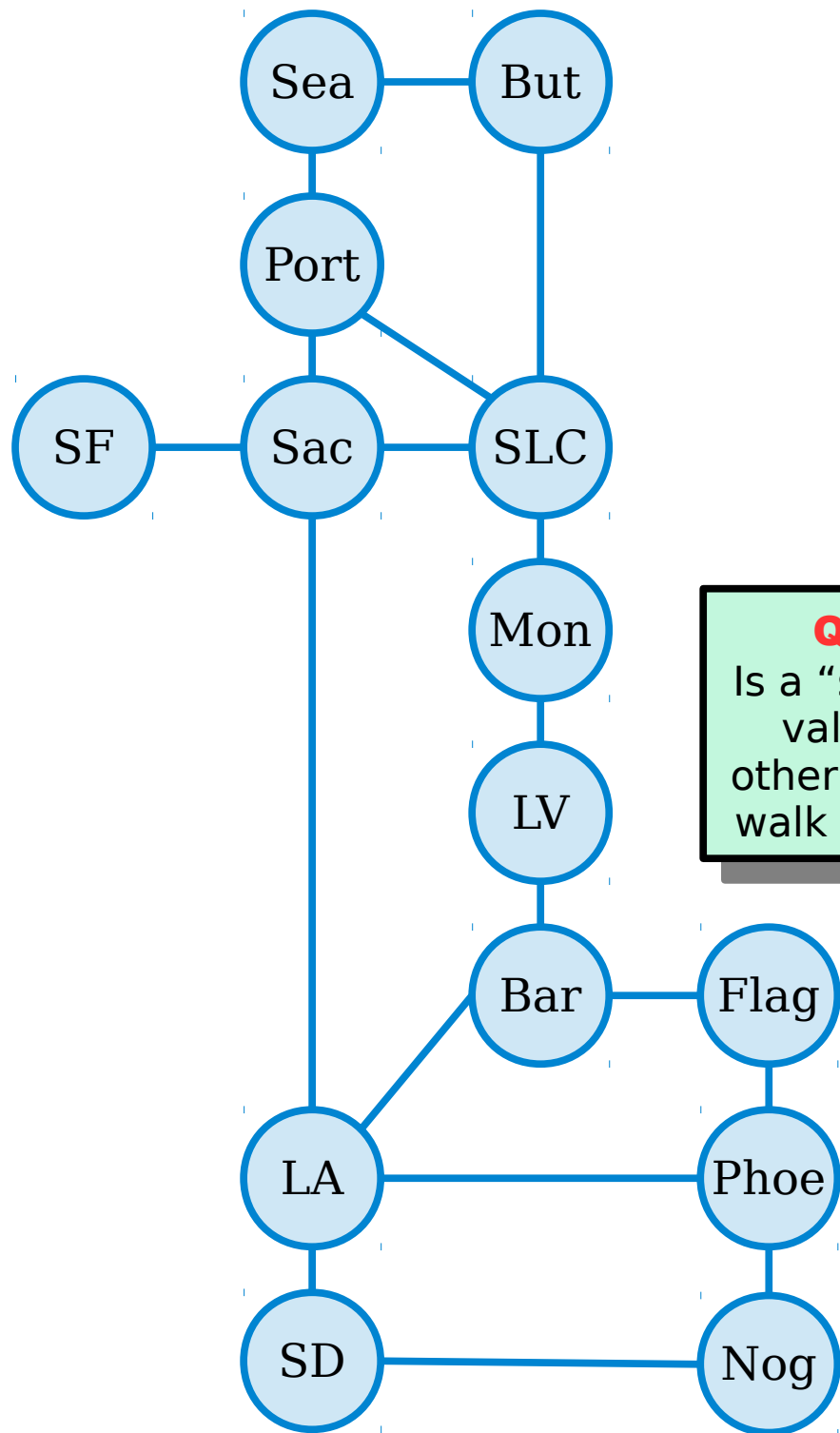
(This walk has length 10, but visits 11 cities.)

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 $n - 1$.



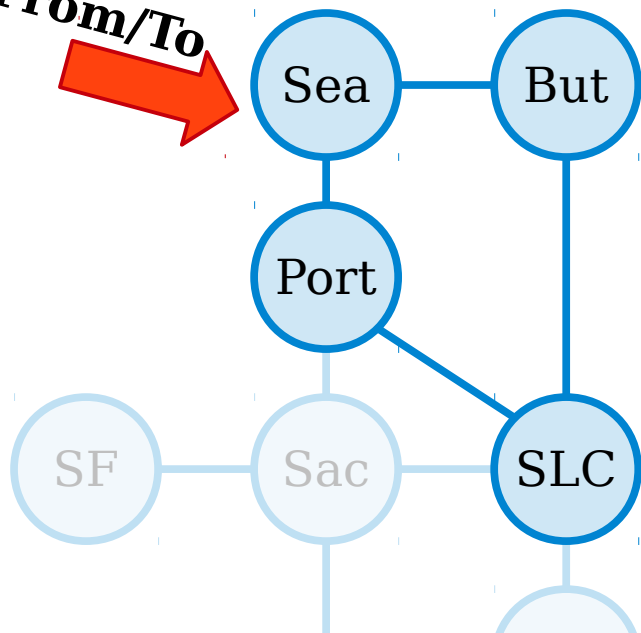
A **walk** in a graph $G = (V, E)$ is a sequence of one or more nodes $v_1, v_2, v_3, \dots, v_n$ such that any two consecutive nodes in the sequence are adjacent.

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

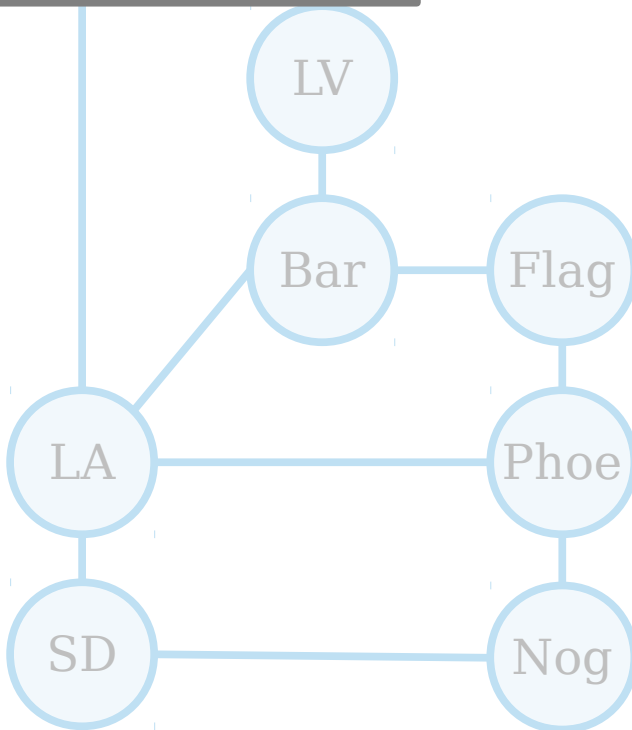
Question:
Is a “staycation” a valid walk? In other words, can a walk be just “SF”?

PollEv.com/
cs103spr26

From/To

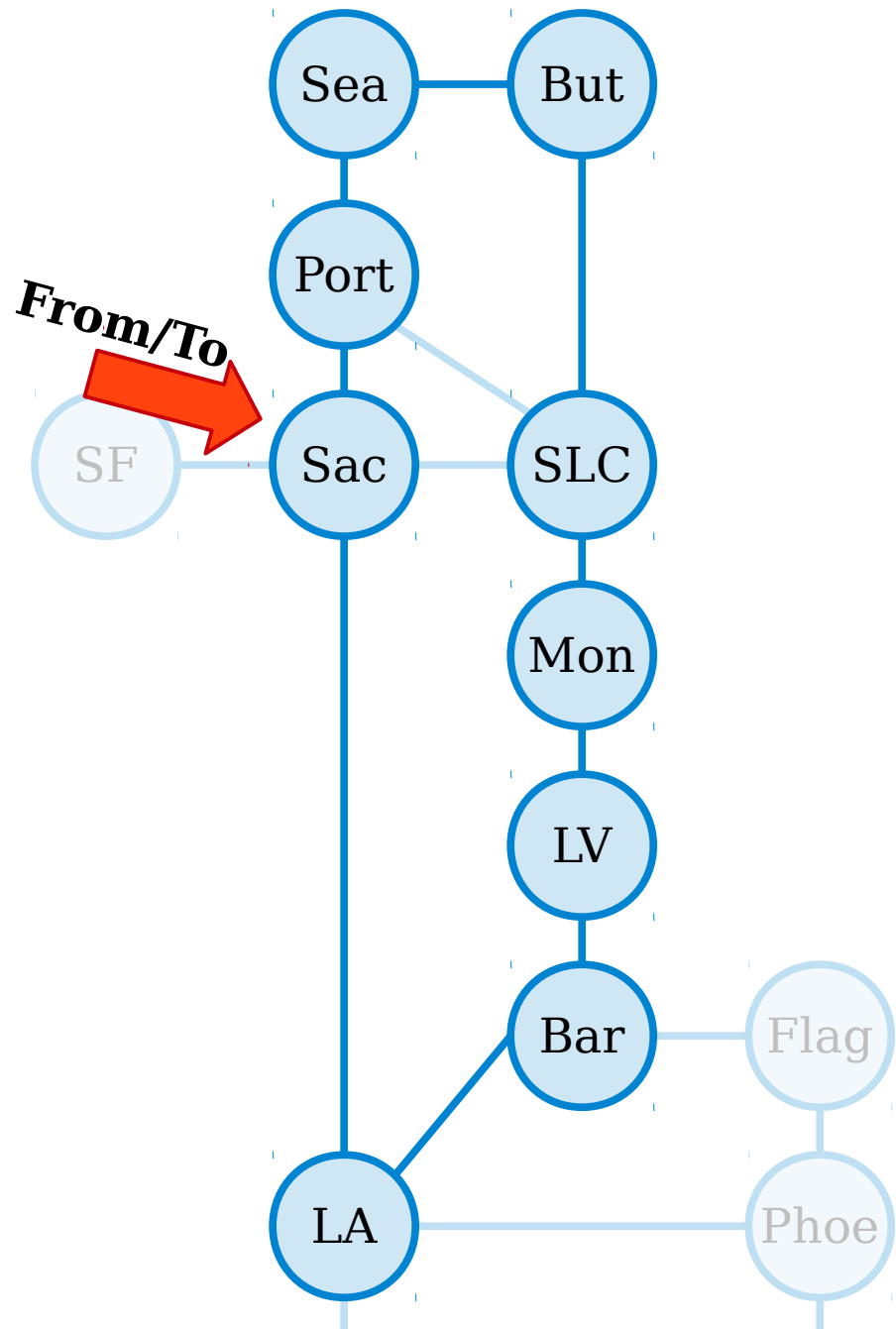


Sea, But, SLC, Port, Sea



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.

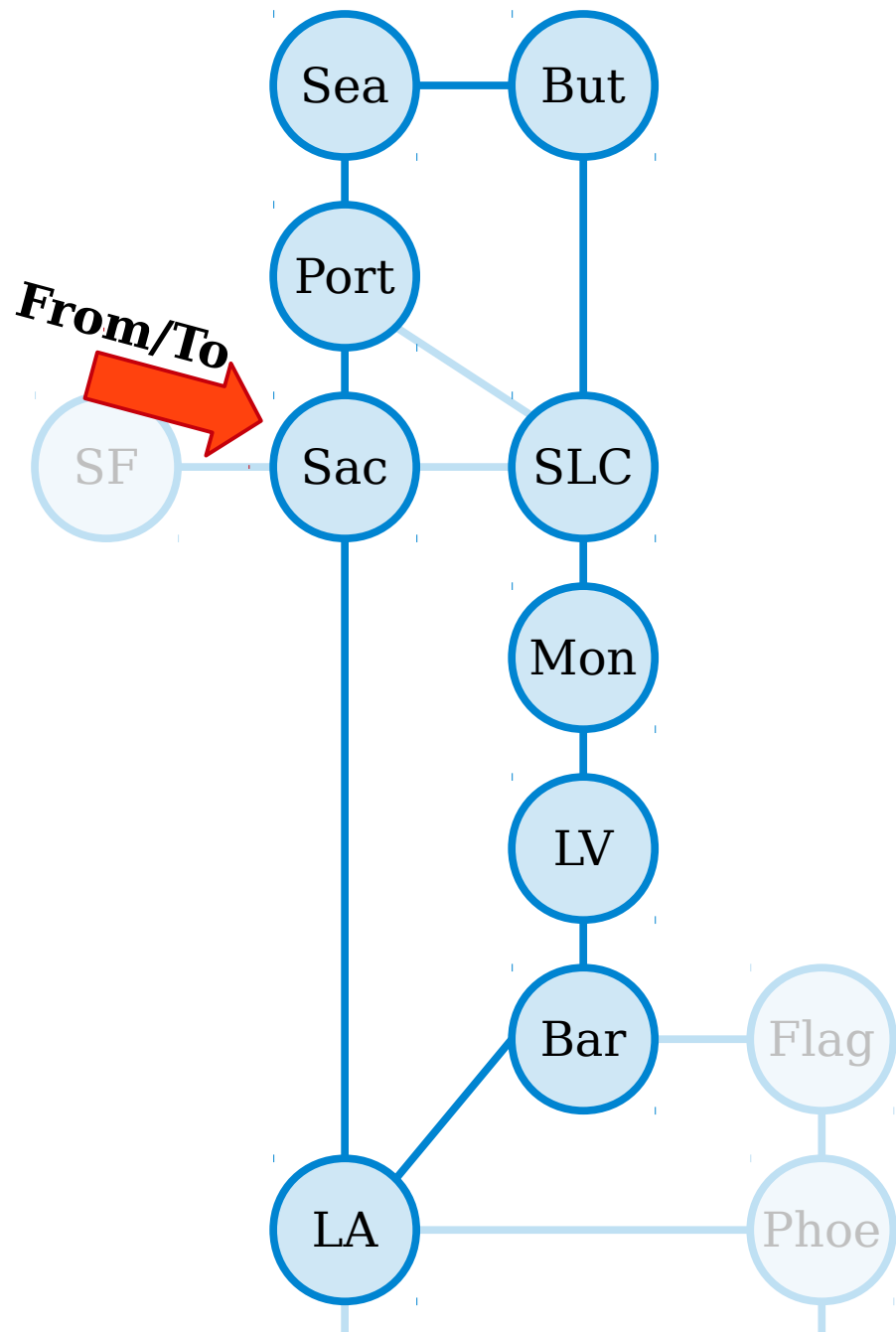
The **length** of the walk v_1, \dots, v_n is $n - 1$.



A **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 $n - 1$.

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



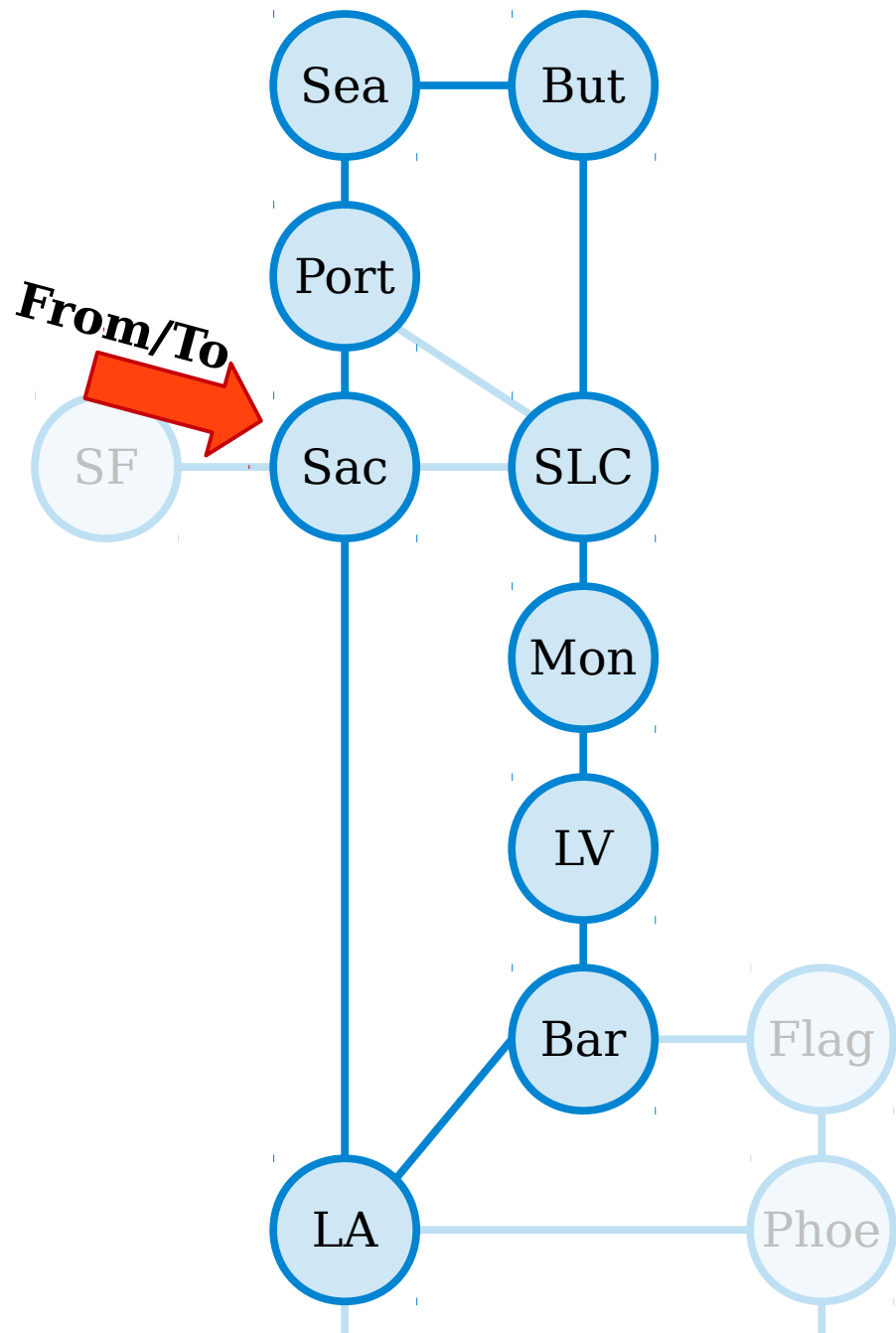
A **walk** in a graph $G = (V, E)$ is a sequence of one or more nodes $v_1, v_2, v_3, \dots, v_n$ such that any two consecutive nodes in the sequence are adjacent.

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.)

(No "staycation" in closed walks, because of this rule.)

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



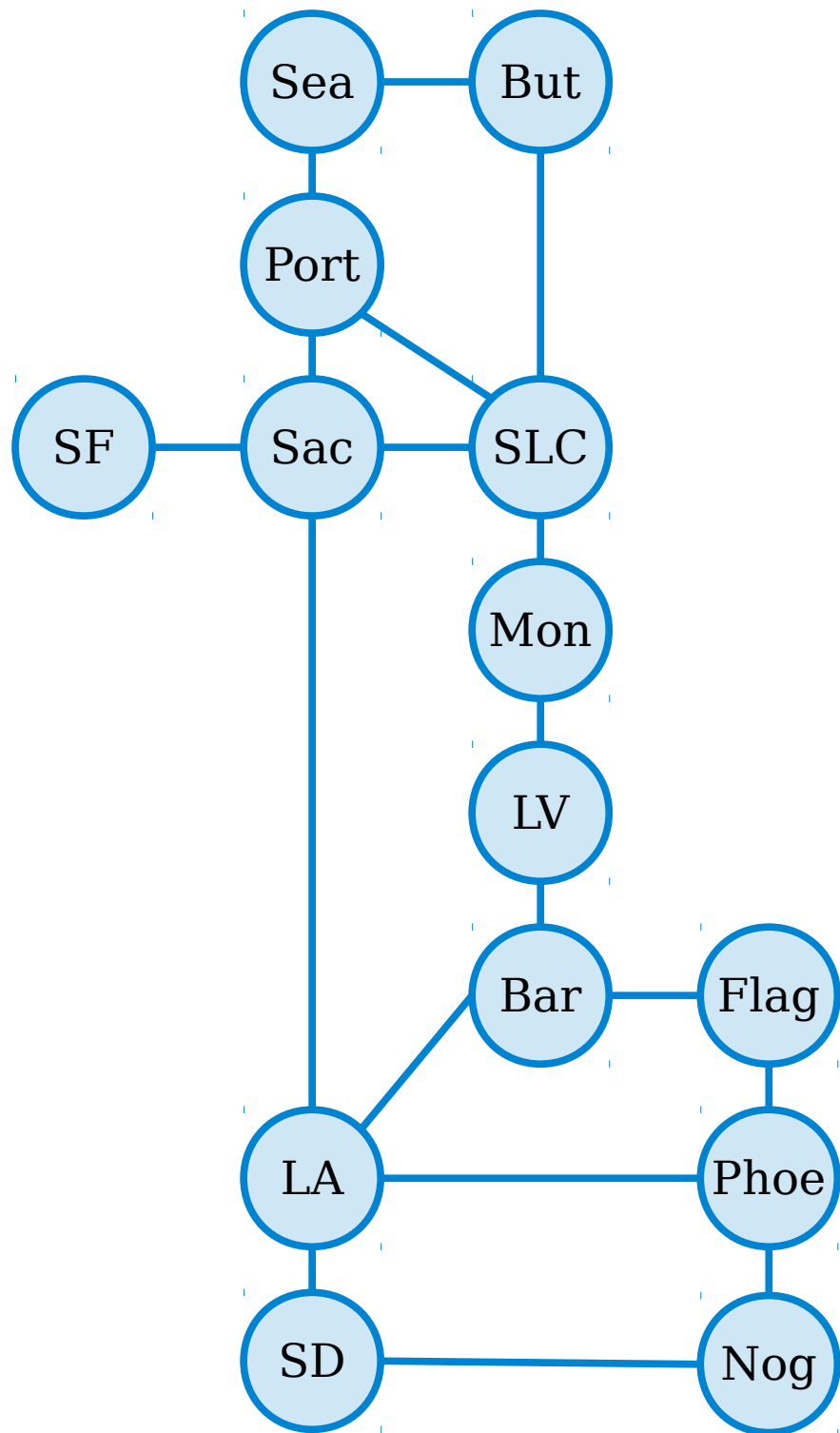
A **walk** in a graph $G = (V, E)$ is a sequence of one or more nodes $v_1, v_2, v_3, \dots, v_n$ such that any two consecutive nodes in the sequence are adjacent.

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.)

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

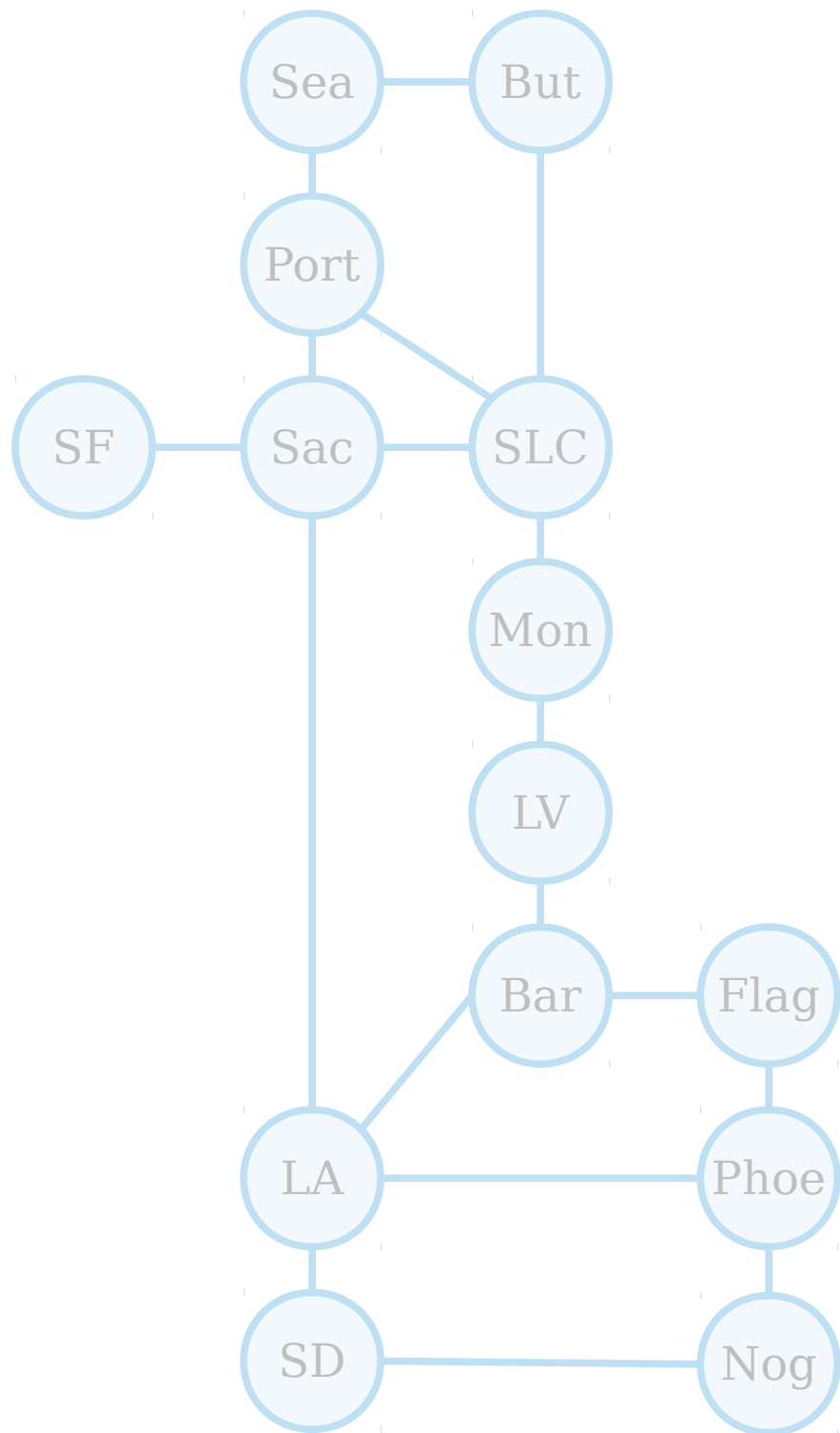
Sac, Port, Sea, But, SLC, Mon, LV, Bar, LA, Sac



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 $n - 1$.

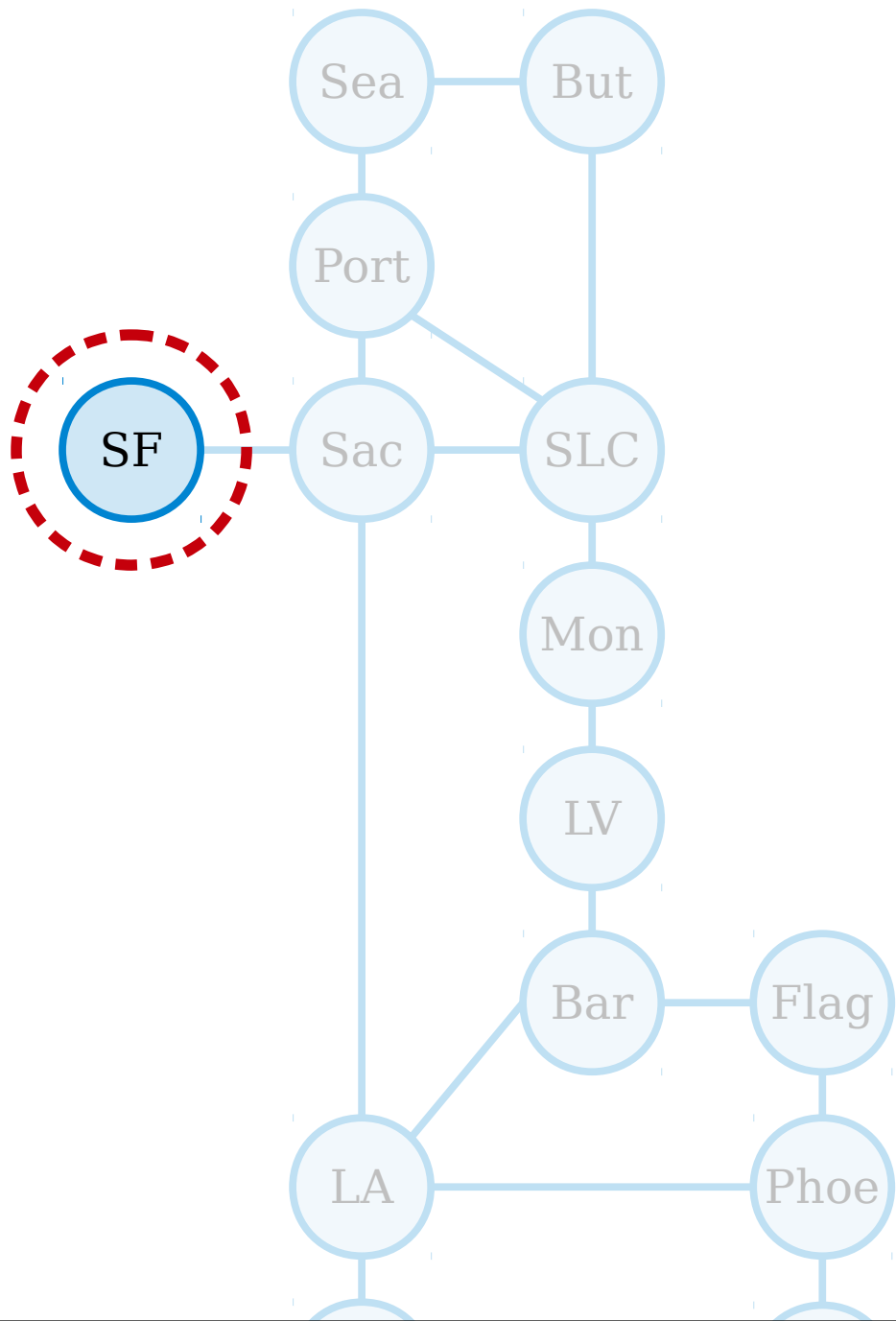
A **closed walk** in a graph is a walk from a node back to itself. (By convention, a closed walk cannot have length zero.)



A **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.)

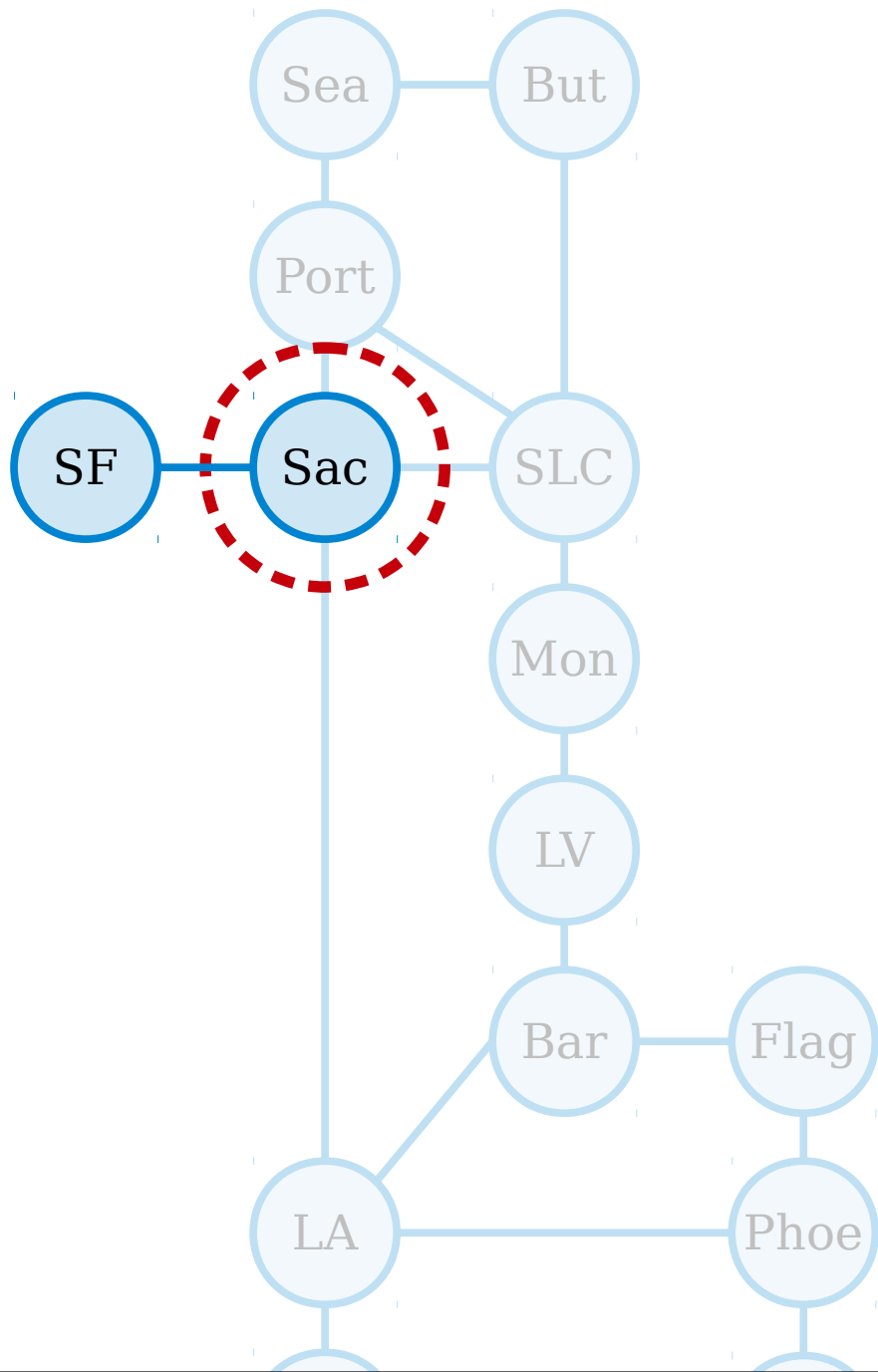


A **walk** in a graph $G = (V, E)$ is a sequence of one or more nodes $v_1, v_2, v_3, \dots, v_n$ such that any two consecutive nodes in the sequence are adjacent.

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.)

SF

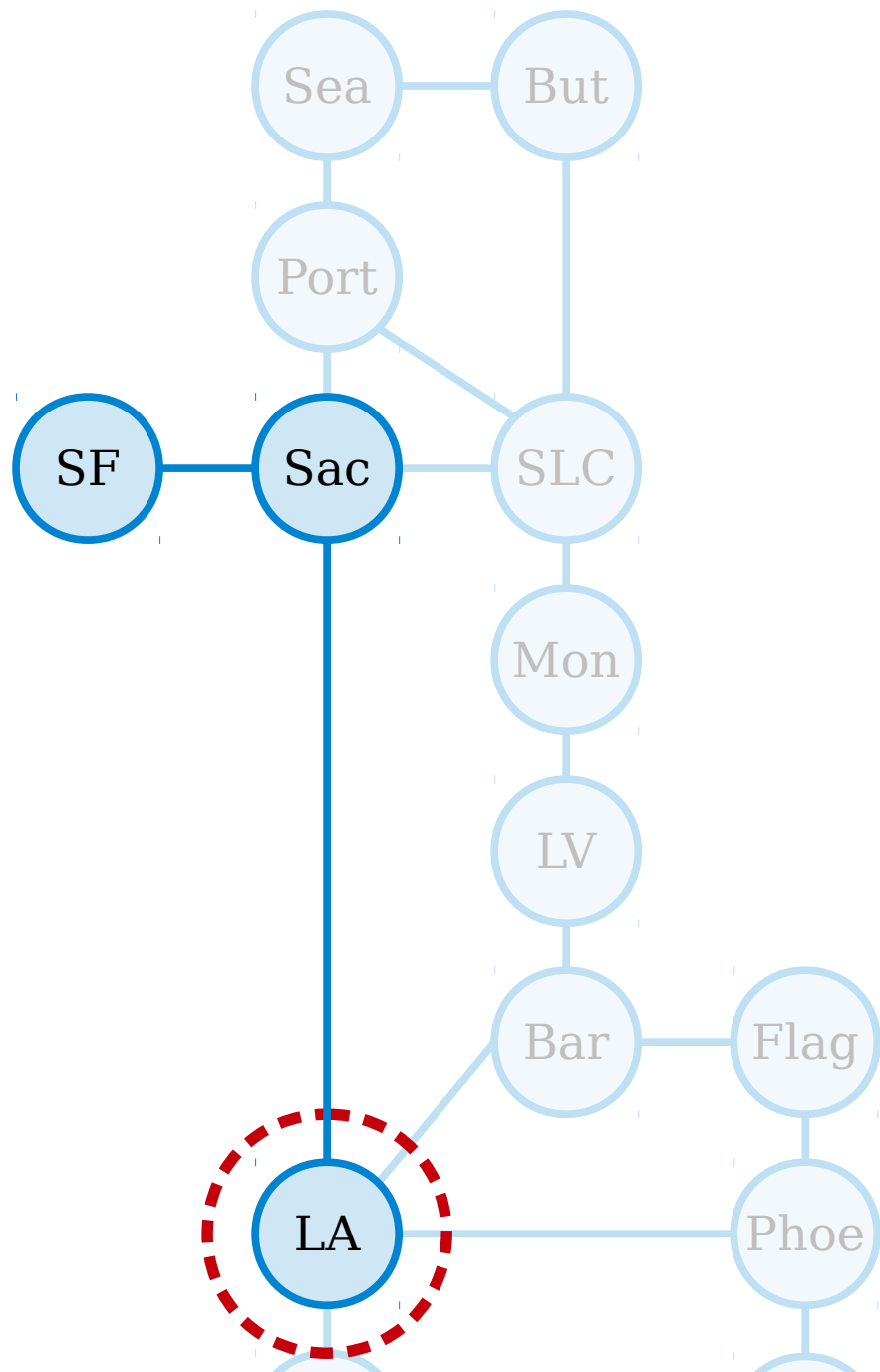


A **walk** in a graph $G = (V, E)$ is a sequence of one or more nodes $v_1, v_2, v_3, \dots, v_n$ such that any two consecutive nodes in the sequence are adjacent.

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.)

SF, Sac

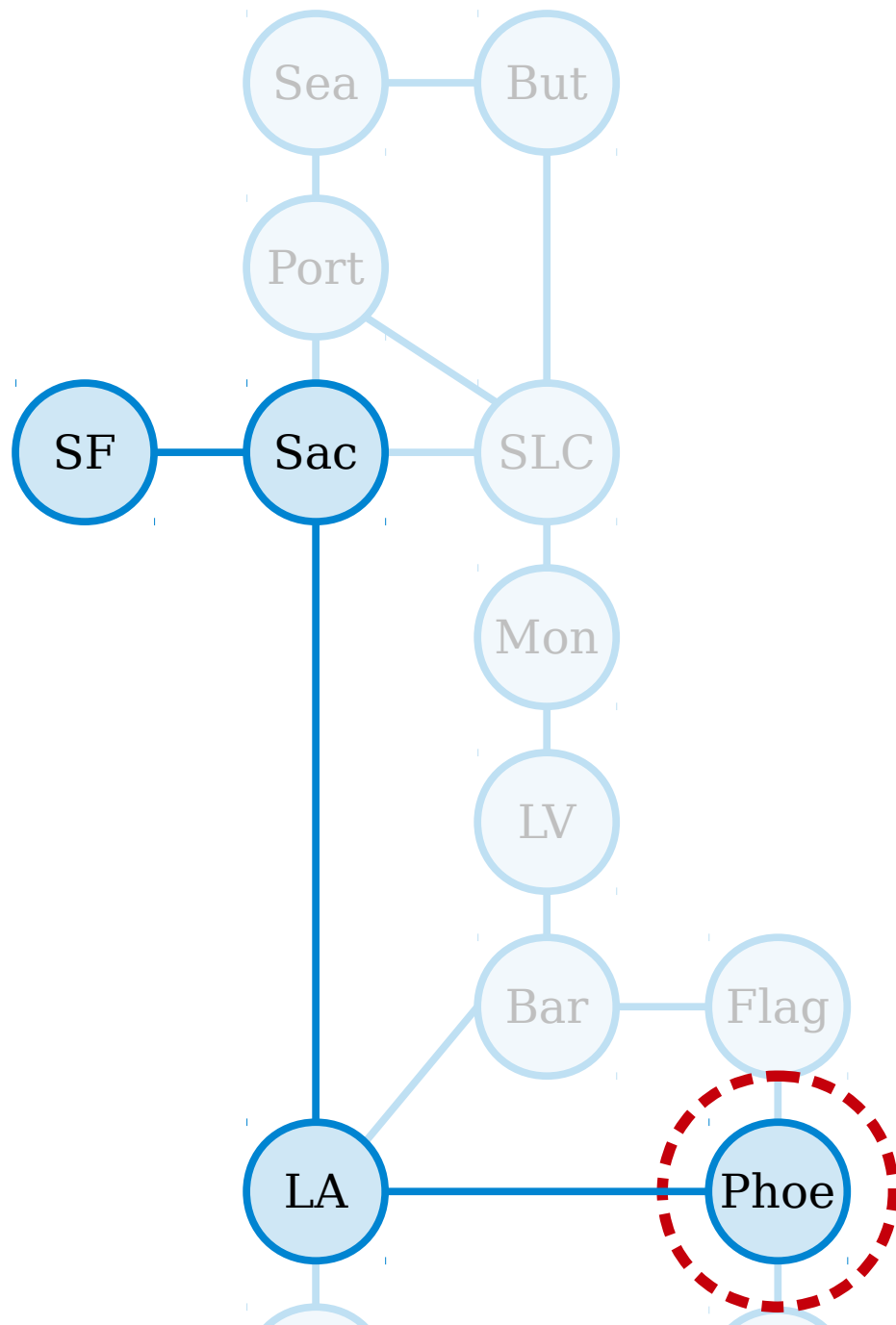


A **walk** in a graph $G = (V, E)$ is a sequence of one or more nodes $v_1, v_2, v_3, \dots, v_n$ such that any two consecutive nodes in the sequence are adjacent.

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.)

SF, Sac, LA

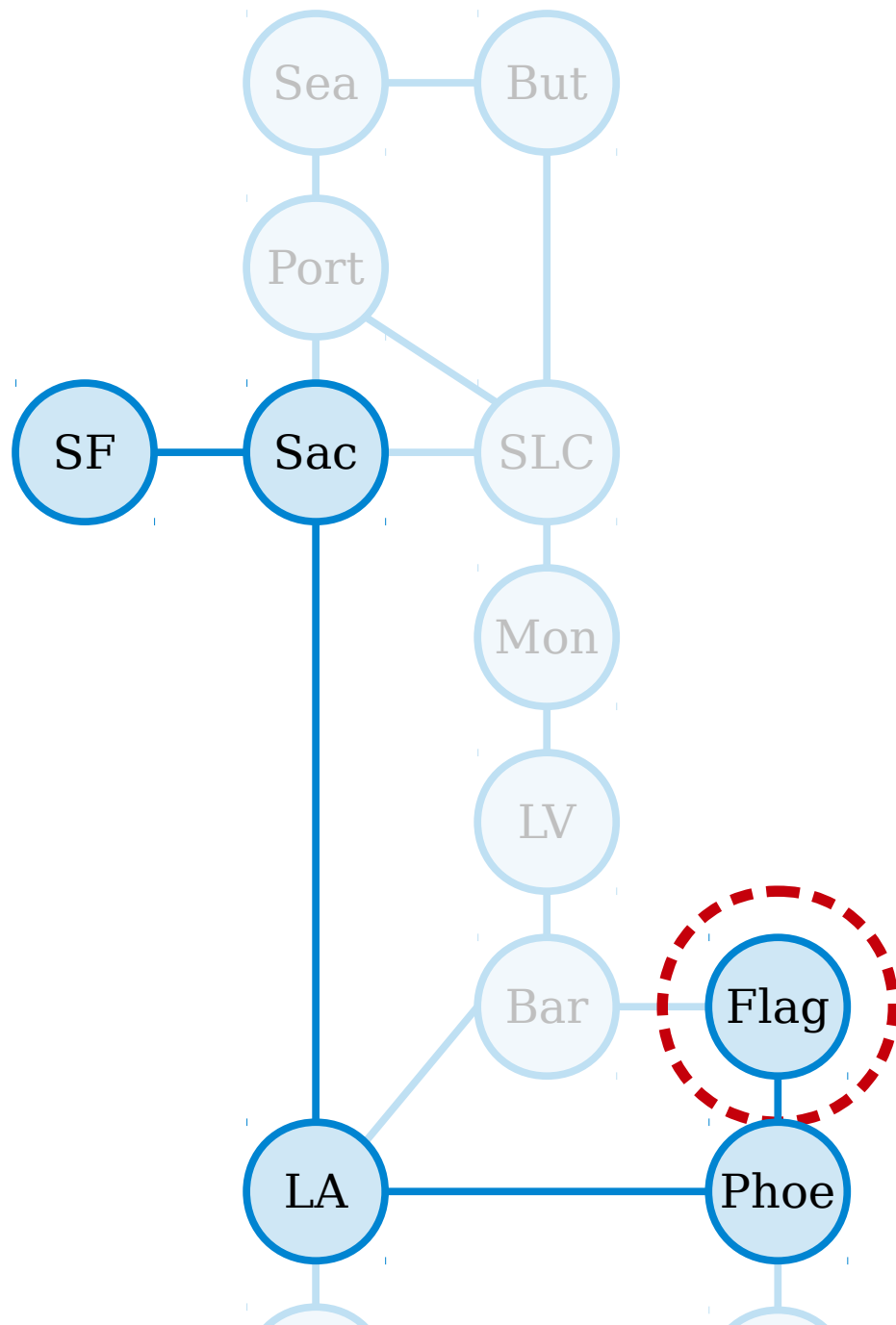


A **walk** in a graph $G = (V, E)$ is a sequence of one or more nodes $v_1, v_2, v_3, \dots, v_n$ such that any two consecutive nodes in the sequence are adjacent.

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.)

SF, Sac, LA, Phoe

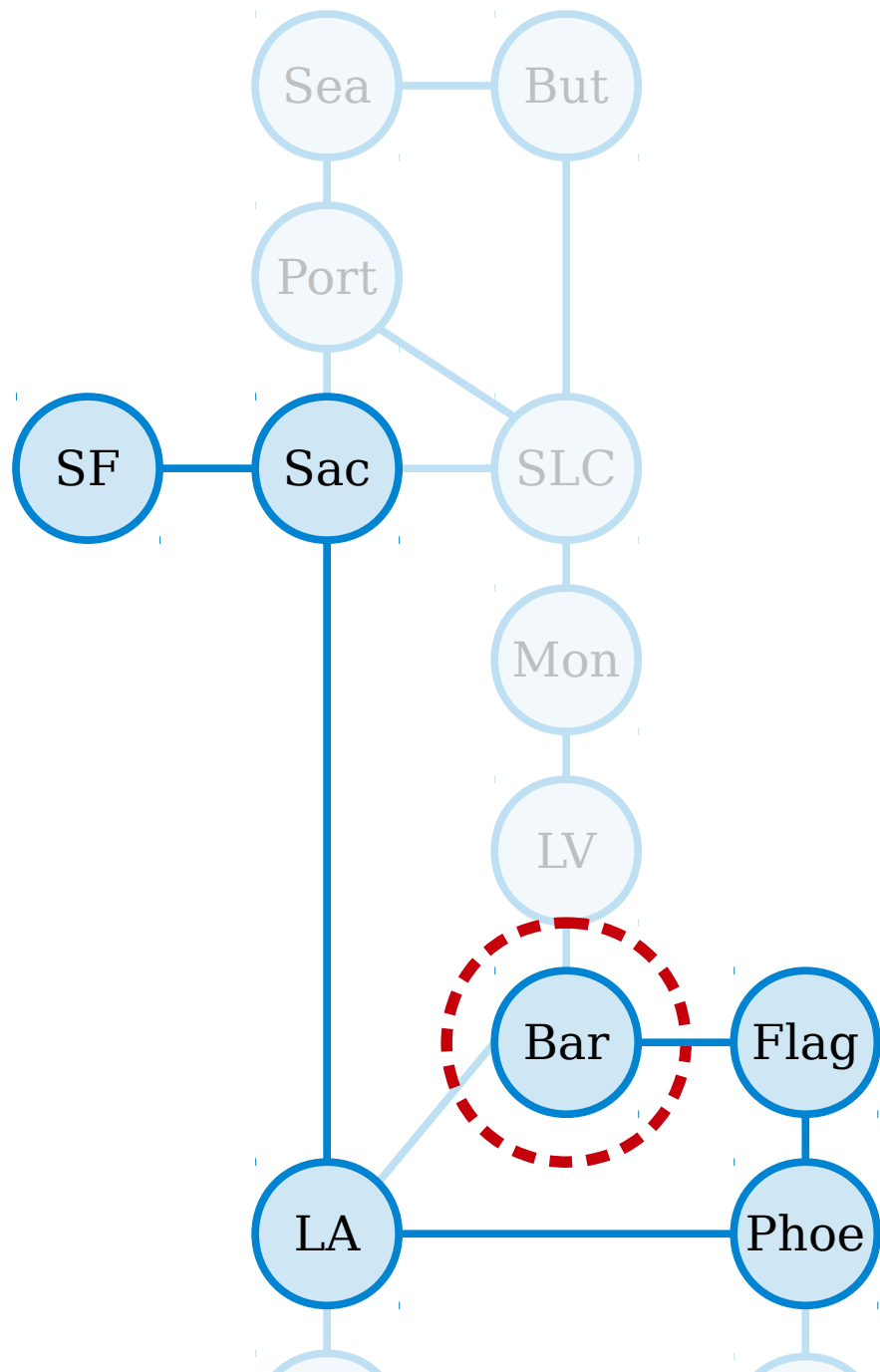


A **walk** in a graph $G = (V, E)$ is a sequence of one or more nodes $v_1, v_2, v_3, \dots, v_n$ such that any two consecutive nodes in the sequence are adjacent.

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.)

SF, Sac, LA, Phoe, Flag

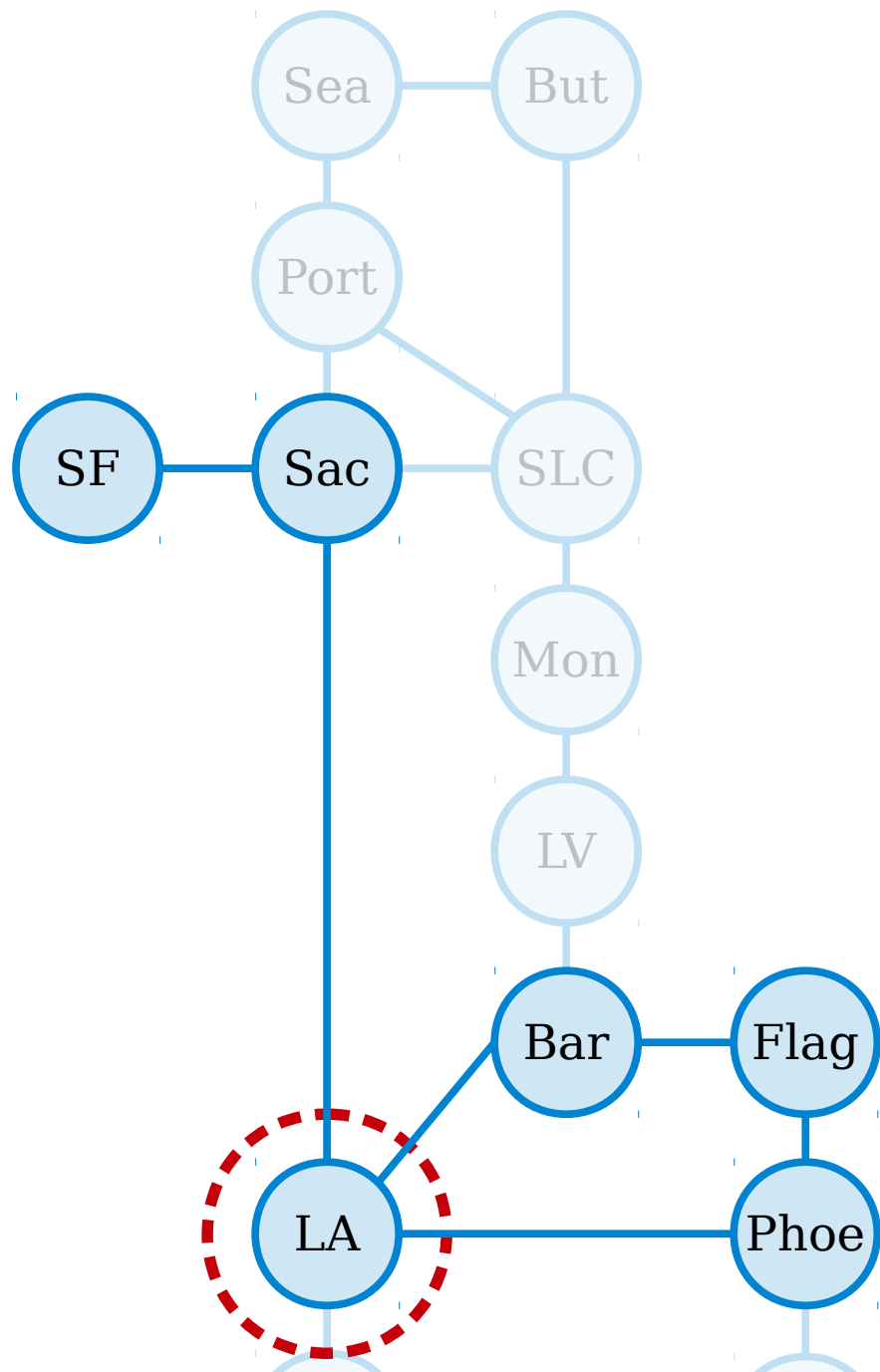


A **walk** in a graph $G = (V, E)$ is a sequence of one or more nodes $v_1, v_2, v_3, \dots, v_n$ such that any two consecutive nodes in the sequence are adjacent.

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.)

SF, Sac, LA, Phoe, Flag, Bar

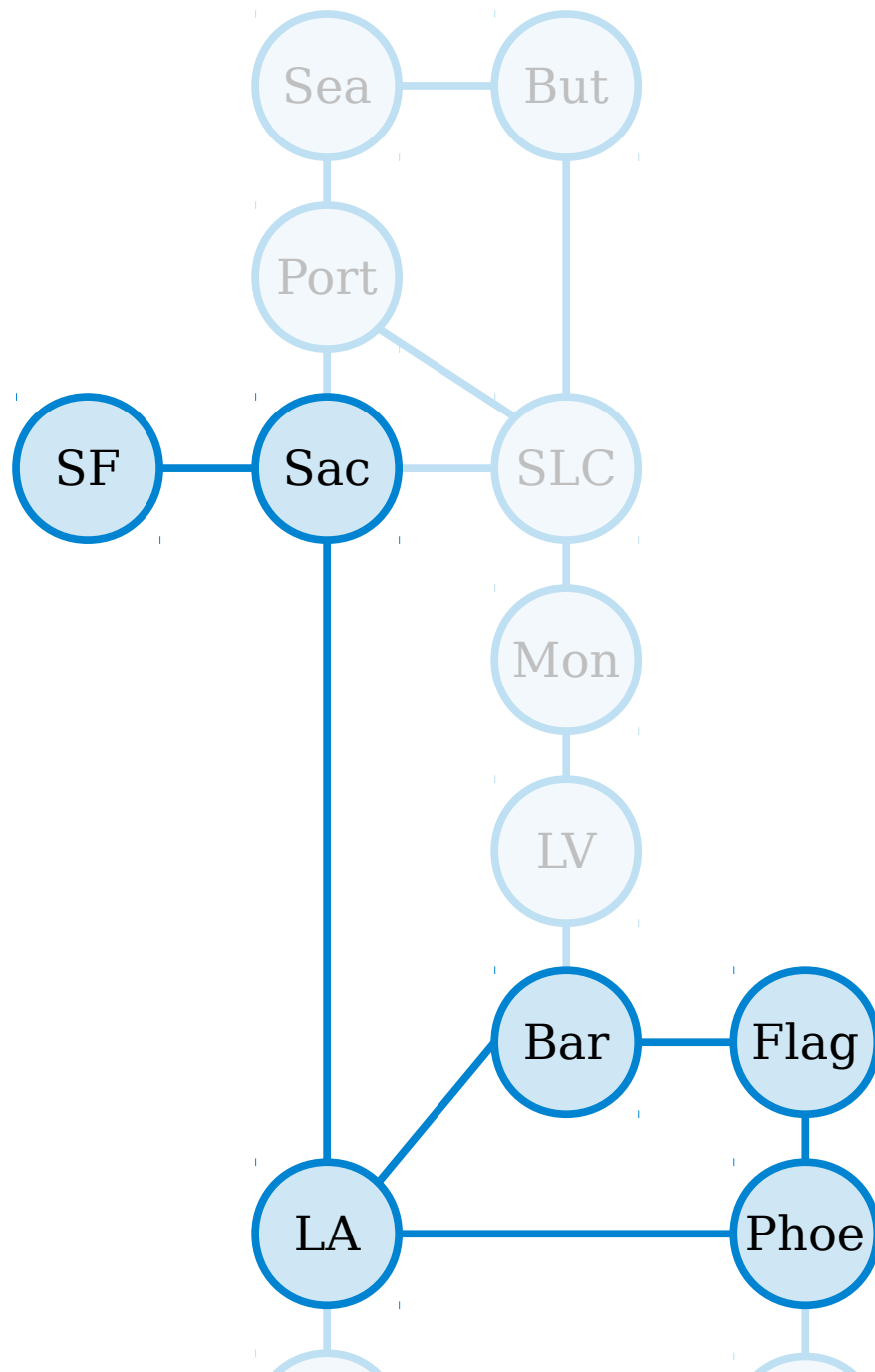


A **walk** in a graph $G = (V, E)$ is a sequence of one or more nodes $v_1, v_2, v_3, \dots, v_n$ such that any two consecutive nodes in the sequence are adjacent.

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.)

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

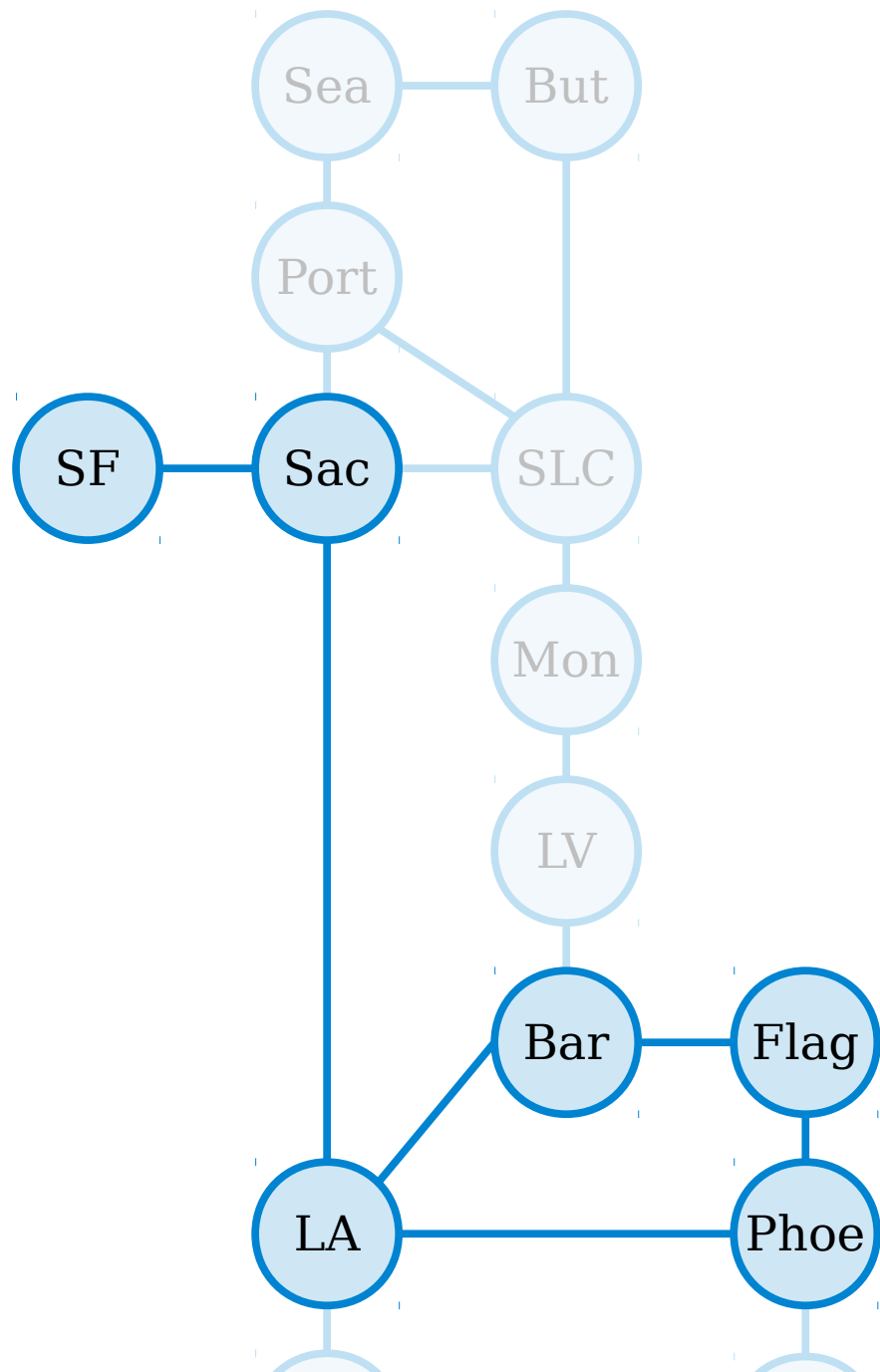


A **walk** in a graph $G = (V, E)$ is a sequence of one or more nodes $v_1, v_2, v_3, \dots, v_n$ such that any two consecutive nodes in the sequence are adjacent.

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.)

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



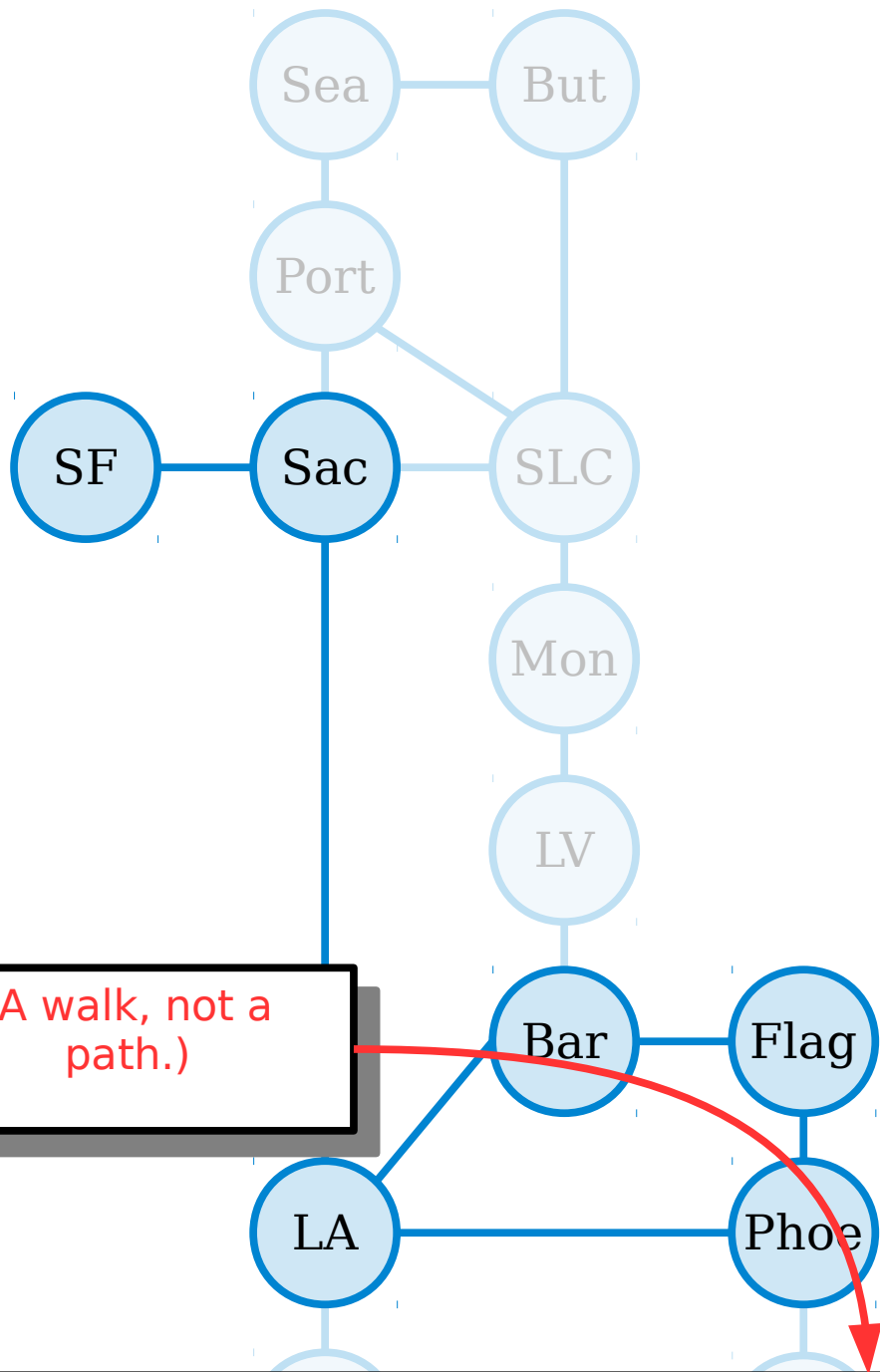
A **walk** in a graph $G = (V, E)$ is a sequence of one or more nodes $v_1, v_2, v_3, \dots, v_n$ such that any two consecutive nodes in the sequence are adjacent.

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.

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.

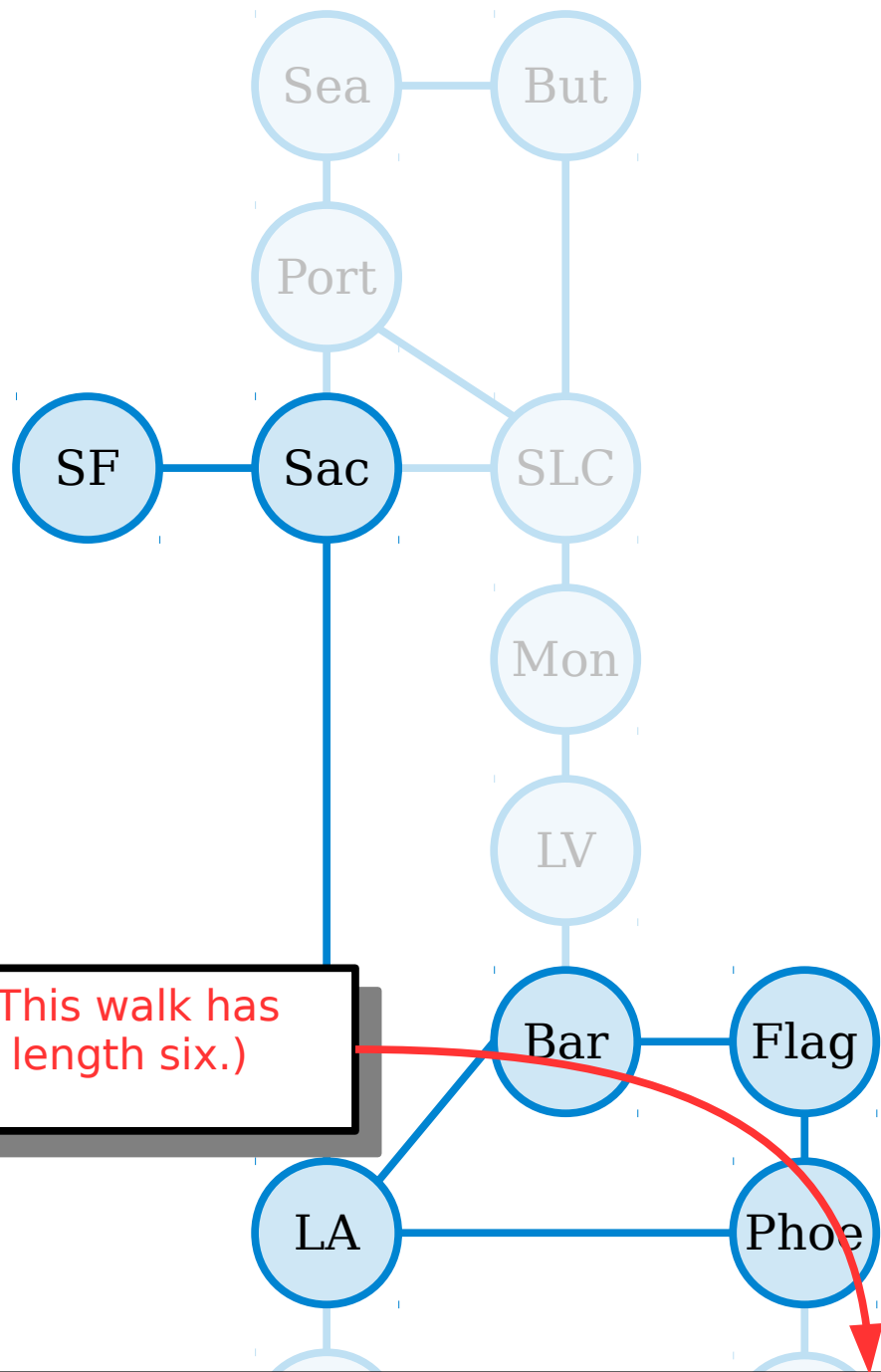
The **length** of the walk v_1, \dots, v_n is $n - 1$.

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 walk, not a path.)

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



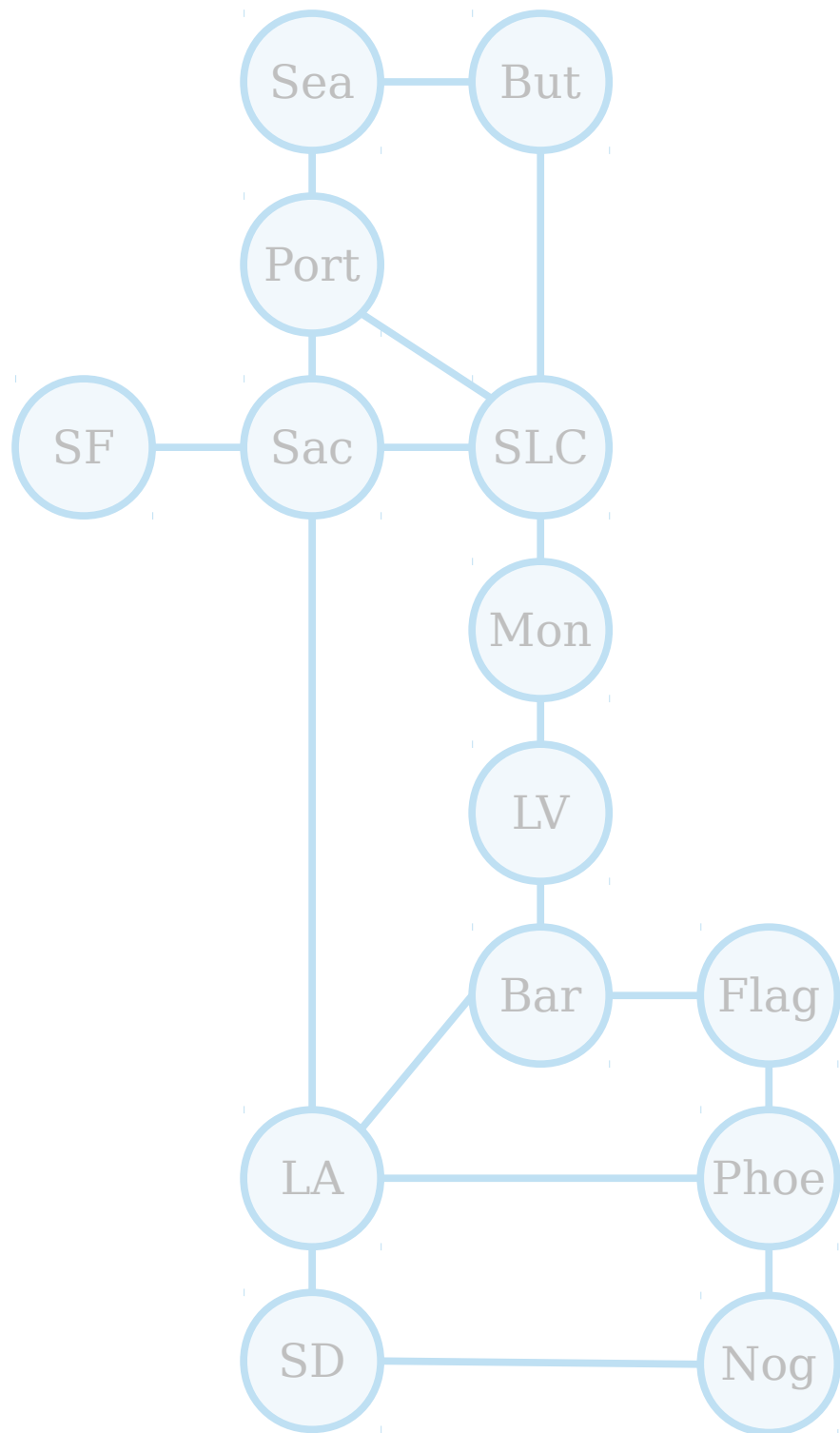
A **walk** in a graph $G = (V, E)$ is a sequence of one or more nodes $v_1, v_2, v_3, \dots, v_n$ such that any two consecutive nodes in the sequence are adjacent.

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.

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

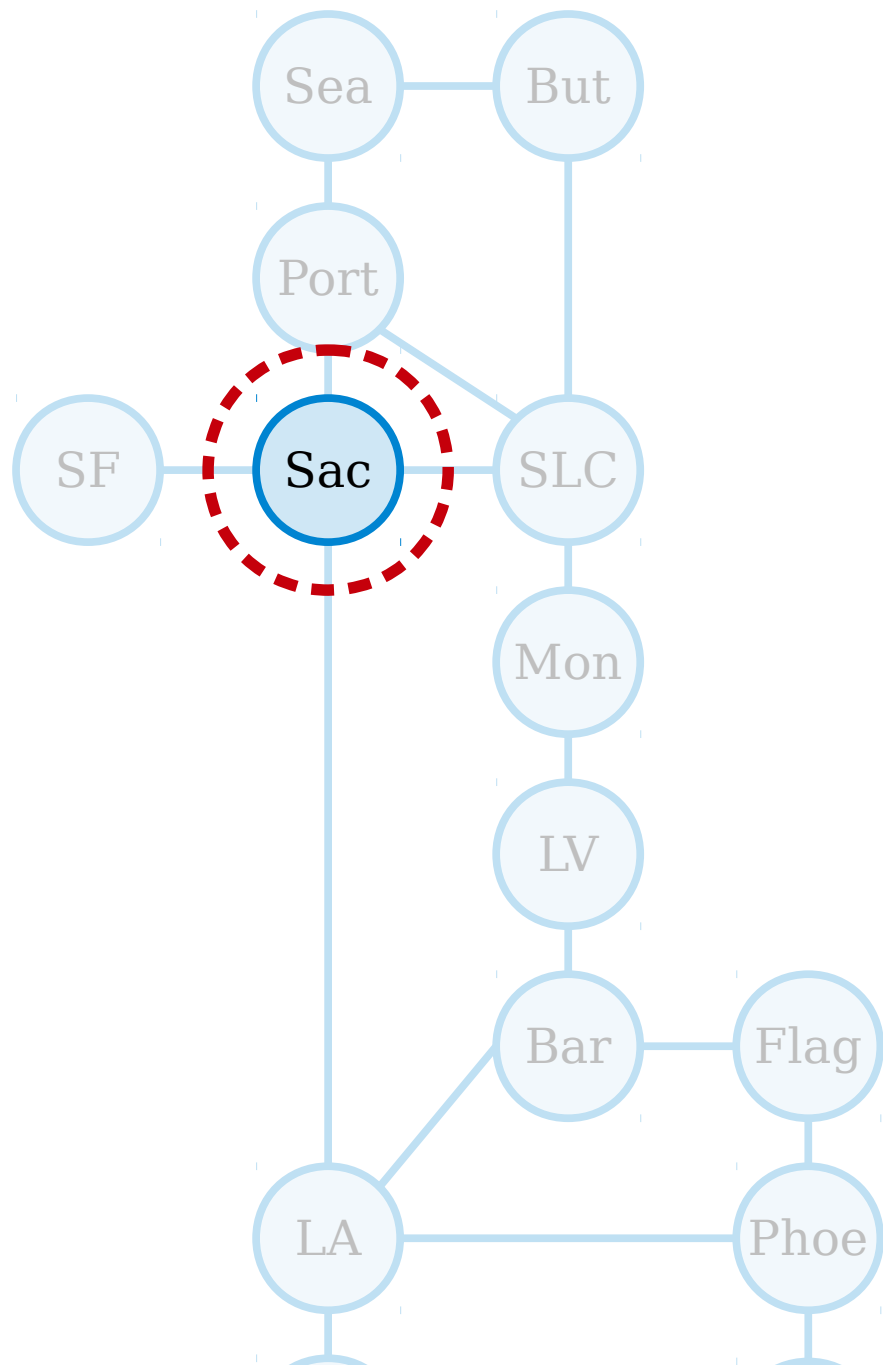


A **walk** in a graph $G = (V, E)$ is a sequence of one or more nodes $v_1, v_2, v_3, \dots, v_n$ such that any two consecutive nodes in the sequence are adjacent.

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.



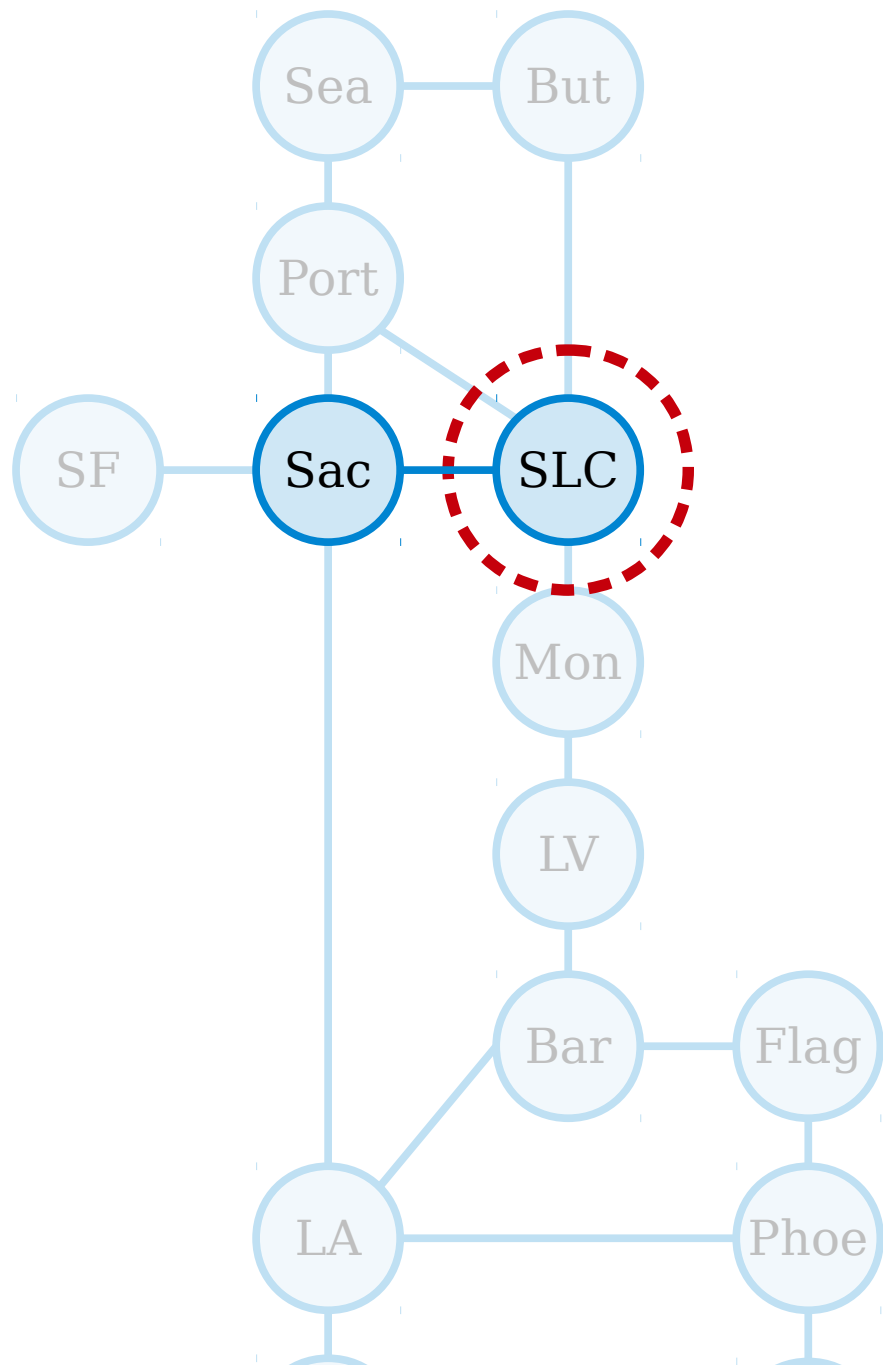
Sac

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.



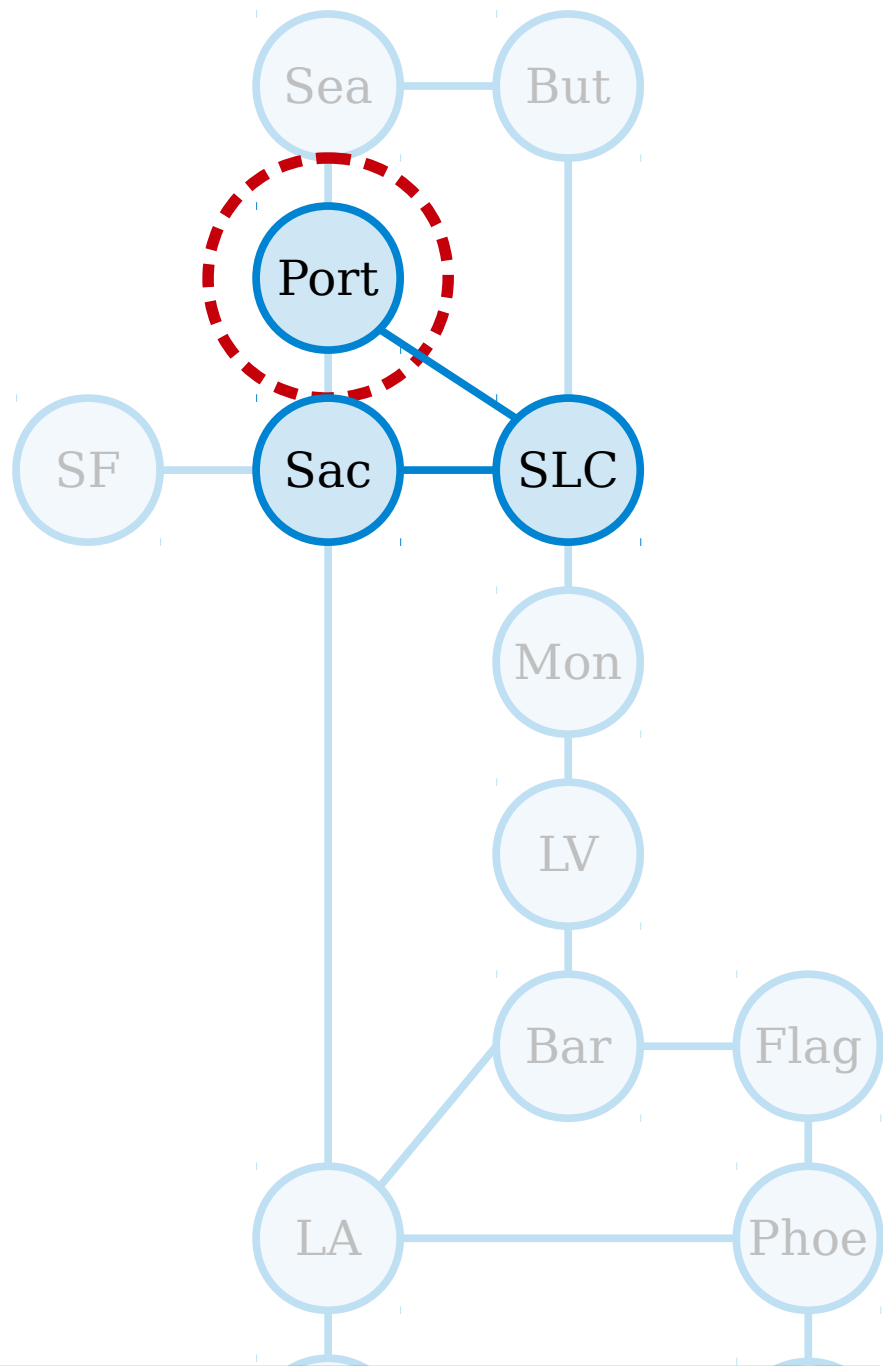
Sac, SLC

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.



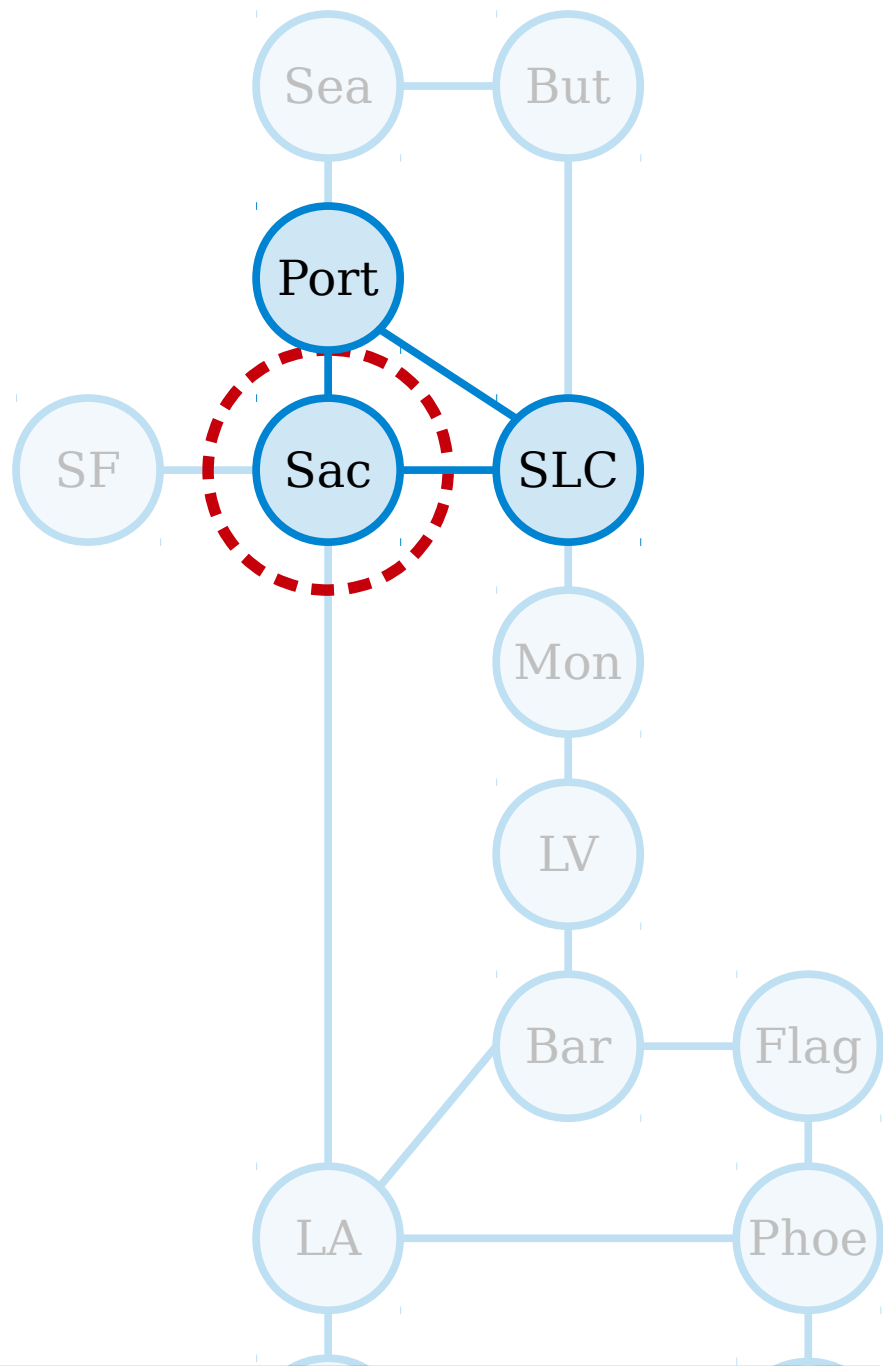
Sac, SLC, Port

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.



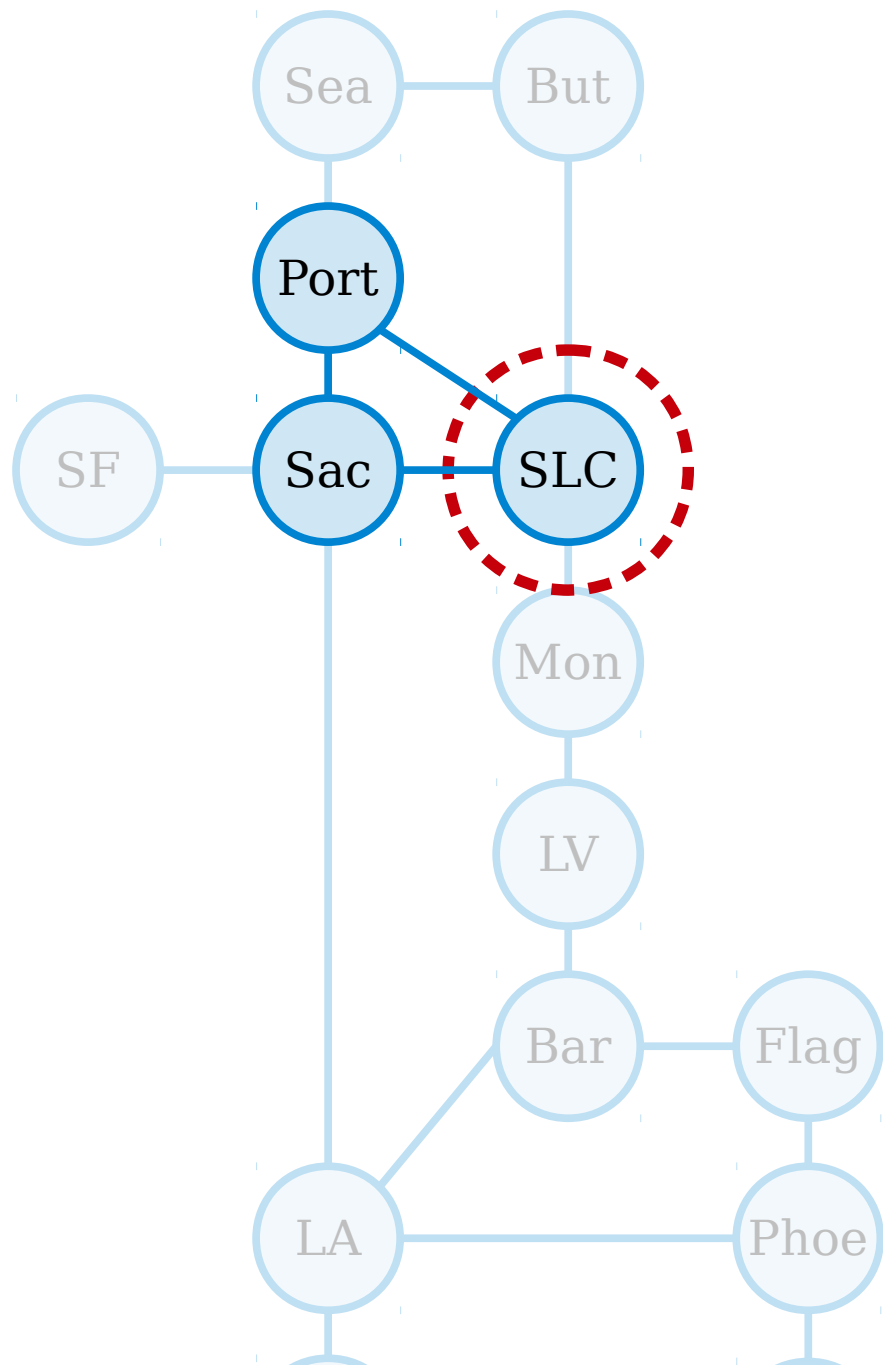
A **walk** in a graph $G = (V, E)$ is a sequence of one or more nodes $v_1, v_2, v_3, \dots, v_n$ such that any two consecutive nodes in the sequence are adjacent.

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.

Sac, SLC, Port, Sac



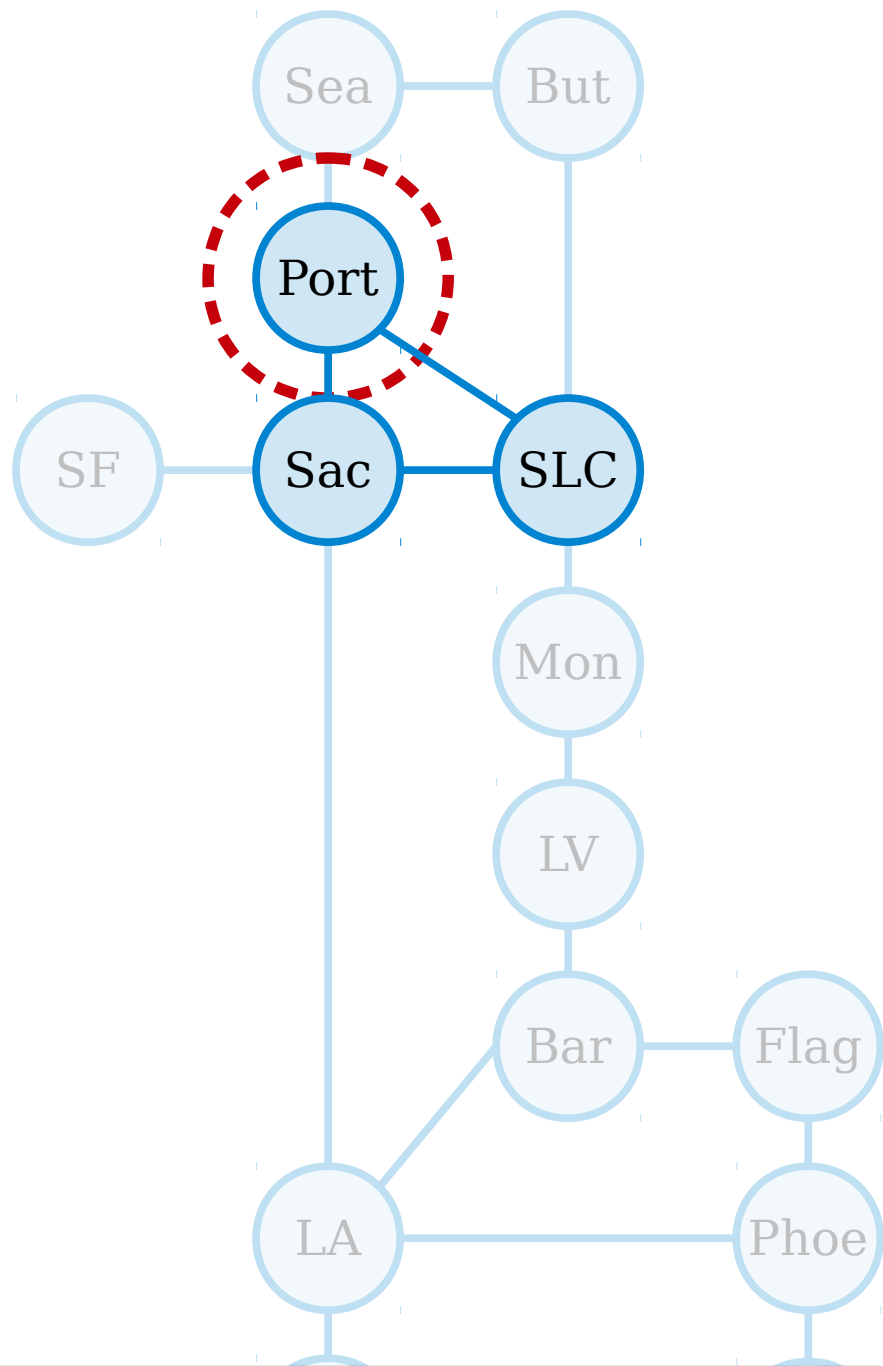
A **walk** in a graph $G = (V, E)$ is a sequence of one or more nodes $v_1, v_2, v_3, \dots, v_n$ such that any two consecutive nodes in the sequence are adjacent.

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.

Sac, SLC, Port, Sac, SLC



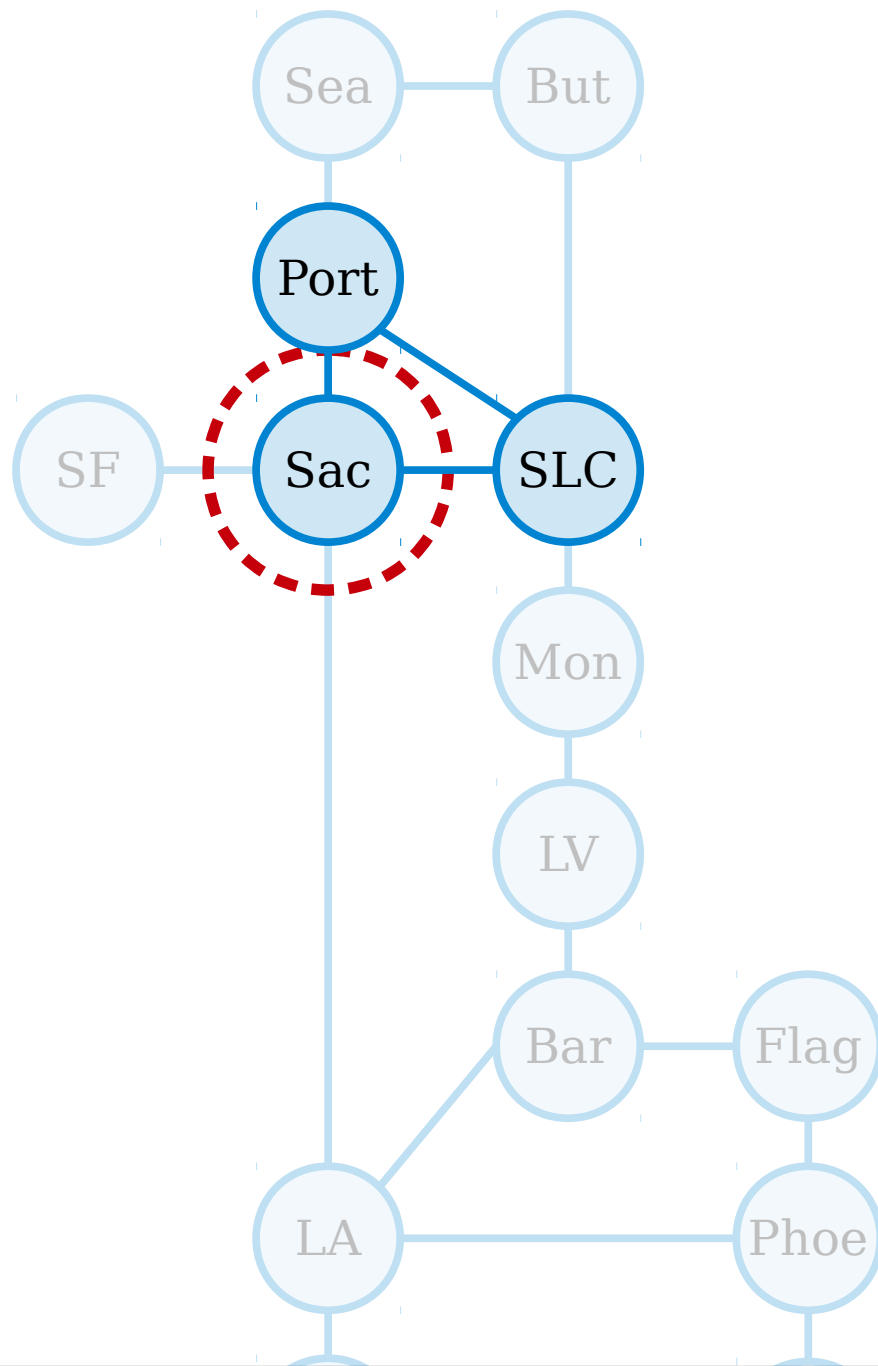
Sac, SLC, Port, Sac, SLC, Port

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.



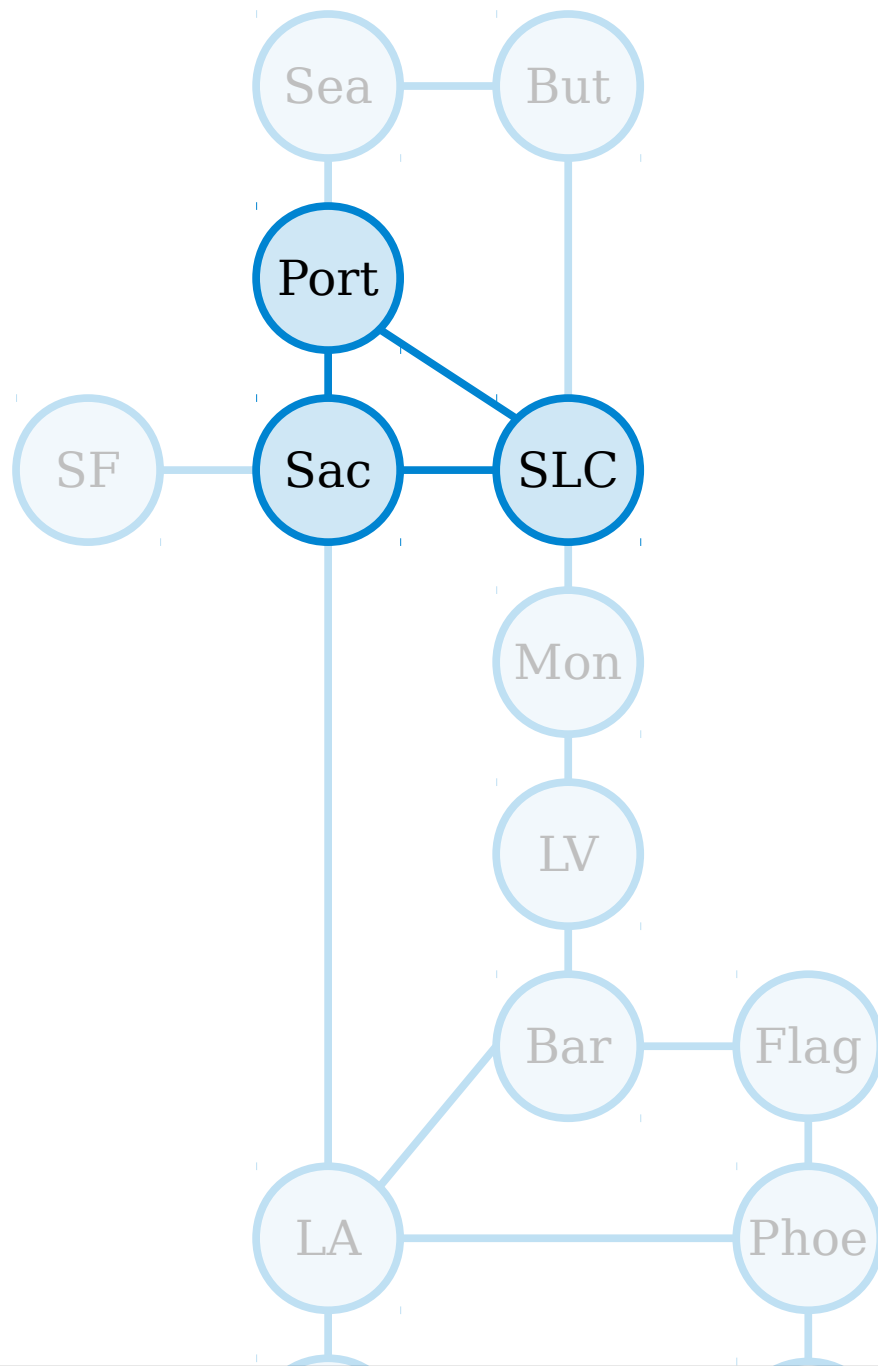
A **walk** in a graph $G = (V, E)$ is a sequence of one or more nodes $v_1, v_2, v_3, \dots, v_n$ such that any two consecutive nodes in the sequence are adjacent.

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.

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



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

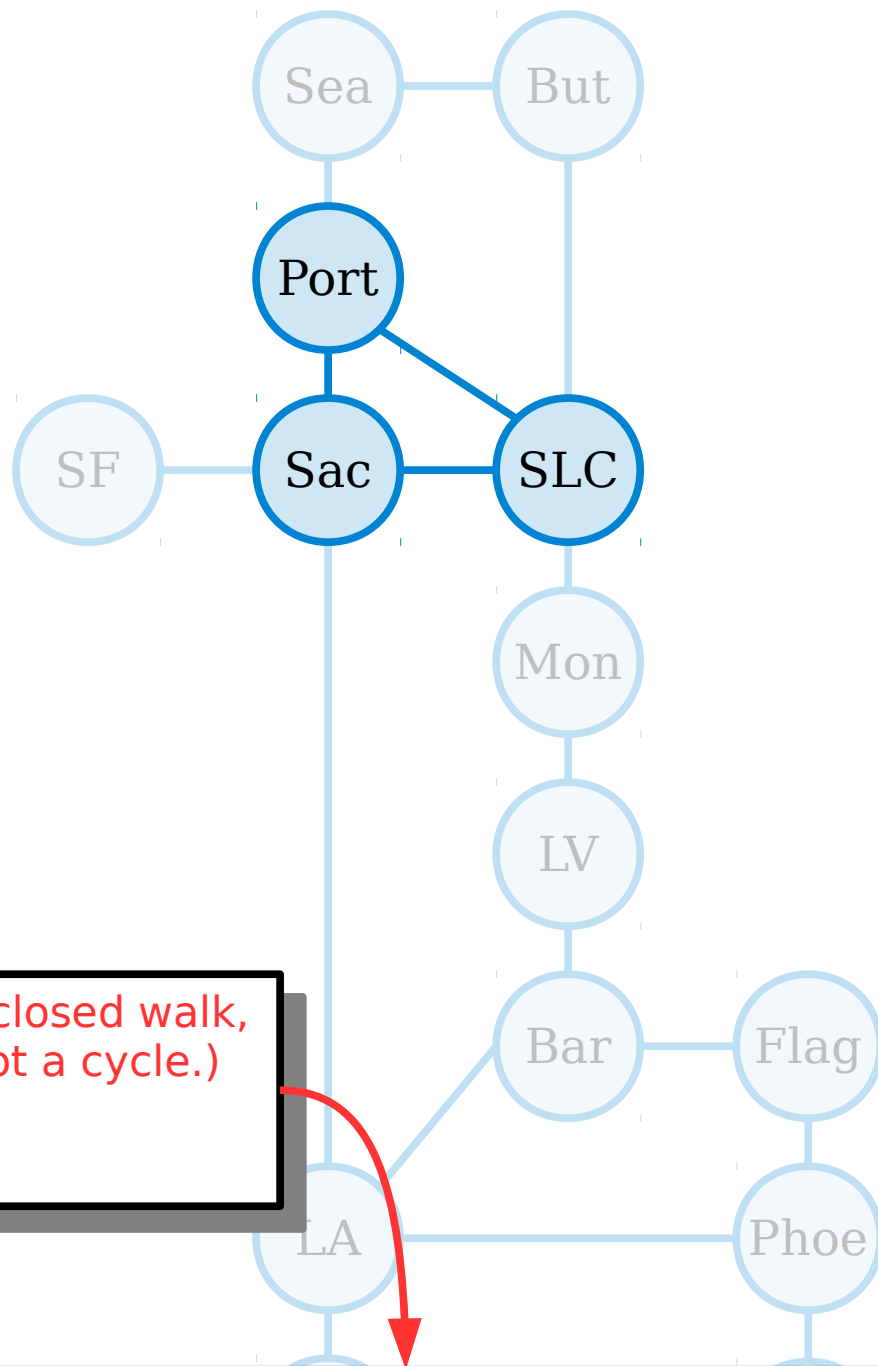
A **walk** in a graph $G = (V, E)$ is a sequence of one or more nodes $v_1, v_2, v_3, \dots, v_n$ such that any two consecutive nodes in the sequence are adjacent.

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.



(A closed walk, not a cycle.)

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

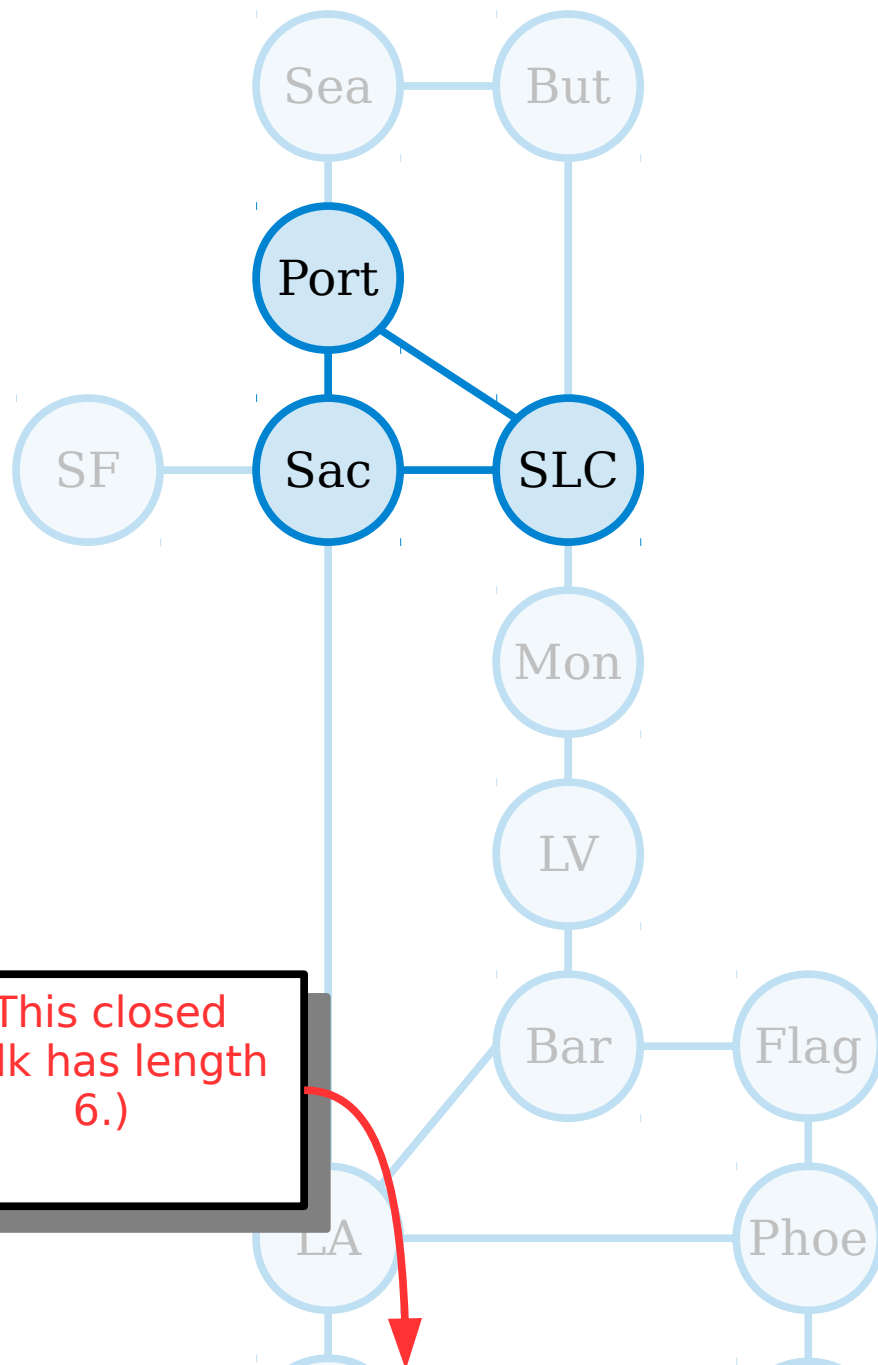
A **walk** in a graph $G = (V, E)$ is a sequence of one or more nodes $v_1, v_2, v_3, \dots, v_n$ such that any two consecutive nodes in the sequence are adjacent.

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 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

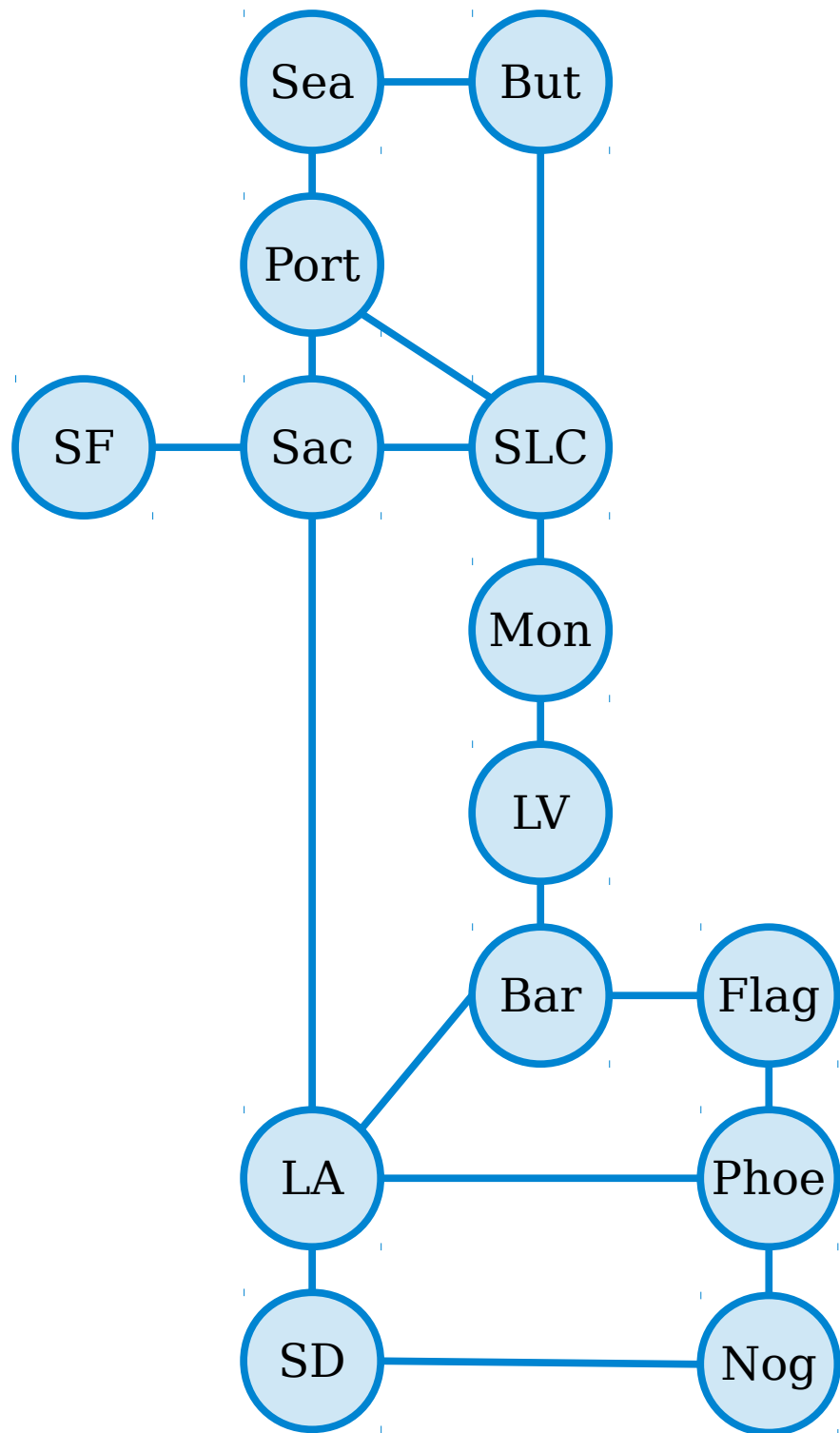
A **walk** in a graph $G = (V, E)$ is a sequence of one or more nodes $v_1, v_2, v_3, \dots, v_n$ such that any two consecutive nodes in the sequence are adjacent.

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.



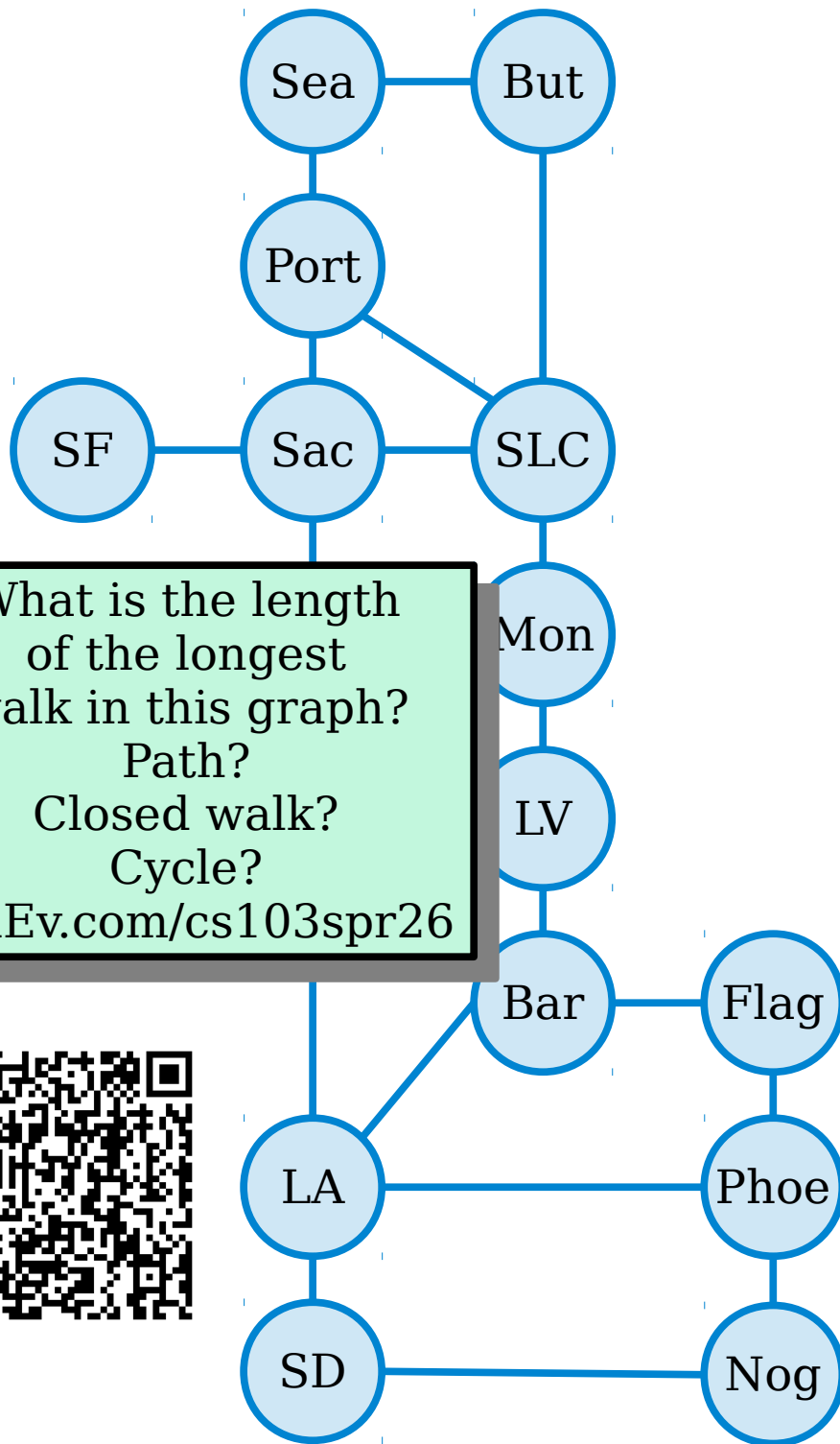
A **walk** in a graph $G = (V, E)$ is a sequence of one or more nodes $v_1, v_2, v_3, \dots, v_n$ such that any two consecutive nodes in the sequence are adjacent.

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.



What is the length of the longest walk in this graph?
 Path?
 Closed walk?
 Cycle?
PollEv.com/cs103spr26



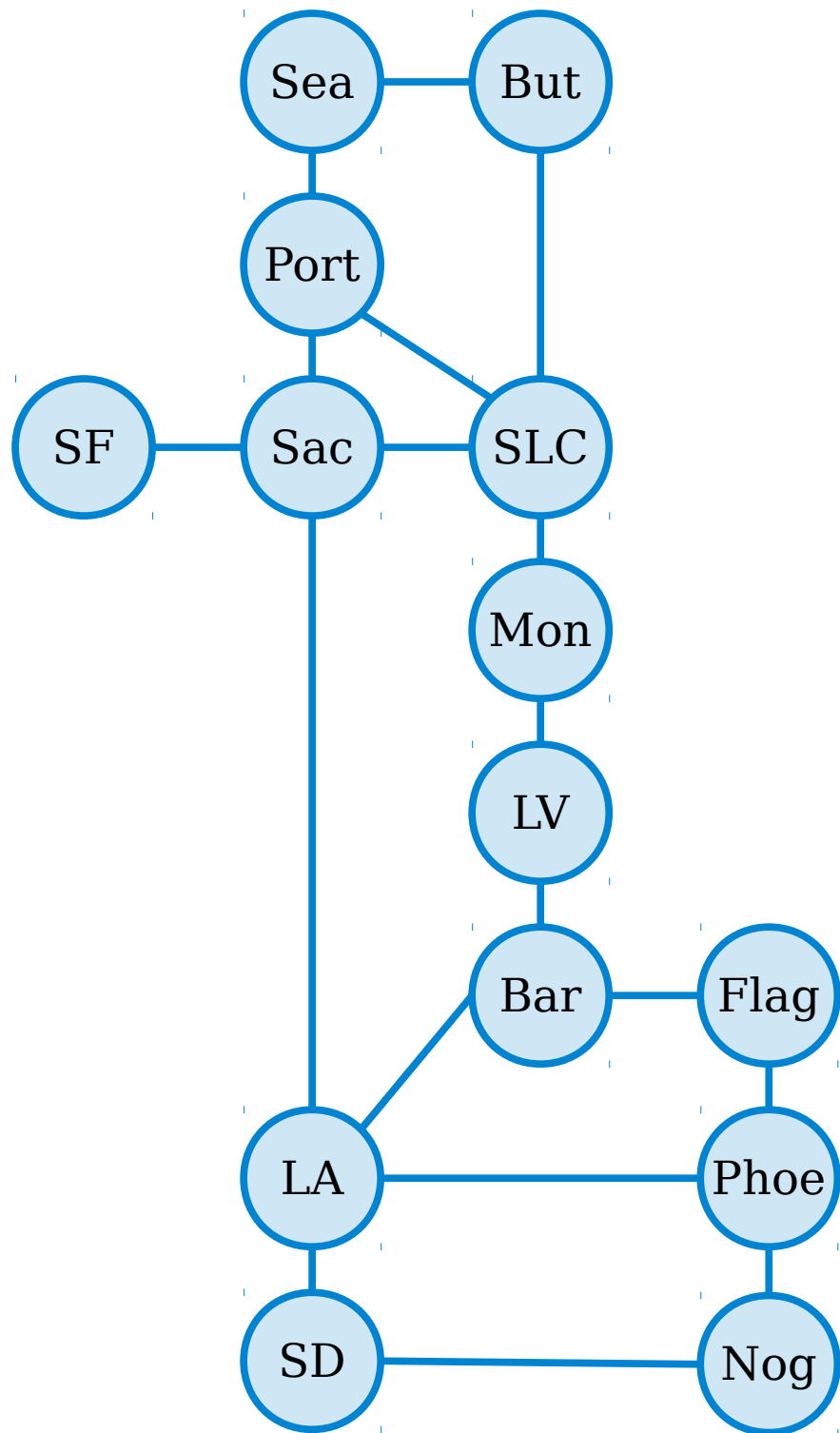
A **walk** in a graph $G = (V, E)$ is a sequence of one or more nodes $v_1, v_2, v_3, \dots, v_n$ such that any two consecutive nodes in the sequence are adjacent.

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.)

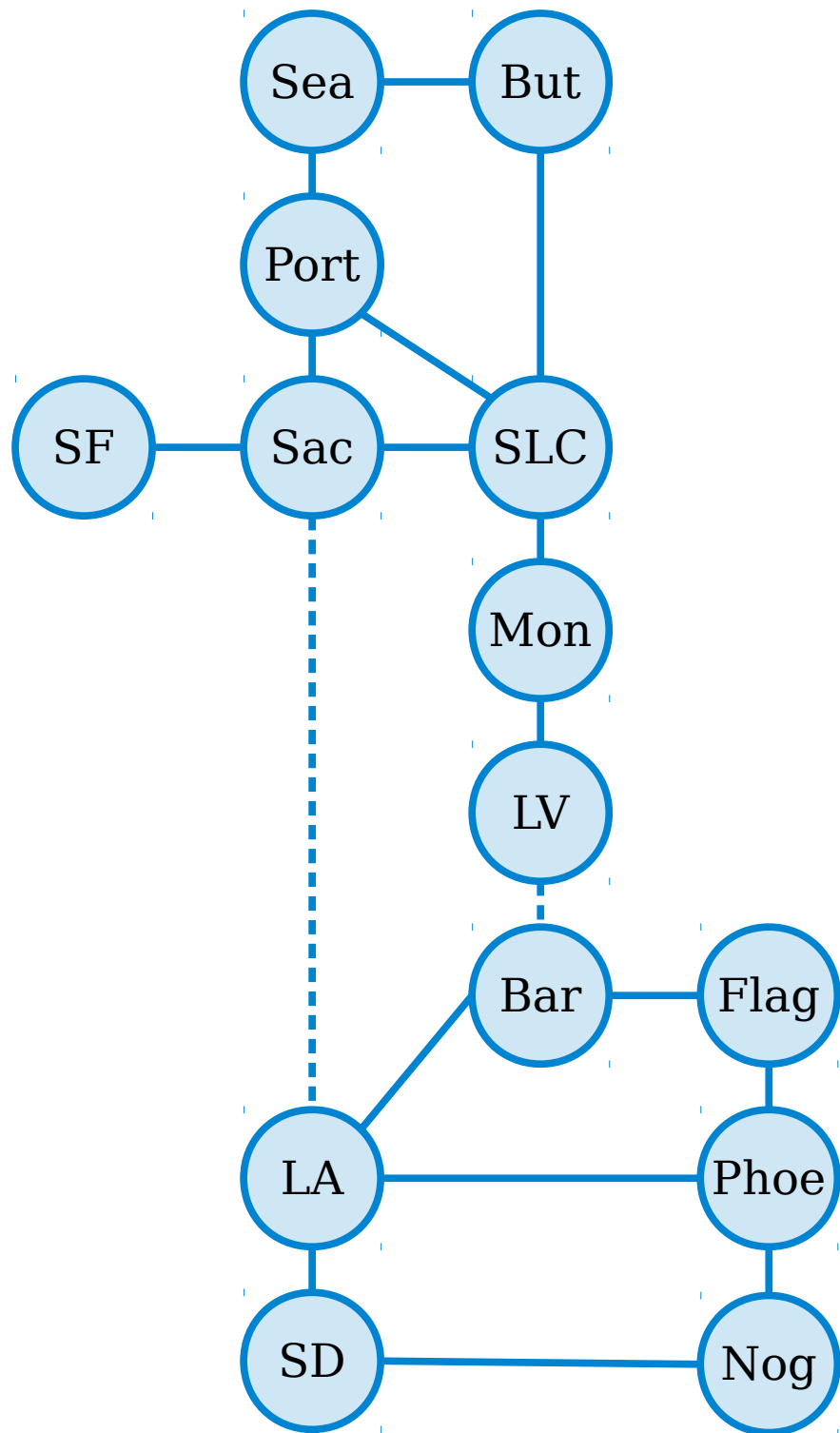
A **path** in a graph is walk that does not repeat any nodes.

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



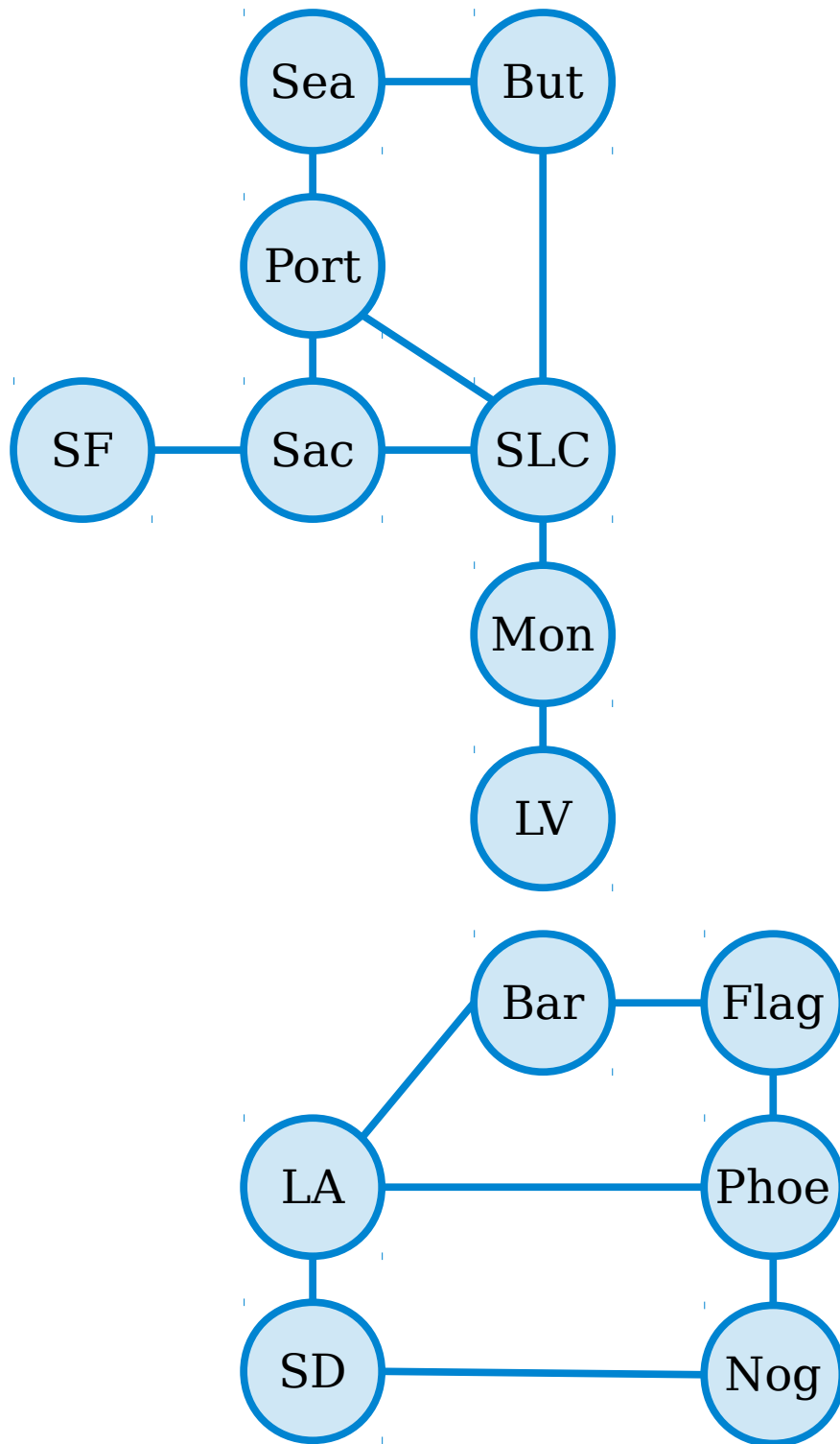
A **walk** in a graph $G = (V, E)$ is a sequence of one or more nodes $v_1, v_2, v_3, \dots, v_n$ such that any two consecutive nodes in the sequence are adjacent.

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



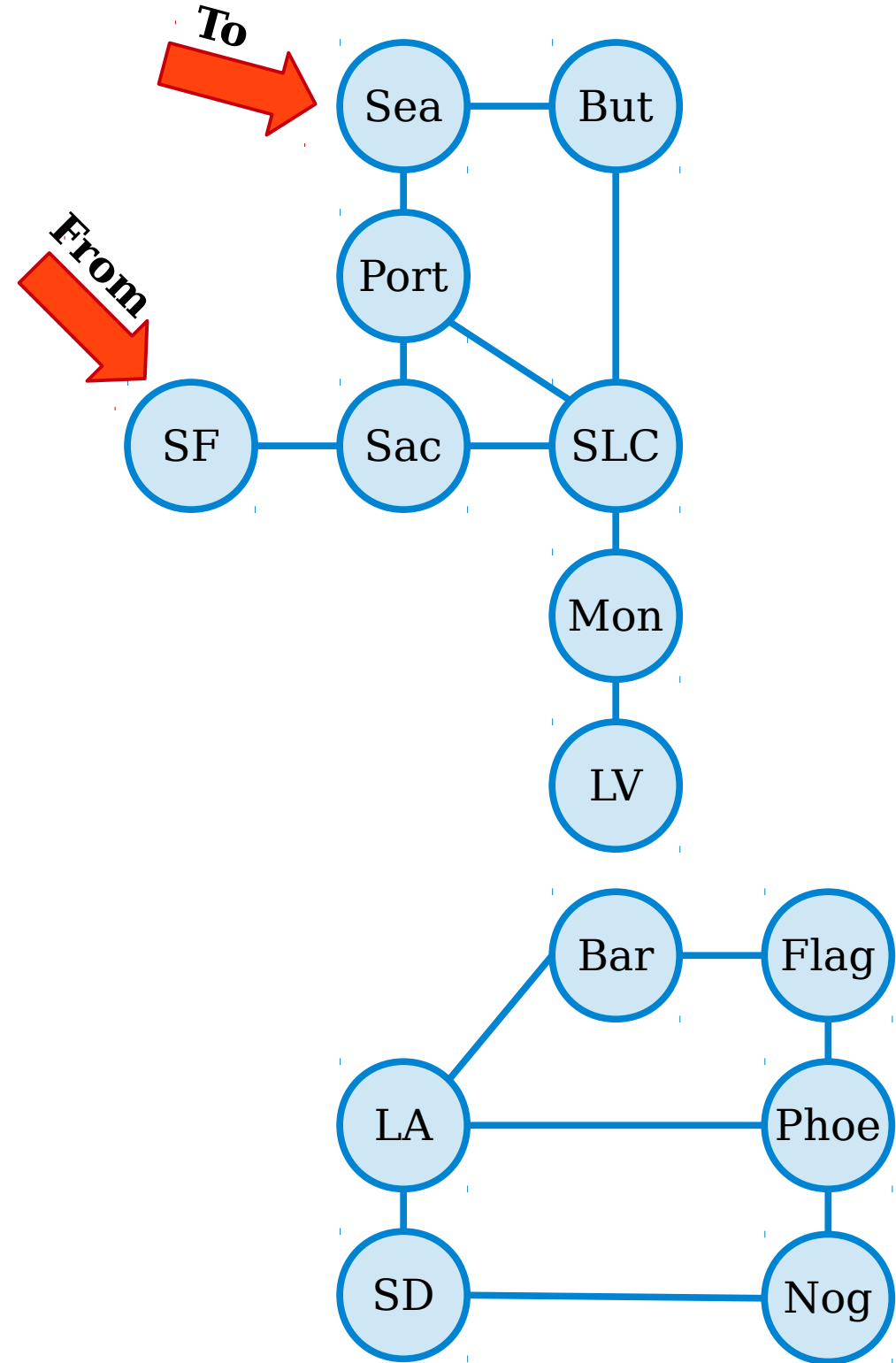
A **walk** in a graph $G = (V, E)$ is a sequence of one or more nodes $v_1, v_2, v_3, \dots, v_n$ such that any two consecutive nodes in the sequence are adjacent.

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



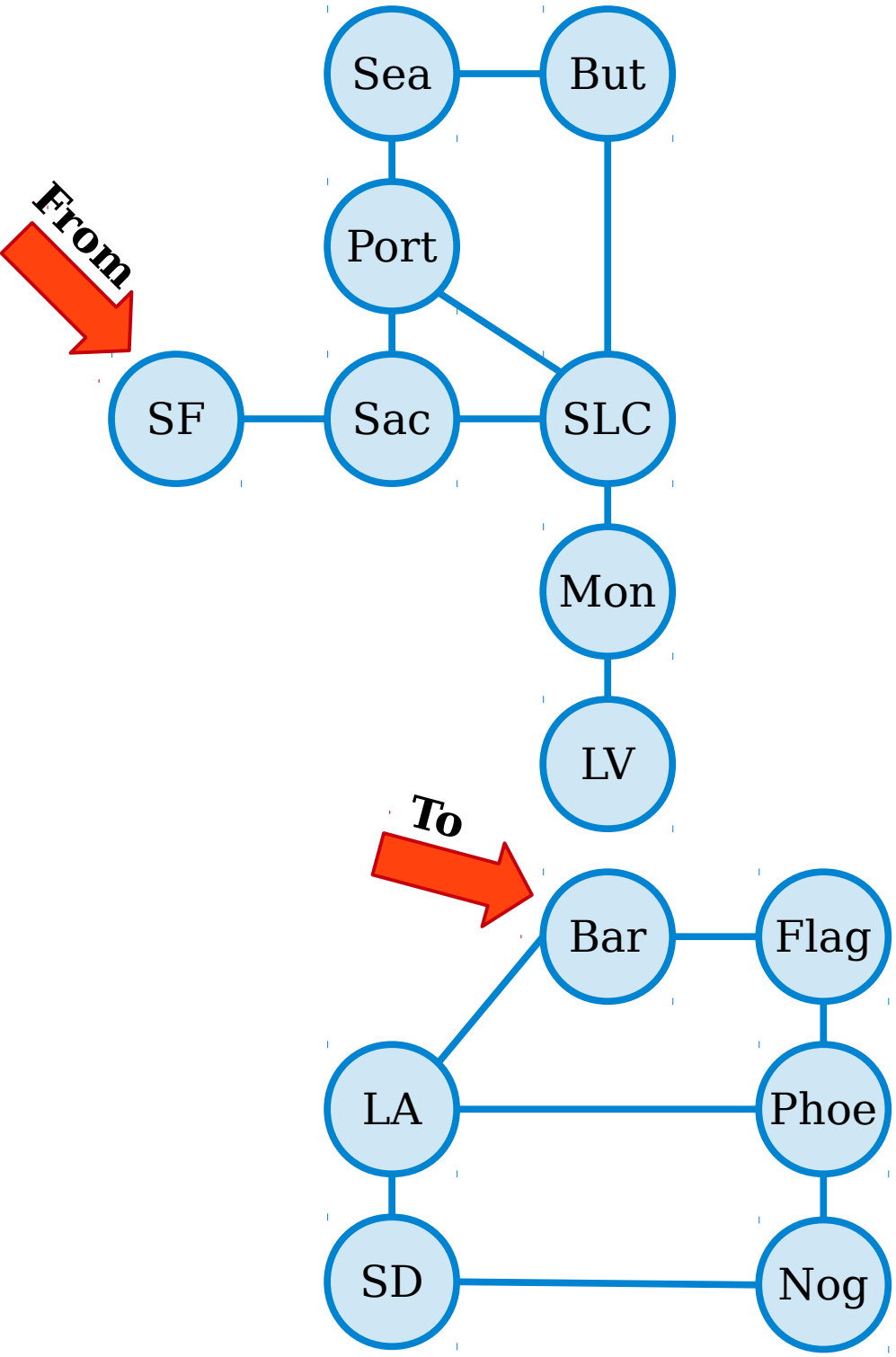
A **walk** in a graph $G = (V, E)$ is a sequence of one or more nodes $v_1, v_2, v_3, \dots, v_n$ such that any two consecutive nodes in the sequence are adjacent.

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



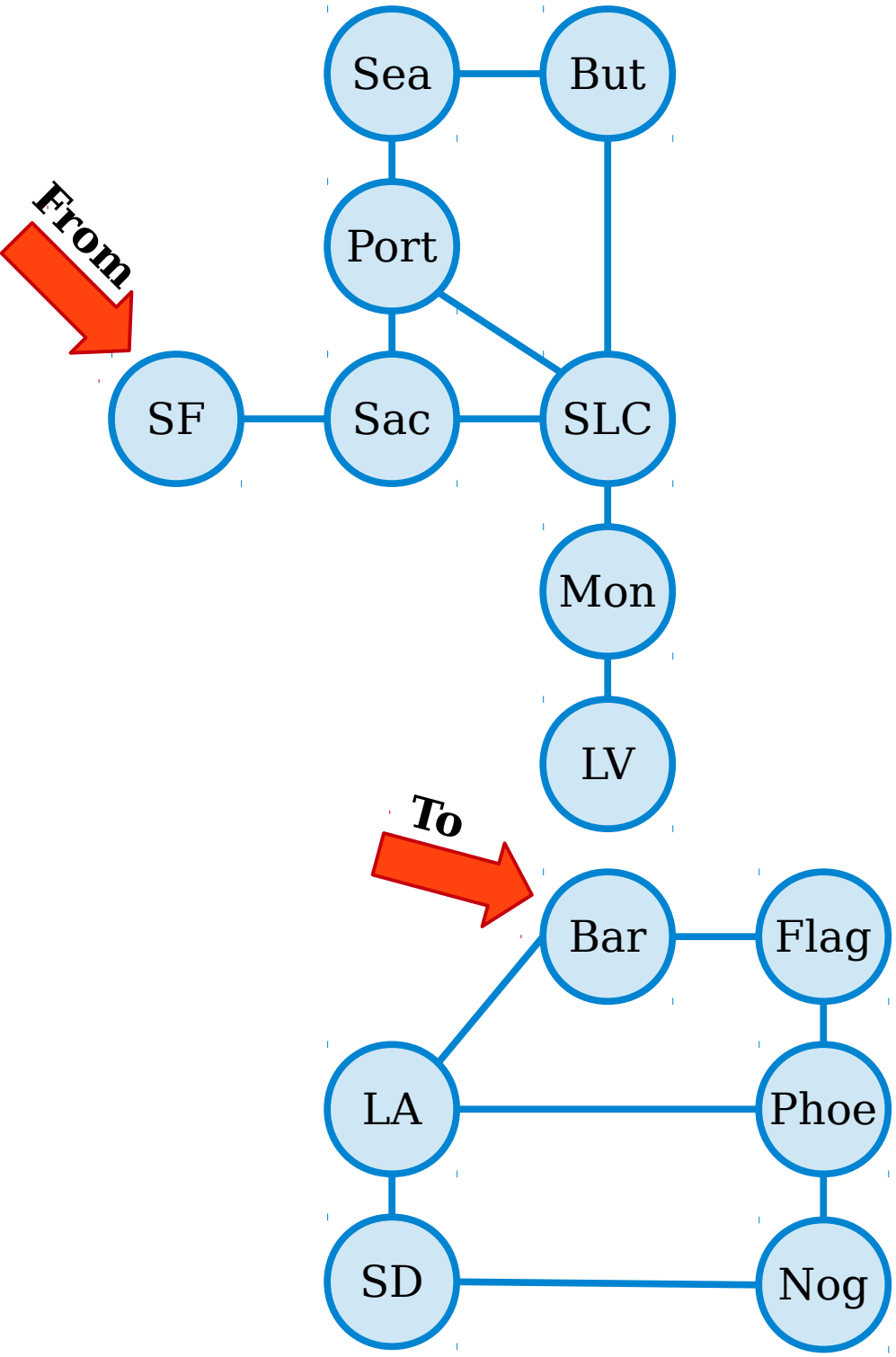
A **walk** in a graph $G = (V, E)$ is a sequence of one or more nodes $v_1, v_2, v_3, \dots, v_n$ such that any two consecutive nodes in the sequence are adjacent.

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



A **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.

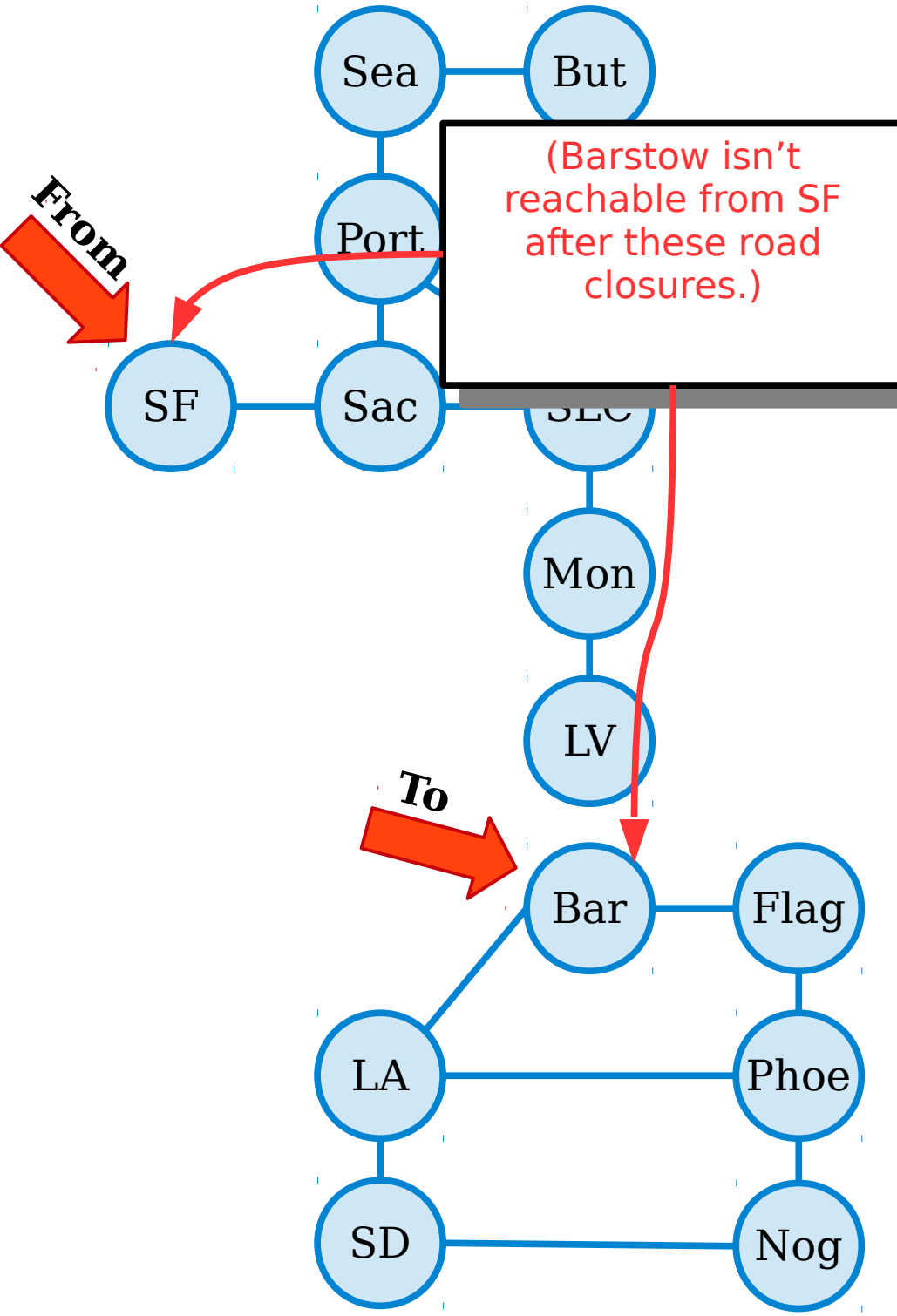
A **path** in a graph is walk that does not repeat any nodes.



A **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.

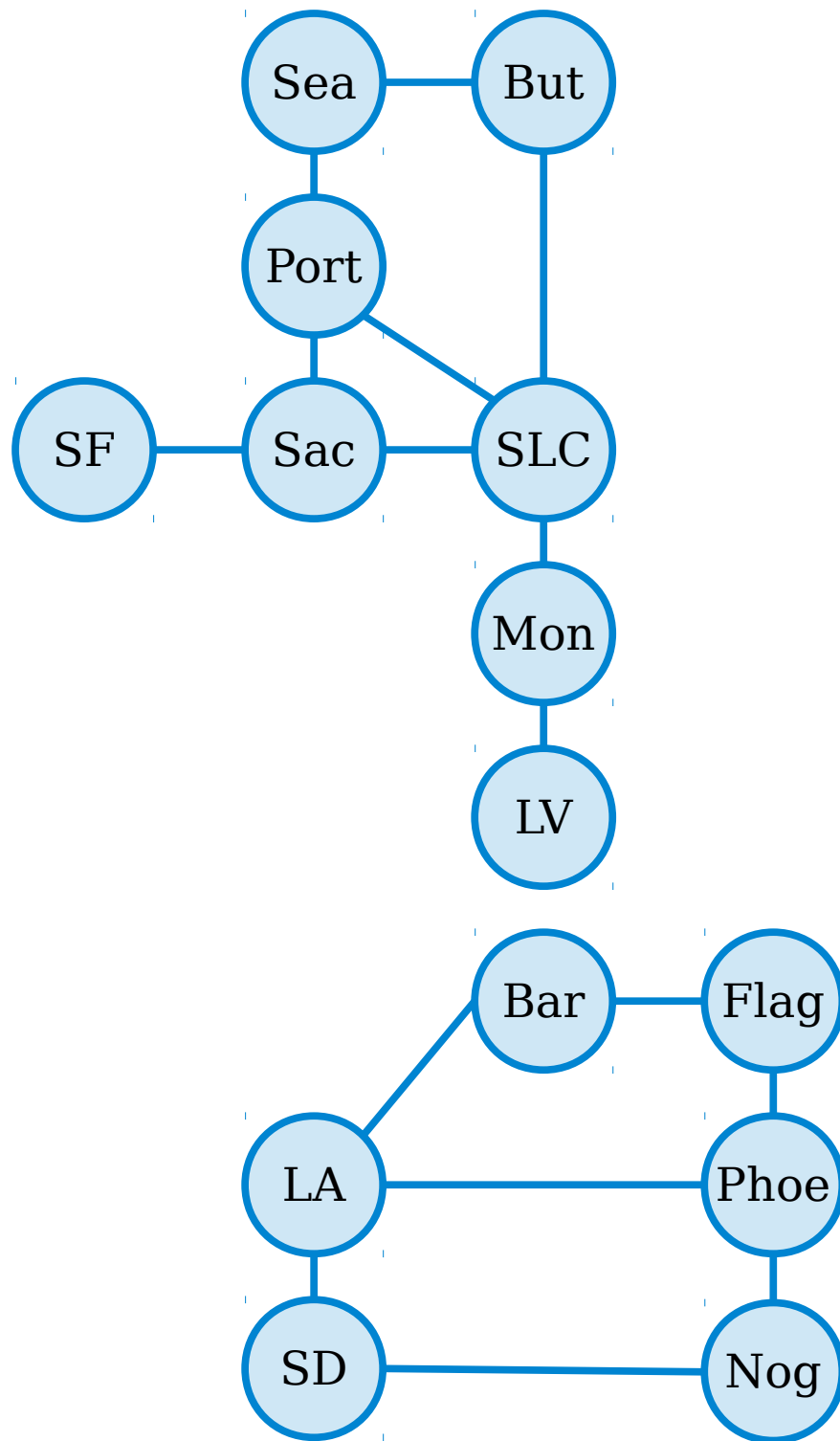
A node v is **reachable** from a node u if there is a path from u to v .



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 if there is a path from u to v .

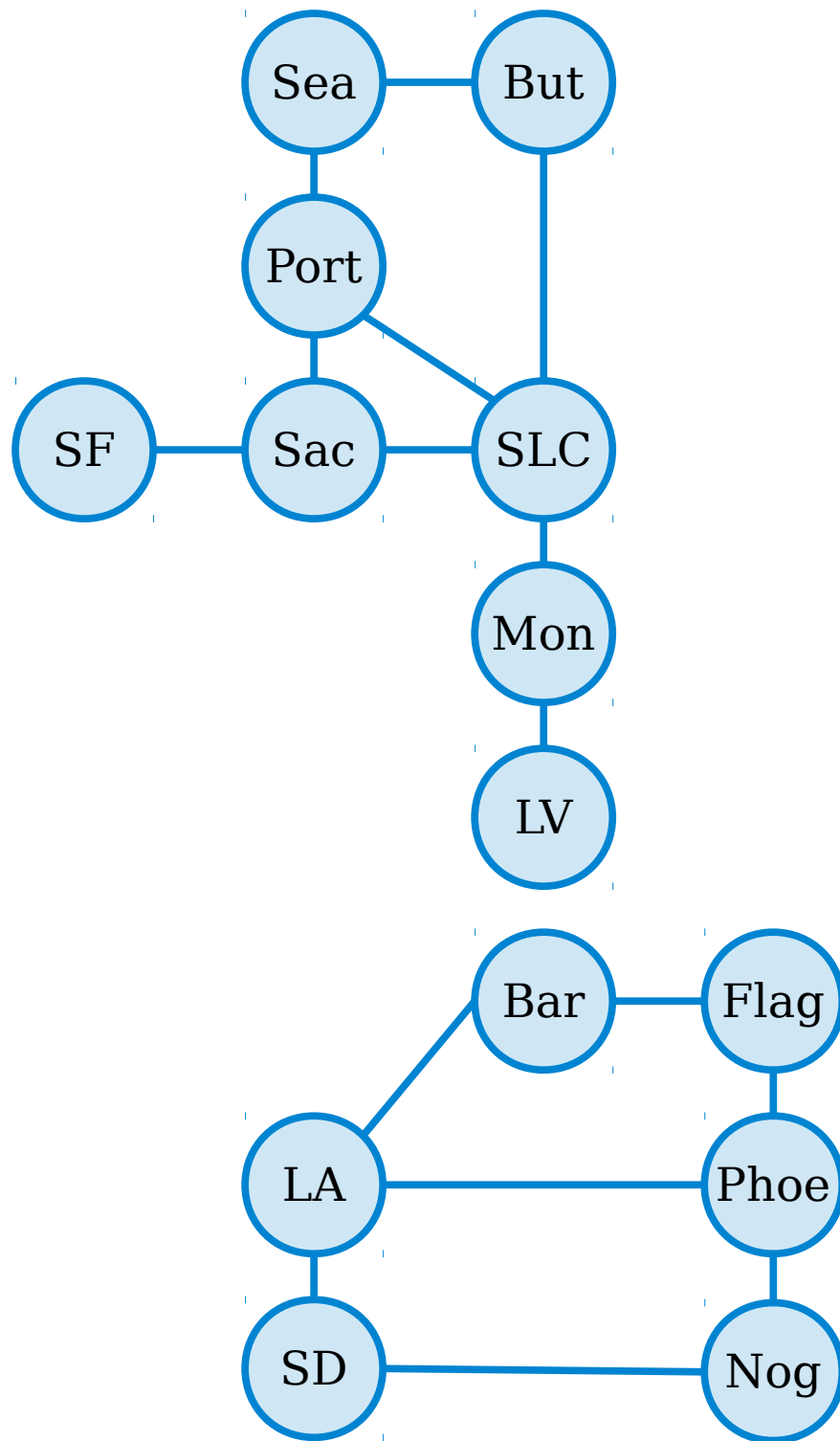


A **walk** in a graph $G = (V, E)$ is a sequence of one or more nodes $v_1, v_2, v_3, \dots, v_n$ such that any two consecutive nodes in the sequence are adjacent.

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

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

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



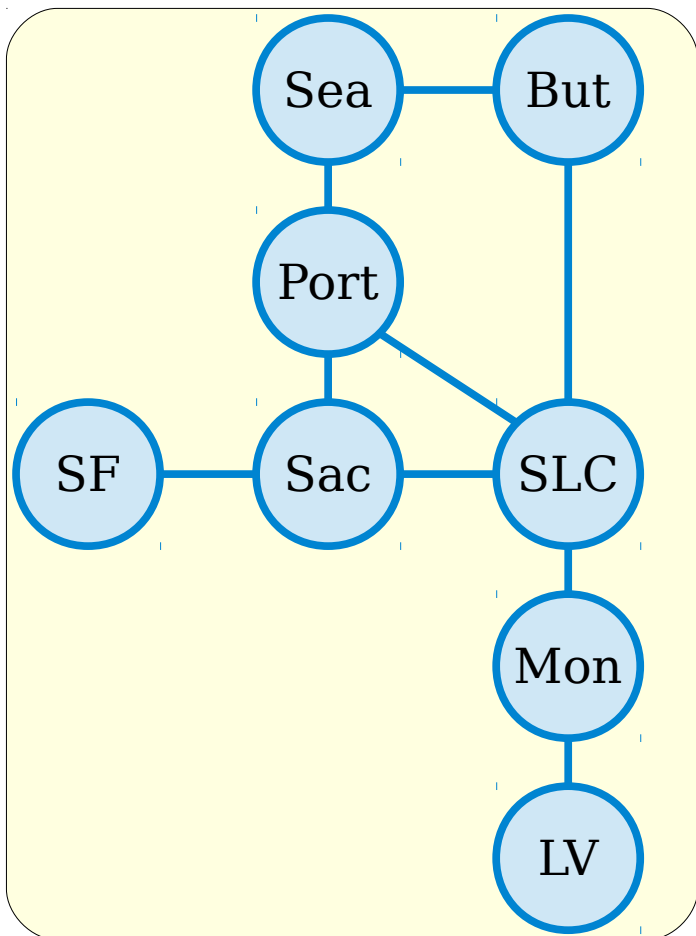
A **walk** in a graph $G = (V, E)$ is a sequence of one or more nodes $v_1, v_2, v_3, \dots, v_n$ such that any two consecutive nodes in the sequence are adjacent.

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

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

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

(This graph is not connected.)

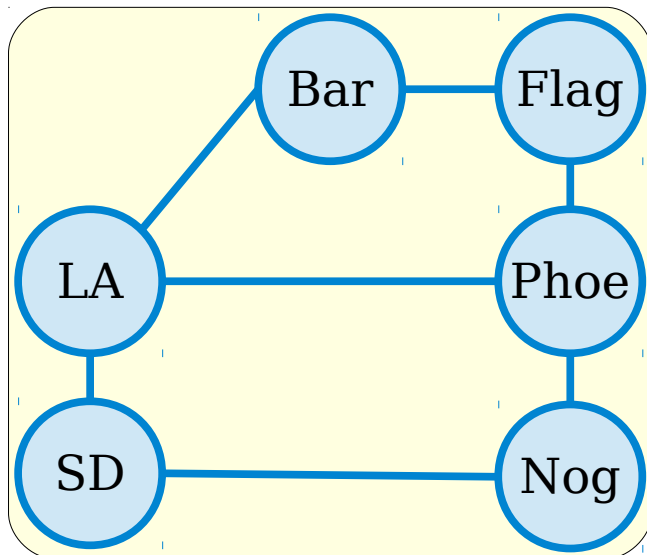


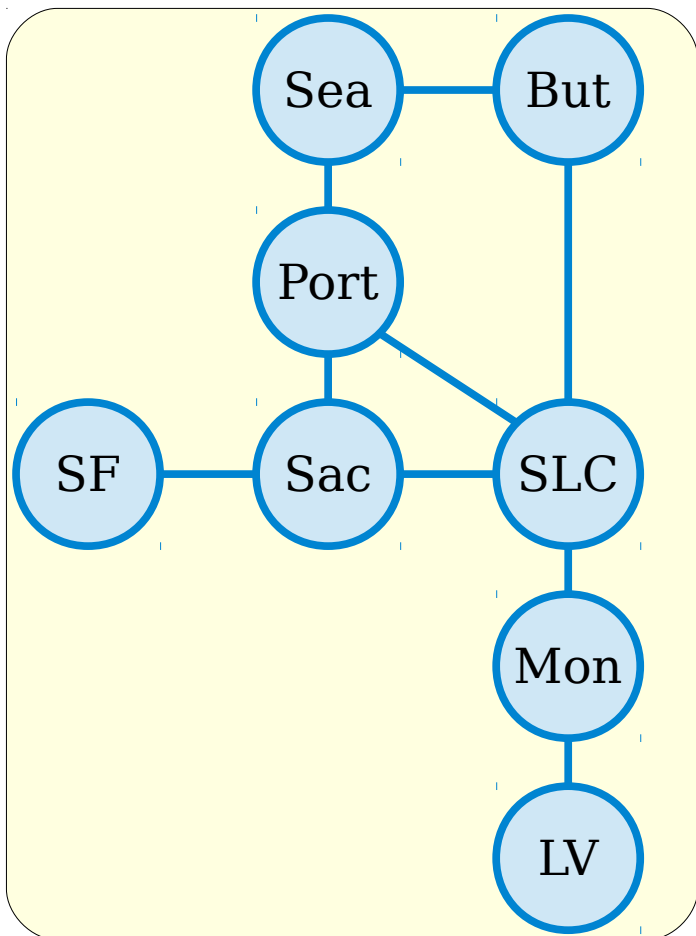
A **walk** in a graph $G = (V, E)$ is a sequence of one or more nodes $v_1, v_2, v_3, \dots, v_n$ such that any two consecutive nodes in the sequence are adjacent.

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

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

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





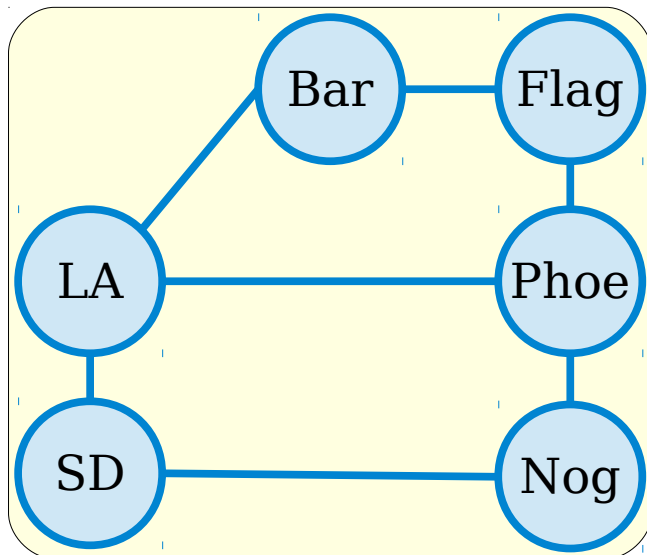
A **walk** in a graph $G = (V, E)$ is a sequence of one or more nodes $v_1, v_2, v_3, \dots, v_n$ such that any two consecutive nodes in the sequence are adjacent.

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

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

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

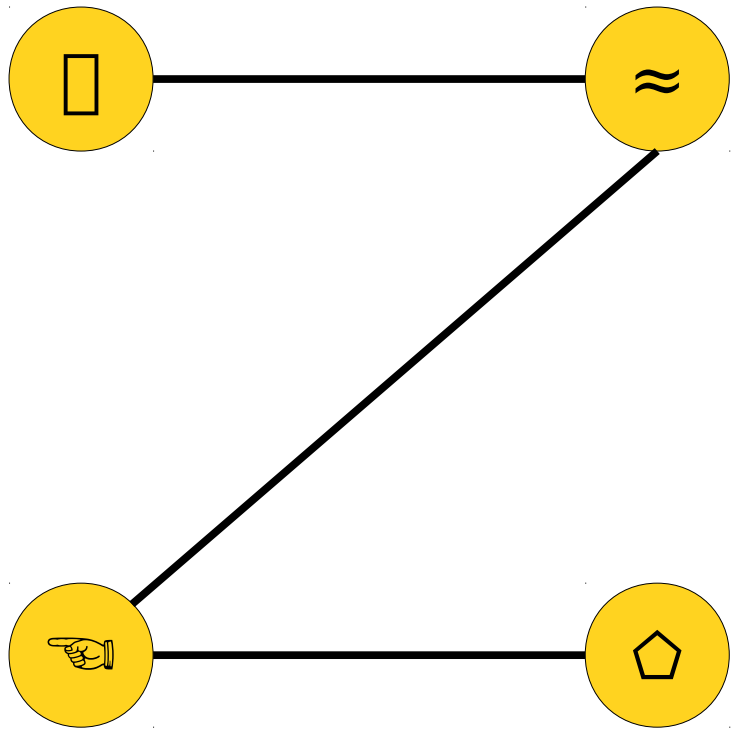
A **connected component** (or **CC**) of G is a maximal set of mutually reachable nodes.



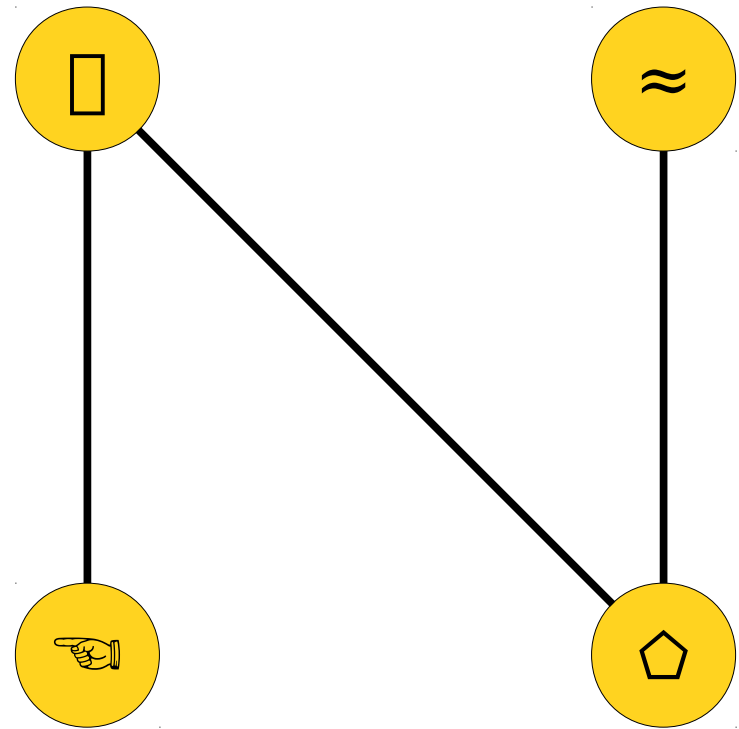
Fun Facts

- Here's a collection of useful facts about graphs that you can take as a given.
 - **Theorem:** If $G = (V, E)$ is a 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 a 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 a graph, then every node in V belongs to exactly one connected component of G .
 - **Theorem:** If $G = (V, E)$ is a graph, then G is connected if and only if G has exactly one connected component.
- Looking for more practice working with formal definitions? Prove these results!

Graph Complements



Graph G



Graph G^c

Let $G = (V, E)$ be an undirected graph.
The **complement of G** is the graph $G^c = (V, E^c)$, where
$$E^c = \{ \{u, v\} \mid u \in V, v \in V, u \neq v, \text{ and } \{u, v\} \notin E \}$$

Theorem: For any graph $G = (V, E)$,
at least one of G and G^c is connected.

Proving a Disjunction

- We need to prove the statement

G is connected $\vee G^c$ is connected.

- Here's a neat observation.
 - If G is connected, we're done.
 - Otherwise, G isn't connected, and we have to prove that G^c is connected.
- We will therefore prove

G is not connected $\rightarrow G^c$ is connected.

For any graph $G = (V, E)$,
at least one of G and G^c is connected.

Proving a Disjunction

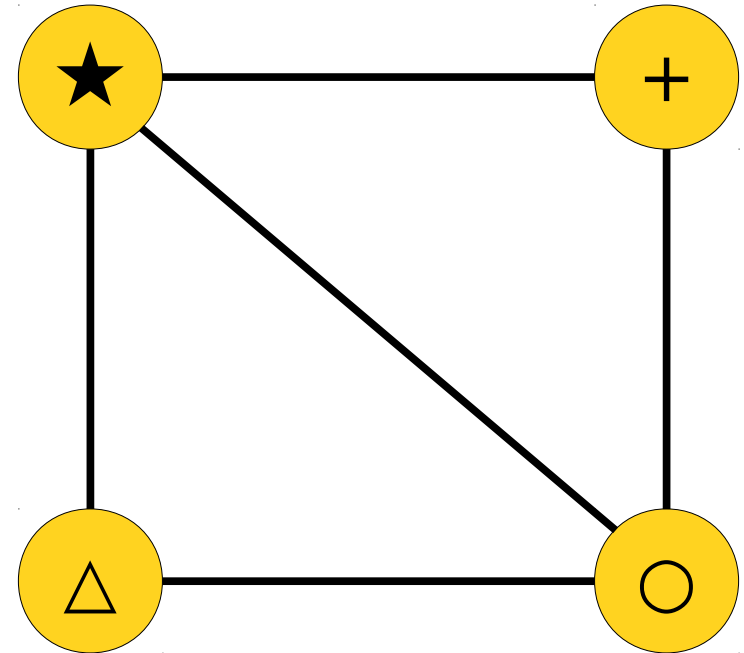
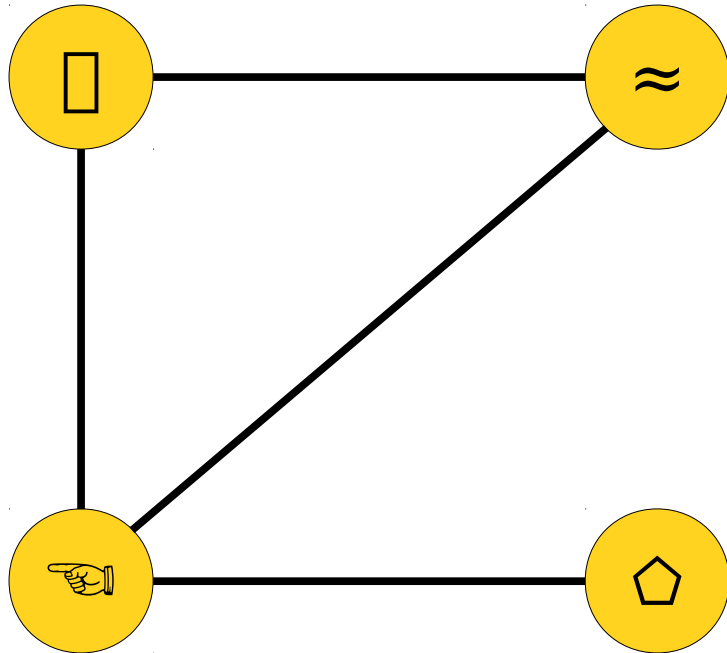
- We need to prove the statement

G is connected $\vee G^c$ is connected.

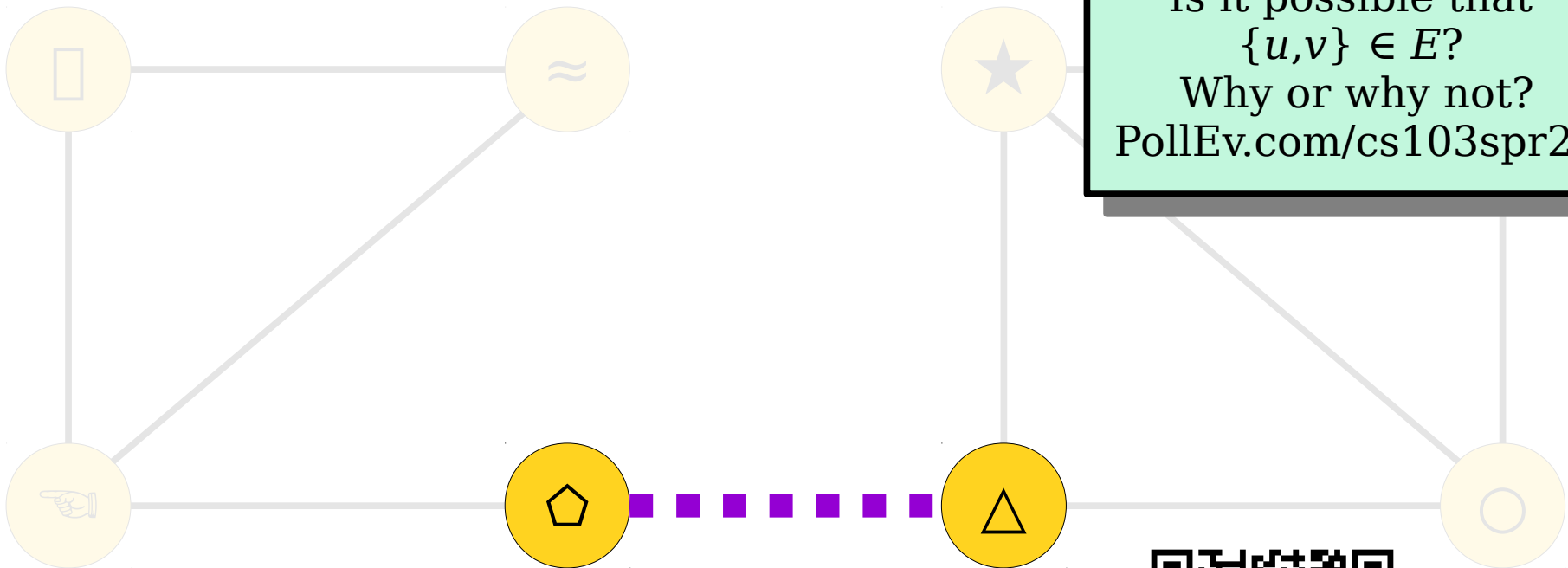
- Here's a neat observation.
 - If G is connected, we're done.
 - Otherwise, G isn't connected, and we have to prove that G^c is connected.
- We will therefore prove

G is not connected $\rightarrow G^c$ is connected.

For any graph $G = (V, E)$,
if G is not connected, then G^c is connected.



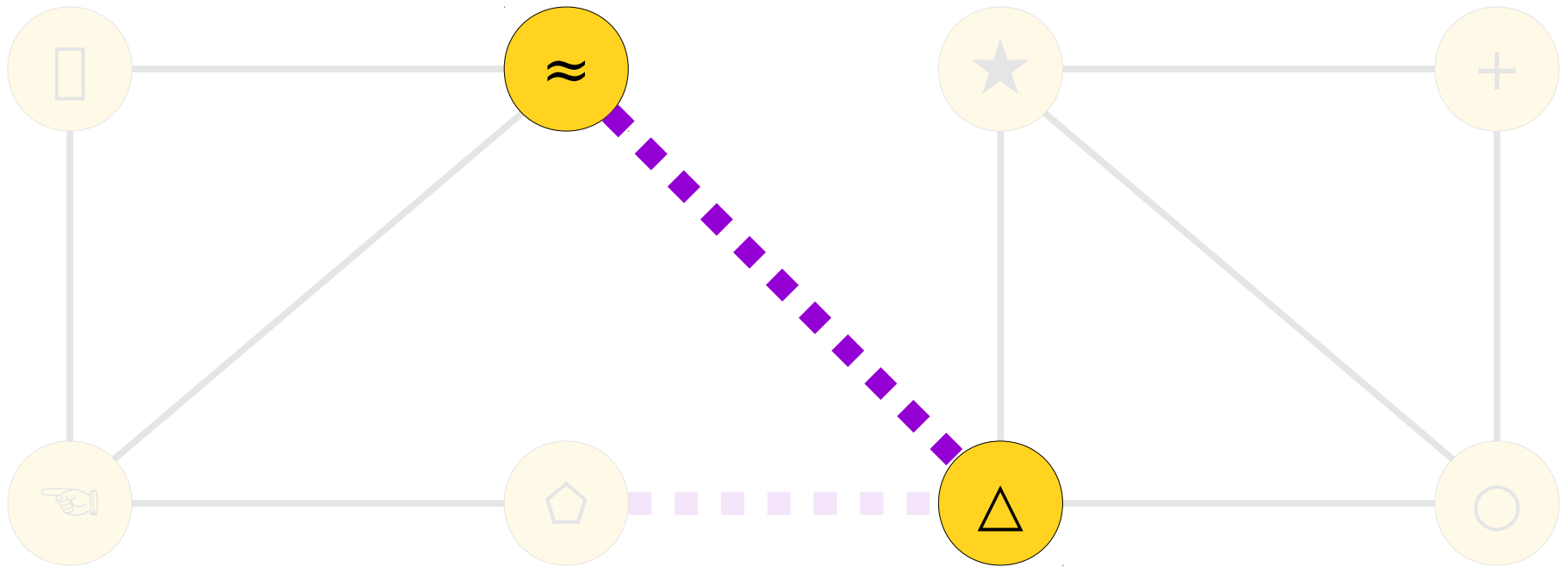
For any graph $G = (V, E)$,
if G is not connected, then G^c is connected.



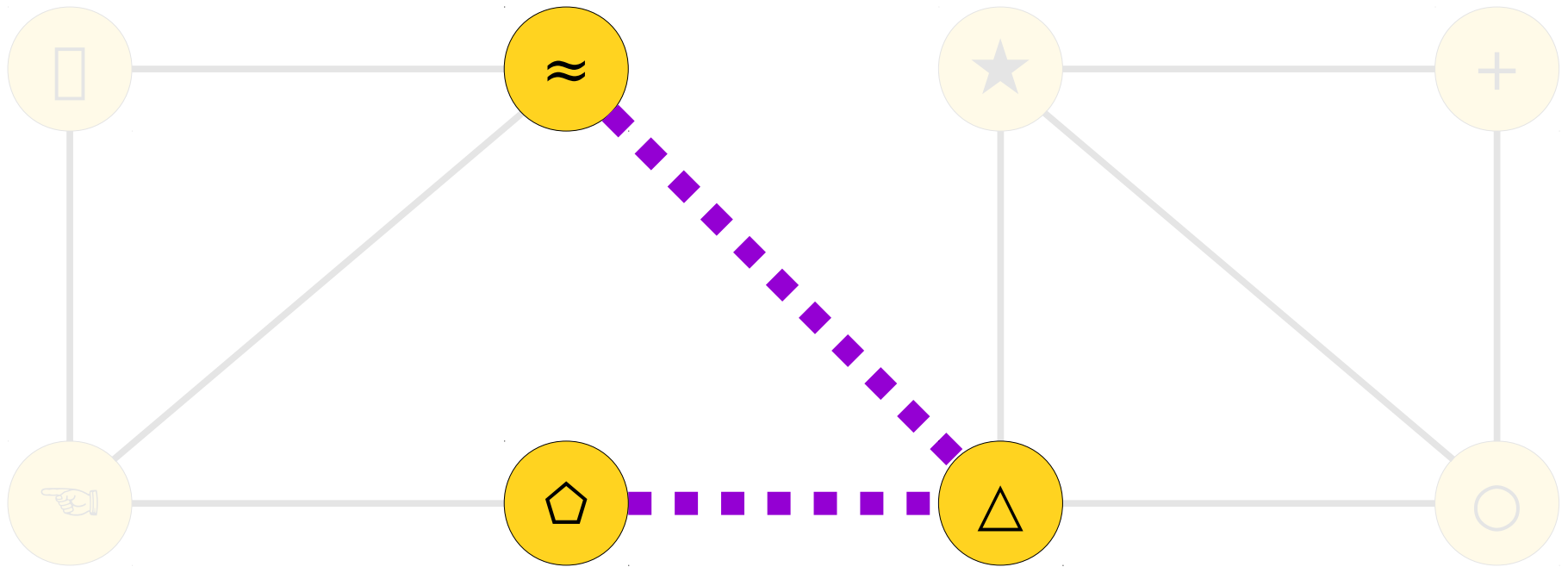
Pick arbitrary nodes u and v from different connected components. Is it possible that $\{u,v\} \in E$? Why or why not? PollEv.com/cs103spr26



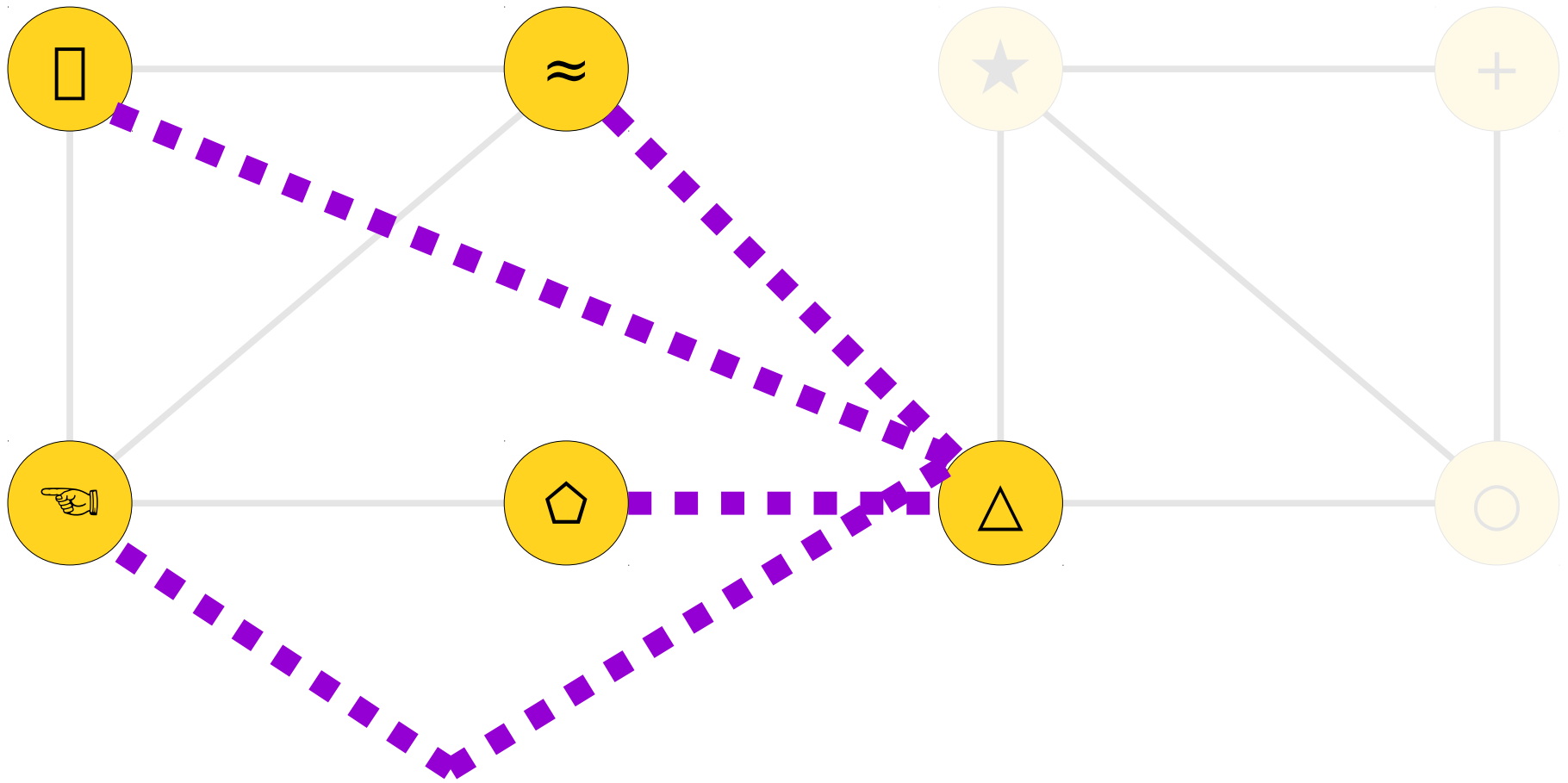
For any graph $G = (V, E)$,
if G is not connected, then G^c is connected.



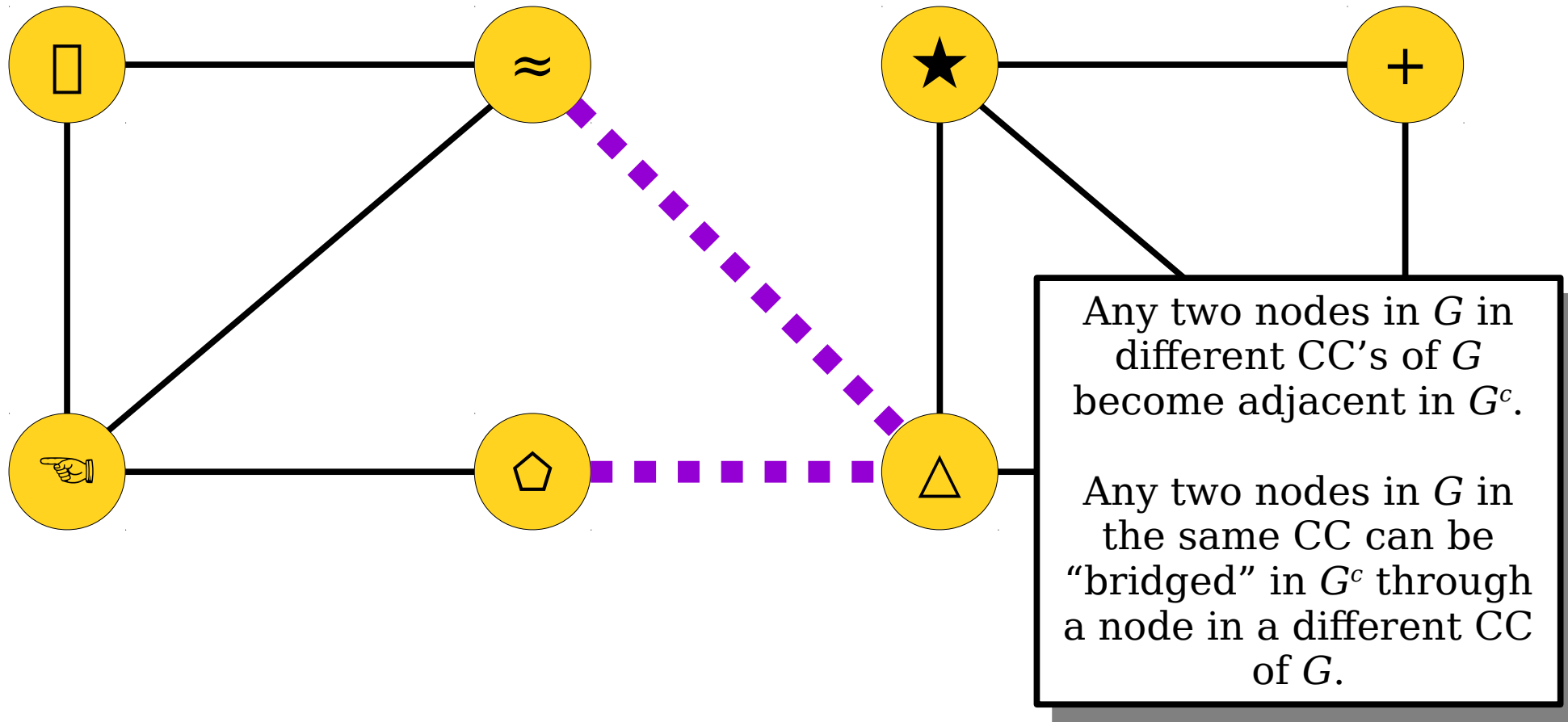
For any graph $G = (V, E)$,
if G is not connected, then G^c is connected.



For any graph $G = (V, E)$,
if G is not connected, then G^c is connected.



For any graph $G = (V, E)$,
if G is not connected, then G^c is connected.



For any graph $G = (V, E)$,
if G is not connected, then G^c is connected.

Theorem: If $G = (V, E)$ is a graph, then at least one of G and G^c is connected.

Theorem: If $G = (V, E)$ is a graph, then at least one of G and G^c is connected.

Proof:

Theorem: If $G = (V, E)$ is a graph, then at least one of G and G^c is connected.

Proof: Let $G = (V, E)$ be an arbitrary graph and assume G is not connected.

Theorem: If $G = (V, E)$ is a graph, then at least one of G and G^c is connected.

Proof: Let $G = (V, E)$ be an arbitrary graph and assume G is not connected. We need to show that $G^c = (V, E^c)$ is connected.

Theorem: If $G = (V, E)$ is a graph, then at least one of G and G^c is connected.

Proof: Let $G = (V, E)$ be an arbitrary graph and assume G is not connected. We need to show that $G^c = (V, E^c)$ is connected. To do so, consider any two distinct nodes $u, v \in V$.

Theorem: If $G = (V, E)$ is a graph, then at least one of G and G^c is connected.

Proof: Let $G = (V, E)$ be an arbitrary graph and assume G is not connected. We need to show that $G^c = (V, E^c)$ is connected. To do so, consider any two distinct nodes $u, v \in V$. We need to show that there is a path from u to v in G^c .

Theorem: If $G = (V, E)$ is a graph, then at least one of G and G^c is connected.

Proof: Let $G = (V, E)$ be an arbitrary graph and assume G is not connected. We need to show that $G^c = (V, E^c)$ is connected. To do so, consider any two distinct nodes $u, v \in V$. We need to show that there is a path from u to v in G^c . We consider two cases:

Case 1: u and v are in different connected components of G .

Case 2: u and v are in the same connected component of G .

Theorem: If $G = (V, E)$ is a graph, then at least one of G and G^c is connected.

Proof: Let $G = (V, E)$ be an arbitrary graph and assume G is not connected. We need to show that $G^c = (V, E^c)$ is connected. To do so, consider any two distinct nodes $u, v \in V$. We need to show that there is a path from u to v in G^c . We consider two cases:

Case 1: u and v are in different connected components of G . This means that $\{u, v\} \notin E$, since otherwise the path u, v would make u reachable from v and they'd be in the same connected component of G .

Case 2: u and v are in the same connected component of G .

Theorem: If $G = (V, E)$ is a graph, then at least one of G and G^c is connected.

Proof: Let $G = (V, E)$ be an arbitrary graph and assume G is not connected. We need to show that $G^c = (V, E^c)$ is connected. To do so, consider any two distinct nodes $u, v \in V$. We need to show that there is a path from u to v in G^c . We consider two cases:

Case 1: u and v are in different connected components of G . This means that $\{u, v\} \notin E$, since otherwise the path u, v would make u reachable from v and they'd be in the same connected component of G . Therefore, we see that $\{u, v\} \in E^c$, and so there is a path (namely, u, v) from u to v in G^c .

Case 2: u and v are in the same connected component of G .

Theorem: If $G = (V, E)$ is a graph, then at least one of G and G^c is connected.

Proof: Let $G = (V, E)$ be an arbitrary graph and assume G is not connected. We need to show that $G^c = (V, E^c)$ is connected. To do so, consider any two distinct nodes $u, v \in V$. We need to show that there is a path from u to v in G^c . We consider two cases:

Case 1: u and v are in different connected components of G . This means that $\{u, v\} \notin E$, since otherwise the path u, v would make u reachable from v and they'd be in the same connected component of G . Therefore, we see that $\{u, v\} \in E^c$, and so there is a path (namely, u, v) from u to v in G^c .

Case 2: u and v are in the same connected component of G . Since G is not connected, there is at least one node that is not connected to u or v —pick one such node and call it z .

Theorem: If $G = (V, E)$ is a graph, then at least one of G and G^c is connected.

Proof: Let $G = (V, E)$ be an arbitrary graph and assume G is not connected. We need to show that $G^c = (V, E^c)$ is connected. To do so, consider any two distinct nodes $u, v \in V$. We need to show that there is a path from u to v in G^c . We consider two cases:

Case 1: u and v are in different connected components of G . This means that $\{u, v\} \notin E$, since otherwise the path u, v would make u reachable from v and they'd be in the same connected component of G . Therefore, we see that $\{u, v\} \in E^c$, and so there is a path (namely, u, v) from u to v in G^c .

Case 2: u and v are in the same connected component of G . Since G is not connected, there is at least one node that is not connected to u or v —pick one such node and call it z . Then by the reasoning from Case 1, we know that $\{u, z\} \in E^c$ and $\{z, v\} \in E^c$.

Theorem: If $G = (V, E)$ is a graph, then at least one of G and G^c is connected.

Proof: Let $G = (V, E)$ be an arbitrary graph and assume G is not connected. We need to show that $G^c = (V, E^c)$ is connected. To do so, consider any two distinct nodes $u, v \in V$. We need to show that there is a path from u to v in G^c . We consider two cases:

Case 1: u and v are in different connected components of G . This means that $\{u, v\} \notin E$, since otherwise the path u, v would make u reachable from v and they'd be in the same connected component of G . Therefore, we see that $\{u, v\} \in E^c$, and so there is a path (namely, u, v) from u to v in G^c .

Case 2: u and v are in the same connected component of G . Since G is not connected, there is at least one node that is not connected to u or v —pick one such node and call it z . Then by the reasoning from Case 1, we know that $\{u, z\} \in E^c$ and $\{z, v\} \in E^c$. This gives a path u, z, v in G^c from u to v .

Theorem: If $G = (V, E)$ is a graph, then at least one of G and G^c is connected.

Proof: Let $G = (V, E)$ be an arbitrary graph and assume G is not connected. We need to show that $G^c = (V, E^c)$ is connected. To do so, consider any two distinct nodes $u, v \in V$. We need to show that there is a path from u to v in G^c . We consider two cases:

Case 1: u and v are in different connected components of G . This means that $\{u, v\} \notin E$, since otherwise the path u, v would make u reachable from v and they'd be in the same connected component of G . Therefore, we see that $\{u, v\} \in E^c$, and so there is a path (namely, u, v) from u to v in G^c .

Case 2: u and v are in the same connected component of G . Since G is not connected, there is at least one node that is not connected to u or v —pick one such node and call it z . Then by the reasoning from Case 1, we know that $\{u, z\} \in E^c$ and $\{z, v\} \in E^c$. This gives a path u, z, v in G^c from u to v .

In either case, we find a path from u to v in G^c , as required.

Theorem: If $G = (V, E)$ is a graph, then at least one of G and G^c is connected.

Proof: Let $G = (V, E)$ be an arbitrary graph and assume G is not connected. We need to show that $G^c = (V, E^c)$ is connected. To do so, consider any two distinct nodes $u, v \in V$. We need to show that there is a path from u to v in G^c . We consider two cases:

Case 1: u and v are in different connected components of G . This means that $\{u, v\} \notin E$, since otherwise the path u, v would make u reachable from v and they'd be in the same connected component of G . Therefore, we see that $\{u, v\} \in E^c$, and so there is a path (namely, u, v) from u to v in G^c .

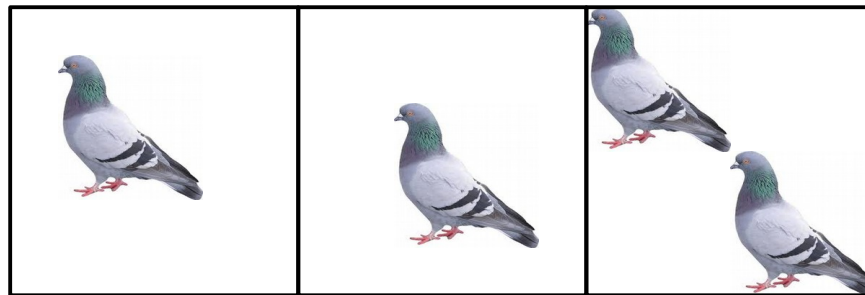
Case 2: u and v are in the same connected component of G . Since G is not connected, there are at least two connected components of G . Pick any node z that belongs to a different connected component of G than u and v . Then by the reasoning from Case 1 we know that $\{u, z\} \in E^c$ and $\{z, v\} \in E^c$. This gives a path u, z, v in G^c from u to v .

In either case, we find a path from u to v in G^c , as required. ■

The Pigeonhole Principle

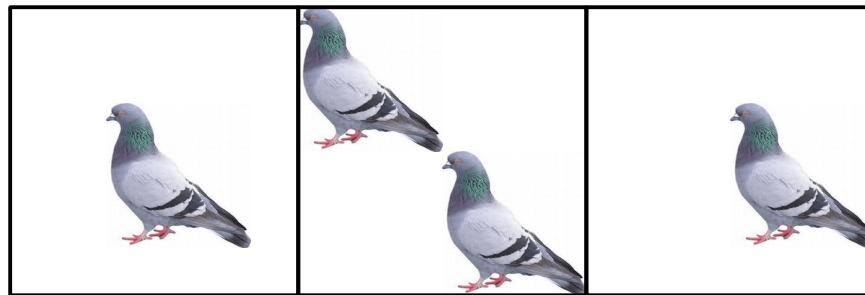
The Pigeonhole Principle

- ***Theorem (The Pigeonhole Principle):***
If m objects are distributed into n bins and $m > n$, then at least one bin will contain at least two objects.



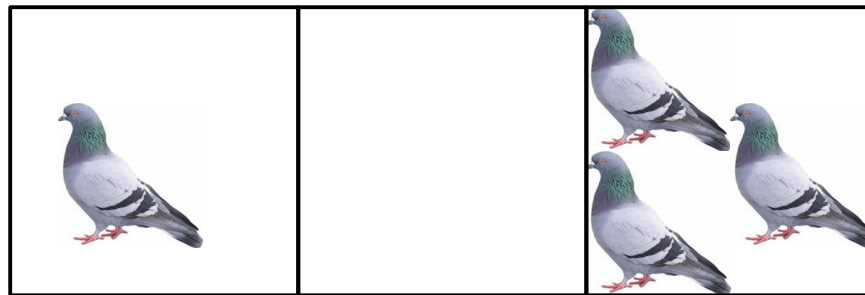
The Pigeonhole Principle

- ***Theorem (The Pigeonhole Principle):***
If m objects are distributed into n bins and $m > n$, then at least one bin will contain at least two objects.



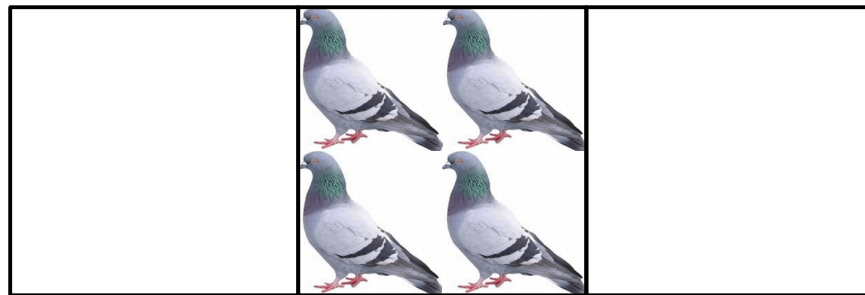
The Pigeonhole Principle

- ***Theorem (The Pigeonhole Principle):***
If m objects are distributed into n bins and $m > n$, then at least one bin will contain at least two objects.



The Pigeonhole Principle

- ***Theorem (The Pigeonhole Principle):***
If m objects are distributed into n bins and $m > n$, then at least one bin will contain at least two objects.



NO MORE
- PIGEON HOLES?!



$$m = 4, n = 3$$

Thanks to Amy Liu for this awesome drawing!

Some Simple Applications

- Any group of 367 people must have a pair of people that share a birthday.
 - 366 possible birthdays (pigeonholes).
 - 367 people (pigeons).
- Two people in San Francisco have the exact same number of hairs on their head.
 - Maximum number of hairs ever found on a human head is no greater than 500,000.
 - There are over 800,000 people in San Francisco.

Theorem (The Pigeonhole Principle): If m objects are distributed into n bins and $m > n$, then at least one bin will contain at least two objects.

Let A and B be finite sets (sets whose cardinalities are natural numbers) and assume $|A| > |B|$. How many of the following statements are true?

- (1) If $f : A \rightarrow B$, then f is injective.
- (2) If $f : A \rightarrow B$, then f is not injective.
- (3) If $f : A \rightarrow B$, then f is surjective.
- (4) If $f : A \rightarrow B$, then f is not surjective.

Proving the Pigeonhole Principle

Theorem: If m objects are distributed into n bins and $m > n$, then there must be some bin that contains at least two objects.

Proof: Suppose for the sake of contradiction that, for some m and n where $m > n$, there is a way to distribute m objects into n bins such that each bin contains at most one object.

Number the bins $1, 2, 3, \dots, n$ and let x_i denote the number of objects in bin i . There are m objects in total, so we know that

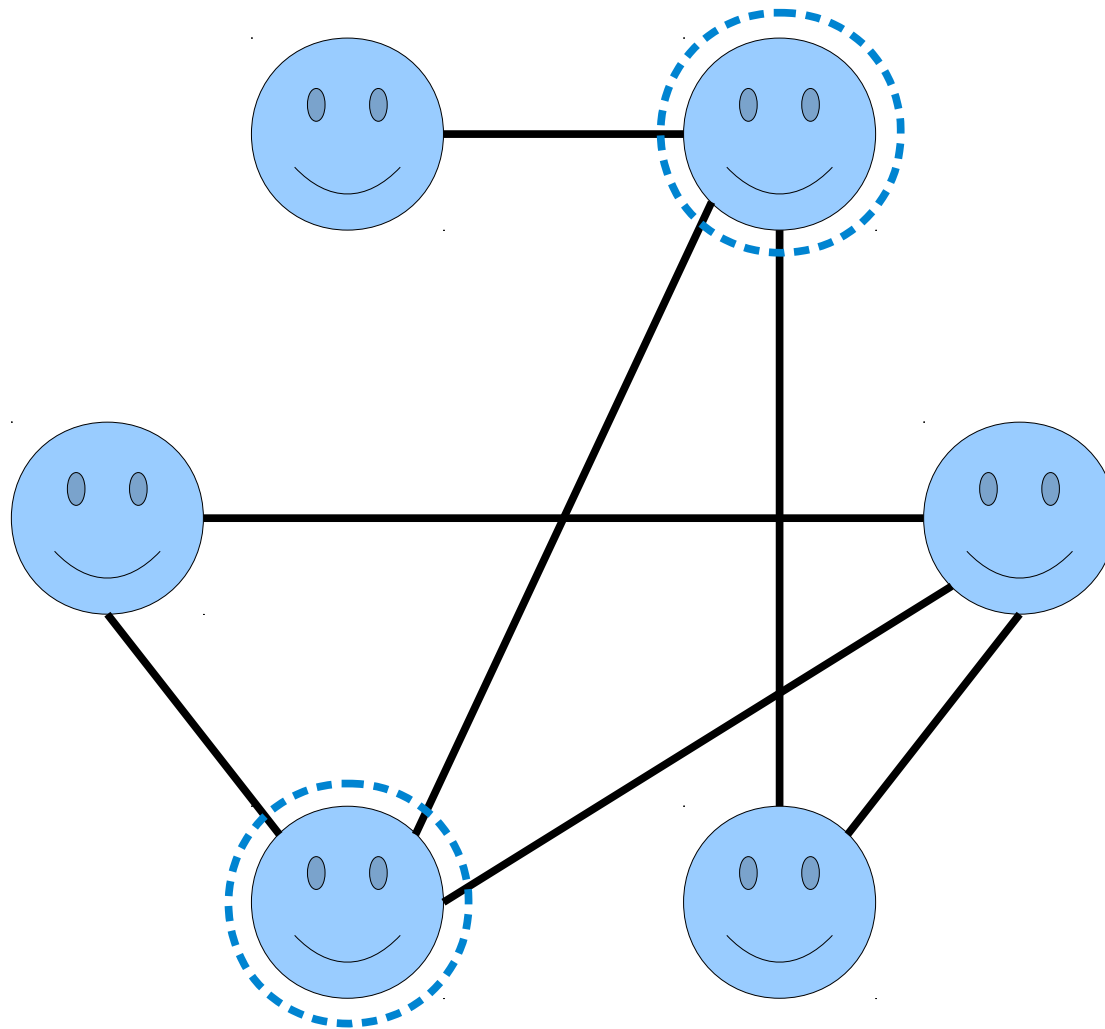
$$m = x_1 + x_2 + \dots + x_n.$$

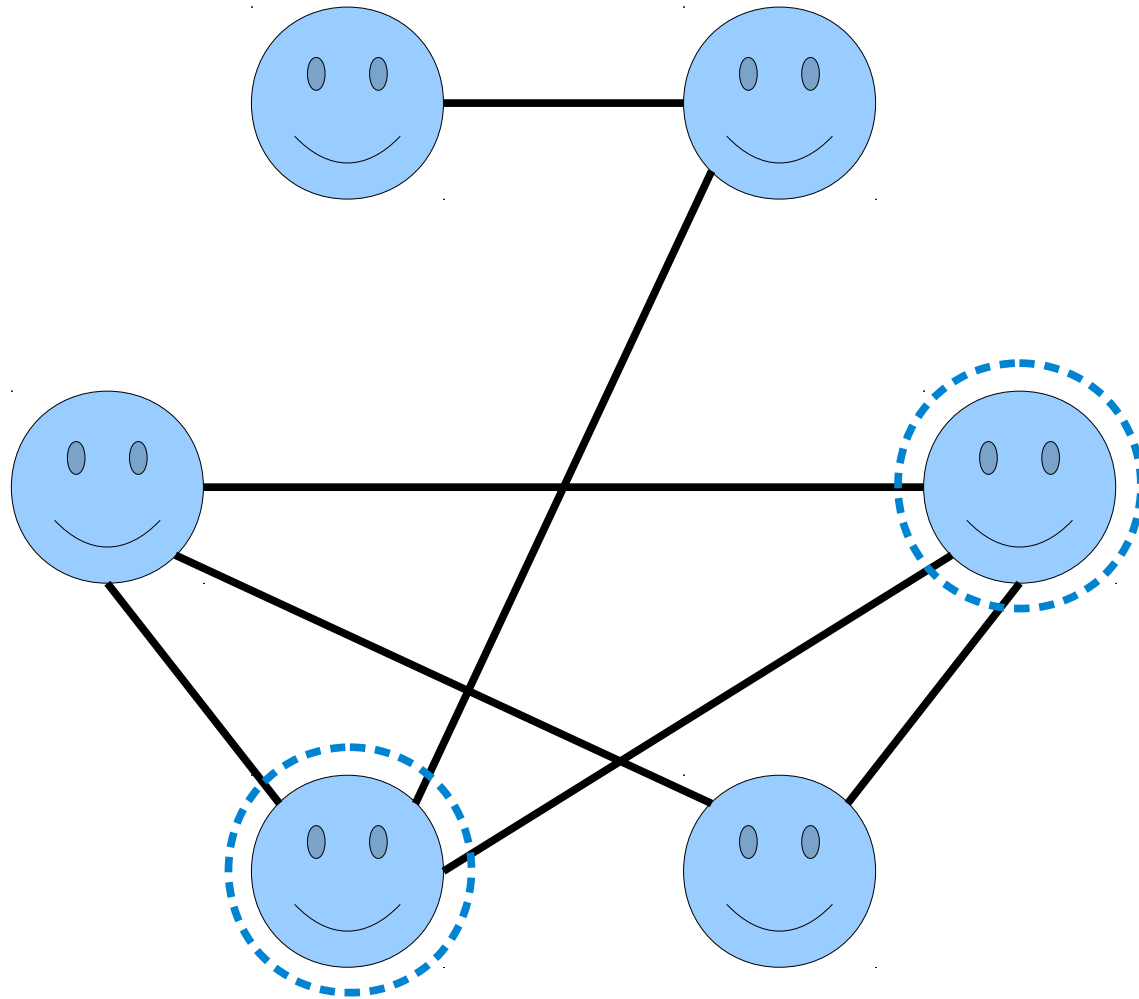
Since each bin has at most one object in it, we know $x_i \leq 1$ for each i . This means that

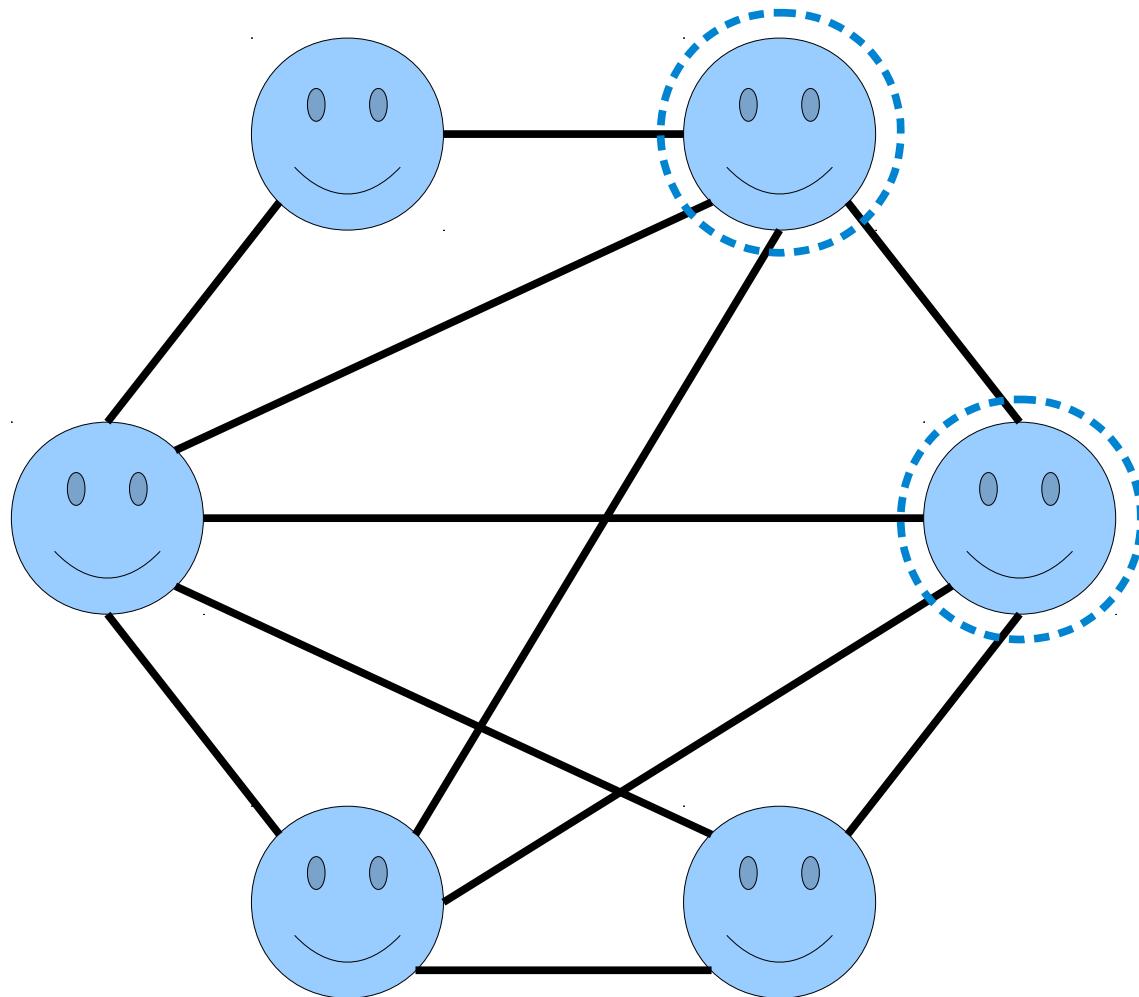
$$\begin{aligned} m &= x_1 + x_2 + \dots + x_n \\ &\leq 1 + 1 + \dots + 1 \quad (n \text{ times}) \\ &= n. \end{aligned}$$

This means that $m \leq n$, contradicting that $m > n$. We've reached a contradiction, so our assumption must have been wrong. Therefore, if m objects are distributed into n bins with $m > n$, some bin must contain at least two objects. ■

Pigeonhole Principle Party Tricks

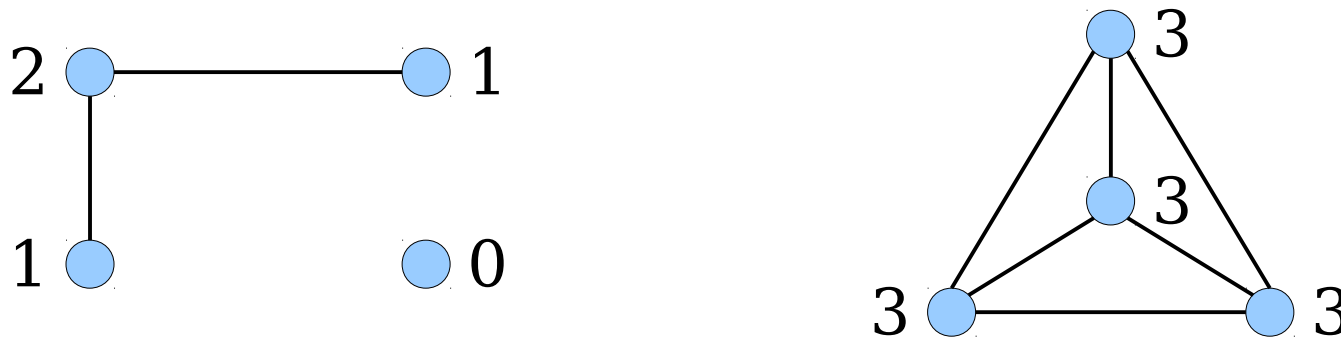




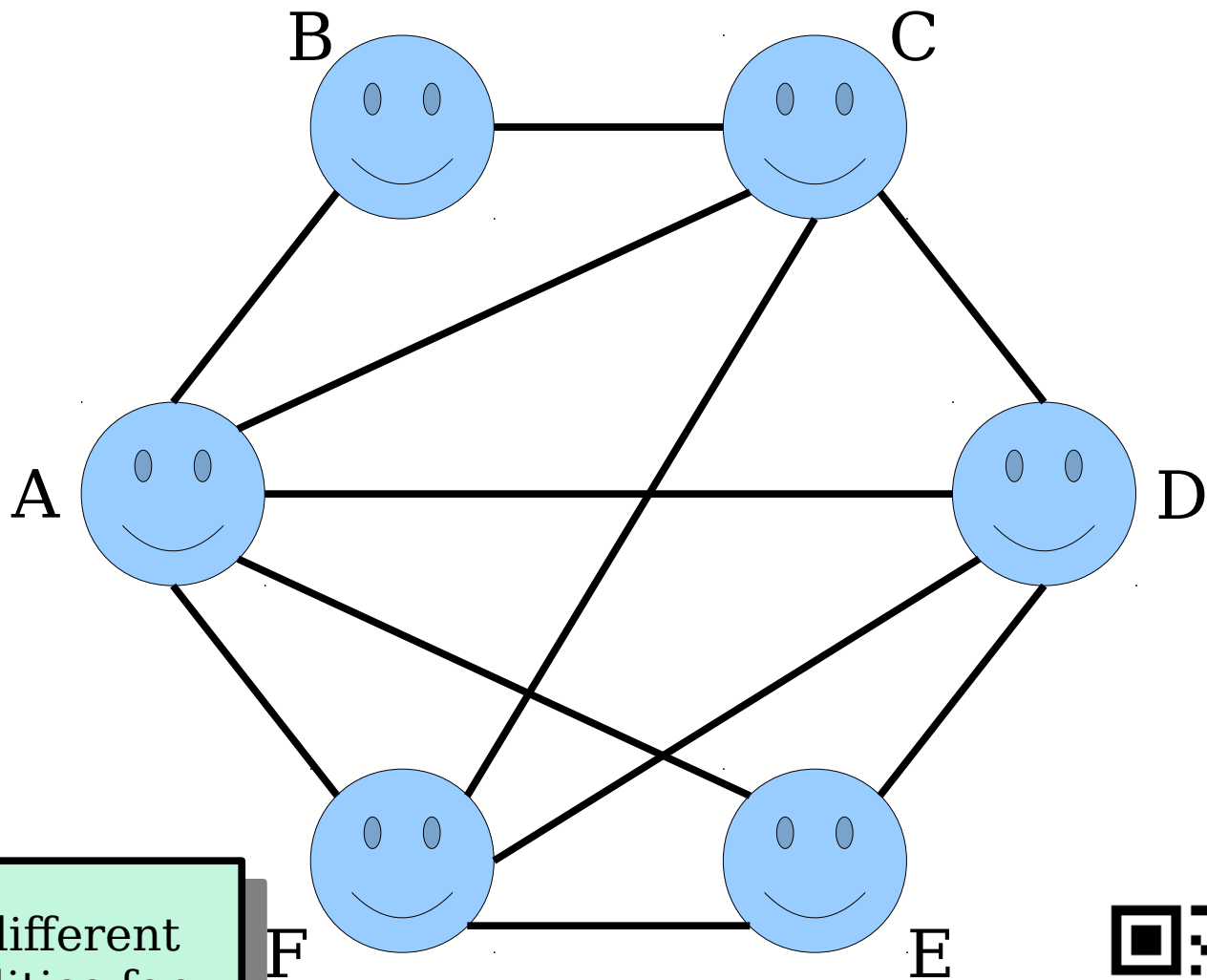


Degrees

- The **degree** of a node v in a graph is the number of nodes that v is adjacent to.



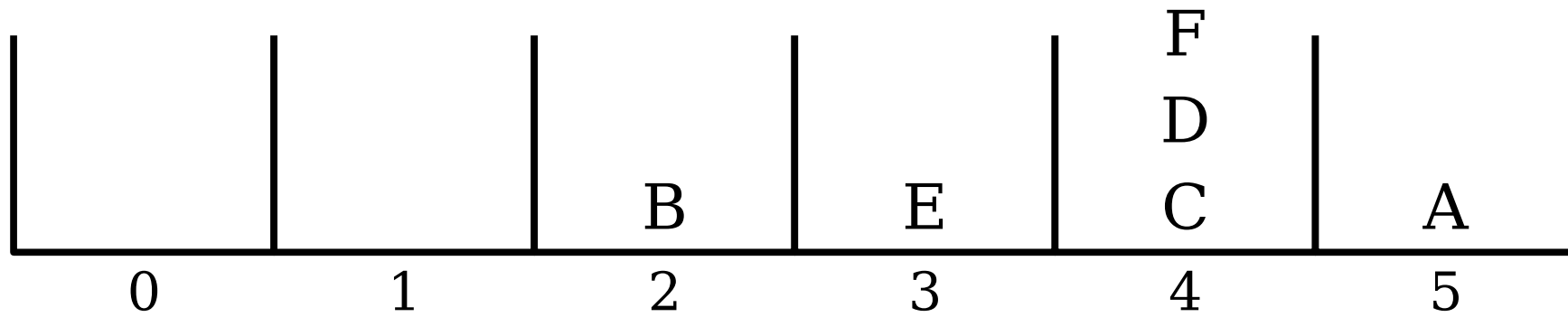
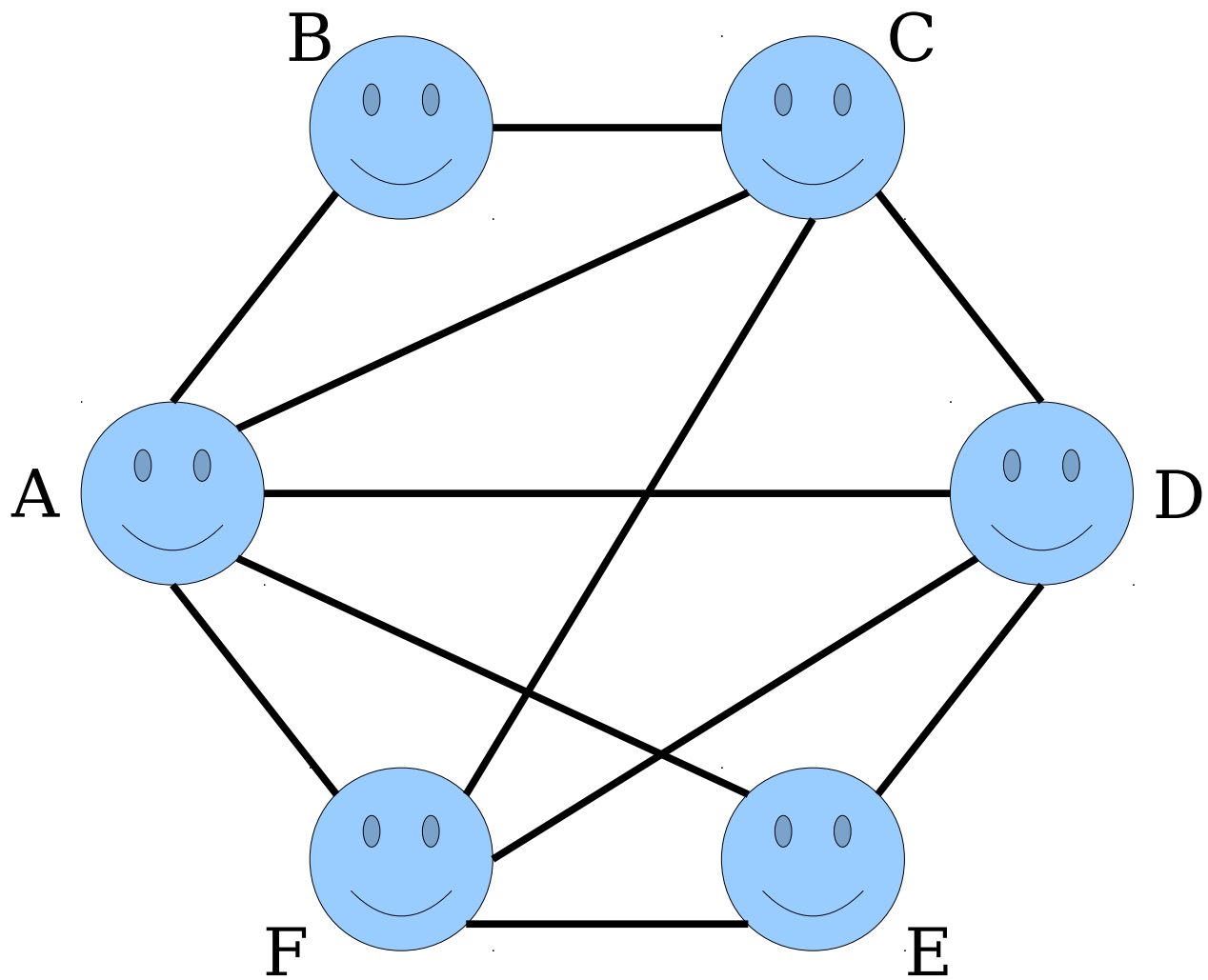
- Theorem:** Every graph with at least two nodes has at least two nodes with the same degree.
 - Equivalently: at any party with at least two people, there are at least two people with the same number of friends at the party.

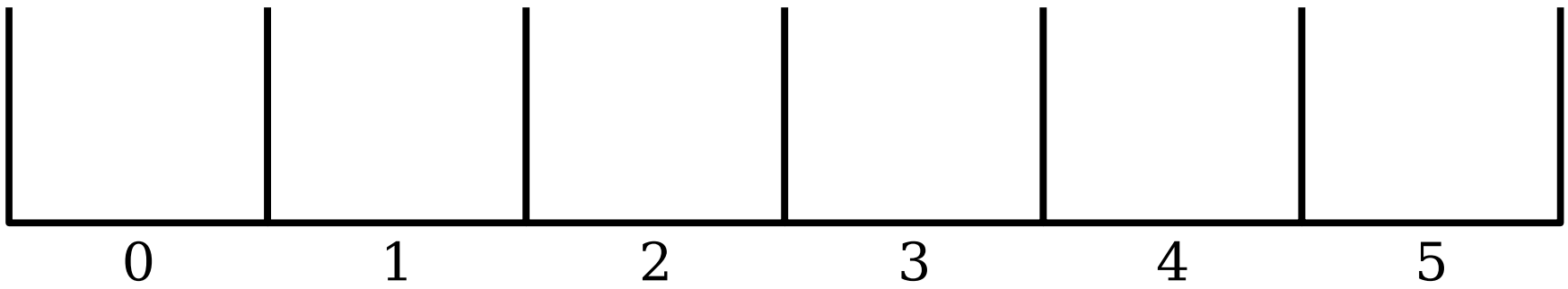
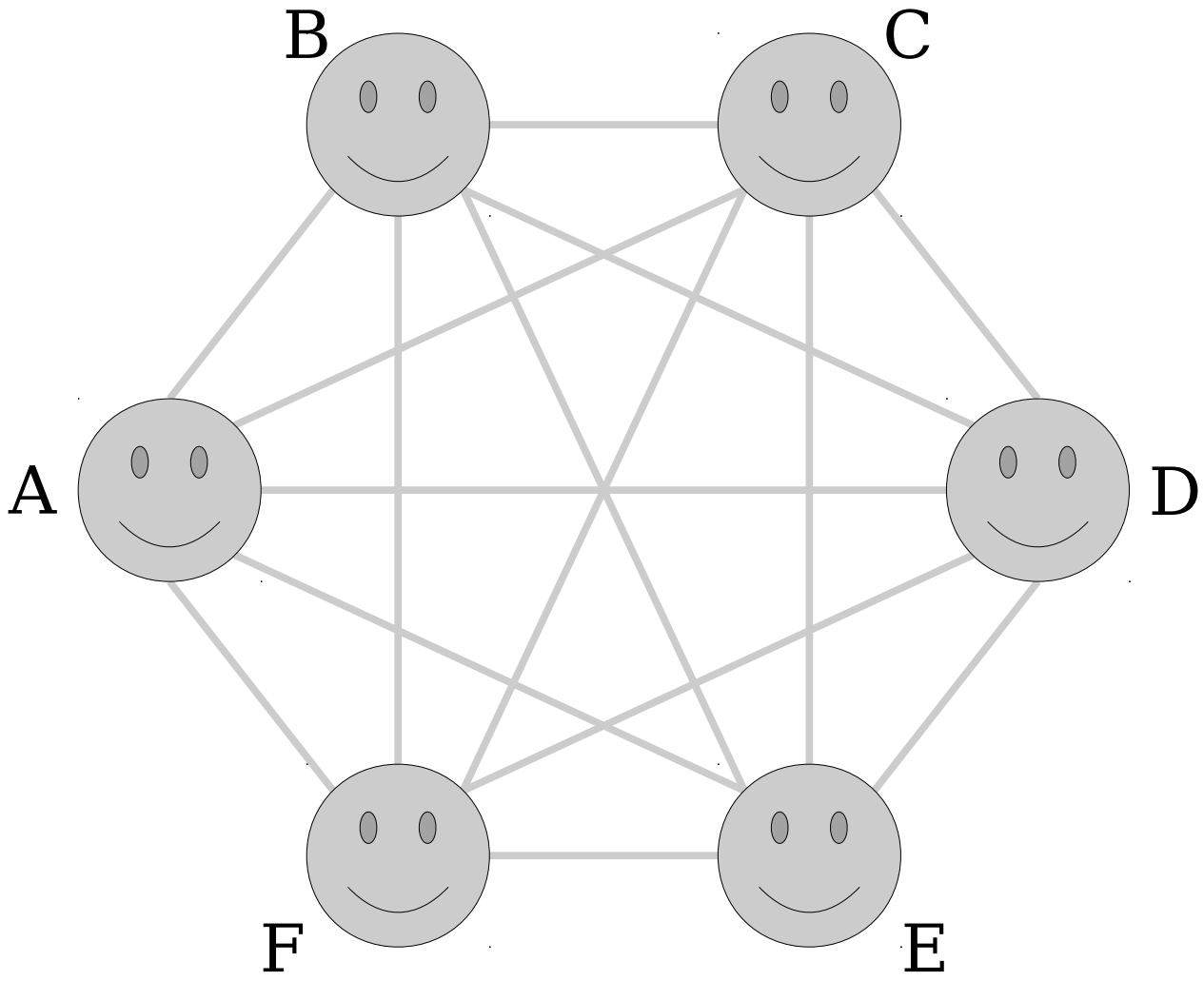


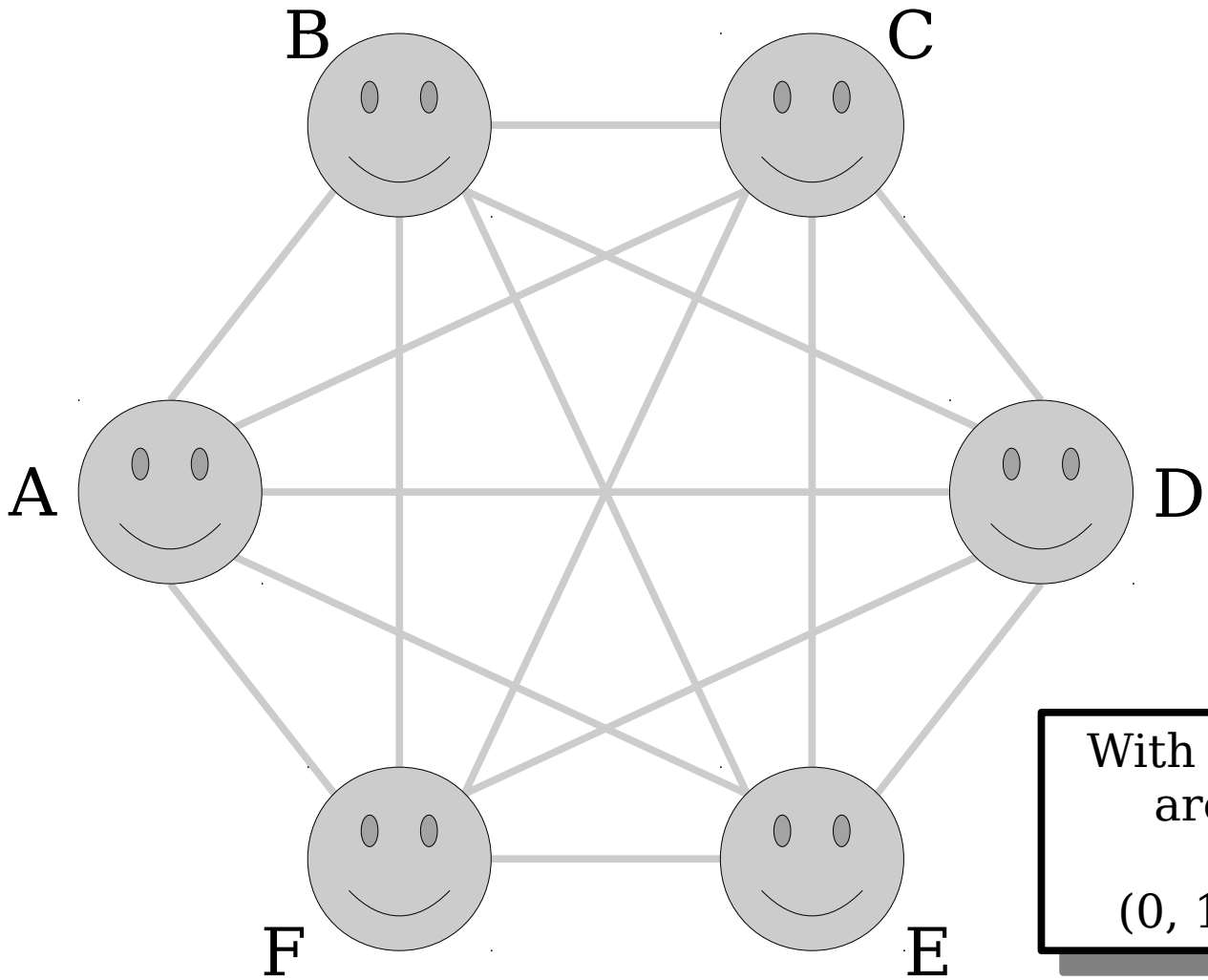
You've seen different edge possibilities for a graph with $|V| = 6$. How many different degrees is it possible for vertex A to have in any such graph?

[PollEv.com/cs103spr26](https://pollEv.com/cs103spr26)

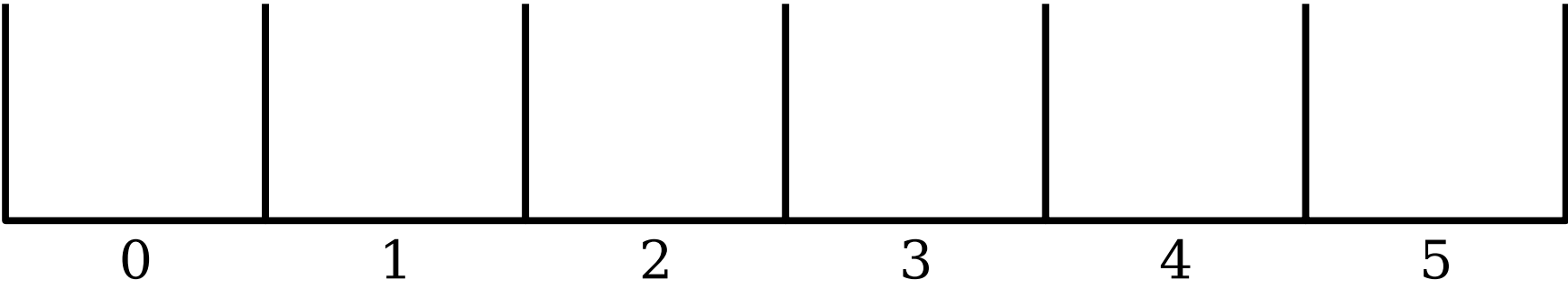


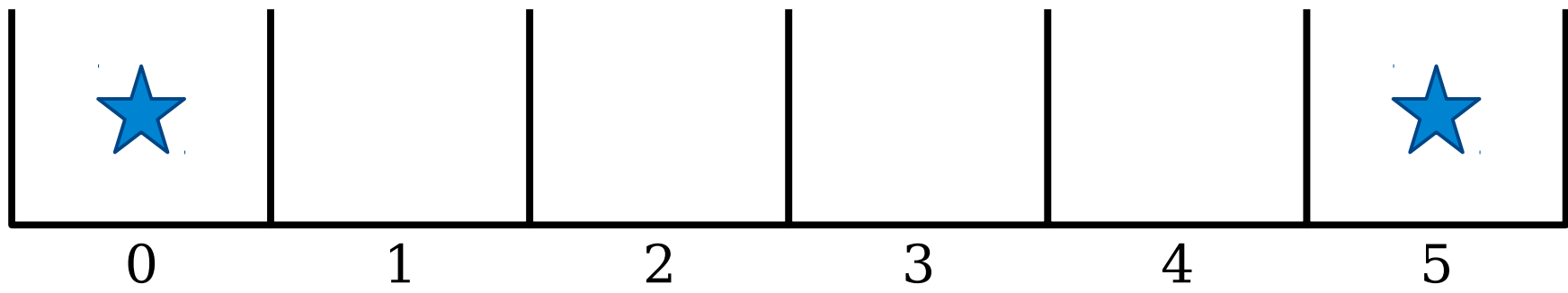
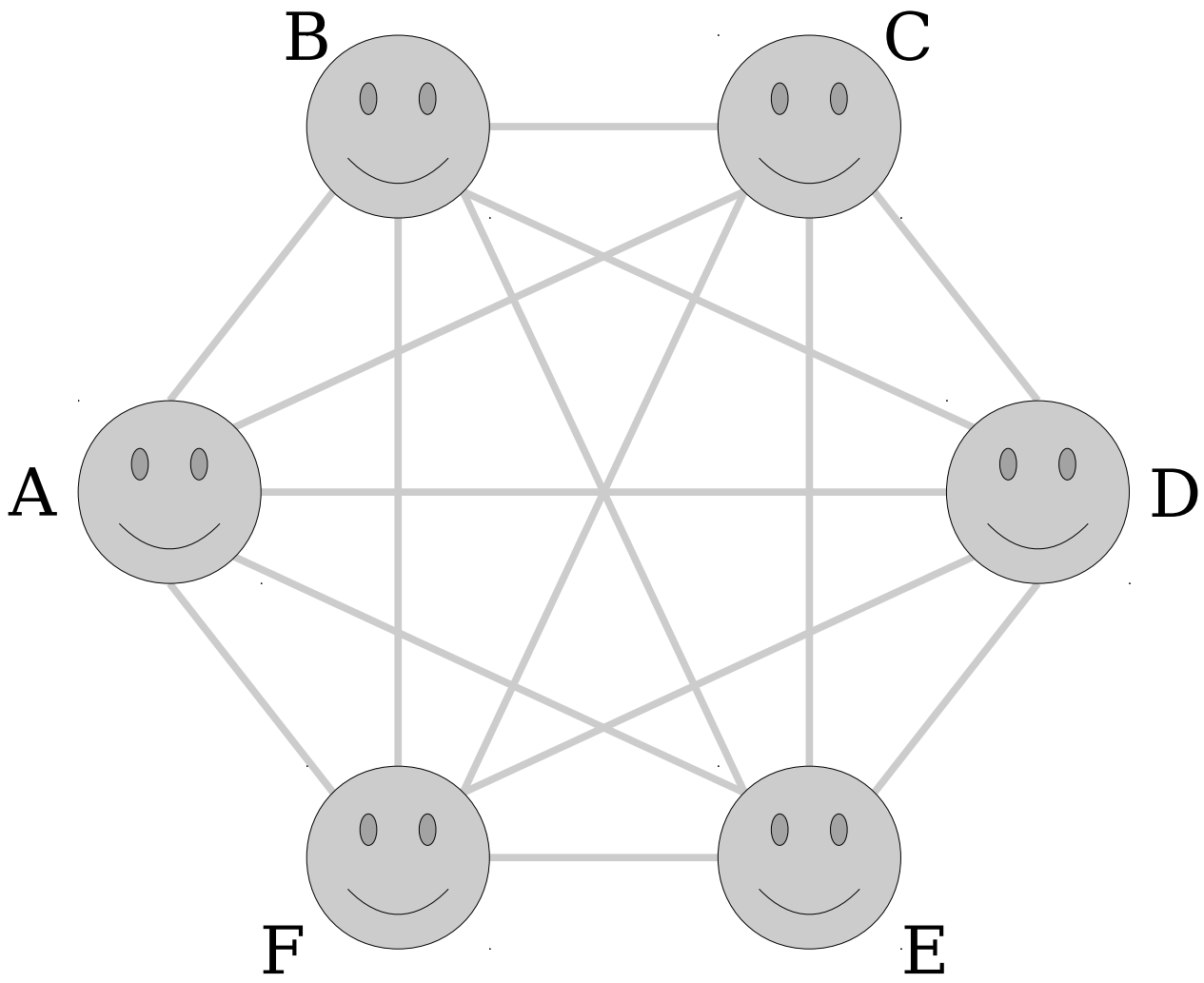


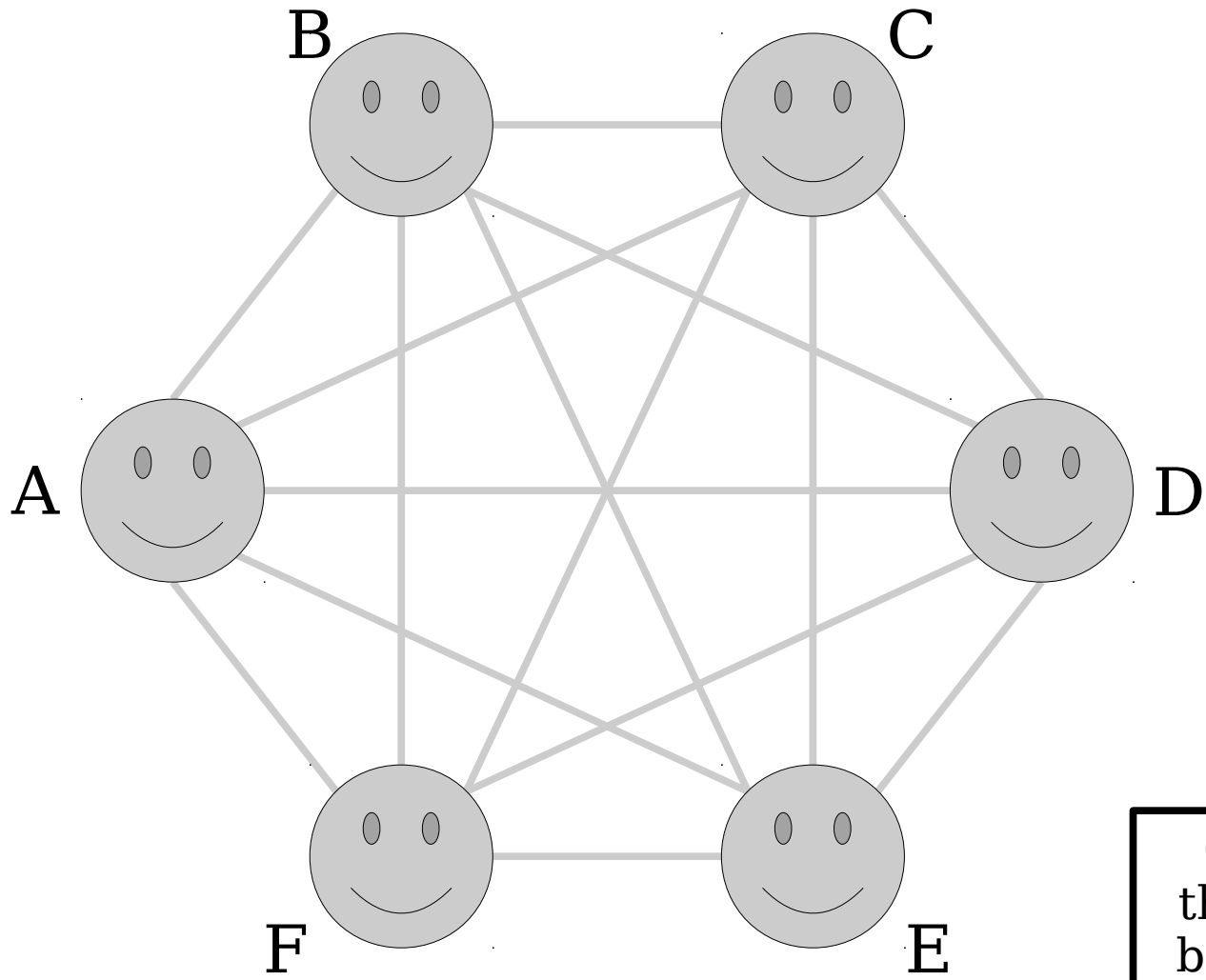




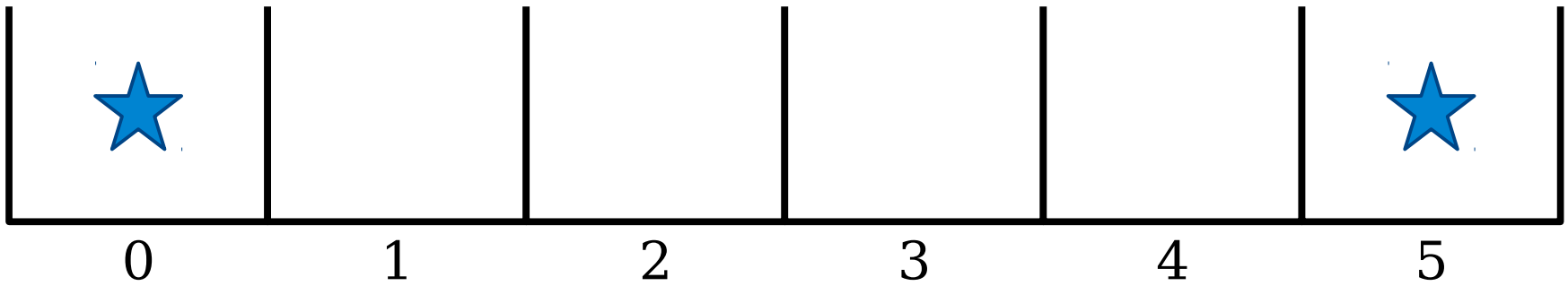
With n nodes, there are n possible degrees
(0, 1, 2, ..., $n - 1$)

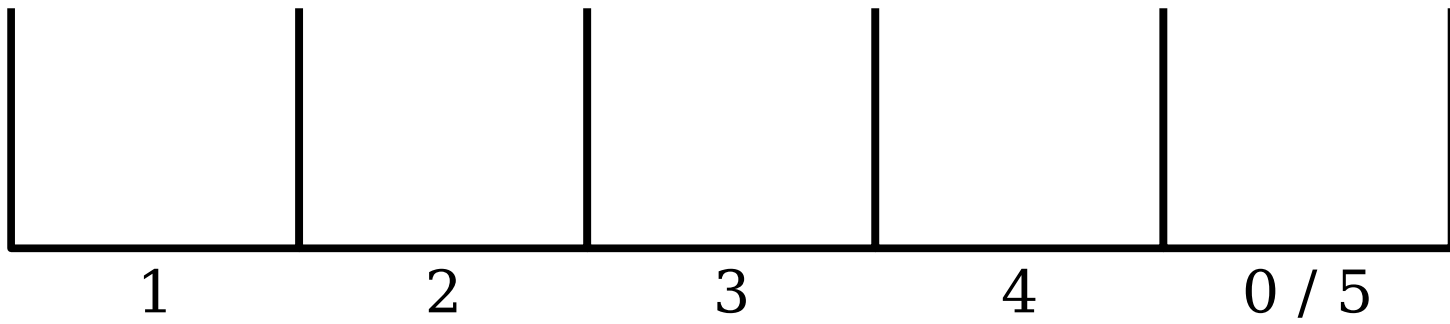
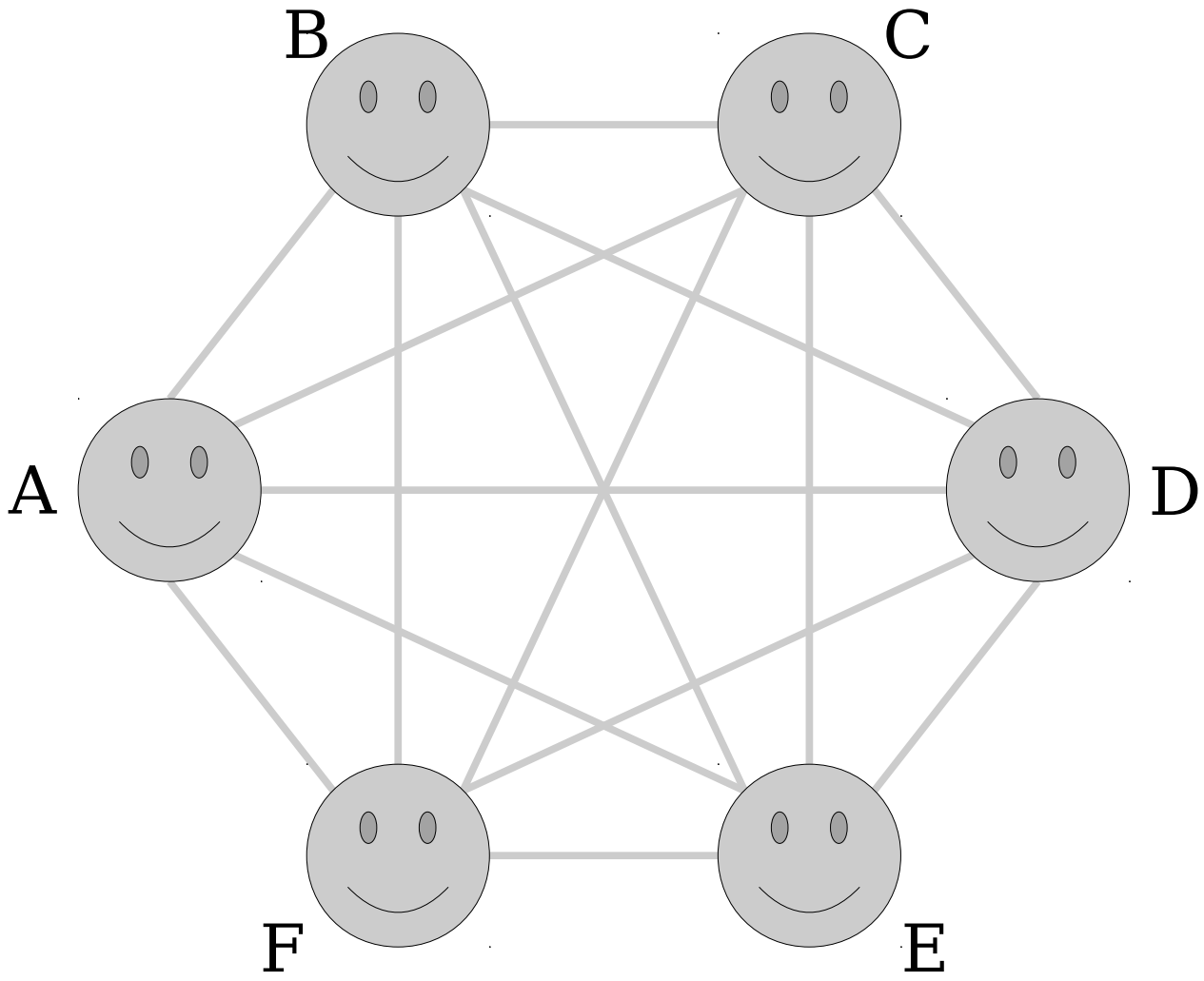






Can both of these buckets be nonempty?





Theorem: In any graph with at least two nodes, there are at least two nodes of the same degree.

Theorem: In any graph with at least two nodes, there are at least two nodes of the same degree.

Proof 1:

Theorem: In any graph with at least two nodes, there are at least two nodes of the same degree.

Proof 1: Let G be a graph with $n \geq 2$ nodes.

Theorem: In any graph with at least two nodes, there are at least two nodes of the same degree.

Proof 1: Let G be a graph with $n \geq 2$ nodes. There are n possible choices for the degrees of nodes in G , namely, $0, 1, 2, \dots$, and $n - 1$.

Theorem: In any graph with at least two nodes, there are at least two nodes of the same degree.

Proof 1: Let G be a graph with $n \geq 2$ nodes. There are n possible choices for the degrees of nodes in G , namely, $0, 1, 2, \dots,$ and $n - 1$.

We claim that G cannot simultaneously have a node u of degree 0 and a node v of degree $n - 1$:

Theorem: In any graph with at least two nodes, there are at least two nodes of the same degree.

Proof 1: Let G be a graph with $n \geq 2$ nodes. There are n possible choices for the degrees of nodes in G , namely, $0, 1, 2, \dots, \text{ and } n - 1$.

We claim that G cannot simultaneously have a node u of degree 0 and a node v of degree $n - 1$: if there were such nodes, then node u would be adjacent to no other nodes and node v would be adjacent to all other nodes, including u . (Note that u and v must be different nodes, since v has degree at least 1 and u has degree 0 .)

Theorem: In any graph with at least two nodes, there are at least two nodes of the same degree.

Proof 1: Let G be a graph with $n \geq 2$ nodes. There are n possible choices for the degrees of nodes in G , namely, $0, 1, 2, \dots, \text{ and } n - 1$.

We claim that G cannot simultaneously have a node u of degree 0 and a node v of degree $n - 1$: if there were such nodes, then node u would be adjacent to no other nodes and node v would be adjacent to all other nodes, including u . (Note that u and v must be different nodes, since v has degree at least 1 and u has degree 0 .)

We therefore see that the possible options for degrees of nodes in G are either drawn from $0, 1, \dots, n - 2$ or from $1, 2, \dots, n - 1$.

Theorem: In any graph with at least two nodes, there are at least two nodes of the same degree.

Proof 1: Let G be a graph with $n \geq 2$ nodes. There are n possible choices for the degrees of nodes in G , namely, $0, 1, 2, \dots, \text{ and } n - 1$.

We claim that G cannot simultaneously have a node u of degree 0 and a node v of degree $n - 1$: if there were such nodes, then node u would be adjacent to no other nodes and node v would be adjacent to all other nodes, including u . (Note that u and v must be different nodes, since v has degree at least 1 and u has degree 0 .)

We therefore see that the possible options for degrees of nodes in G are either drawn from $0, 1, \dots, n - 2$ or from $1, 2, \dots, n - 1$. In either case, there are n nodes and $n - 1$ possible degrees, so by the pigeonhole principle two nodes in G must have the same degree.

Theorem: In any graph with at least two nodes, there are at least two nodes of the same degree.

Proof 1: Let G be a graph with $n \geq 2$ nodes. There are n possible choices for the degrees of nodes in G , namely, $0, 1, 2, \dots$, and $n - 1$.

We claim that G cannot simultaneously have a node u of degree 0 and a node v of degree $n - 1$: if there were such nodes, then node u would be adjacent to no other nodes and node v would be adjacent to all other nodes, including u . (Note that u and v must be different nodes, since v has degree at least 1 and u has degree 0.)

We therefore see that the possible options for degrees of nodes in G are either drawn from $0, 1, \dots, n - 2$ or from $1, 2, \dots, n - 1$. In either case, there are n nodes and $n - 1$ possible degrees, so by the pigeonhole principle two nodes in G must have the same degree. ■

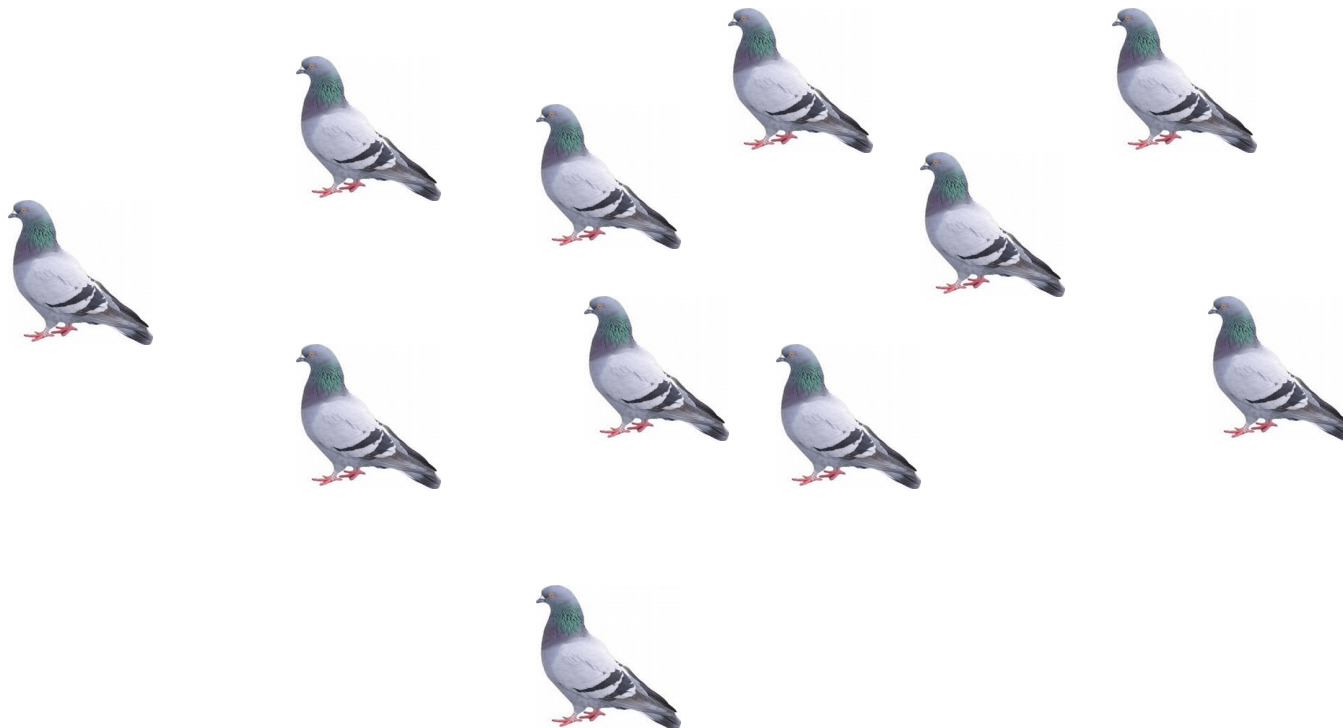
Theorem: In any graph with at least two nodes, there are at least two nodes of the same degree.

Proof 2: Assume for the sake of contradiction that there is a graph G with $n \geq 2$ nodes where no two nodes have the same degree. There are n possible choices for the degrees of nodes in G , namely $0, 1, 2, \dots, n - 1$, so this means that G must have exactly one node of each degree. However, this means that G has a node of degree 0 and a node of degree $n - 1$. (These can't be the same node, since $n \geq 2$.) This first node is adjacent to no other nodes, but this second node is adjacent to every other node, which is impossible.

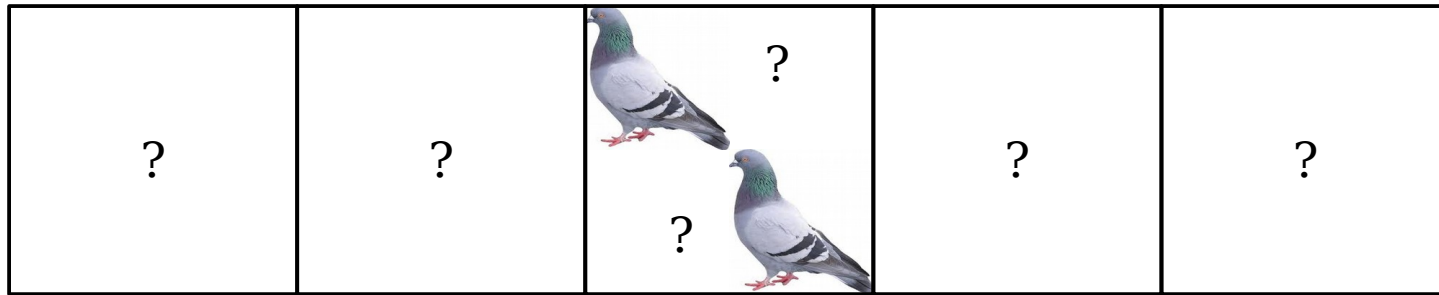
We have reached a contradiction, so our assumption must have been wrong. Thus if G is a graph with at least two nodes, G must have at least two nodes of the same degree. ■

The Generalized Pigeonhole Principle

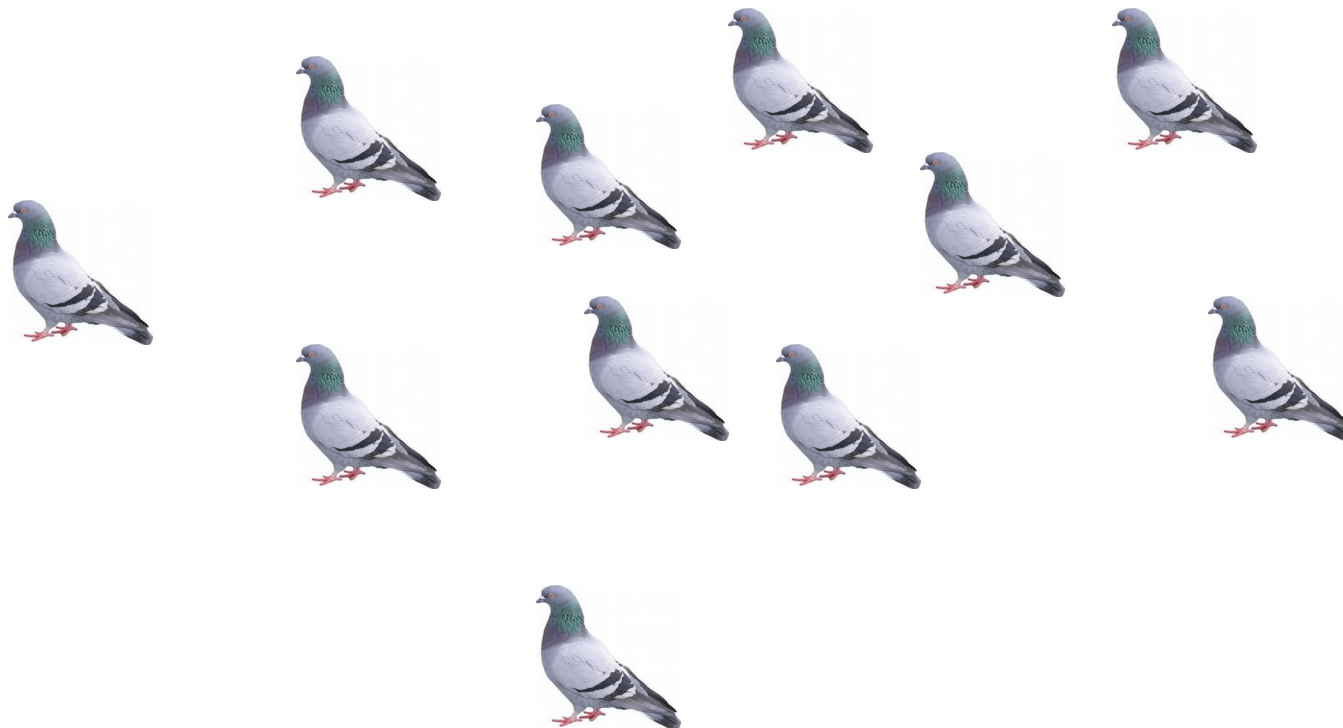
The Pigeonhole Principle



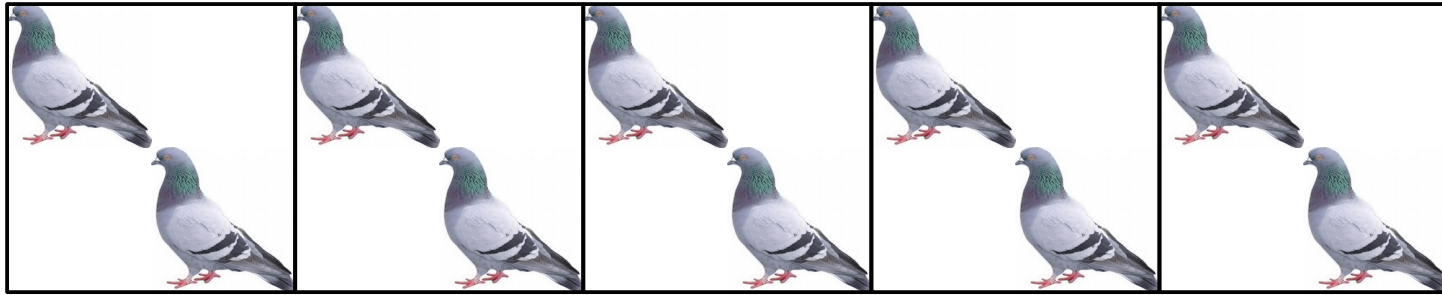
The Pigeonhole Principle



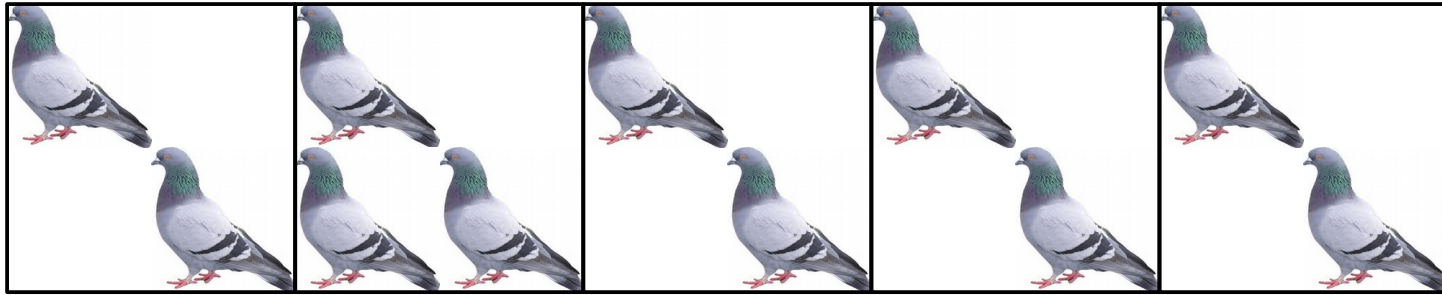
The Pigeonhole Principle



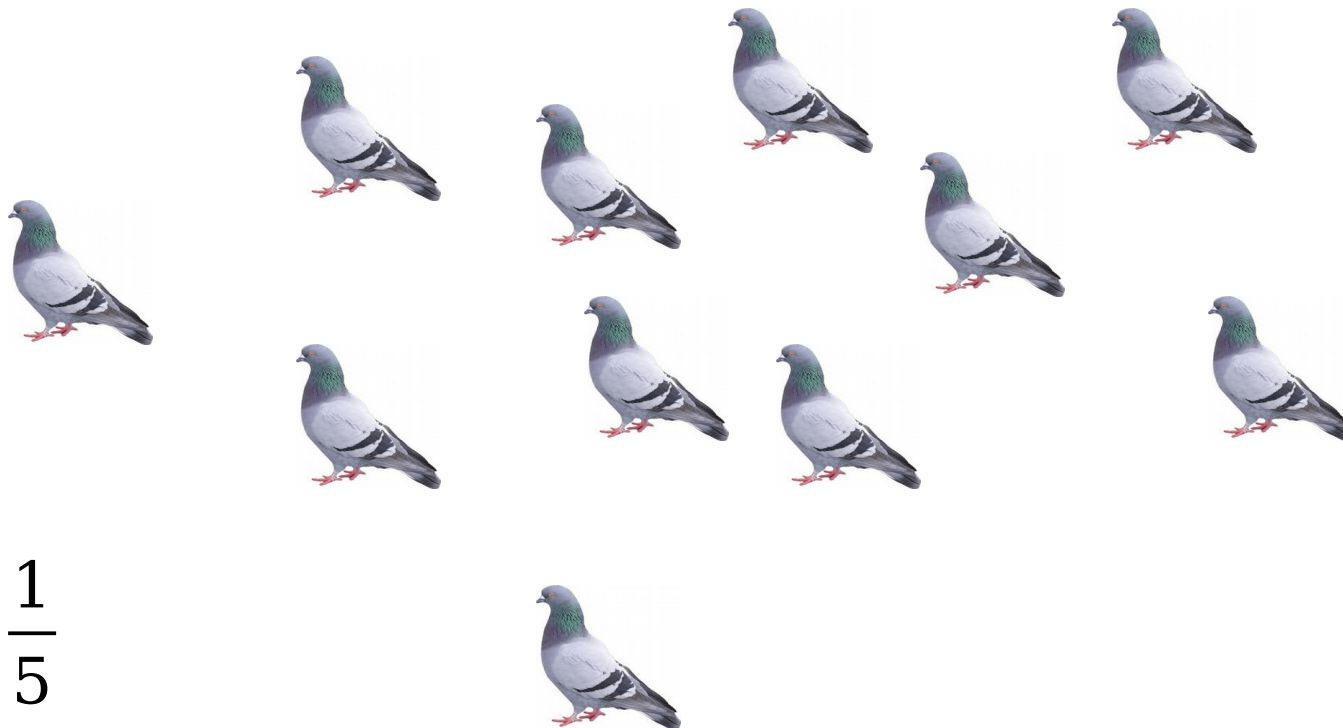
The Pigeonhole Principle



The Pigeonhole Principle



The Pigeonhole Principle



$$\frac{11}{5} = 2\frac{1}{5}$$

A More General Version

- The ***generalized pigeonhole principle*** says that if you distribute m objects into n bins, then
 - some bin will have at least $\lceil m/n \rceil$ objects in it, and
 - some bin will have at most $\lfloor m/n \rfloor$ objects in it.

$\lceil m/n \rceil$ means “ m/n , rounded up.”
 $\lfloor m/n \rfloor$ means “ m/n , rounded down.”

A More General Version

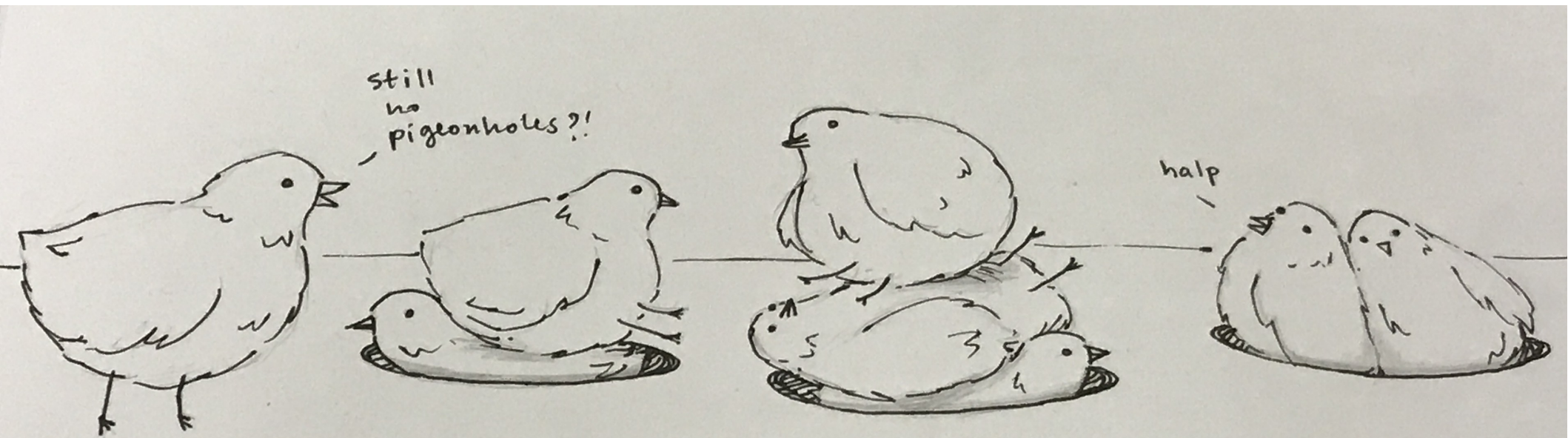
- The ***generalized pigeonhole principle*** says that if you distribute m objects into n bins, then
 - some bin will have at least $\lceil m/n \rceil$ objects in it, and
 - some bin will have at most $\lfloor m/n \rfloor$ objects in it.

$\lceil m/n \rceil$ means “ m/n , rounded up.”
 $\lfloor m/n \rfloor$ means “ m/n , rounded down.”

A More General Version

- The ***generalized pigeonhole principle*** says that if you distribute m objects into n bins, then
 - some bin will have at least $\lceil m/n \rceil$ objects in it, and
 - some bin will have at most $\lfloor m/n \rfloor$ objects in it.

$\lceil m/n \rceil$ means “ m/n , rounded up.”
 $\lfloor m/n \rfloor$ means “ m/n , rounded down.”



$$m = 8, n = 3$$

Thanks to Amy Liu for this awesome drawing!

Theorem: If m objects are distributed into $n > 0$ bins, then some bin will contain at least $\lceil m/n \rceil$ objects.

Proof: We will prove that if m objects are distributed into n bins, then some bin contains at least m/n objects. Since the number of objects in each bin is an integer, this will prove that some bin must contain at least $\lceil m/n \rceil$ objects.

To do this, we proceed by contradiction. Suppose that, for some m and n , there is a way to distribute m objects into n bins such that each bin contains fewer than m/n objects.

Number the bins $1, 2, 3, \dots, n$ and let x_i denote the number of objects in bin i . Since there are m objects in total, we know that

$$m = x_1 + x_2 + \dots + x_n.$$

Since each bin contains fewer than m/n objects, we see that $x_i < m/n$ for each i . Therefore, we have that

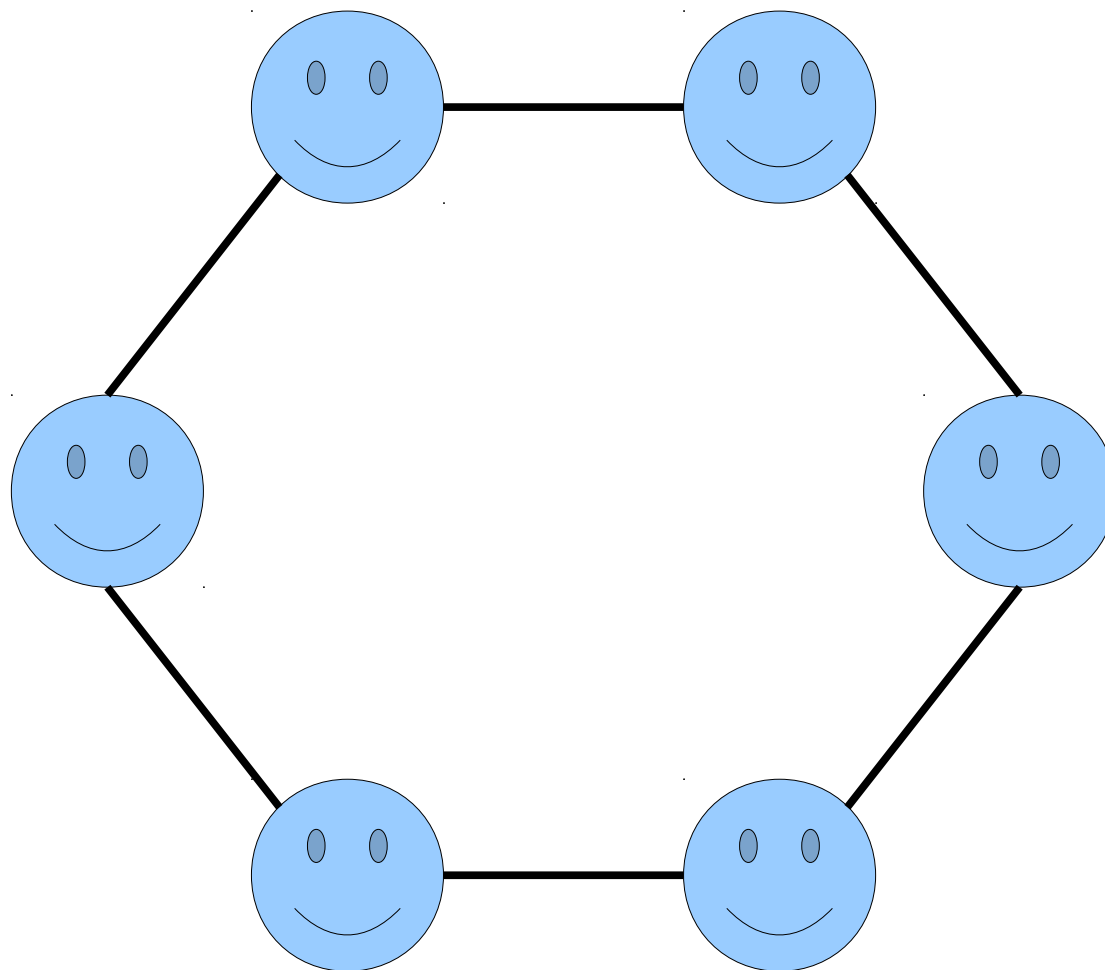
$$\begin{aligned} m &= x_1 + x_2 + \dots + x_n \\ &< m/n + m/n + \dots + m/n \quad (n \text{ times}) \\ &= m. \end{aligned}$$

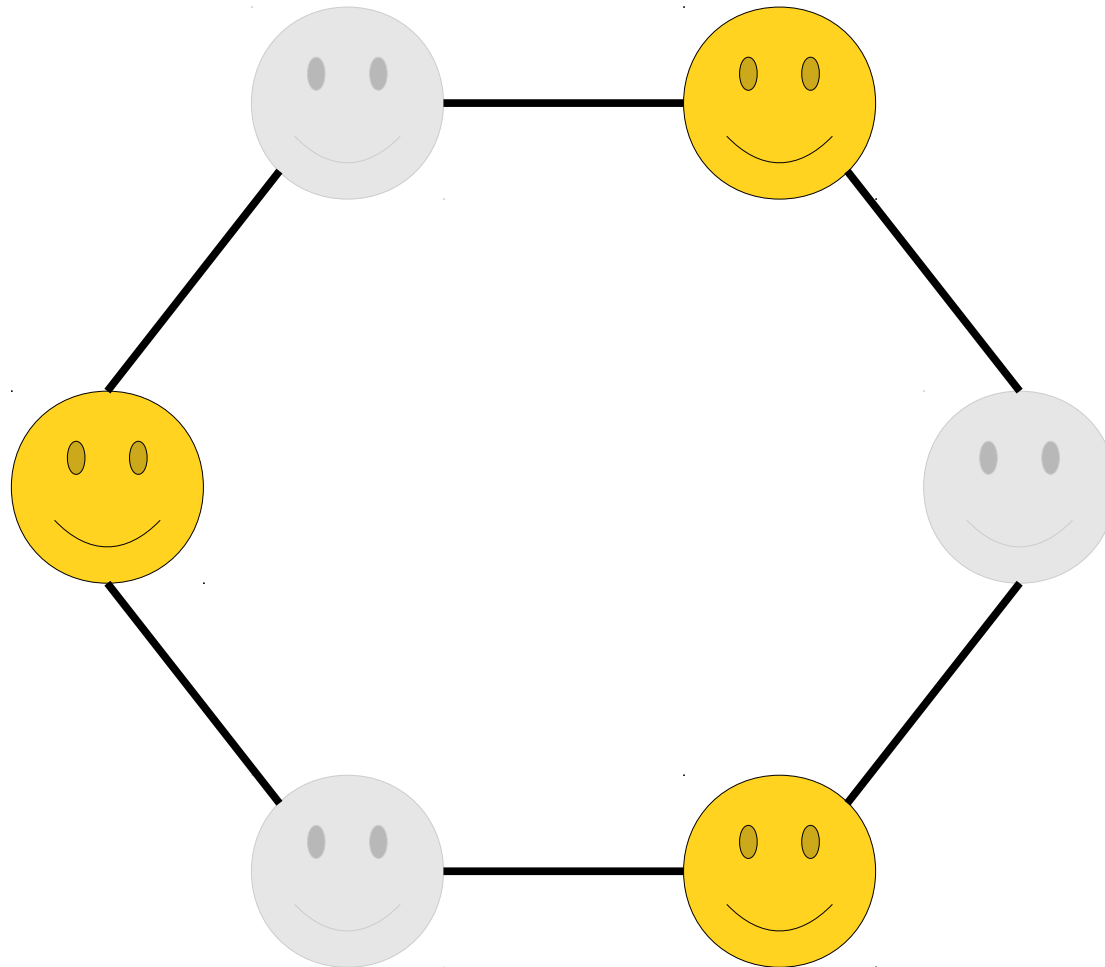
But this means that $m < m$, which is impossible. We have reached a contradiction, so our initial assumption must have been wrong. Therefore, if m objects are distributed into n bins, some bin must contain at least $\lceil m/n \rceil$ objects. ■

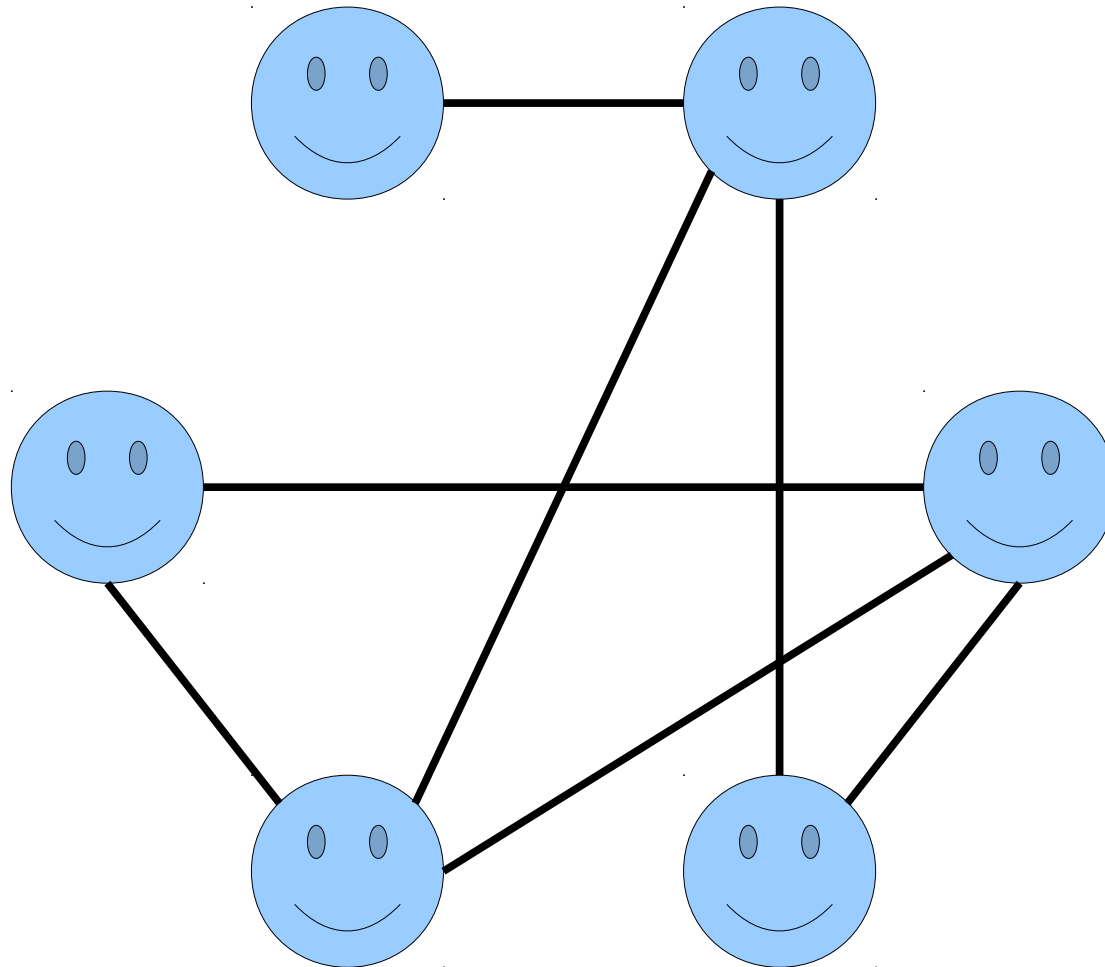
An Application: Friends and Strangers

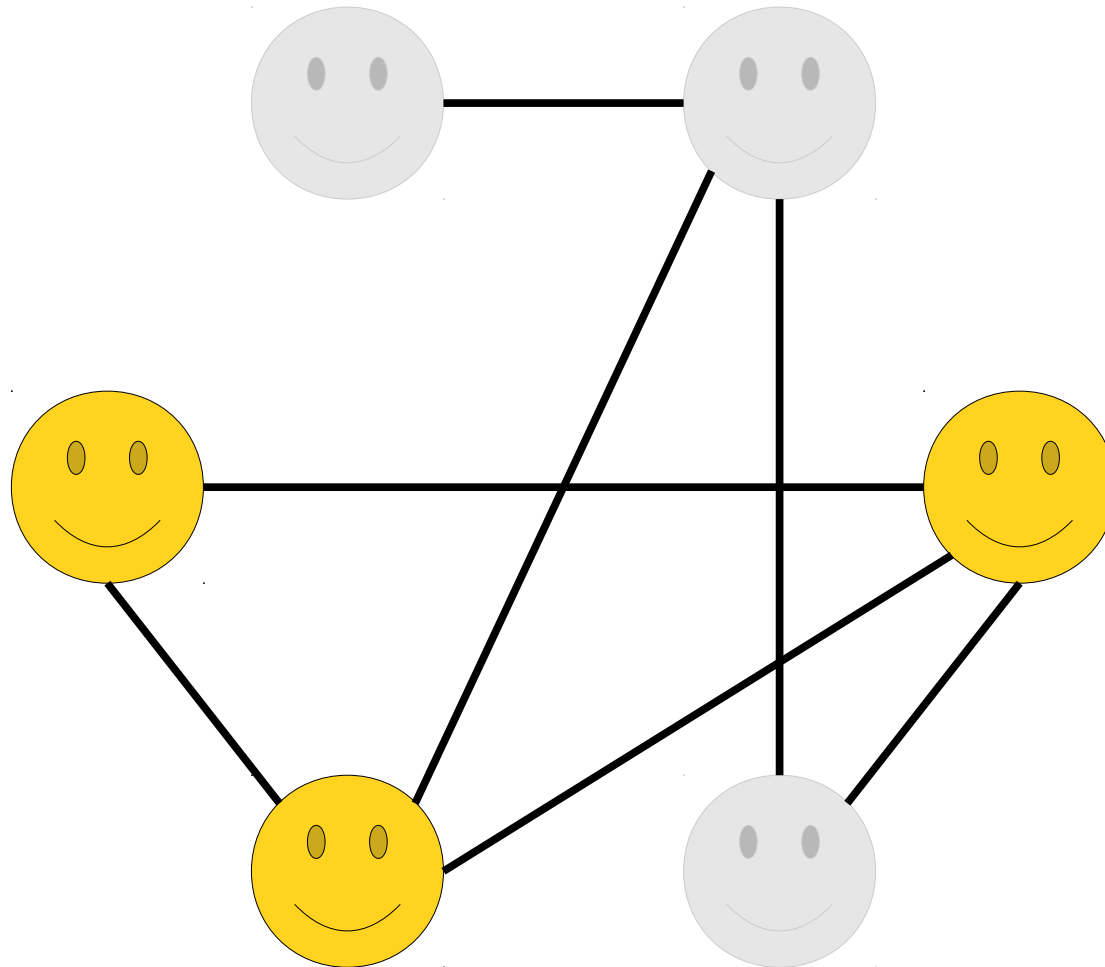
Friends and Strangers

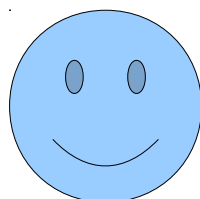
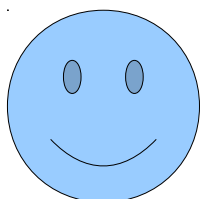
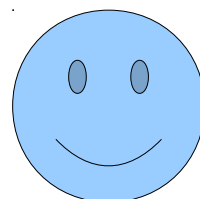
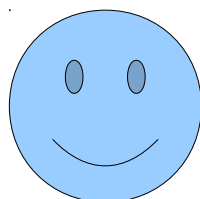
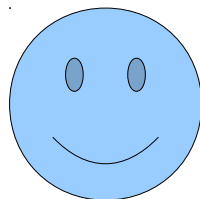
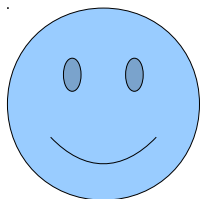
- Suppose you have a party of six people. Each pair of people are either friends (they know each other) or strangers (they do not).
- ***Theorem:*** Any such party must have a group of three mutual friends (three people who all know one another) or three mutual strangers (three people, none of whom know any of the others).

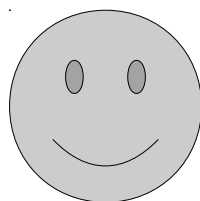
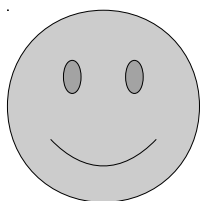
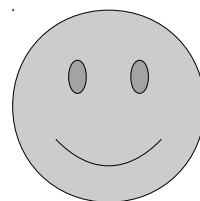
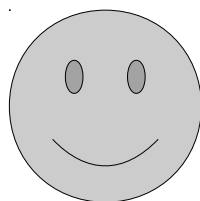
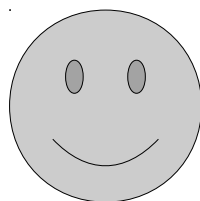
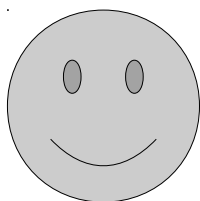


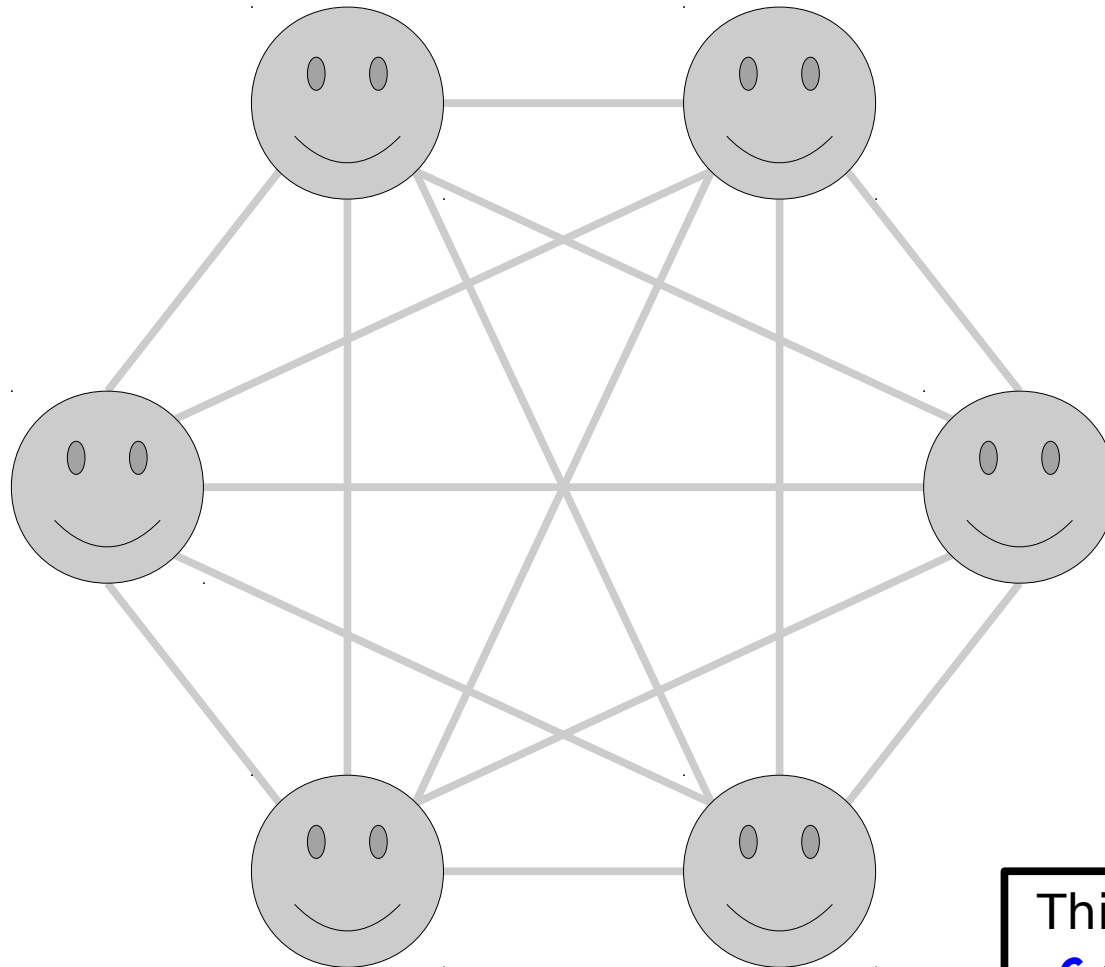




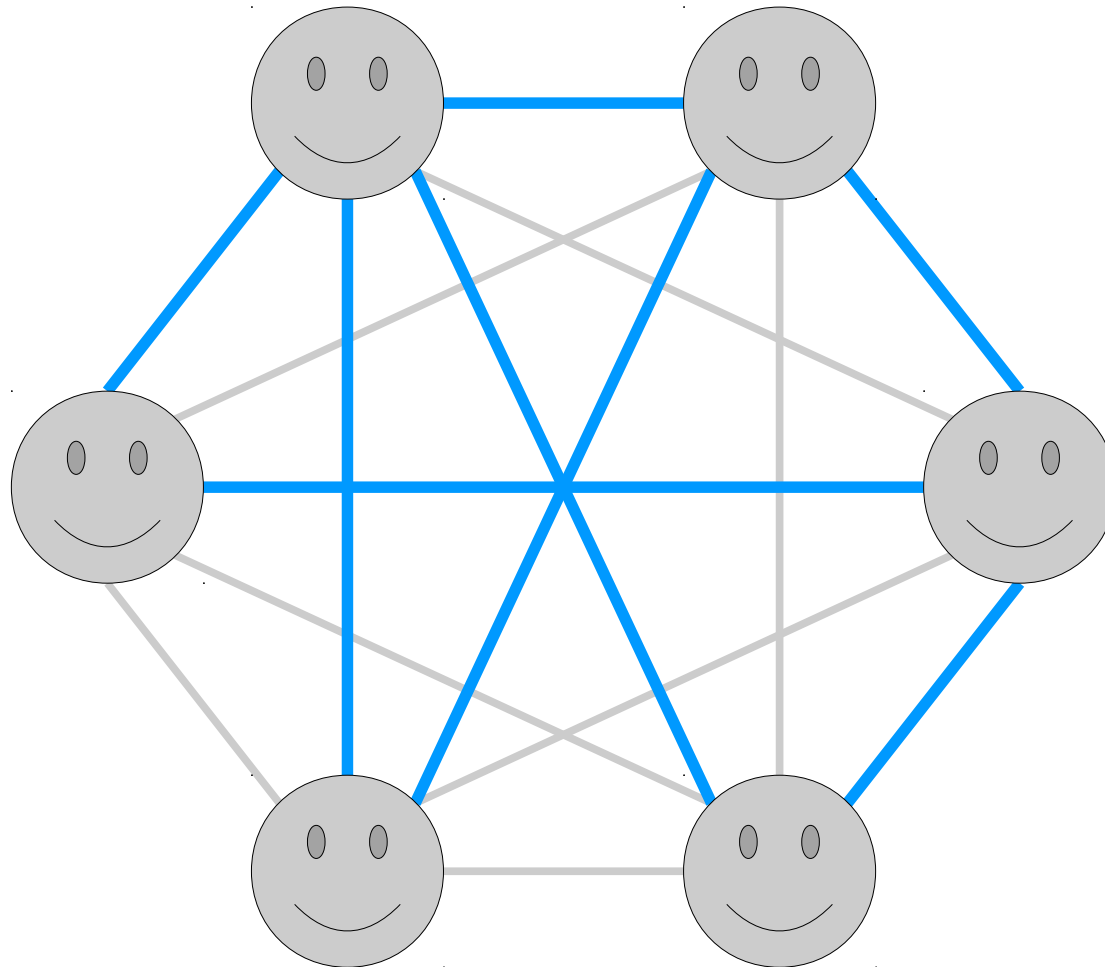


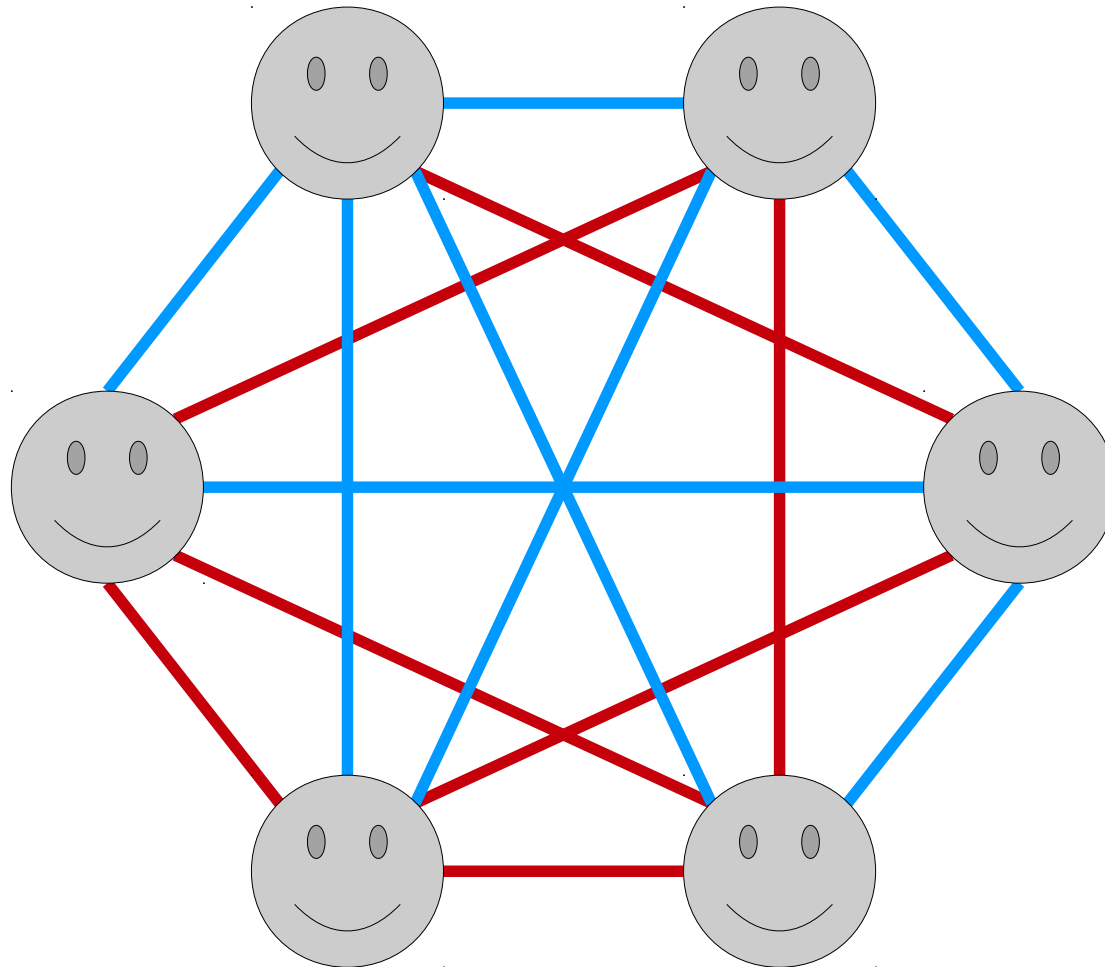


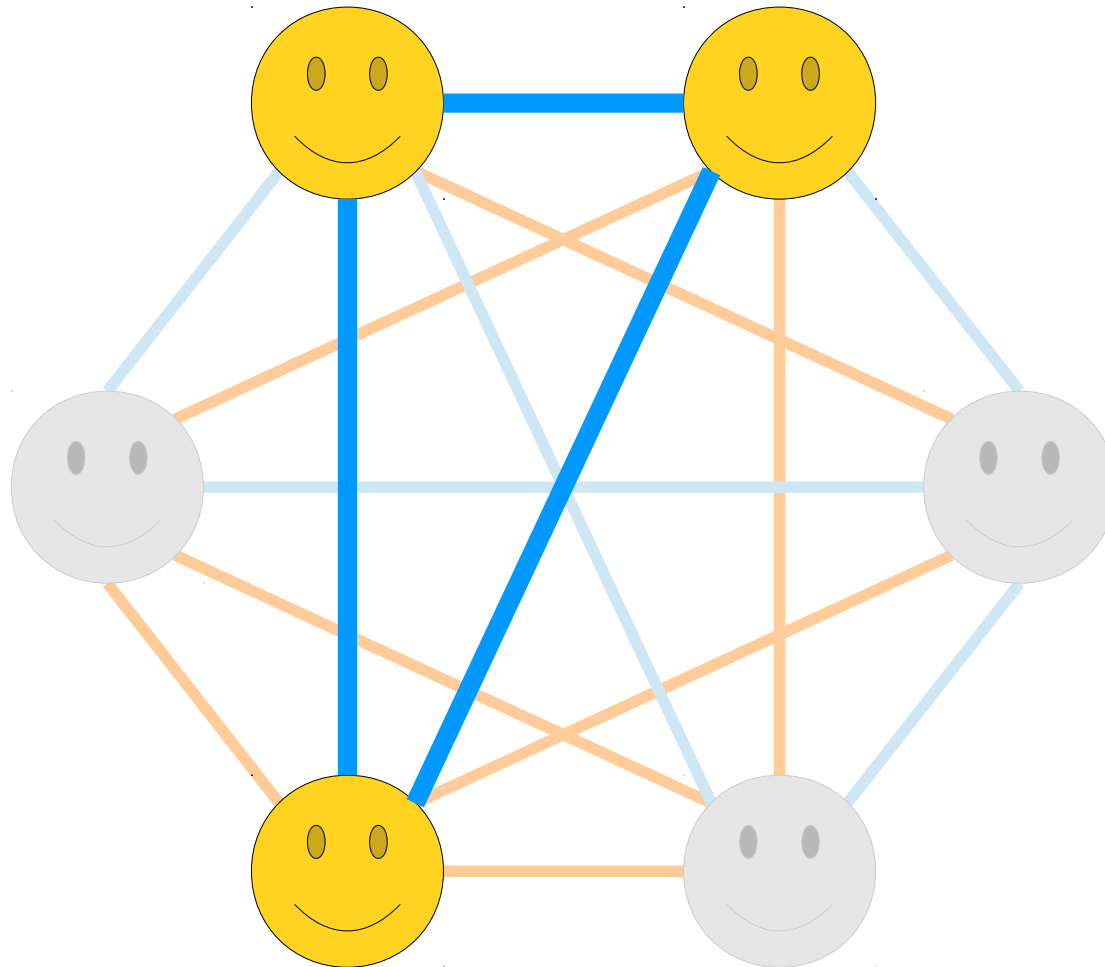


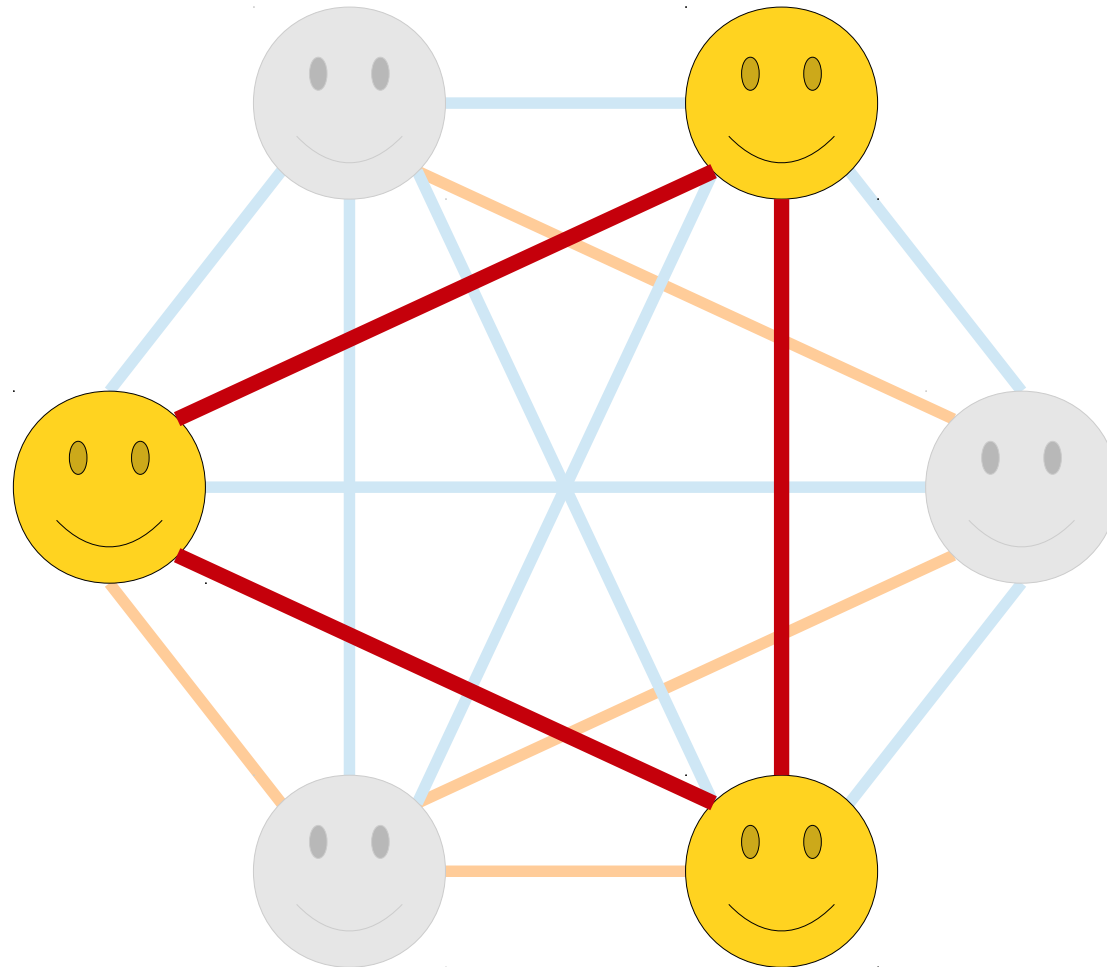


This graph is called a **6-clique**, by the way.







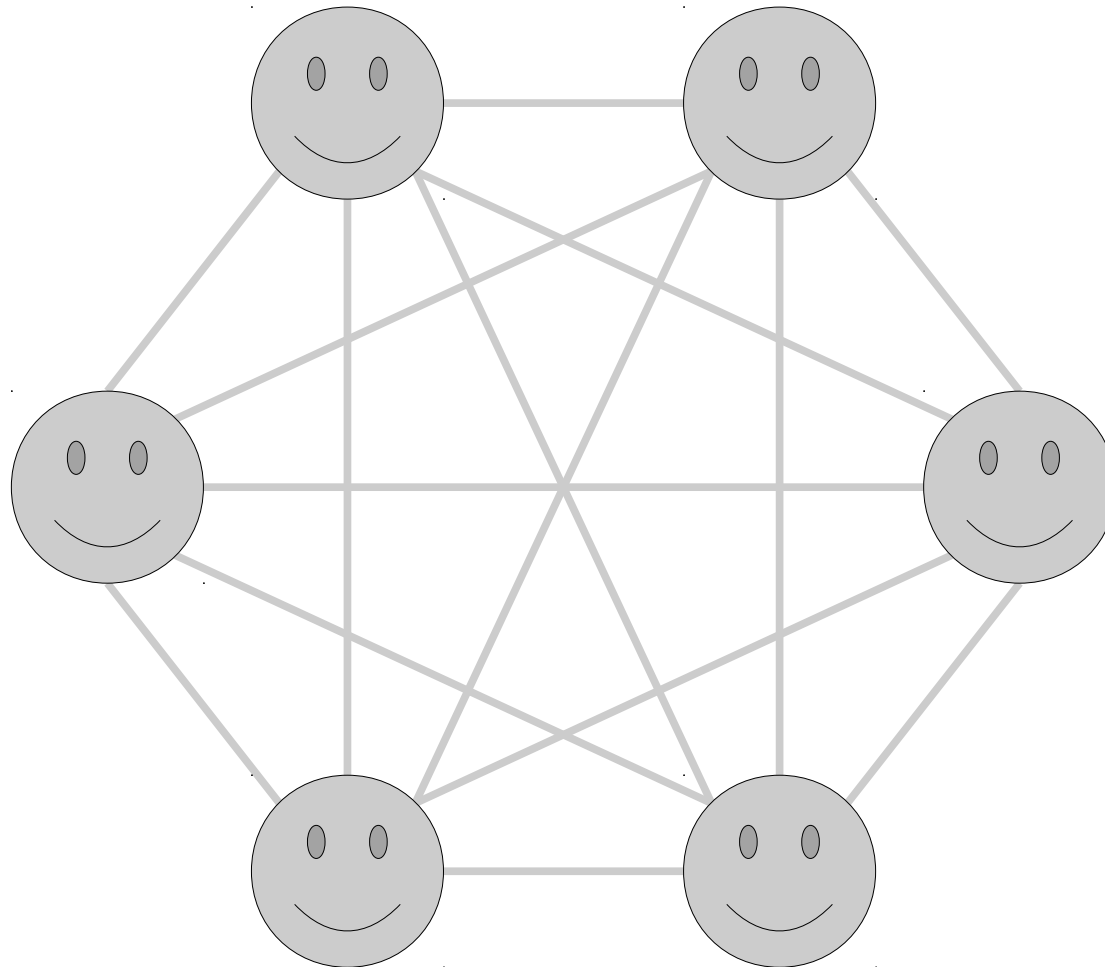


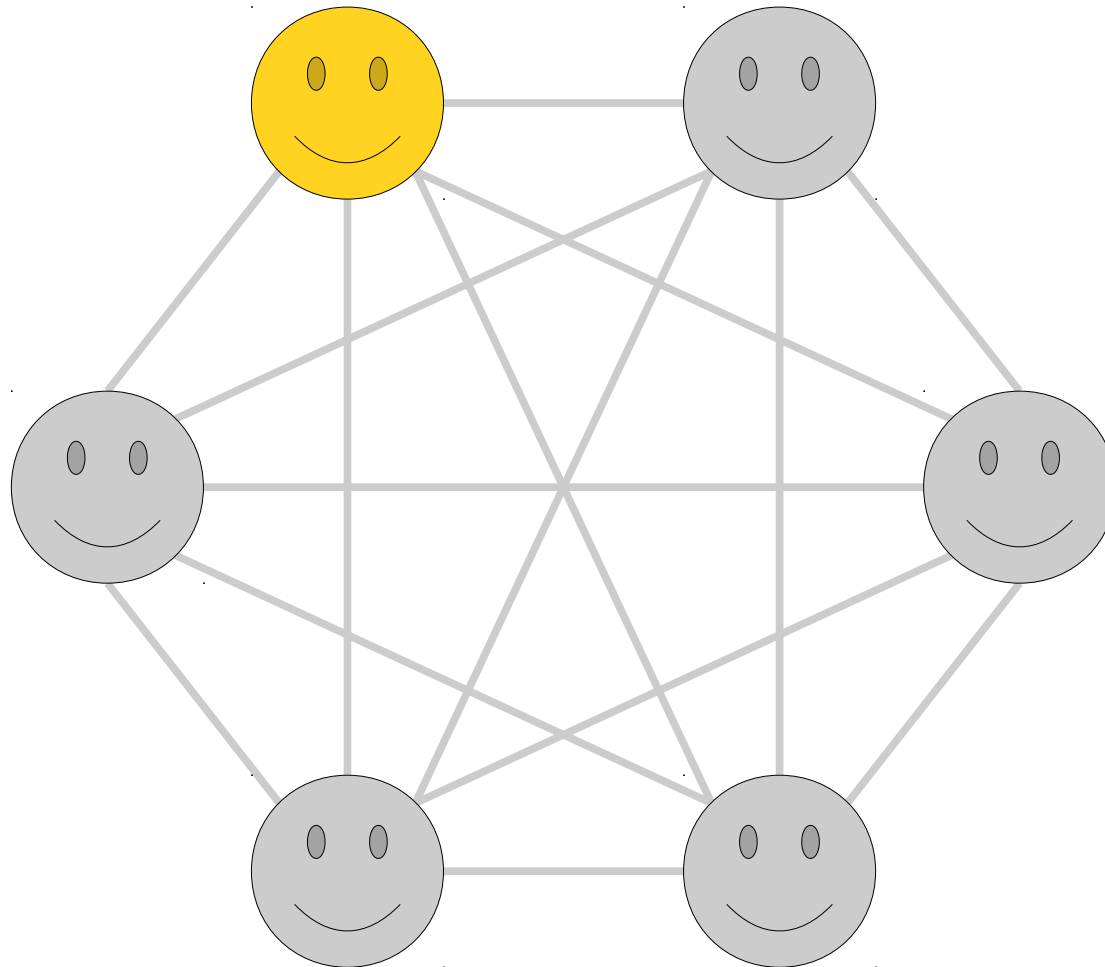
Friends and Strangers Restated

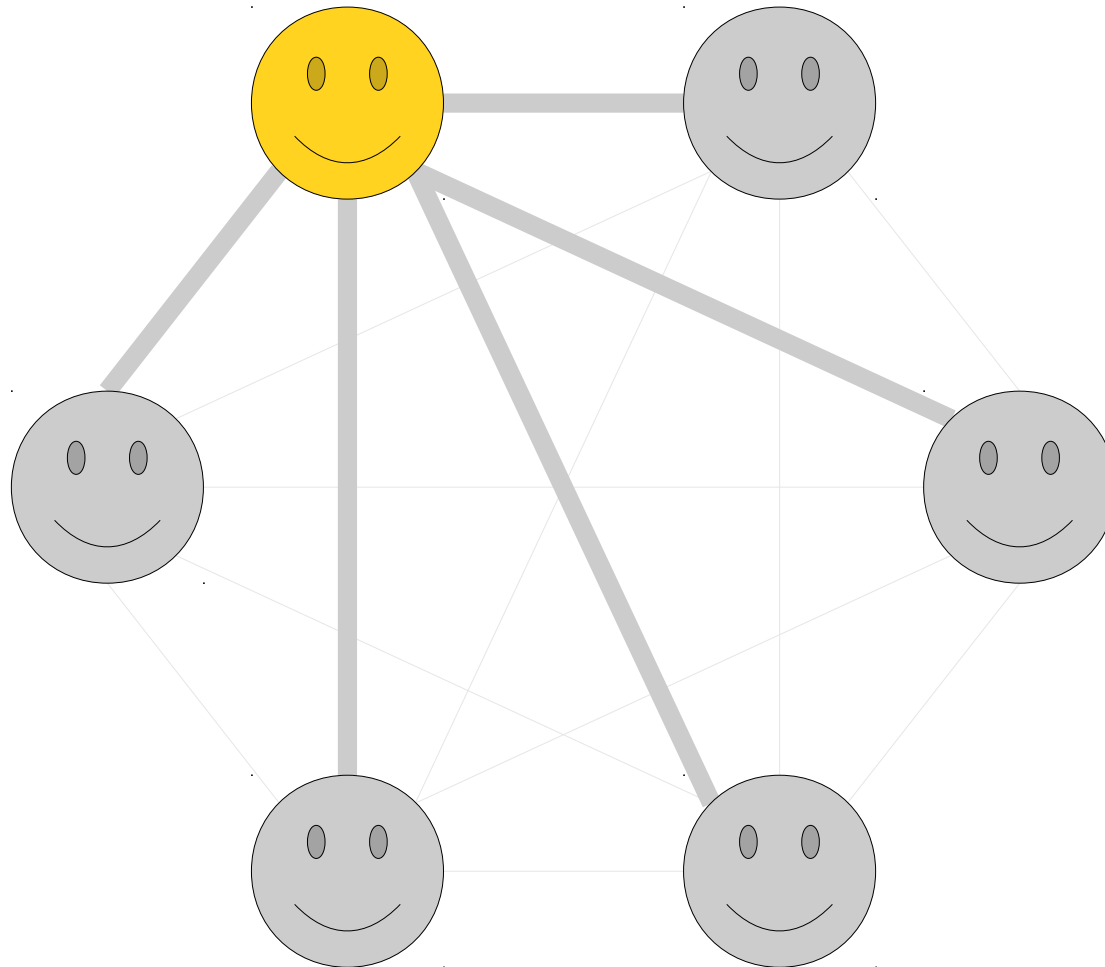
- From a graph-theoretic perspective, the Theorem on Friends and Strangers can be restated as follows:

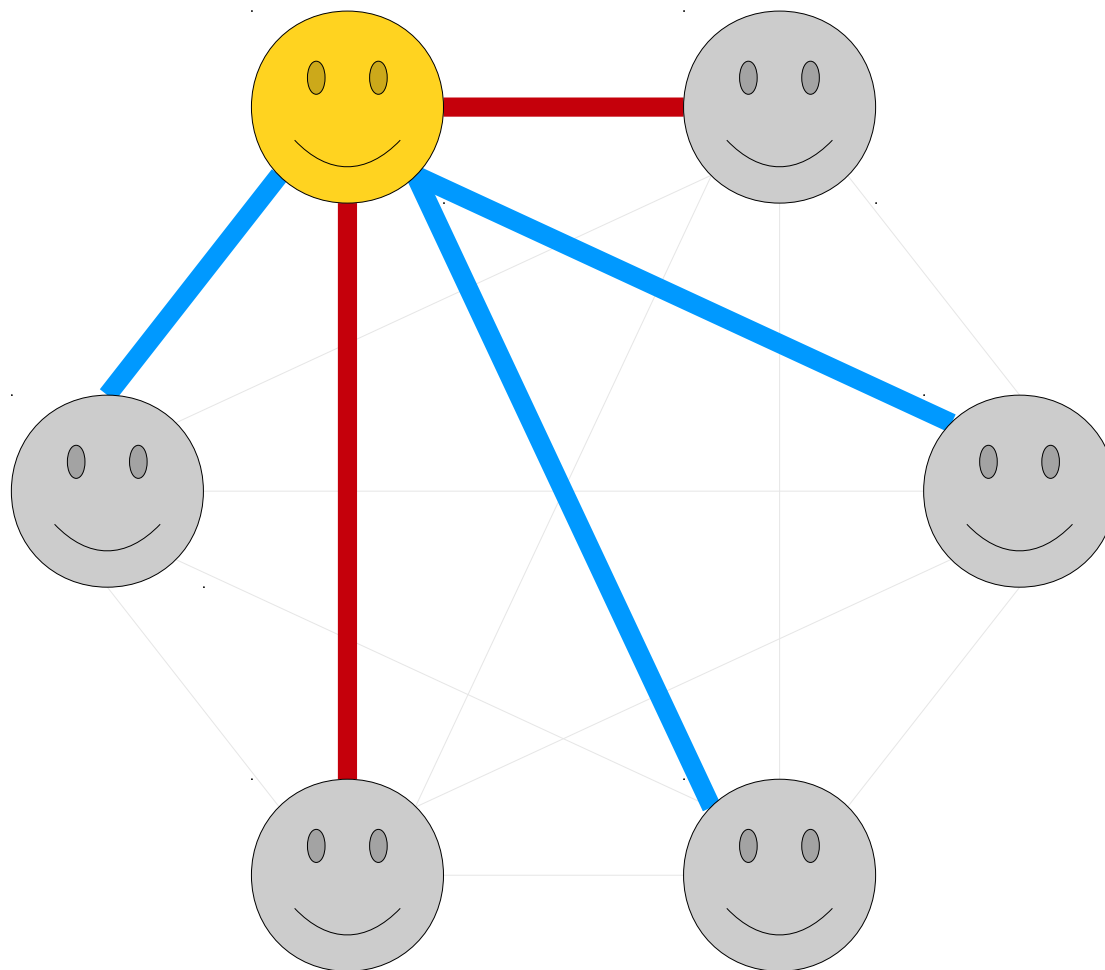
Theorem: Any 6-clique whose edges are colored red and blue contains a red triangle or a blue triangle (or both).

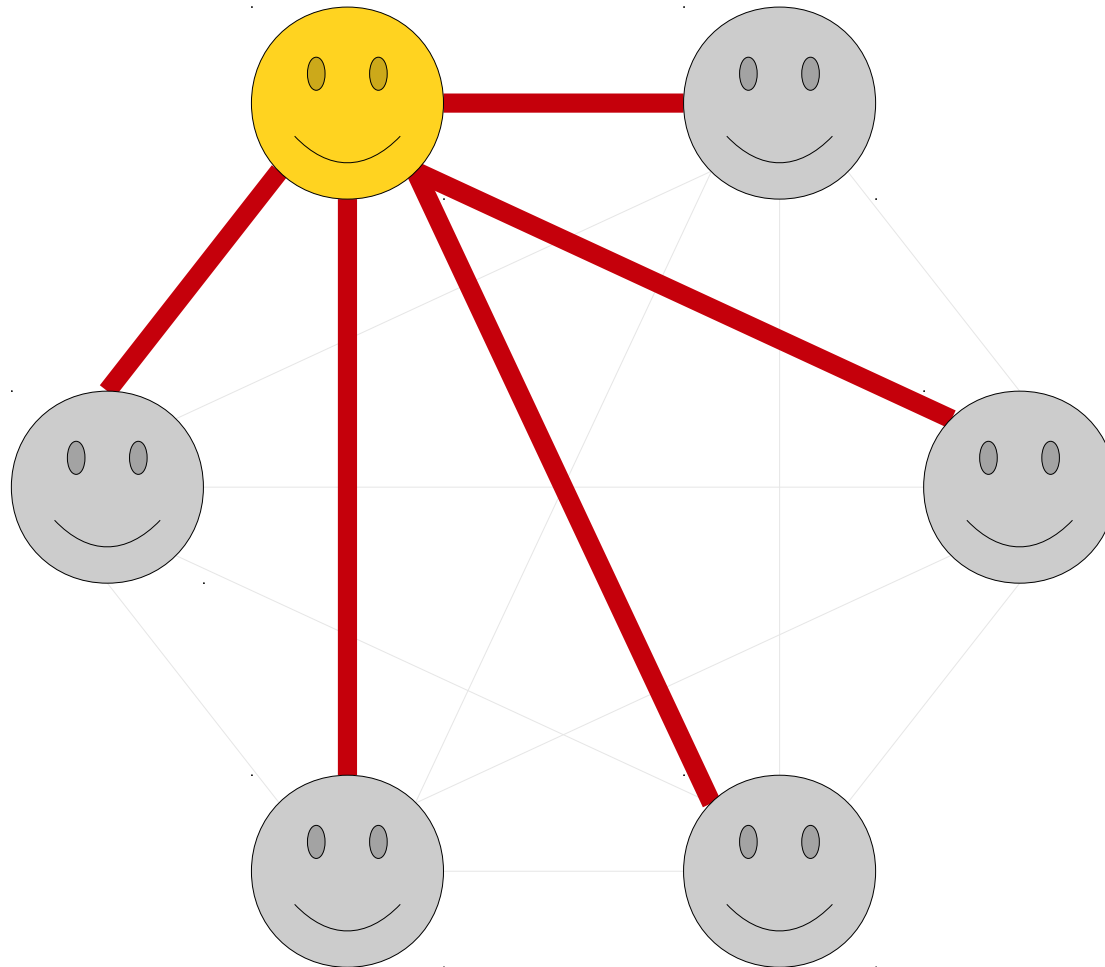
- How can we prove this?

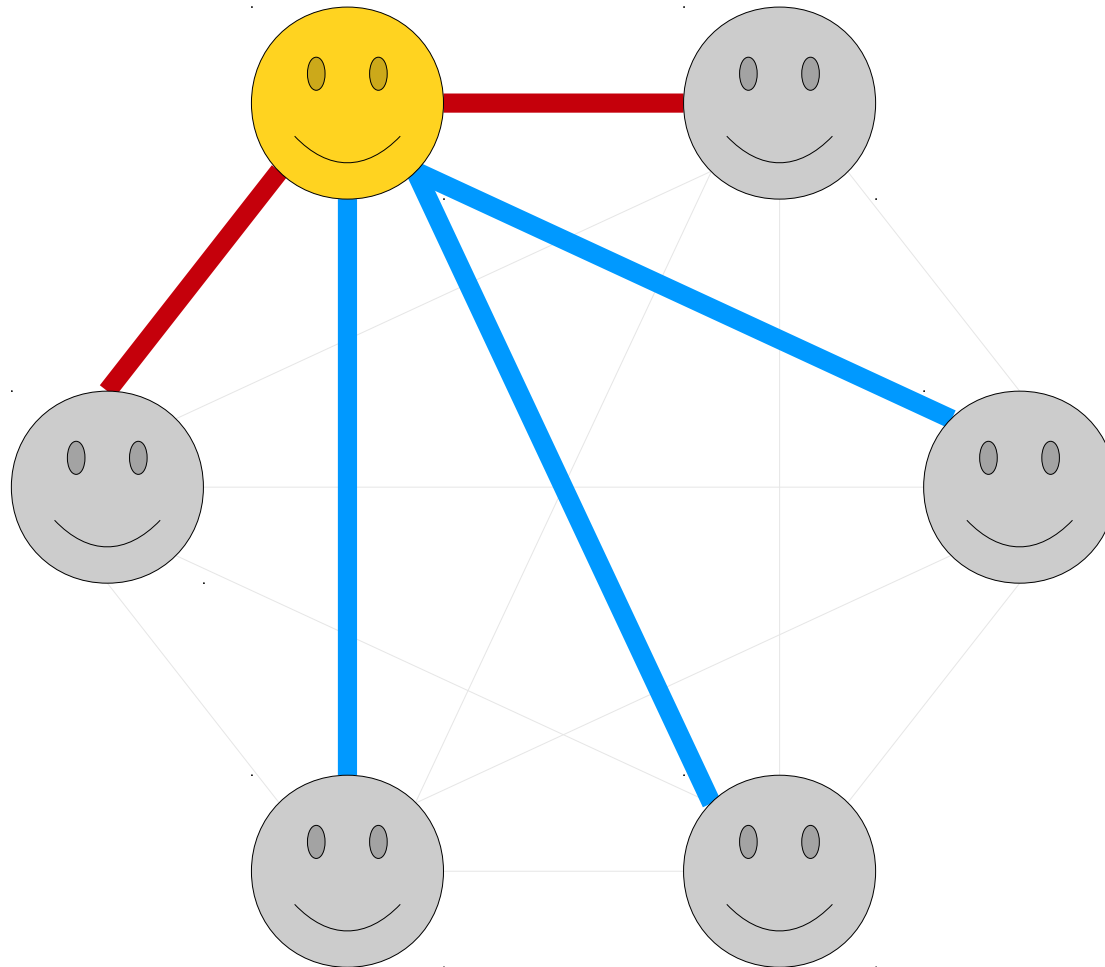


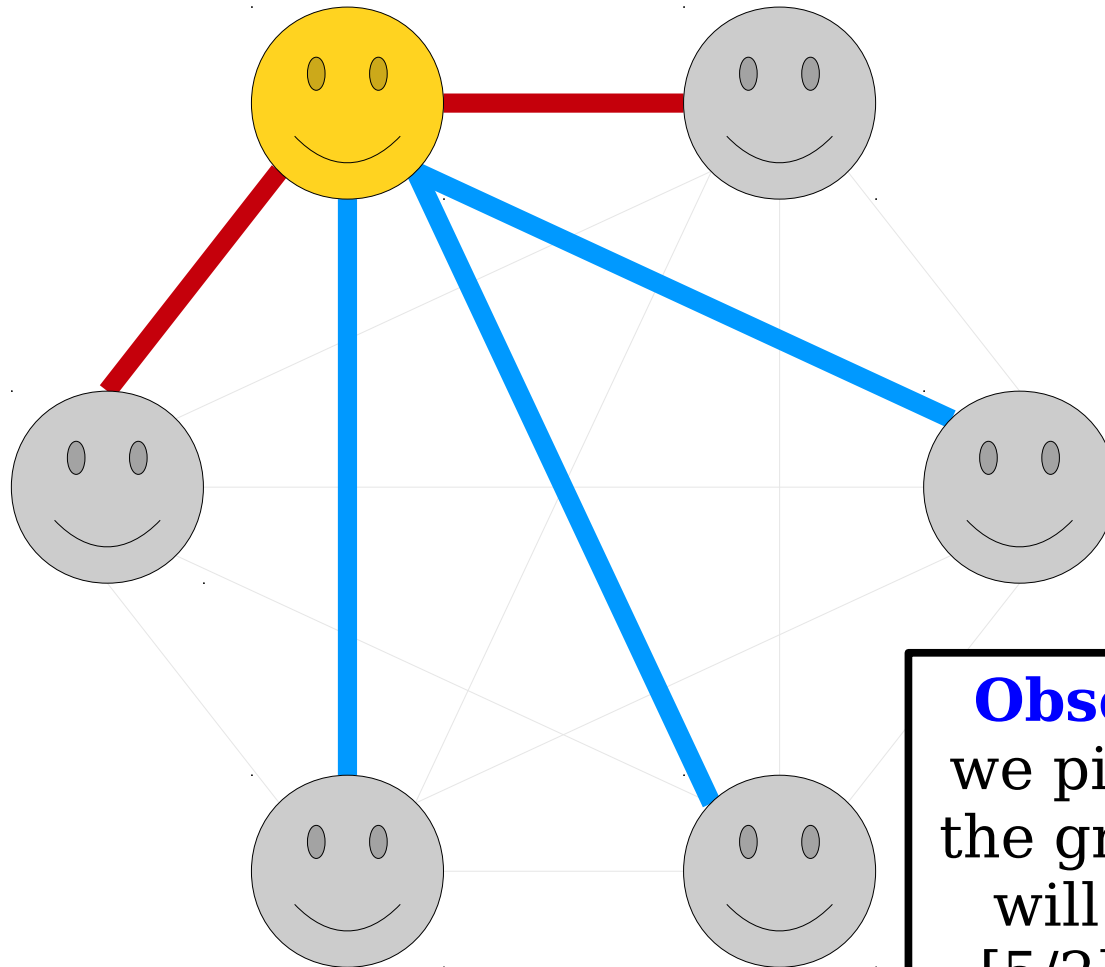




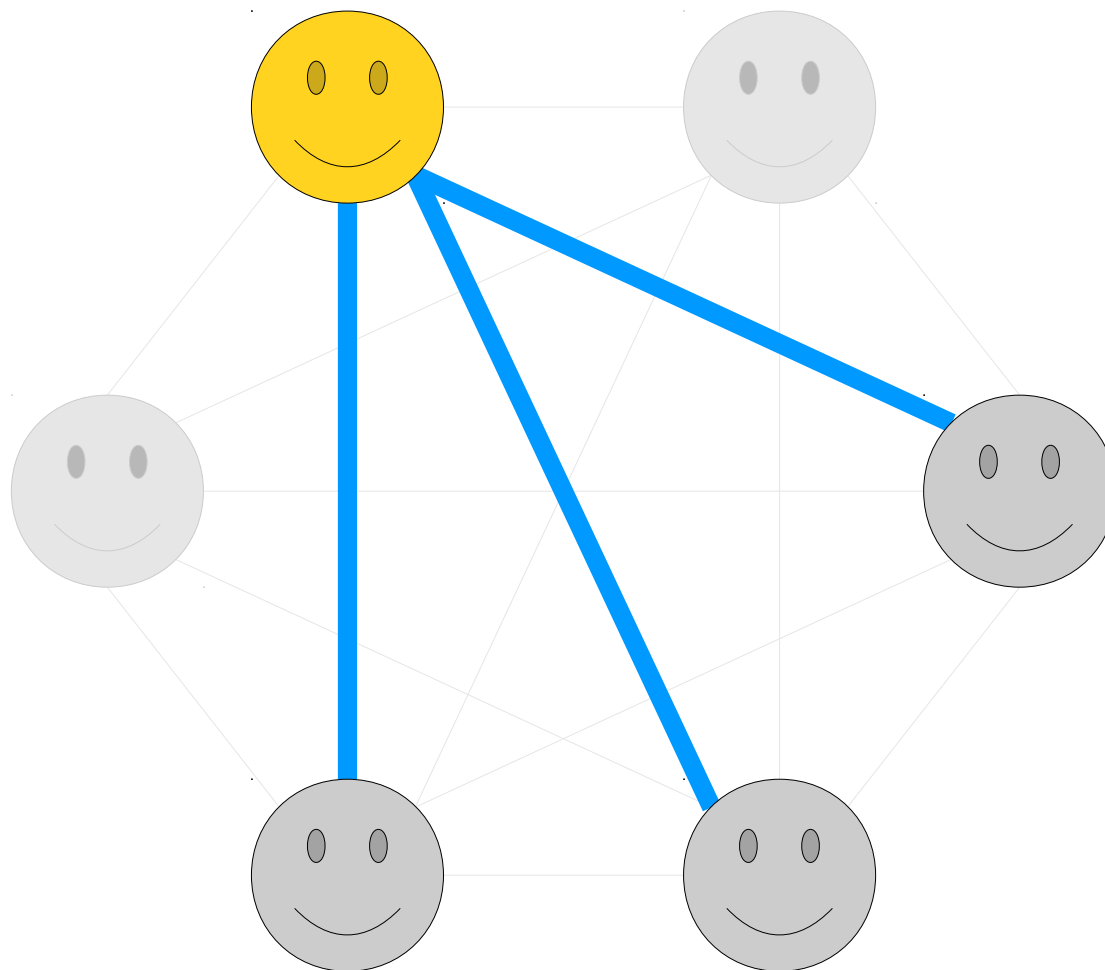


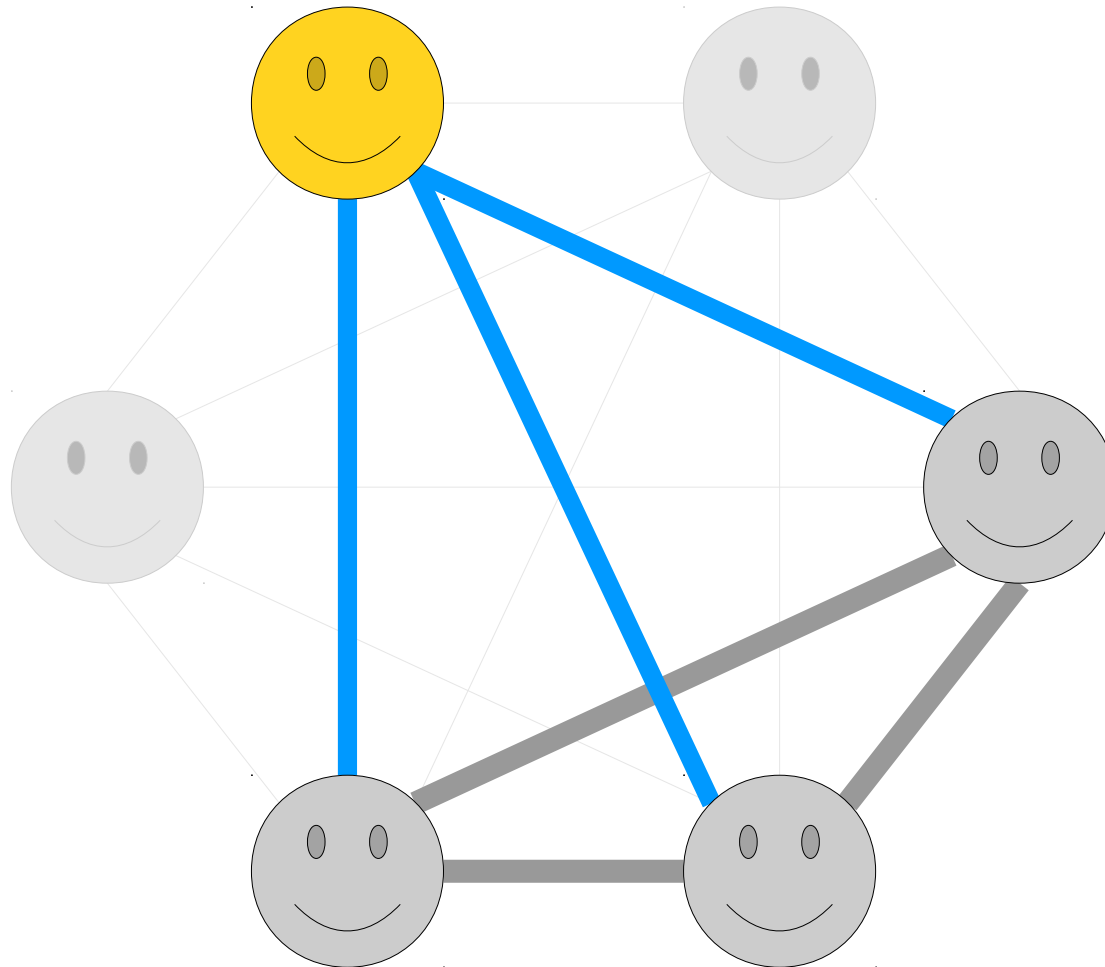


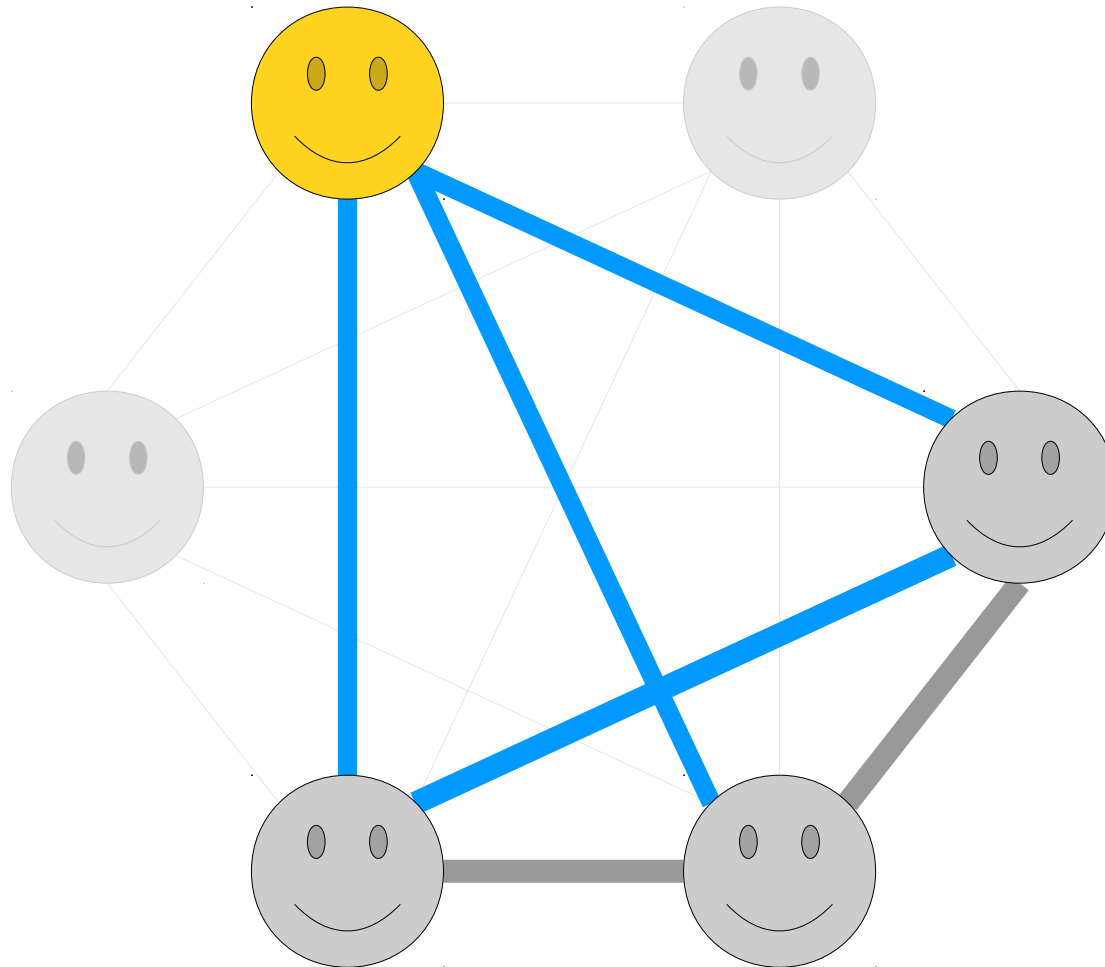


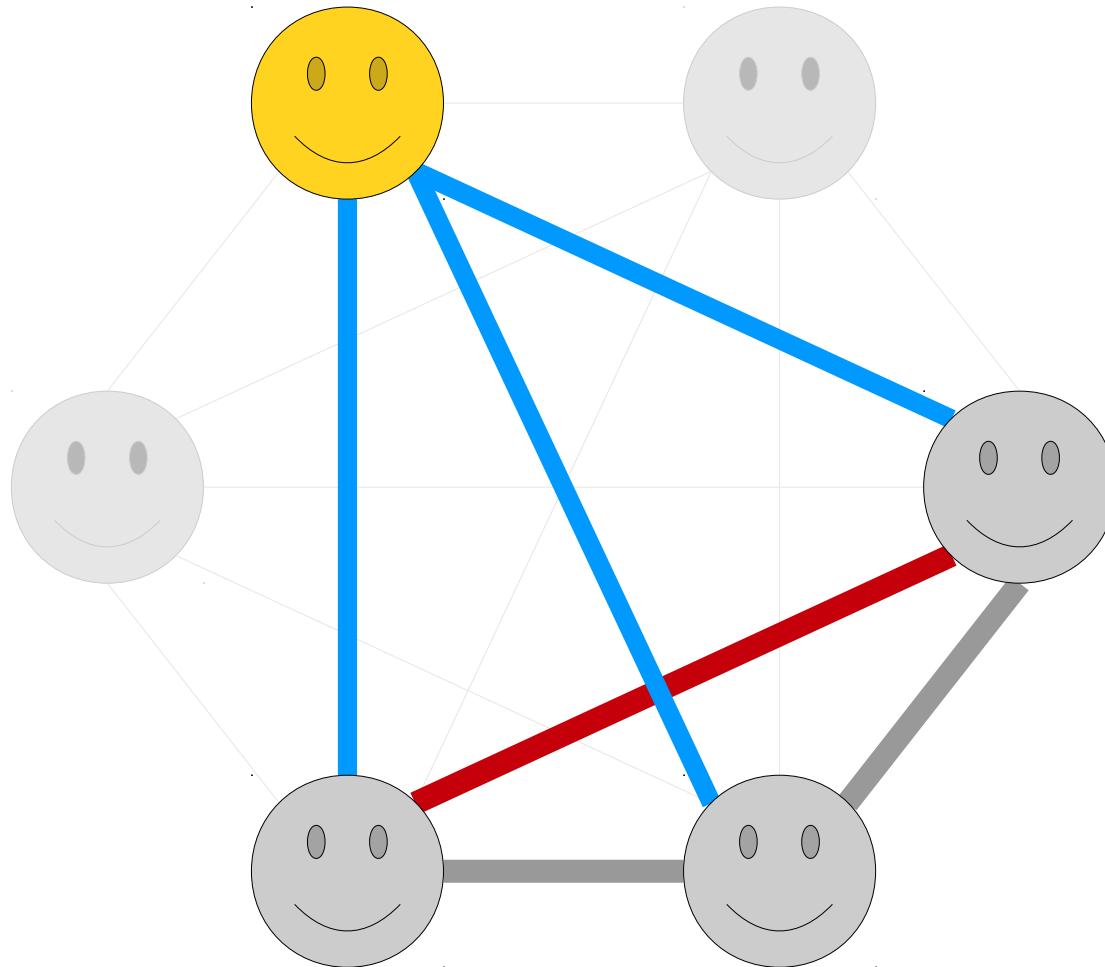


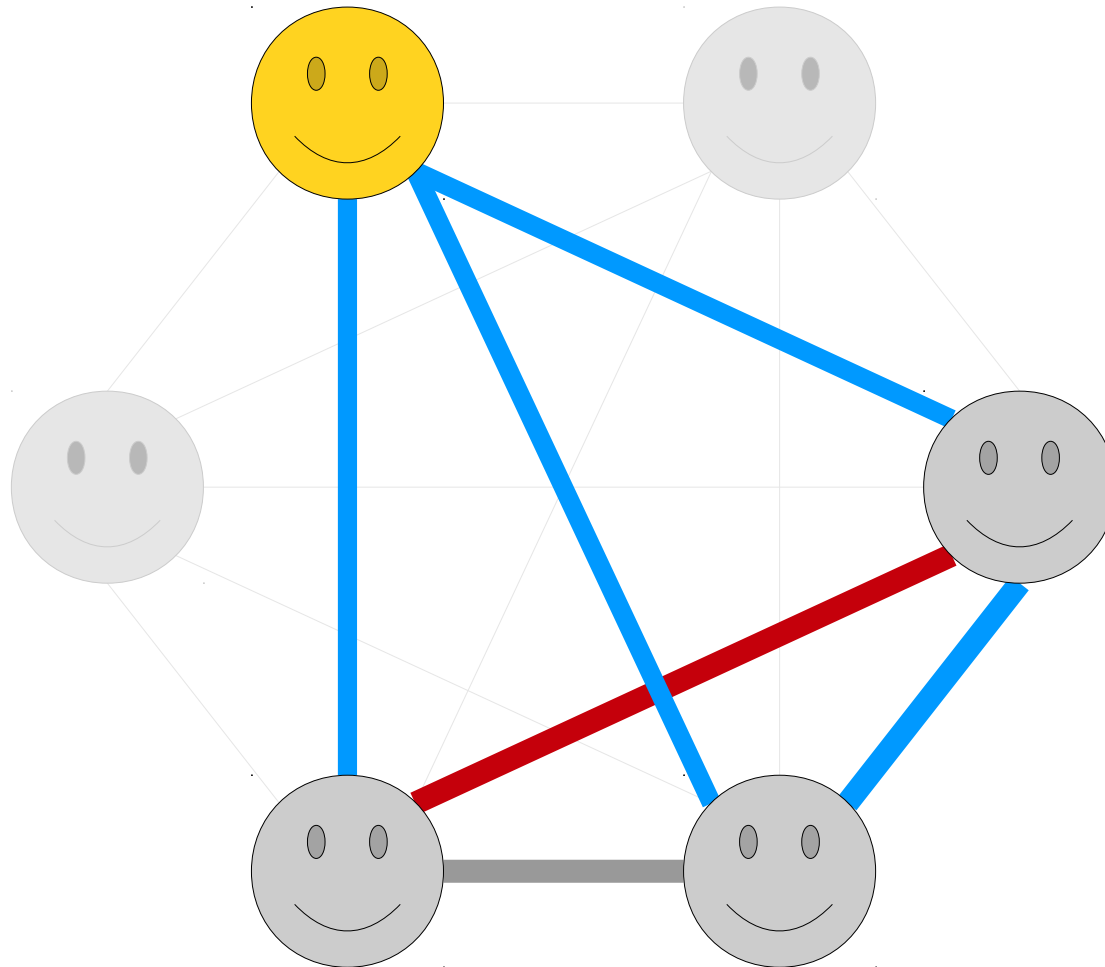
Observation 1: If we pick any node in the graph, that node will have at least $\lceil 5/2 \rceil = 3$ edges of the same color incident to it.

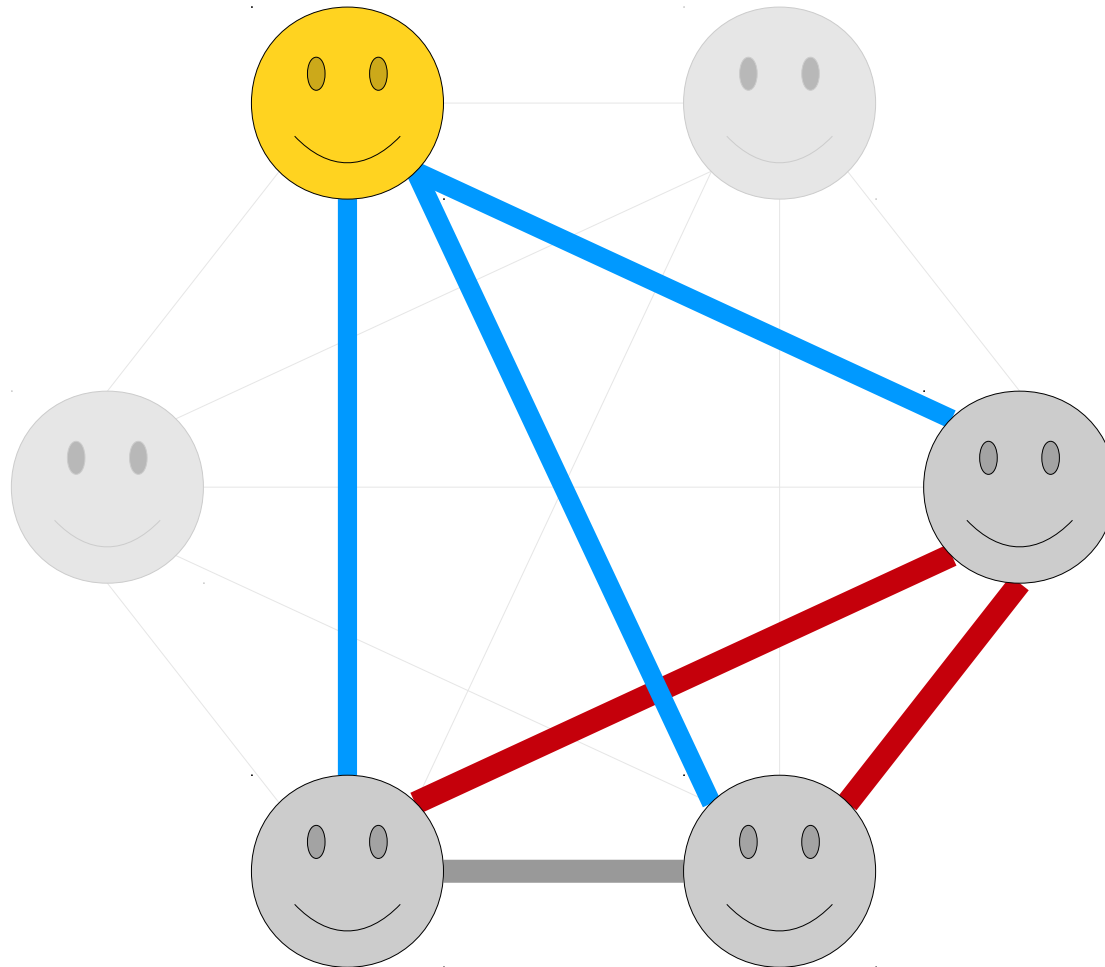


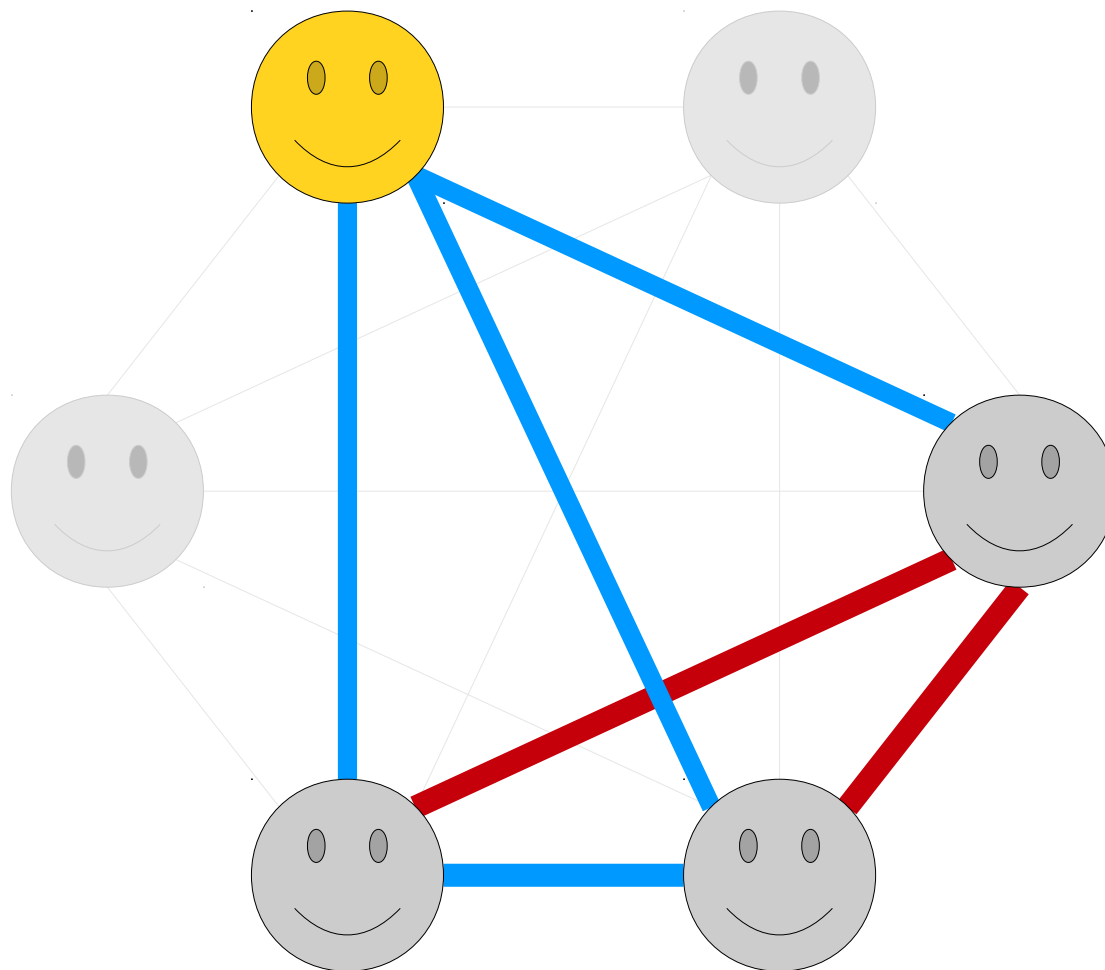


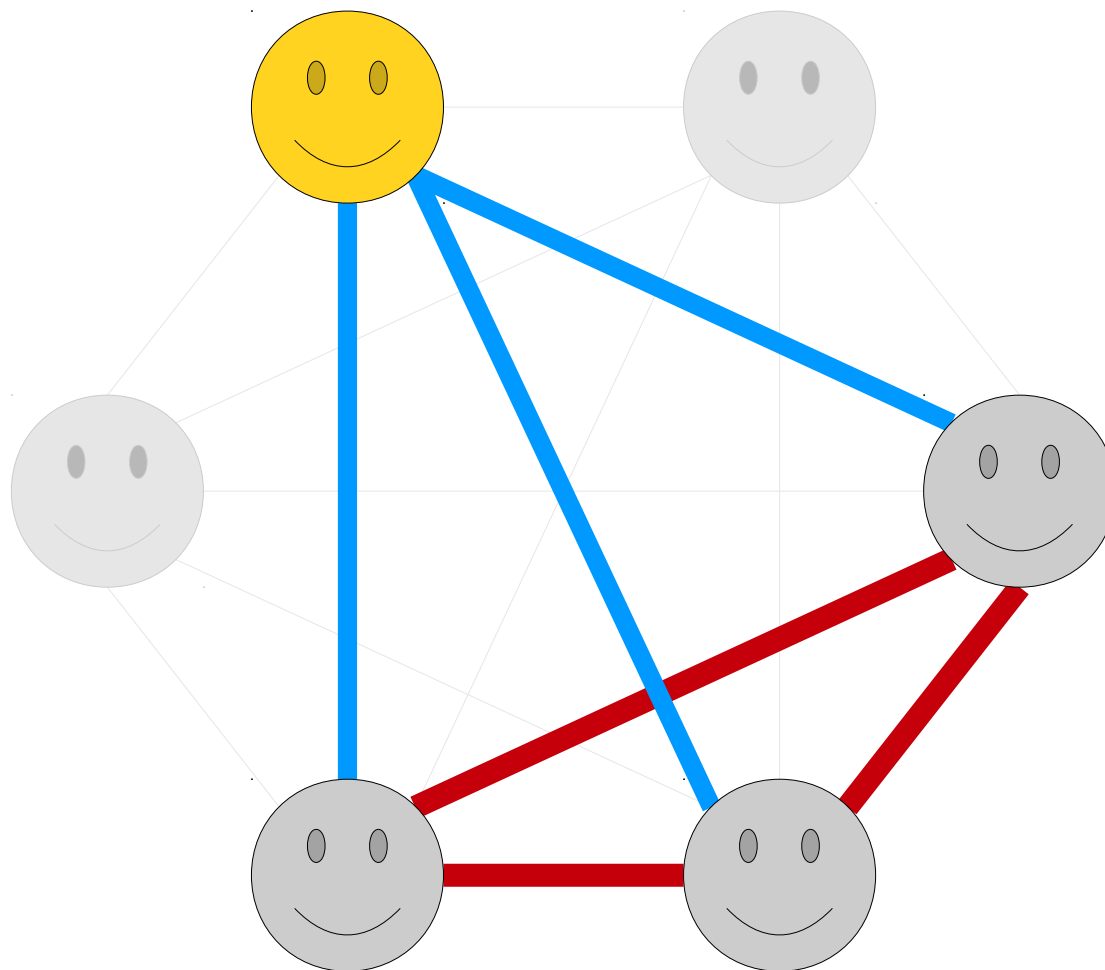












Theorem: Consider a 6-clique in which every edge is colored either red or blue. Then there must be a triangle of red edges, a triangle of blue edges, or both.

Proof: We need to show that the colored 6-clique contains a red triangle or a blue triangle.

Let x be any node in the 6-clique. It is incident to five edges and there are two possible colors for those edges. Therefore, by the generalized pigeonhole principle, at least $\lceil 5/2 \rceil = 3$ of those edges must be the same color. Without loss of generality, assume those edges are blue.

Let r , s , and t be three of the nodes adjacent to node x along a blue edge. If any of the edges $\{r, s\}$, $\{r, t\}$, or $\{s, t\}$ are blue, then one of those edges plus the two edges connecting back to node x form a blue triangle. Otherwise, all three of those edges are red, and they form a red triangle. Overall, this gives a red triangle or a blue triangle, as required. ■

Ramsey Theory

- The theorem we just proved is a special case of a broader result.
- ***Theorem (Ramsey's Theorem):*** For any natural number n , there is a number $R(n)$ where for any clique with $R(n)$ or more nodes that's painted red or blue, that clique has either a red n -clique or a blue n -clique, and for all cliques with fewer than $R(n)$ nodes, there's a way to paint it red and blue so it has no red n -cliques and no blue n -cliques.
 - Our proof was that $R(3) \leq 6$.
- A more philosophical take on this theorem: true disorder is impossible at a large scale, since no matter how you organize things, you're guaranteed to find some interesting substructure.

Going Further

- The pigeonhole principle can be used to prove a *ton* of amazing theorems. Here's a sampler:
 - There is always a way to fairly split rent among multiple people, even if different people want different rooms. (*Sperner's lemma*)
 - You and a friend can climb any mountain from two different starting points so that the two of you maintain the same altitude at each point in time. (*Mountain-climbing theorem*)
 - If you model coffee in a cup as a collection of infinitely many points and then stir the coffee, some point is always where it initially started. (*Brower's fixed-point theorem*)
 - A complex process that doesn't parallelize well must contain a large serial subprocess. (*Mirksy's theorem*)
 - Any positive integer n has a nonzero multiple that can be written purely using the digits 1 and 0. (*Doesn't have a name, but still cool!*)

More to Explore

- Interested in more about graphs and the pigeonhole principle? Check out...
 - ... **Math 107** (Graph Theory), a deep dive into graph theory.
 - ... **Math 108** (Combinatorics), which explores a bunch of results pertaining to graphs and counting things.
 - ... **CS161** (Algorithms), which explores algorithms for computing important properties of graphs.
 - ... **CS224W** (Deep Learning on Graphs), which uses a mix of mathematical and statistical techniques to explore graphs.

Next Time

- ***Mathematical Induction***
 - Reasoning about stepwise processes!
- ***Applications of Induction***
 - To numbers!
 - To anticounterfeiting!
 - To puzzles!