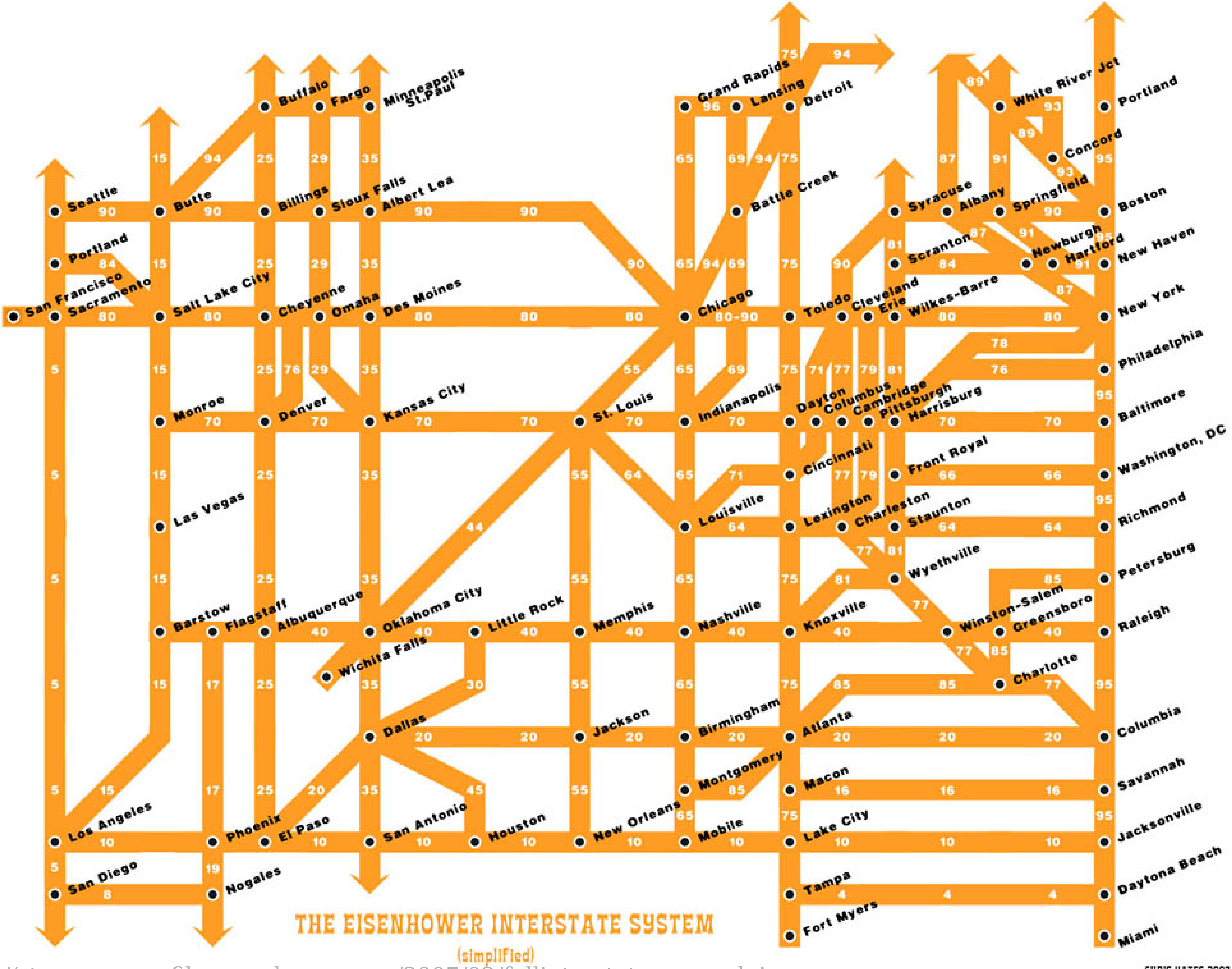
Graph Theory

Part One

Outline for Today

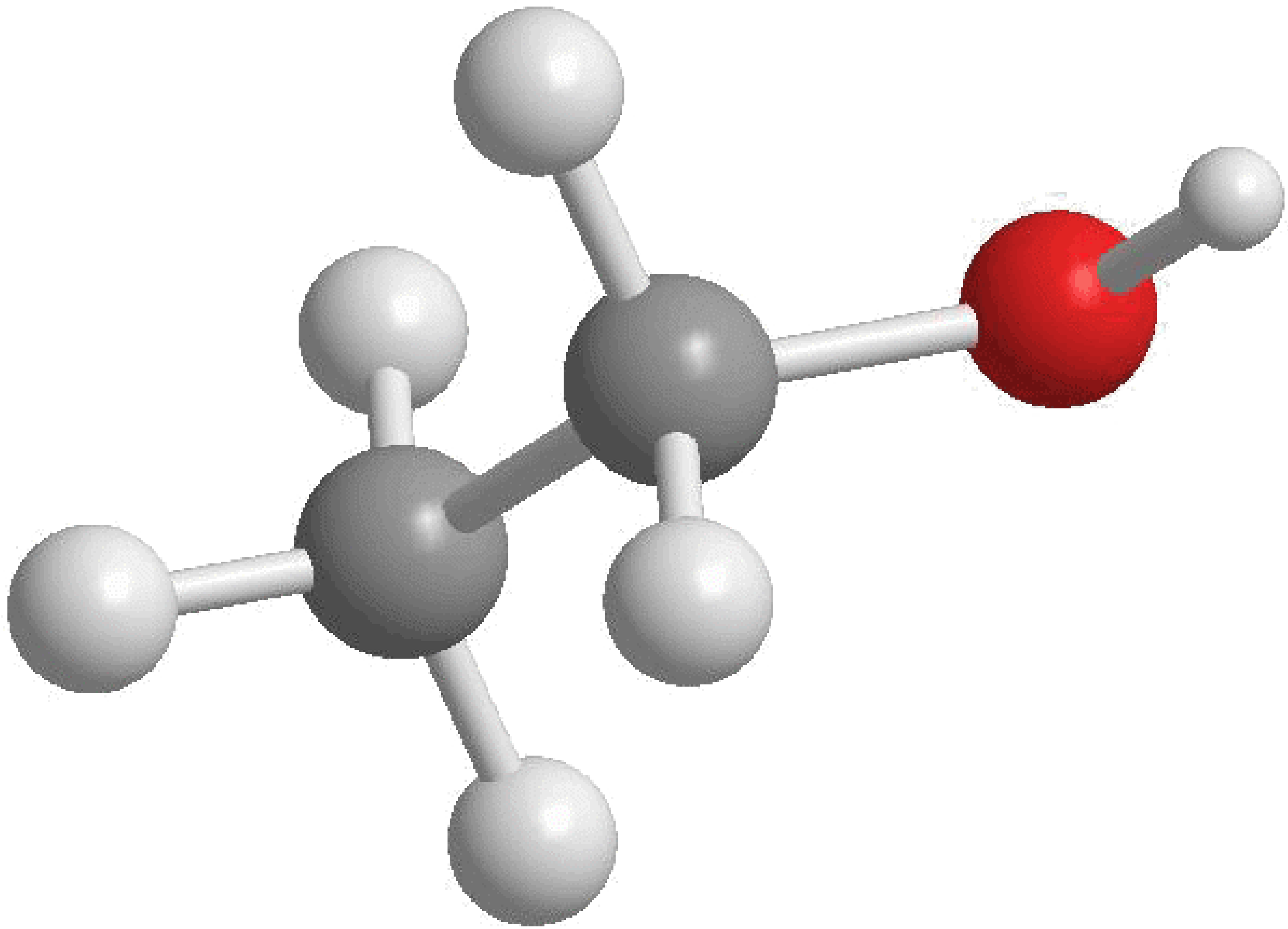
- ***Graphs and Digraphs***
  - Two fundamental mathematical structures.
- ***Graphs Meet FOL***
  - Building visual intuitions.
- ***Independent Sets and Vertex Covers***
  - Two structures in graphs.

Graphs and Digraphs

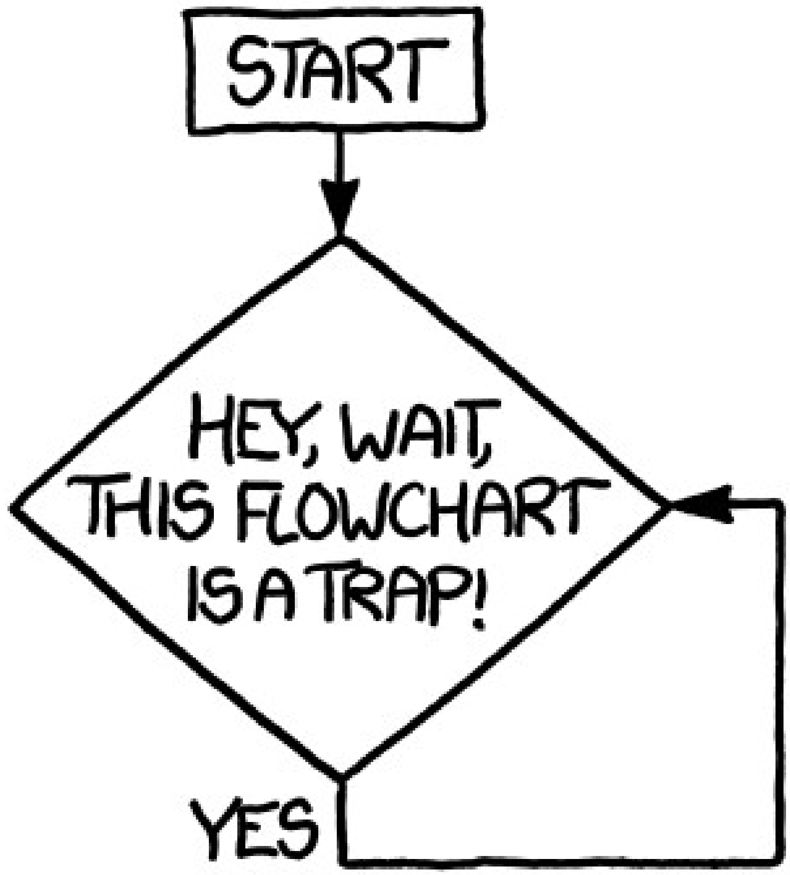


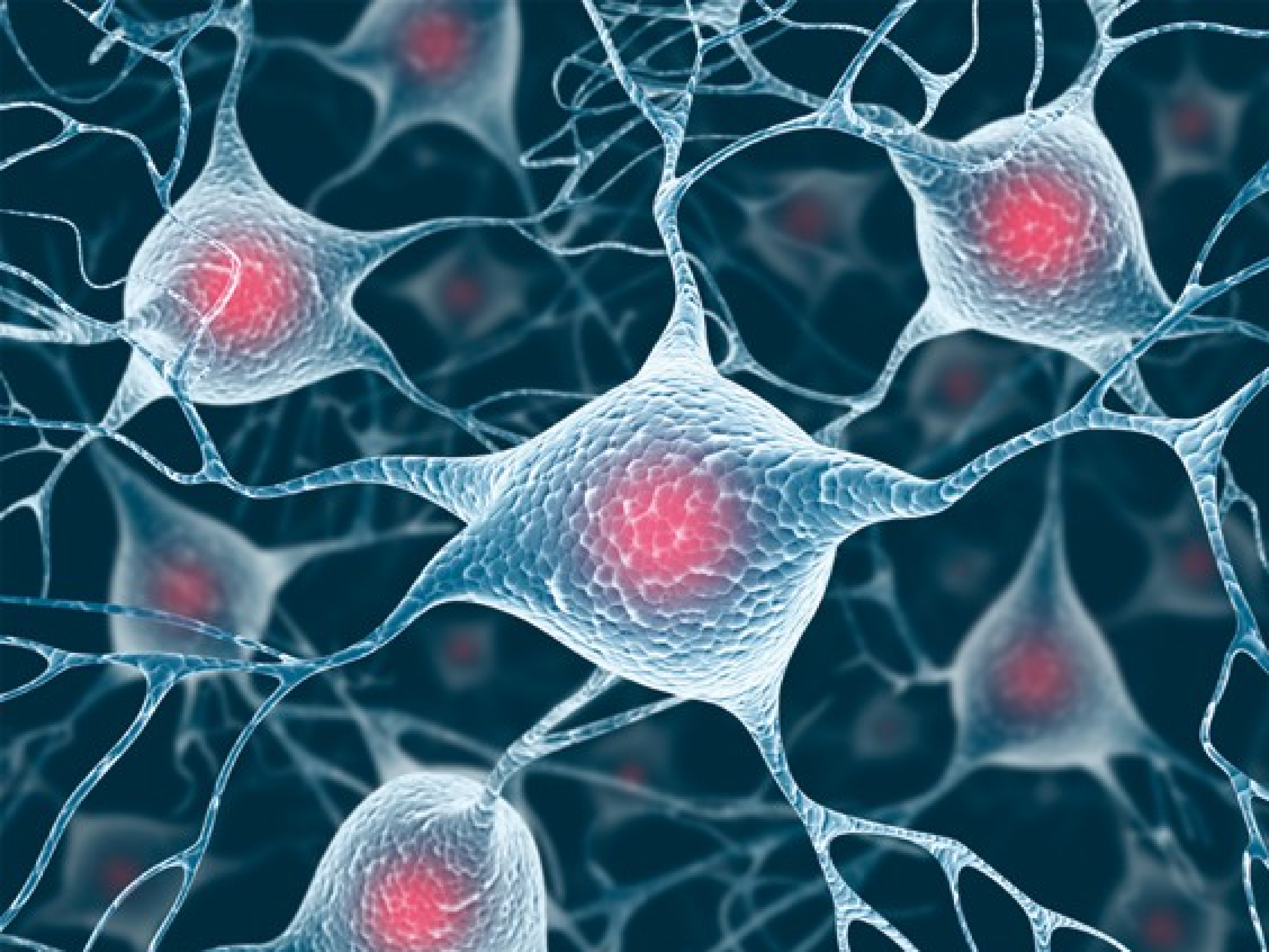
http://strangemaps.files.wordpress.com/2007/02/fullinterstatemap-web.jpg

Chemical Bonds



http://4.bp.blogspot.com/-xCtBJ8lKHqA/Tjm0BONWBRI/AAAAAAAAAK4/-mHrbAUOHHg/s1600/


START
HEY, WAIT, THIS FLOWCHART IS A TRAP!
YES

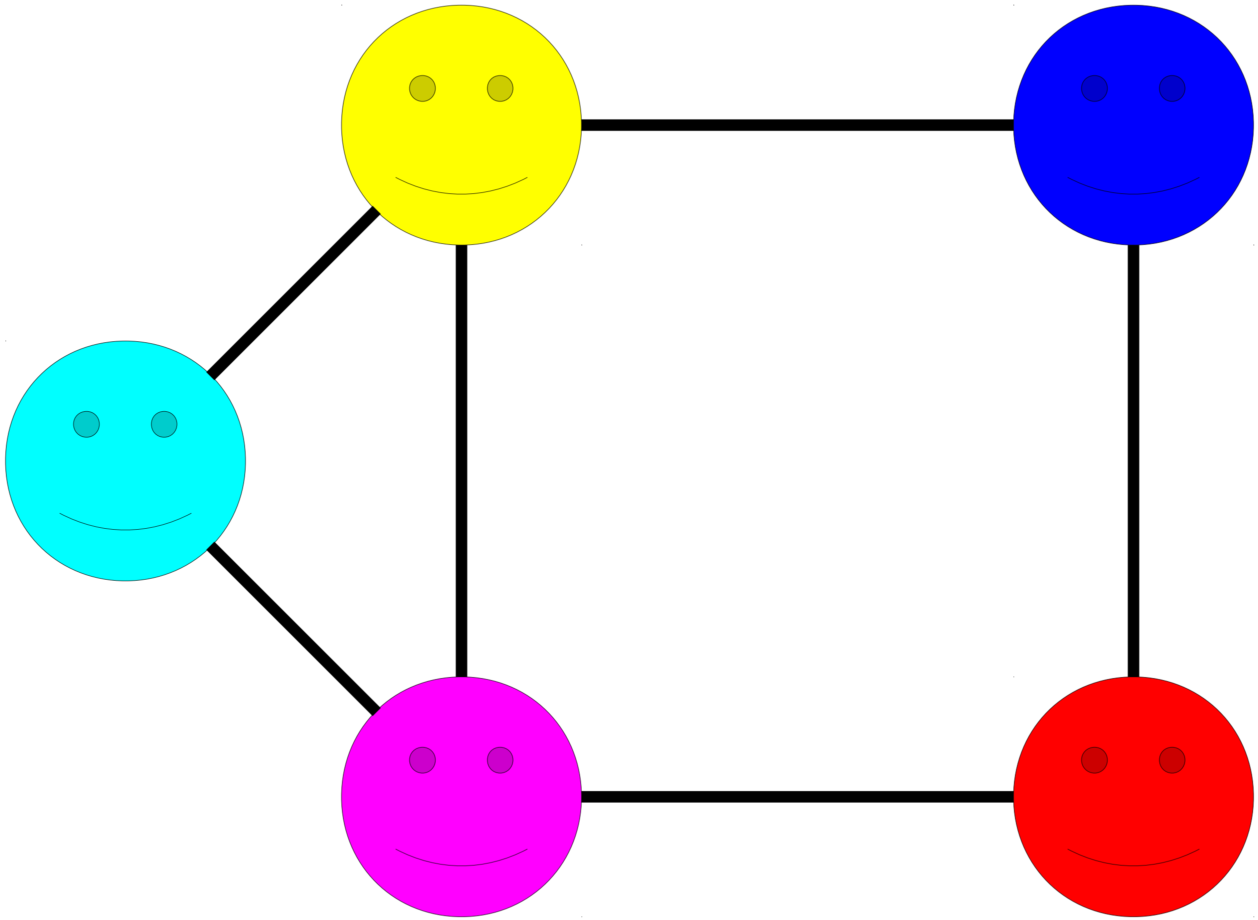


LinkedIn®

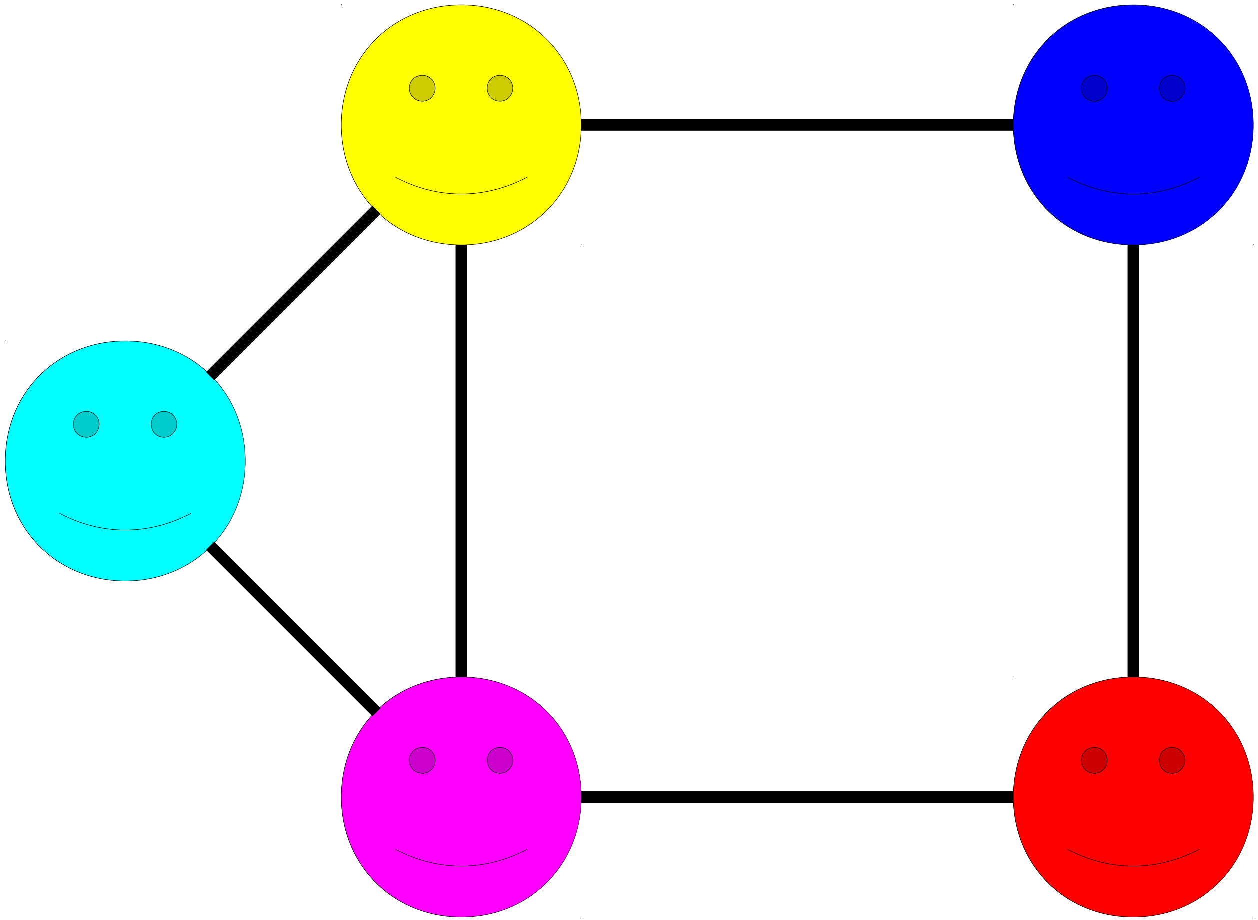
What's in Common

- Each of these structures consists of
  - a collection of objects and
  - links between those objects.
- ***Goal:*** find a general framework for describing these objects and their properties.

A ***graph*** is a mathematical structure for representing relationships.

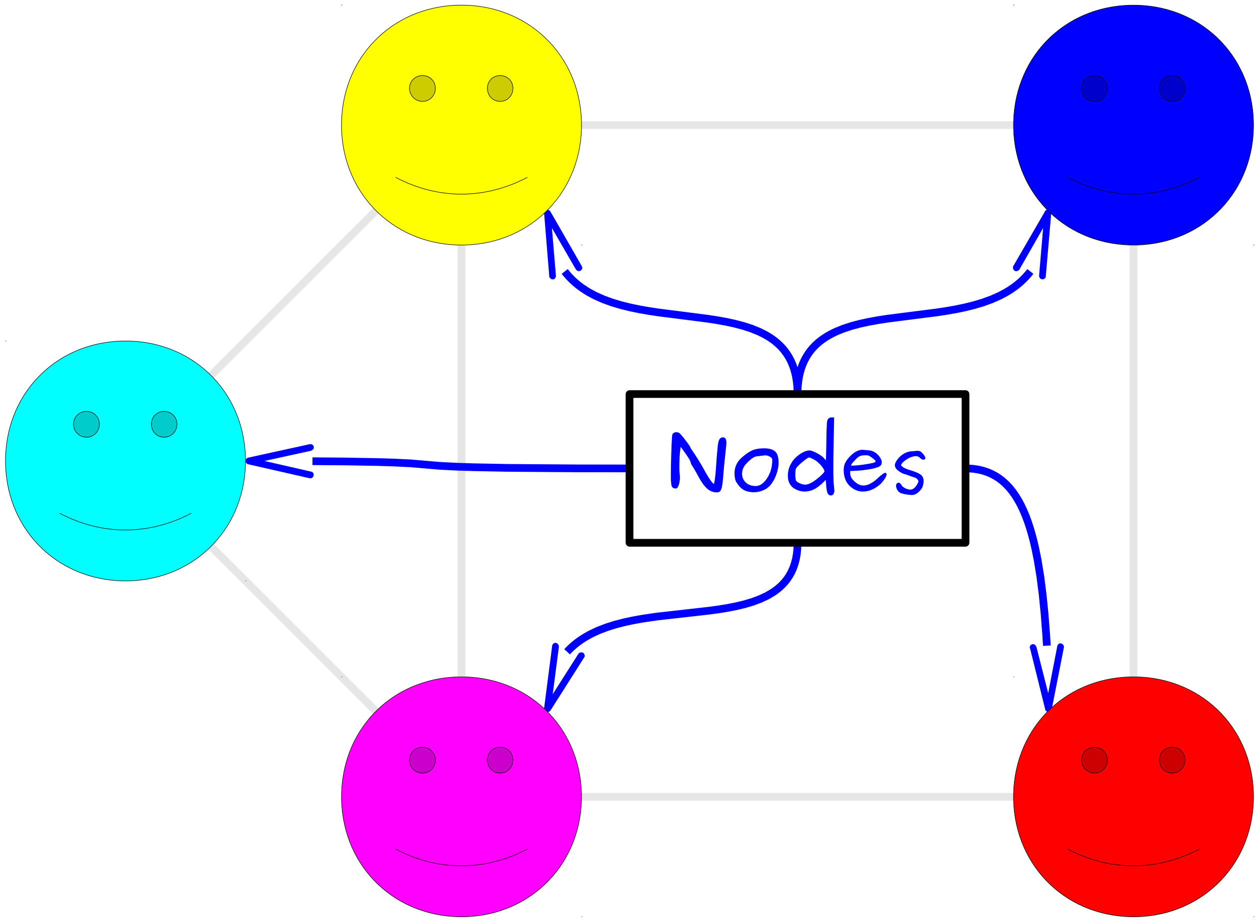


A ***graph*** is a mathematical structure for representing relationships.



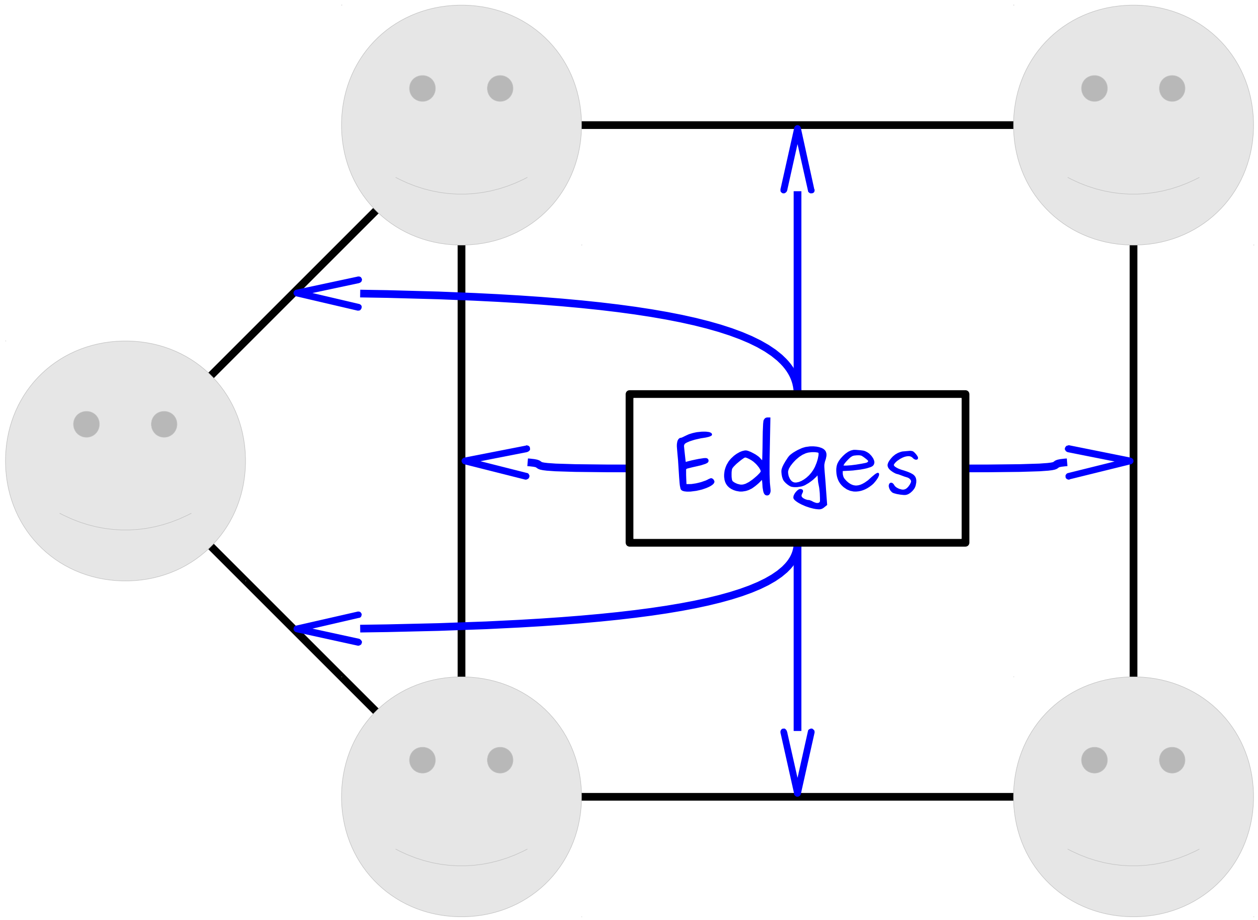
A graph consists of a set of ***nodes*** (or ***vertices***) connected by ***edges*** (or ***arcs***)

A ***graph*** is a mathematical structure for representing relationships.



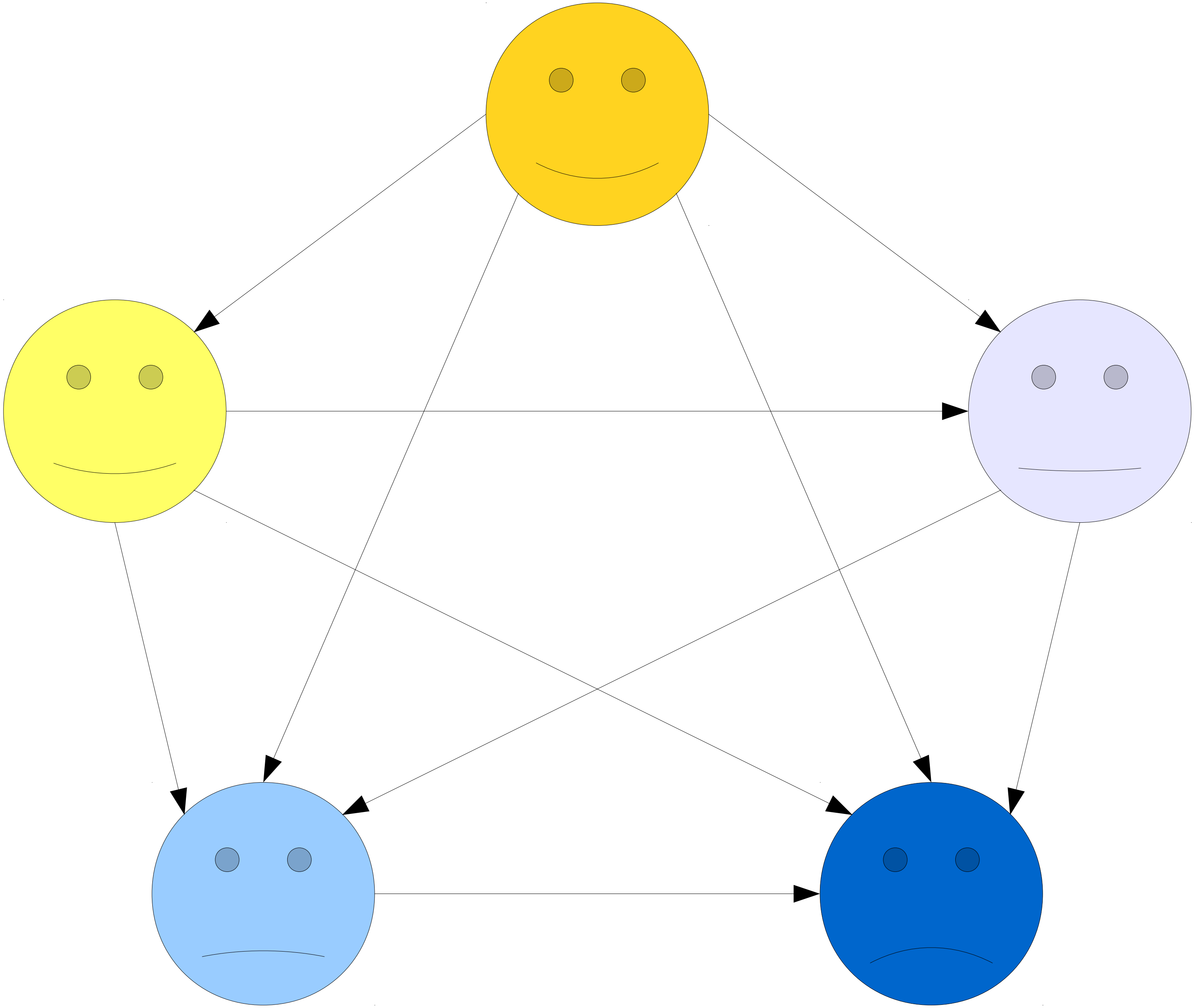
A graph consists of a set of ***nodes*** (or ***vertices***) connected by ***edges*** (or ***arcs***)

A ***graph*** is a mathematical structure for representing relationships.

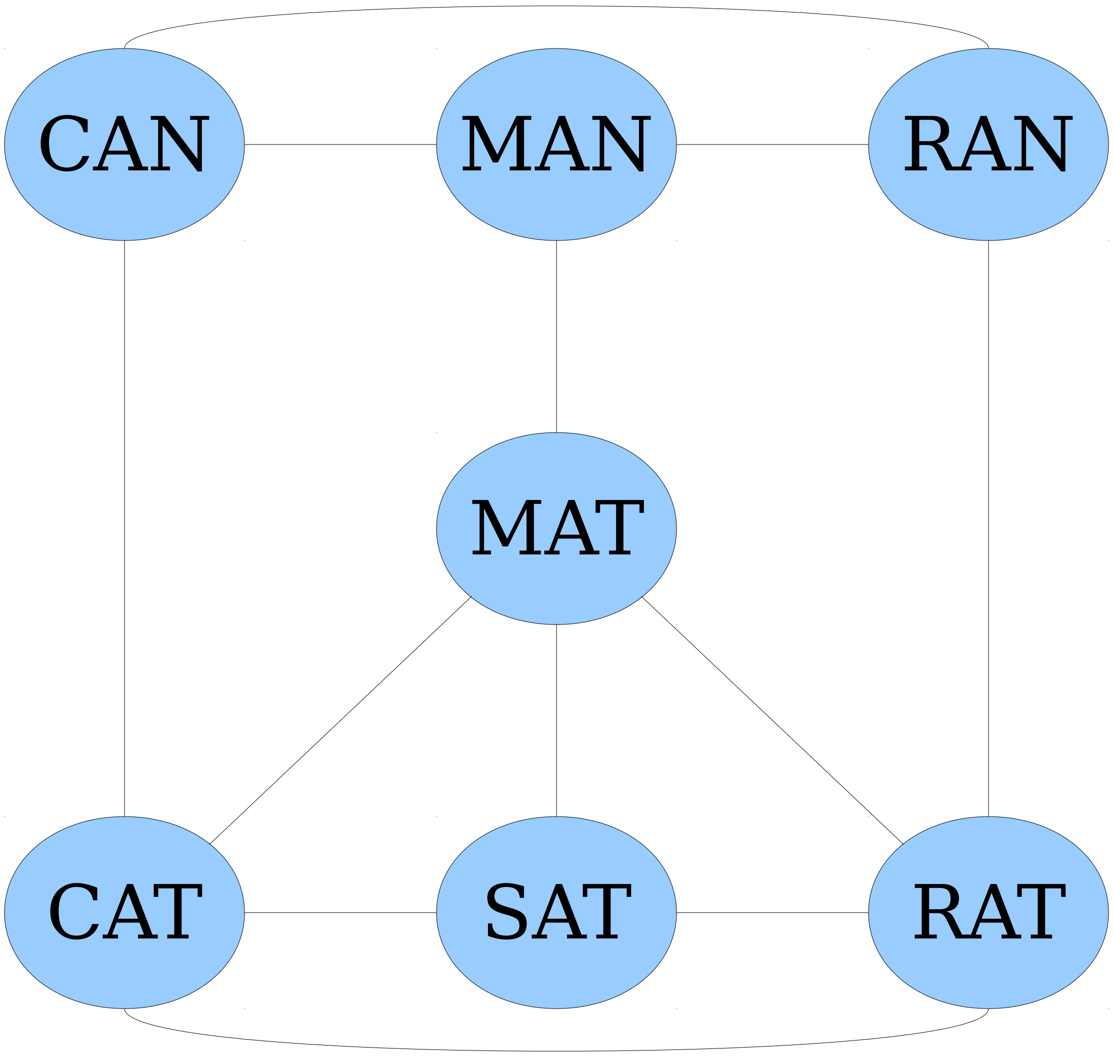


A graph consists of a set of ***nodes*** (or ***vertices***) connected by ***edges*** (or ***arcs***)

Some graphs are ***directed***.



Some graphs are ***undirected***.



Graphs and Digraphs

- An ***undirected graph*** is one where edges link nodes, with no endpoint preferred over the other.
- A ***directed graph*** (or ***digraph***) is one where edges have an associated direction.
- (There's something called a ***mixed graph*** that allows for both, but they're fairly uncommon and we won't talk about them.)
- Unless specified otherwise:

☞ ***"Graph" means "undirected graph"*** ☜

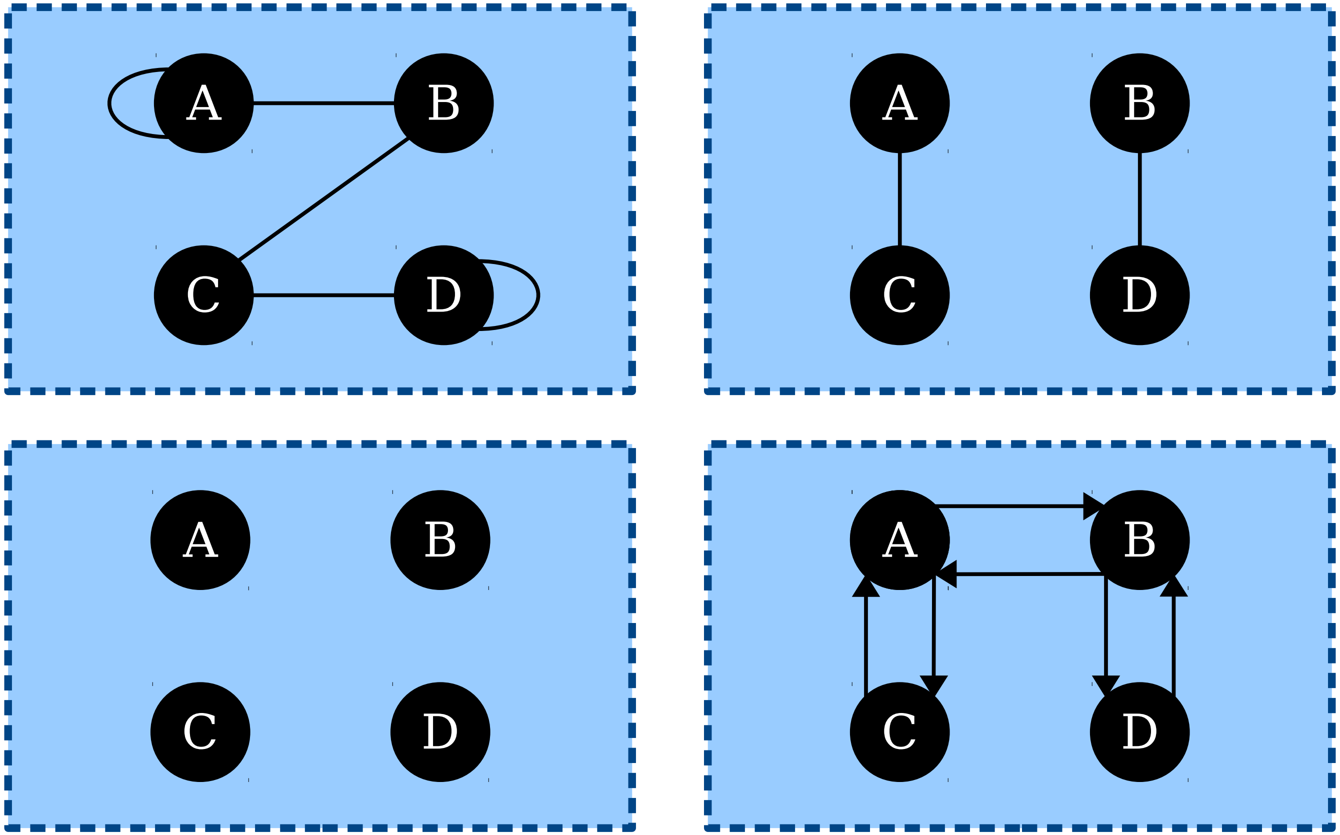
Formalizing Graphs

- How might we define a graph mathematically?
- We need to specify
  - what the nodes in the graph are, and
  - which edges are in the graph.
- The nodes can be pretty much anything.
- What about the edges?

Formalizing Graphs

- An ***unordered pair*** is a set $\{a, b\}$ of two elements $a \neq b$. (Remember that sets are unordered.)
  - For example, $\{0, 1\} = \{1, 0\}$
- An ***undirected graph*** is an ordered pair $G = (V, E)$, where
  - V is a set of nodes, which can be anything, and
  - E is a set of edges, which are *unordered* pairs of nodes drawn from V.
- A ***directed graph*** (or ***digraph***) is an ordered pair $G = (V, E)$, where
  - V is a set of nodes, which can be anything, and
  - E is a set of edges, which are *ordered* pairs of nodes drawn from V.

- An ***unordered pair*** is a set $\{a, b\}$ of two elements $a \neq b$.
- An ***undirected graph*** is an ordered pair $G = (V, E)$, where
  - V is a set of nodes, which can be anything, and
  - E is a set of edges, which are unordered pairs of nodes drawn from V.

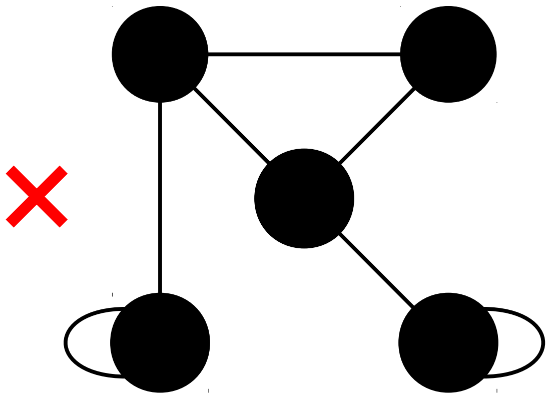


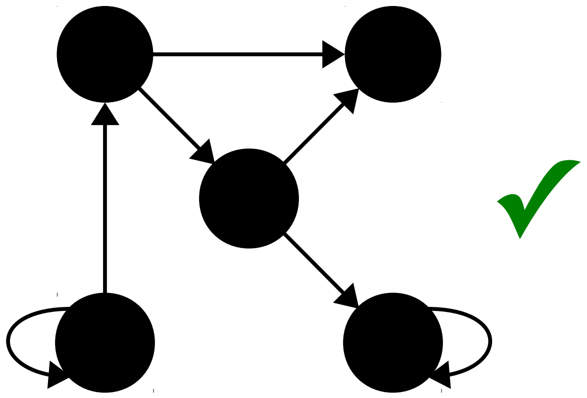
How many of these drawings are of valid undirected graphs?

Respond at pollev.com/cs103

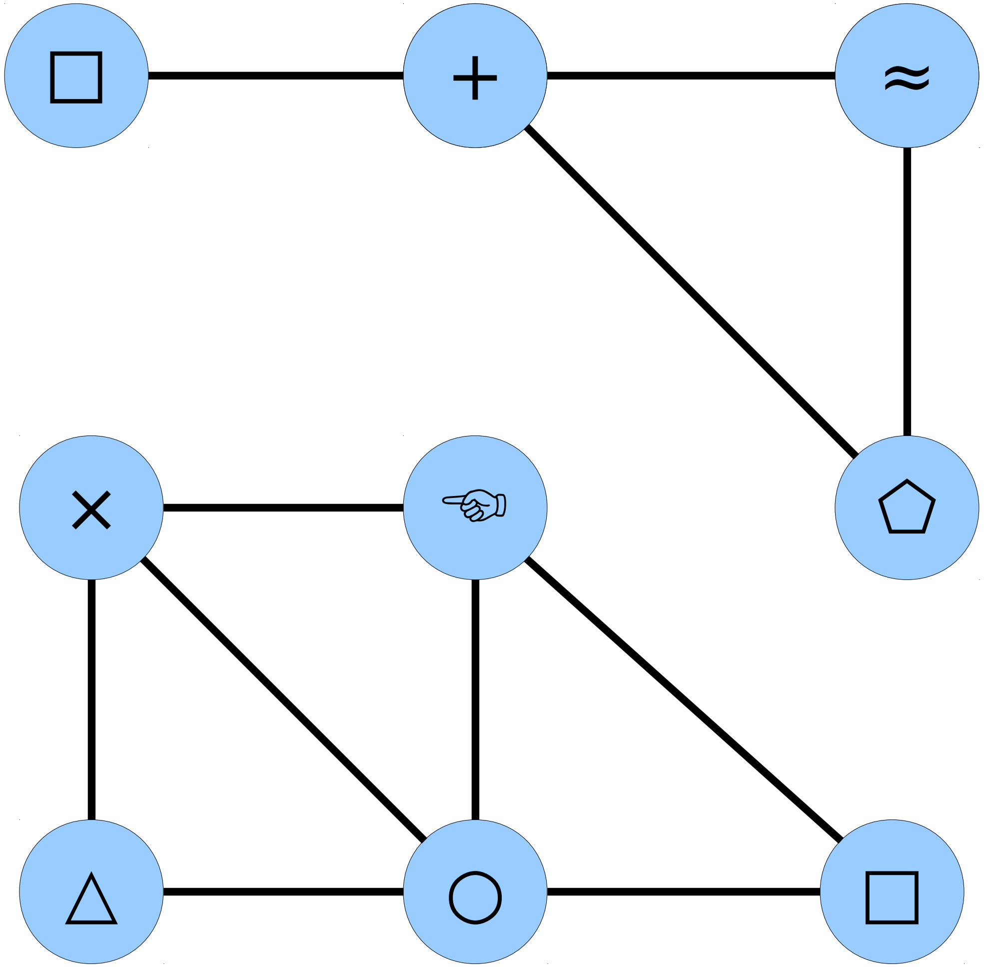
Self-Loops

- An edge from a node to itself is called a ***self-loop***.
- In (undirected) graphs, self-loops are generally not allowed.
    - Can you see how this follows from the definition?
- In digraphs, self-loops are generally allowed unless specified otherwise.



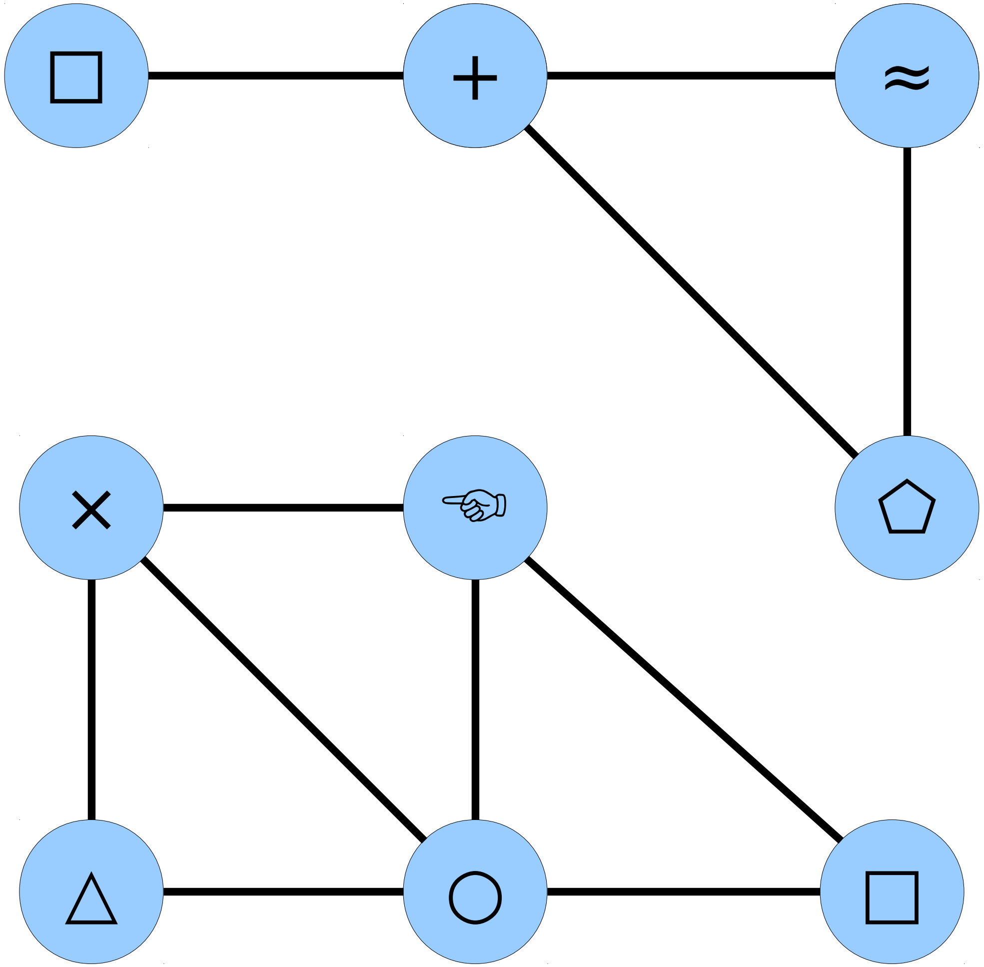


The Great Graph Gallery

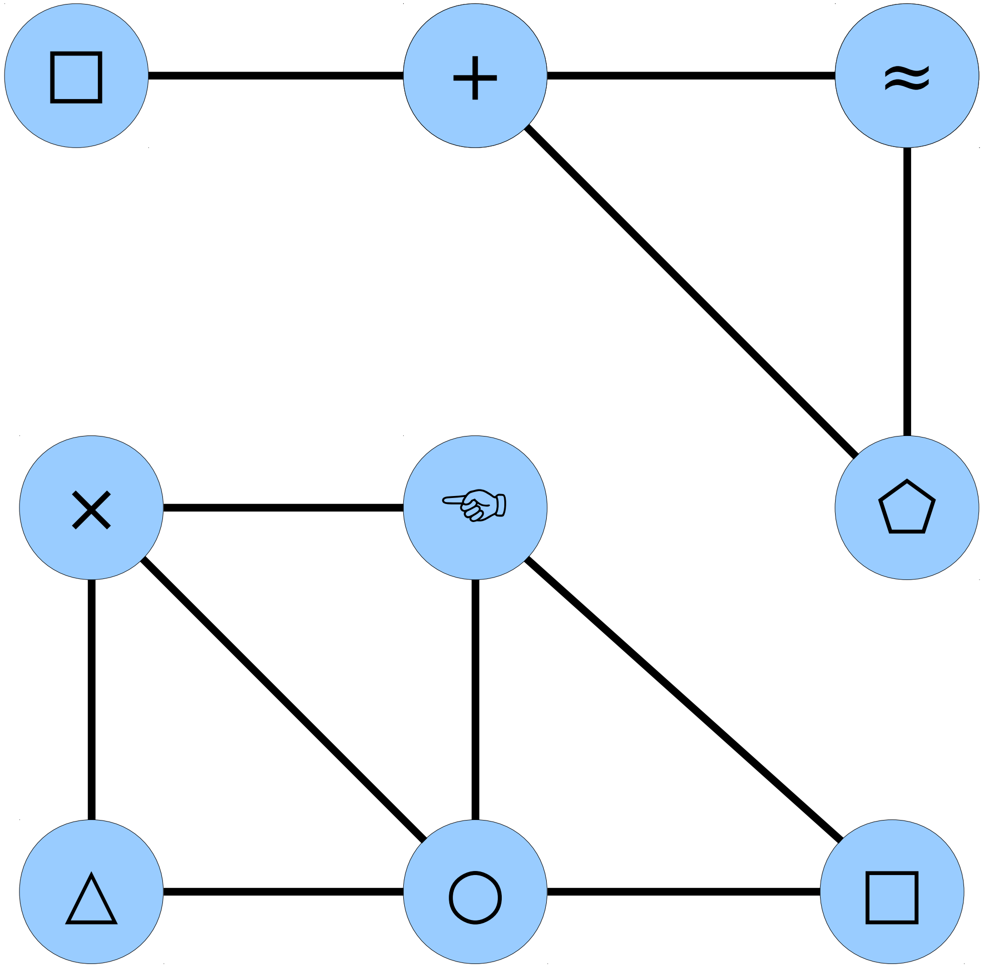


Is this formula true about this graph?

$$\forall u \in V.\ \exists v \in V.\ \{u, v\} \in E$$

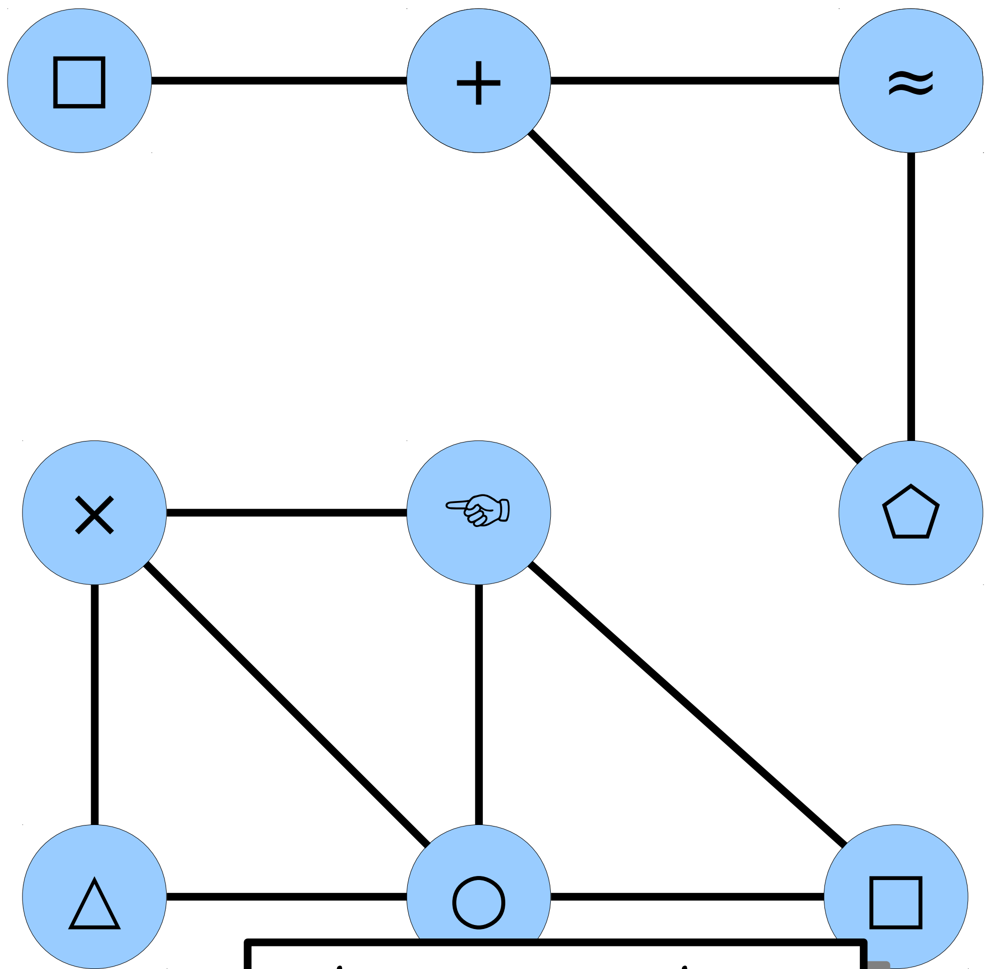


$$\forall u \in V.\ \exists v \in V.\ \{u, v\} \in E$$



"for any node u"

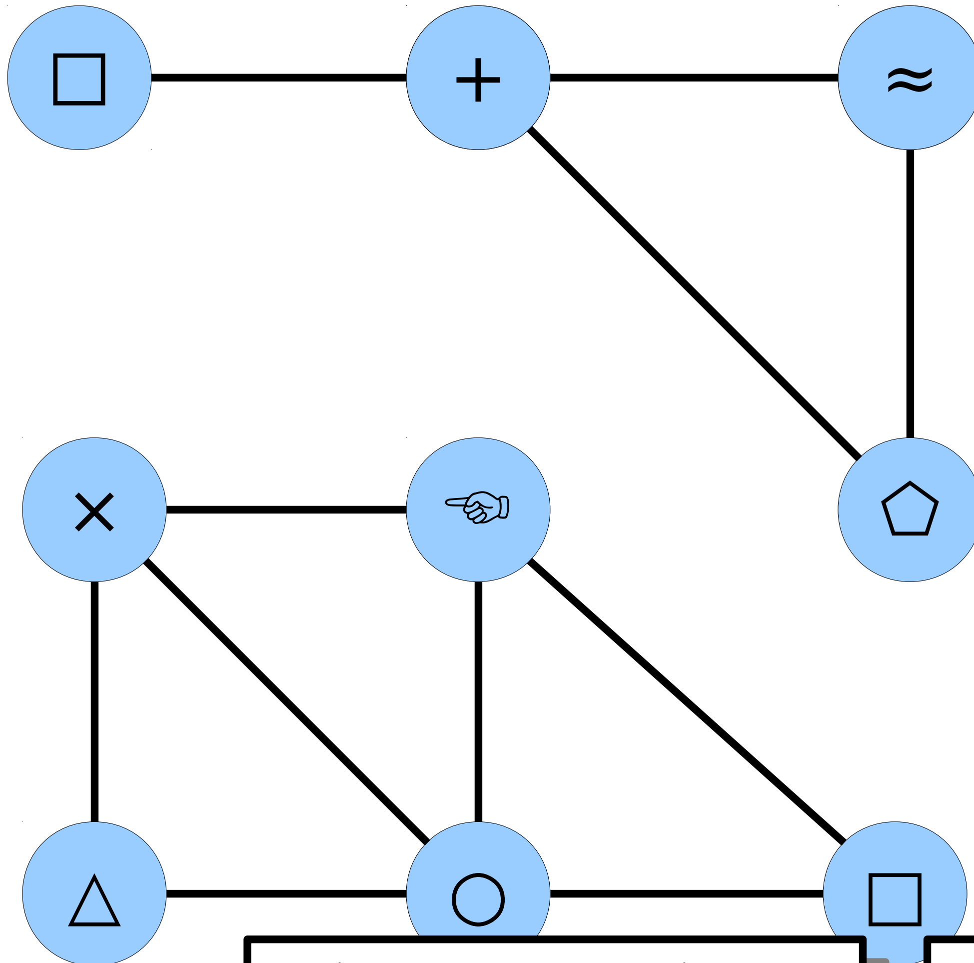
$\forall u \in V. \; \exists v \in V. \; \{u, v\} \in E$



"for any node u"

"there exists a node v"

$\forall u \in V.\ \exists v \in V.\ \{u, v\} \in E$



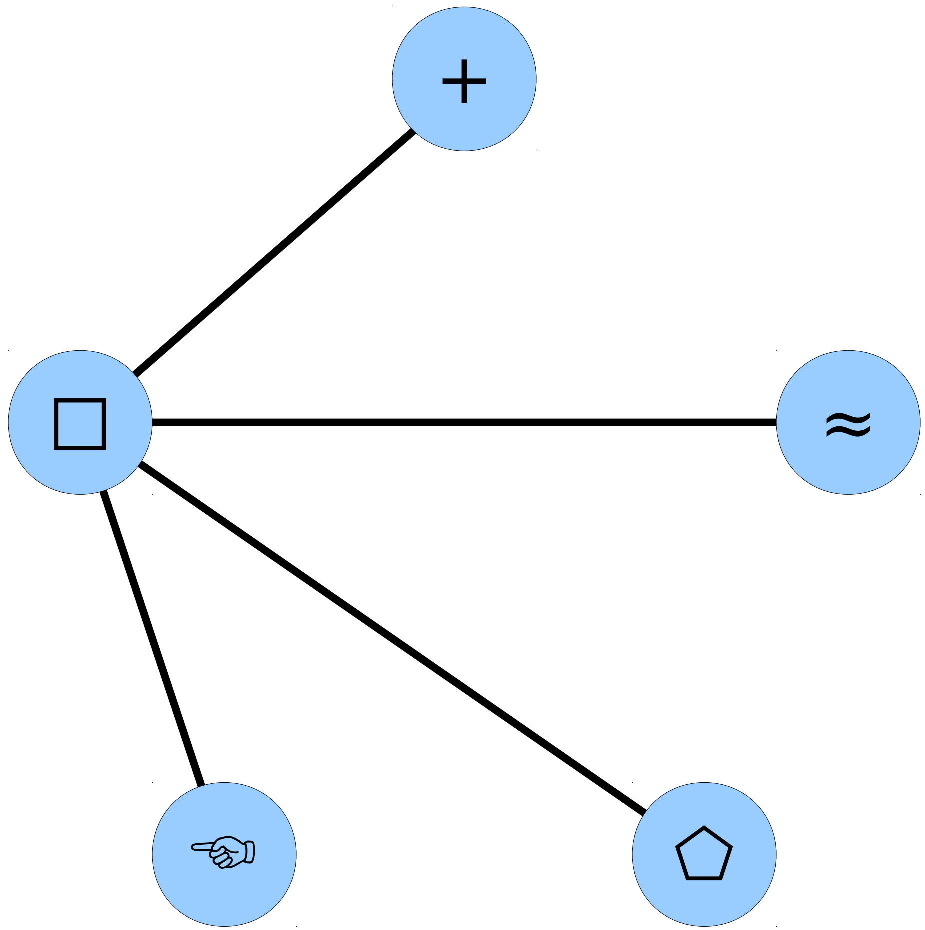
"for any node u"

"there exists a node v"

"where u and v are ***adjacent***"

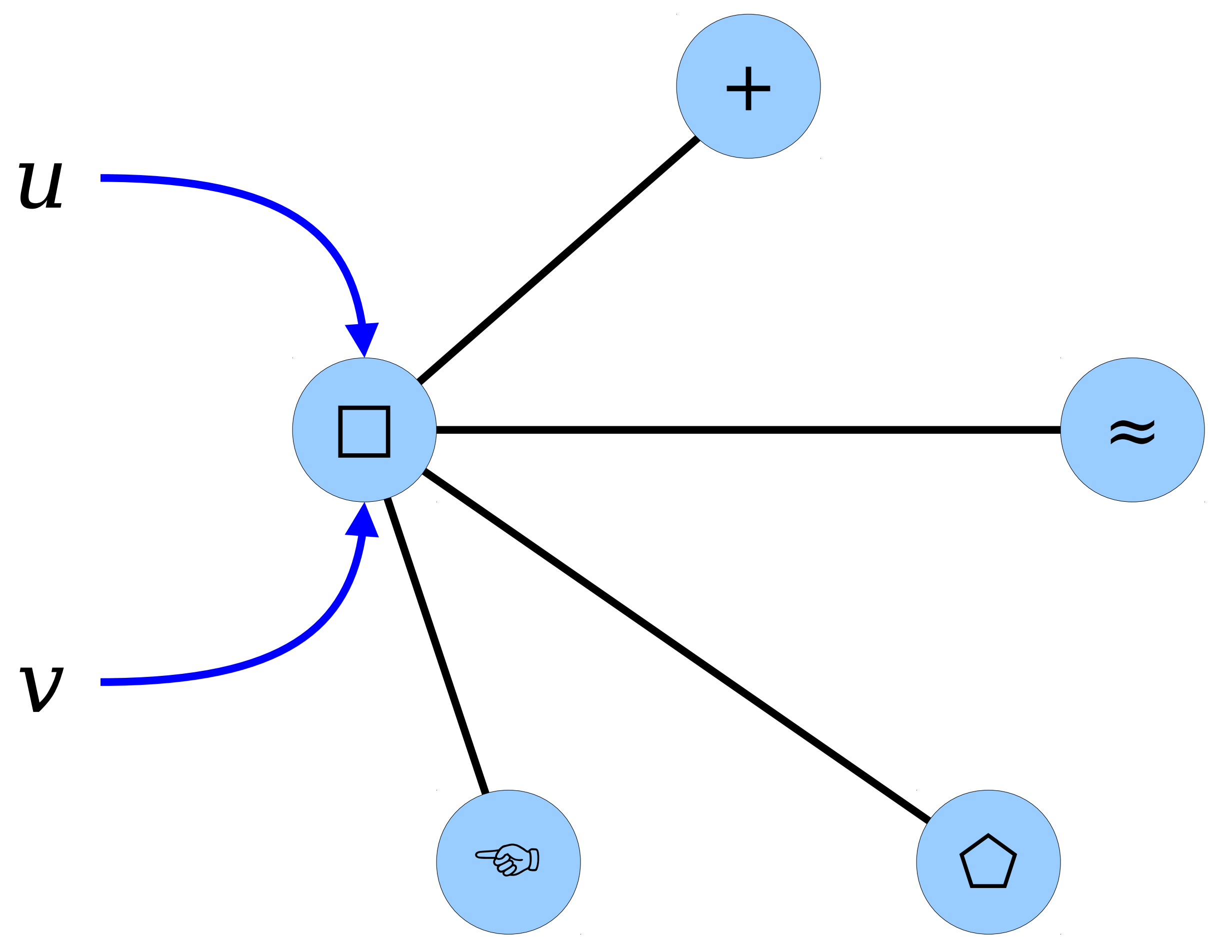
$$\forall u \in V.\ \exists v \in V.\ \{u, v\} \in E$$

Respond at pollev.com/cs103



Is this formula true about this graph?

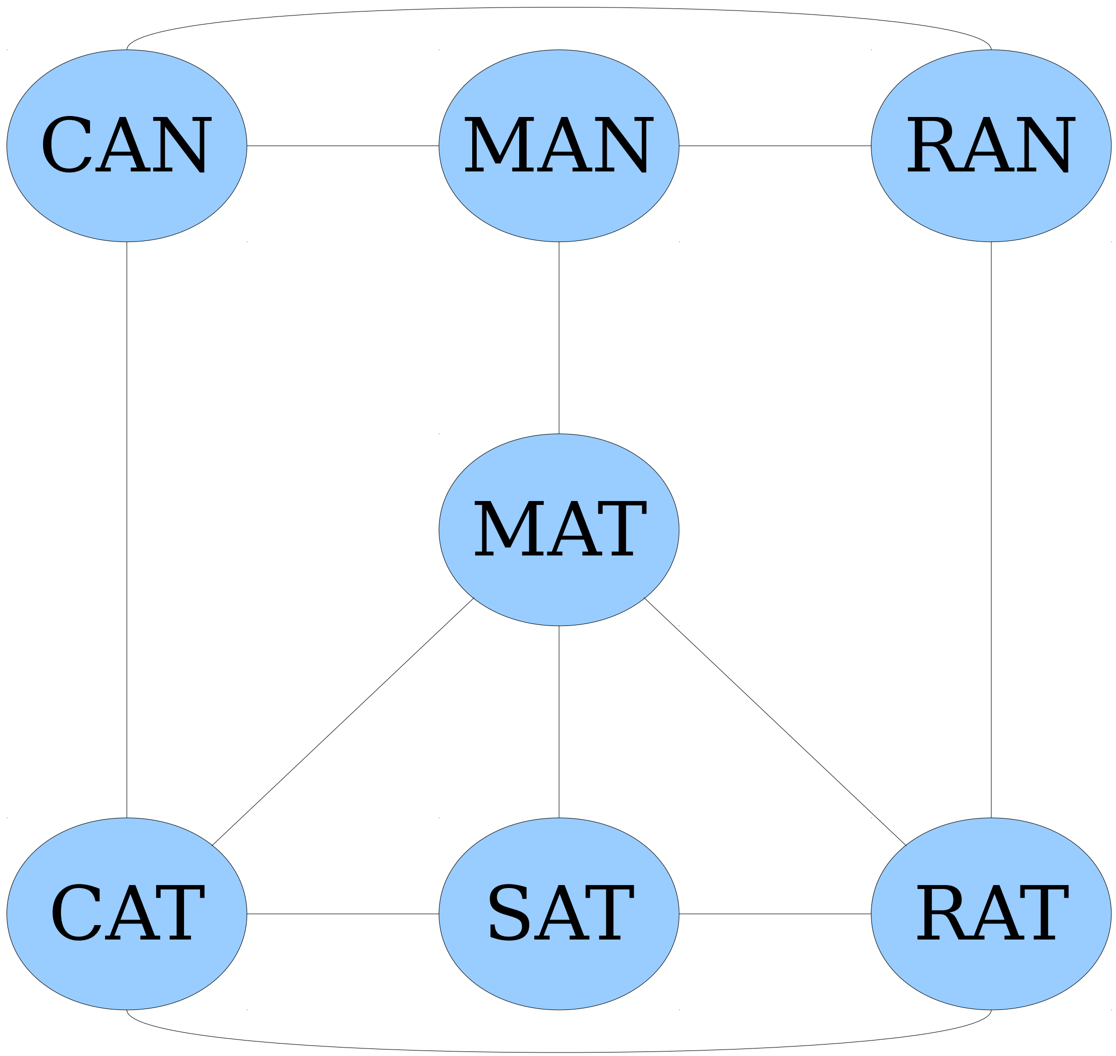
$$\exists u \in V.\ \forall v \in V.\ \{u, v\} \in E$$



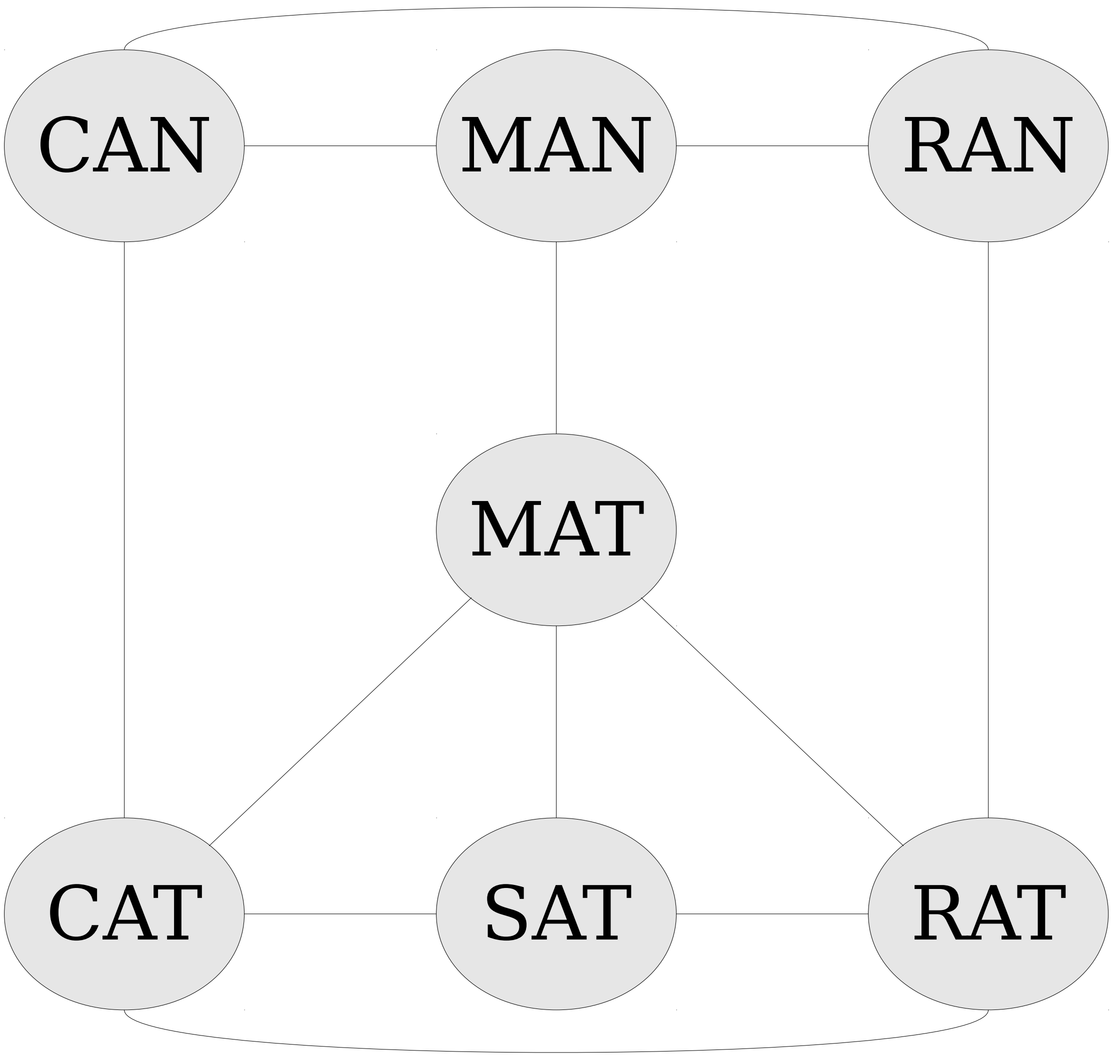
Is this formula true about this graph?

$$\exists u \in V.\ \forall v \in V.\ \{u, v\} \in E$$

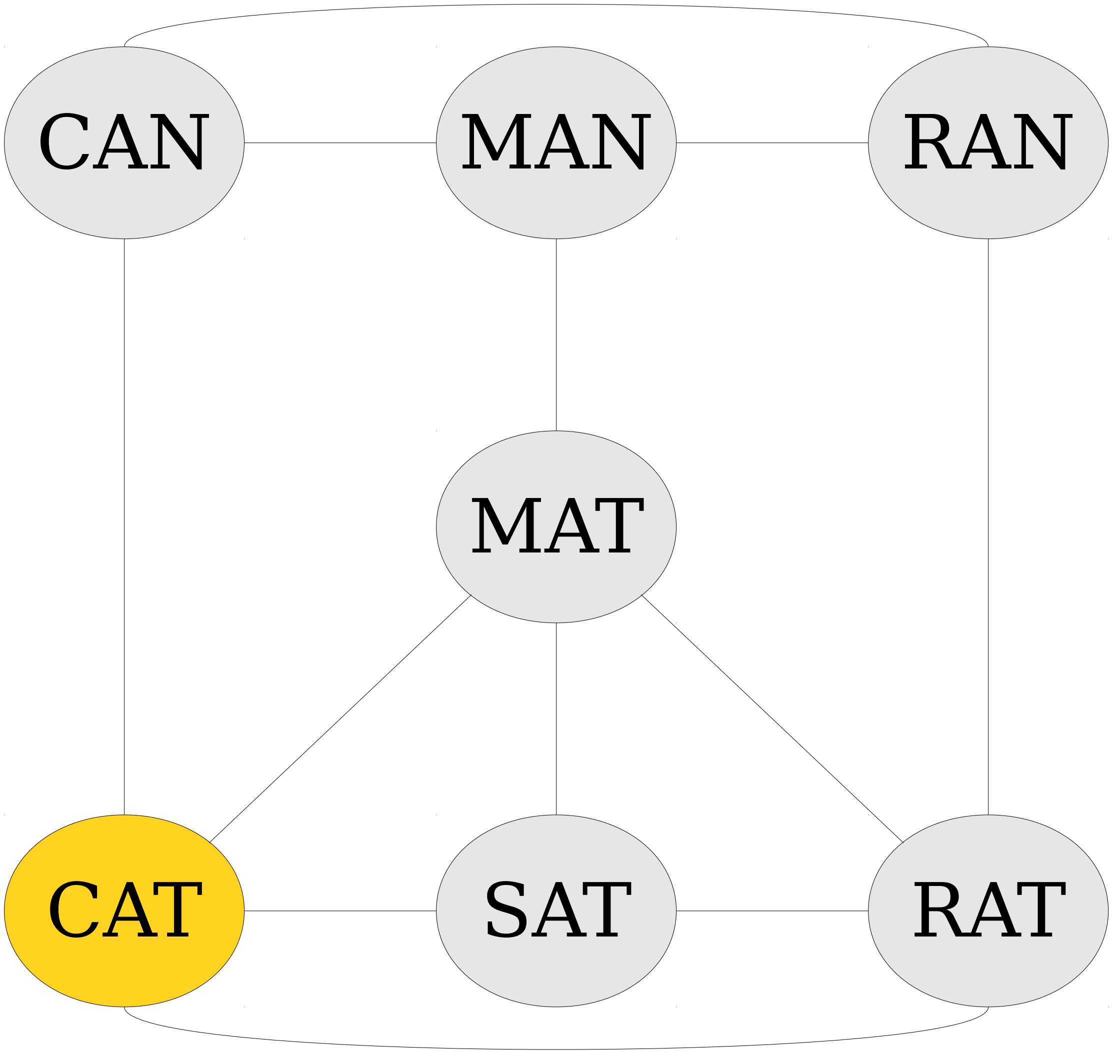
Walks, Paths, and Reachability



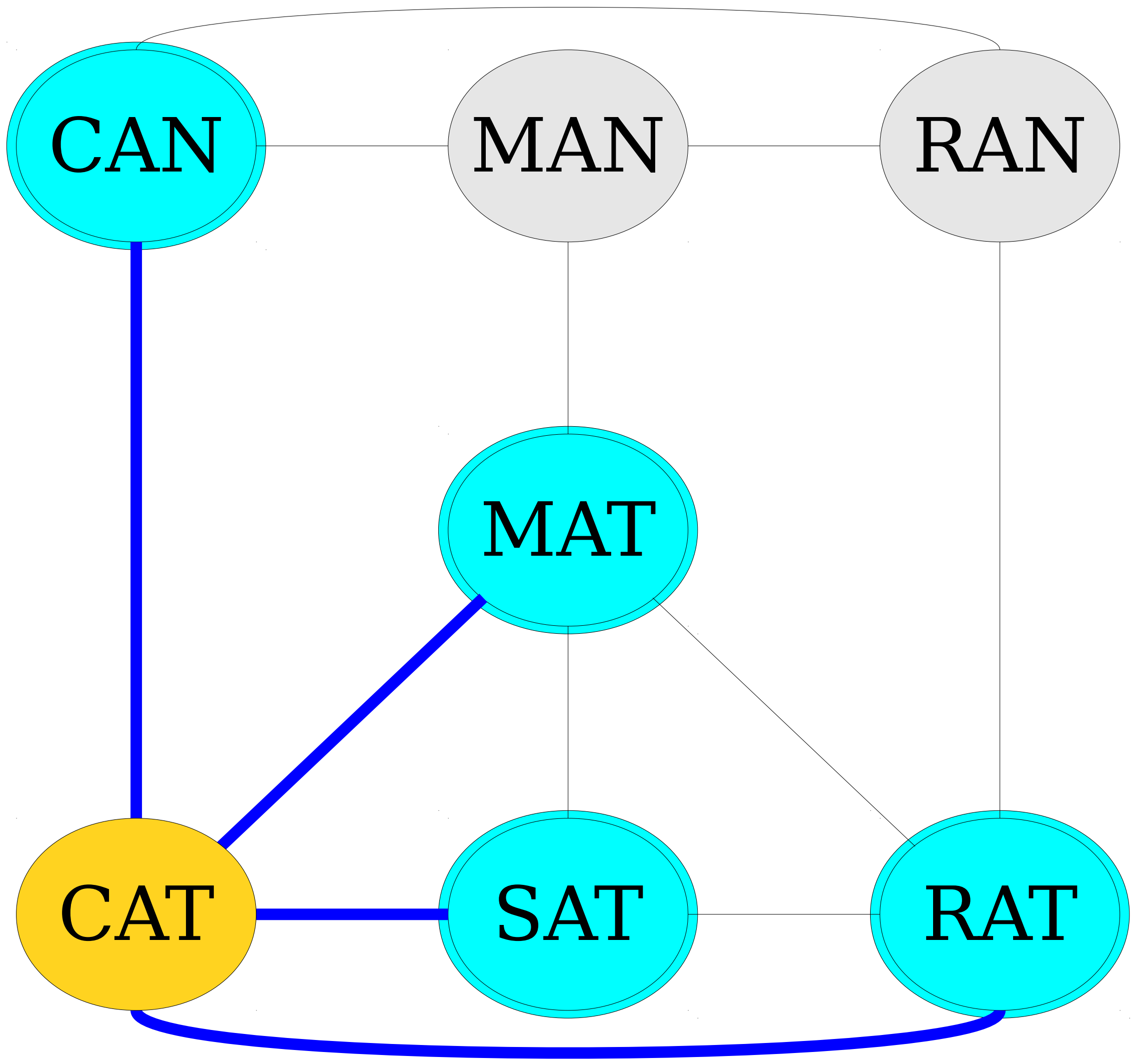
Two nodes are called ***adjacent*** if there is an edge between them.



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



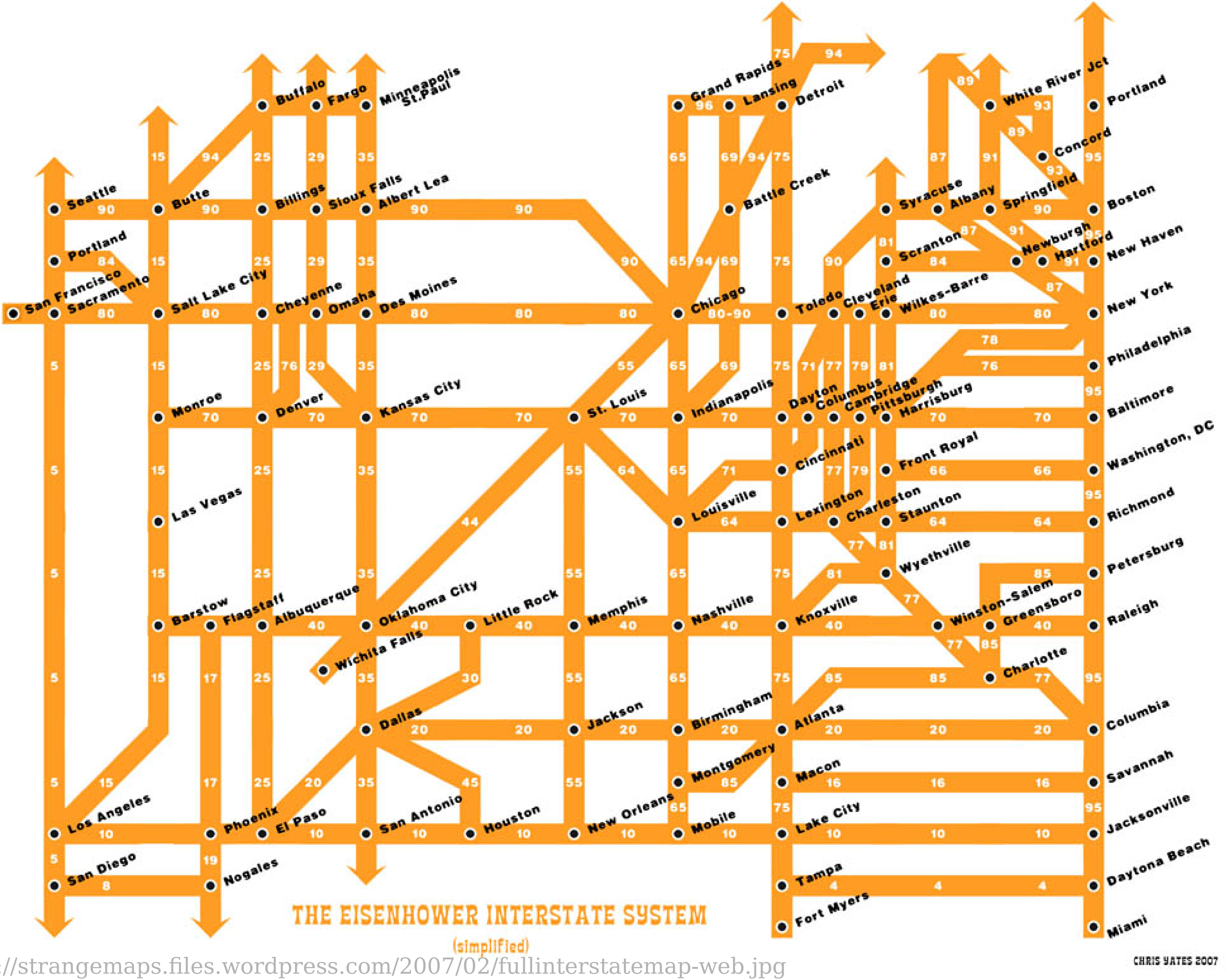
Two nodes are called ***adjacent*** if there is an edge between them.



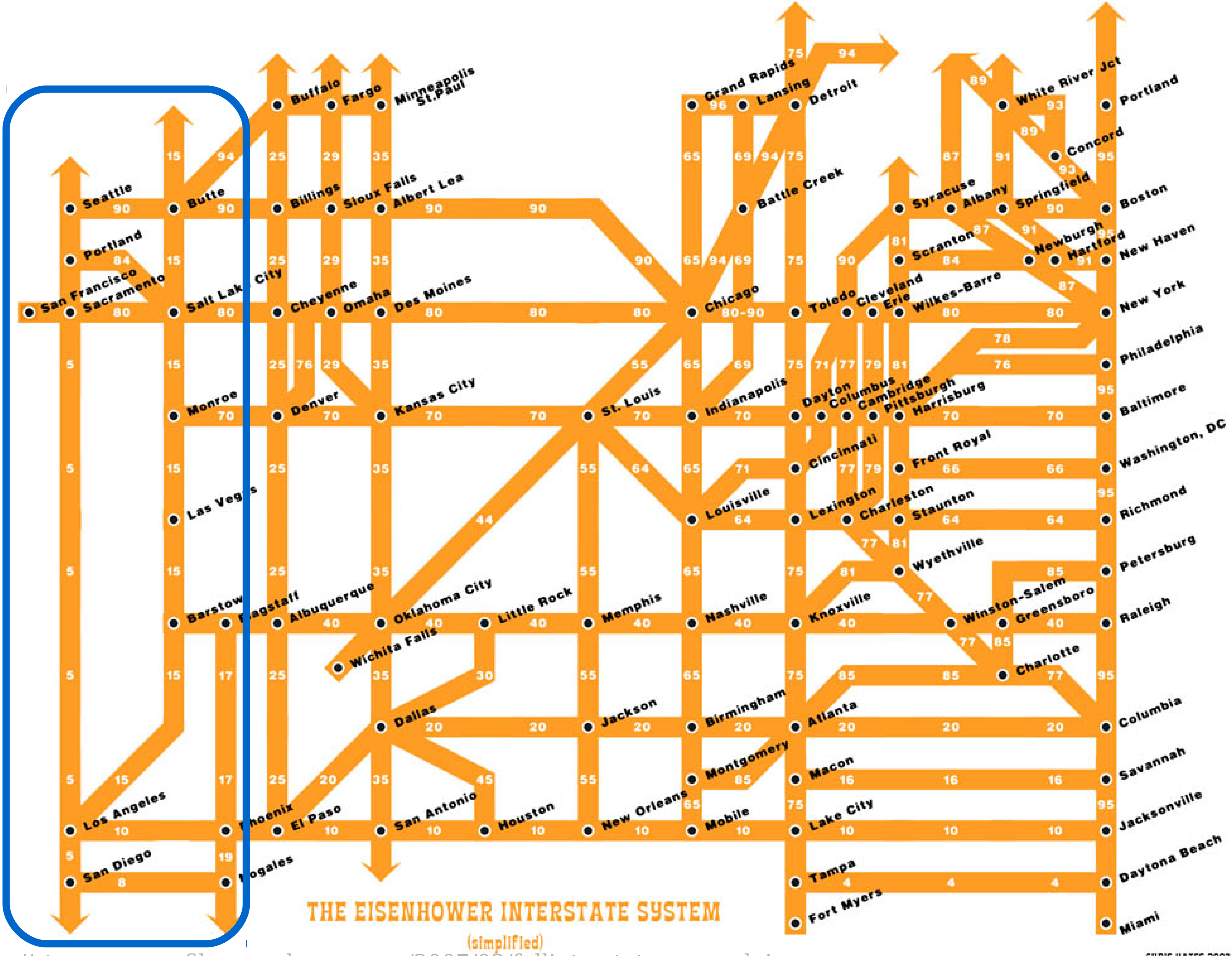
Two nodes are called ***adjacent*** if there is an edge between them.

Using our Formalisms

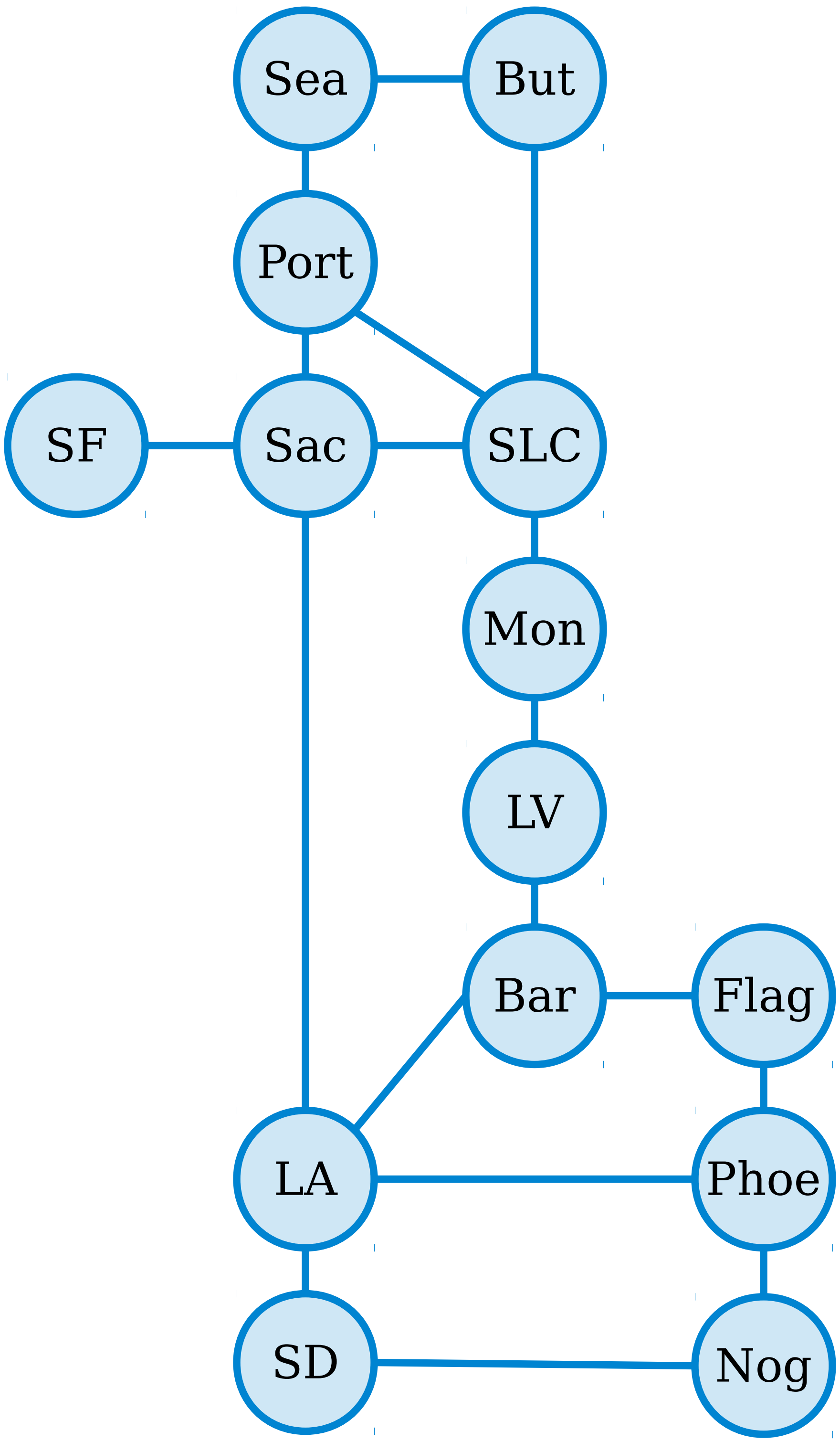
- Let $G = (V, E)$ be an (undirected) graph.
- Intuitively, two nodes are adjacent if they're linked by an edge.
- Formally speaking, we say that two nodes $u, v \in V$ are ***adjacent*** if we have $\{u, v\} \in E$.
- There isn’t an analogous notion for directed graphs. We usually just say “there’s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

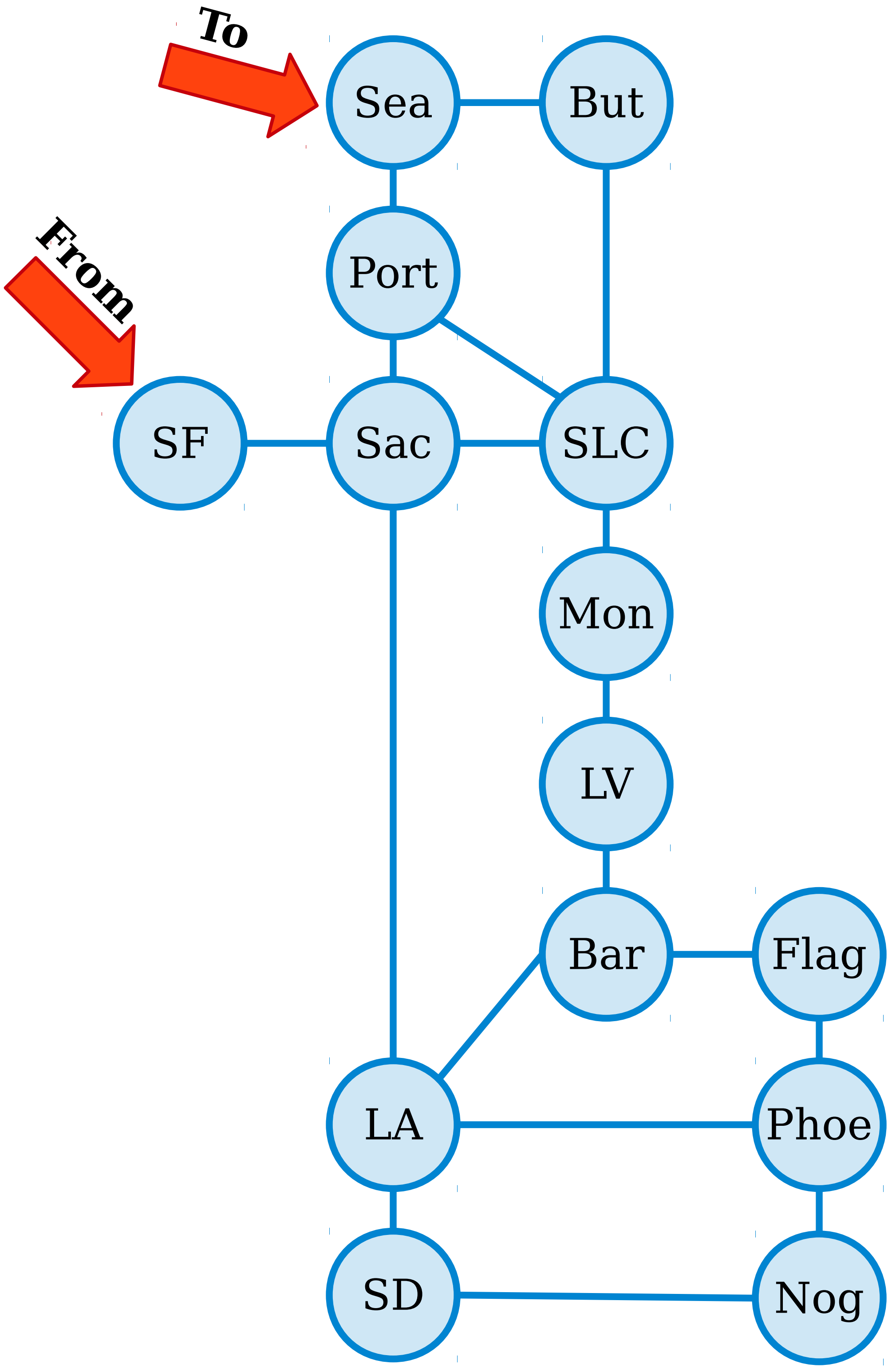


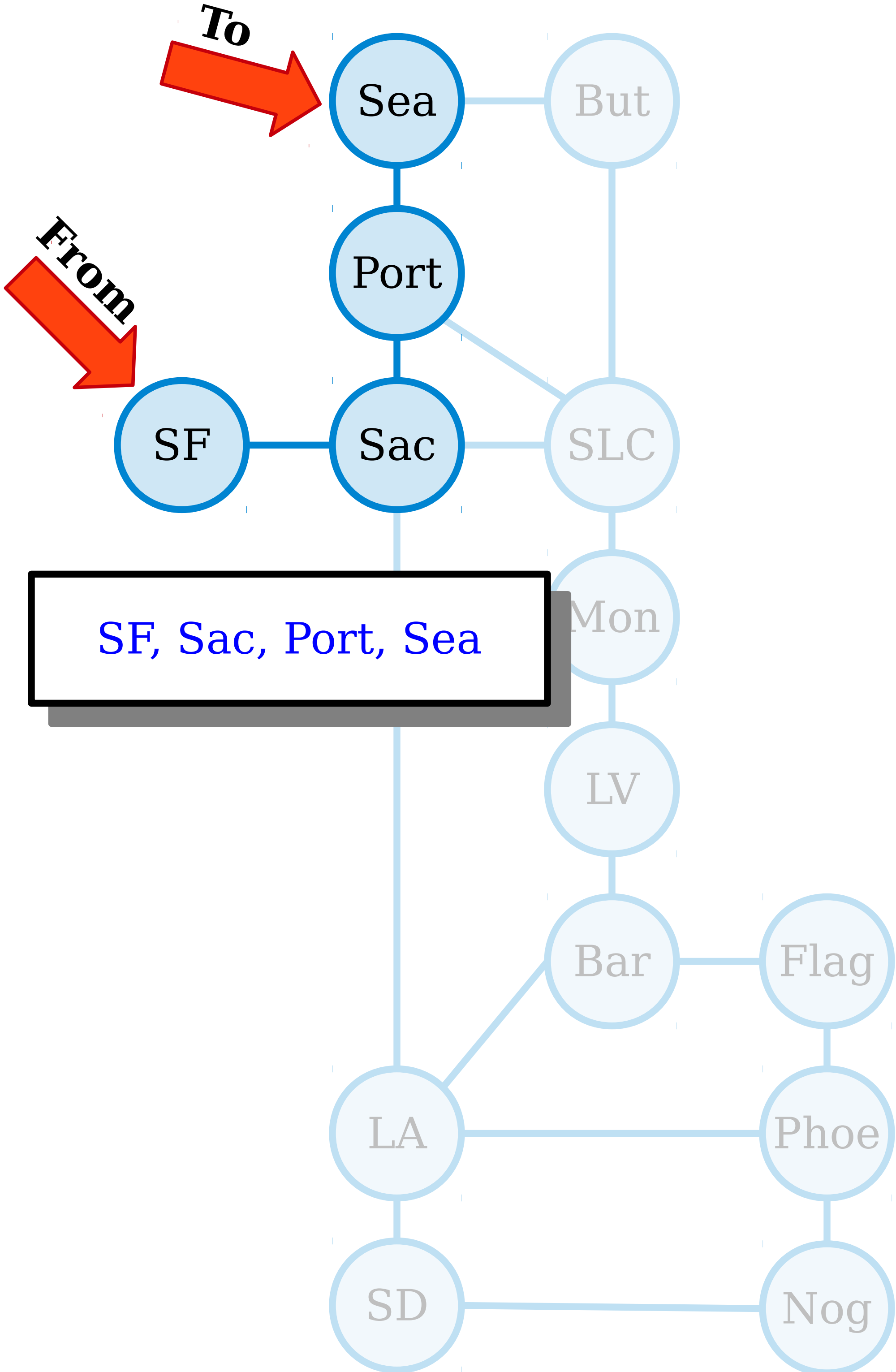
http://strangemaps.files.wordpress.com/2007/02/fullinterstatemap-web.jpg

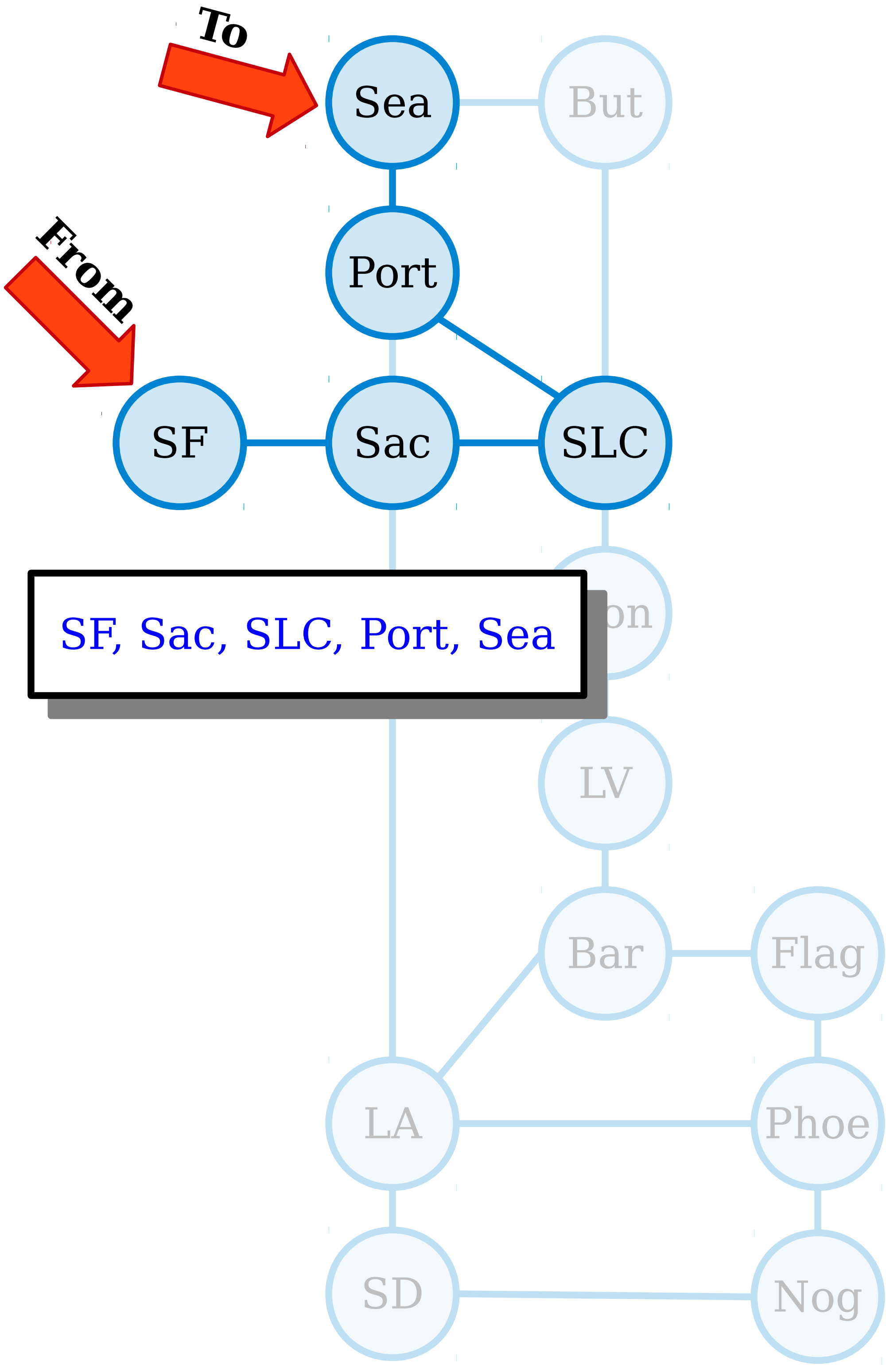


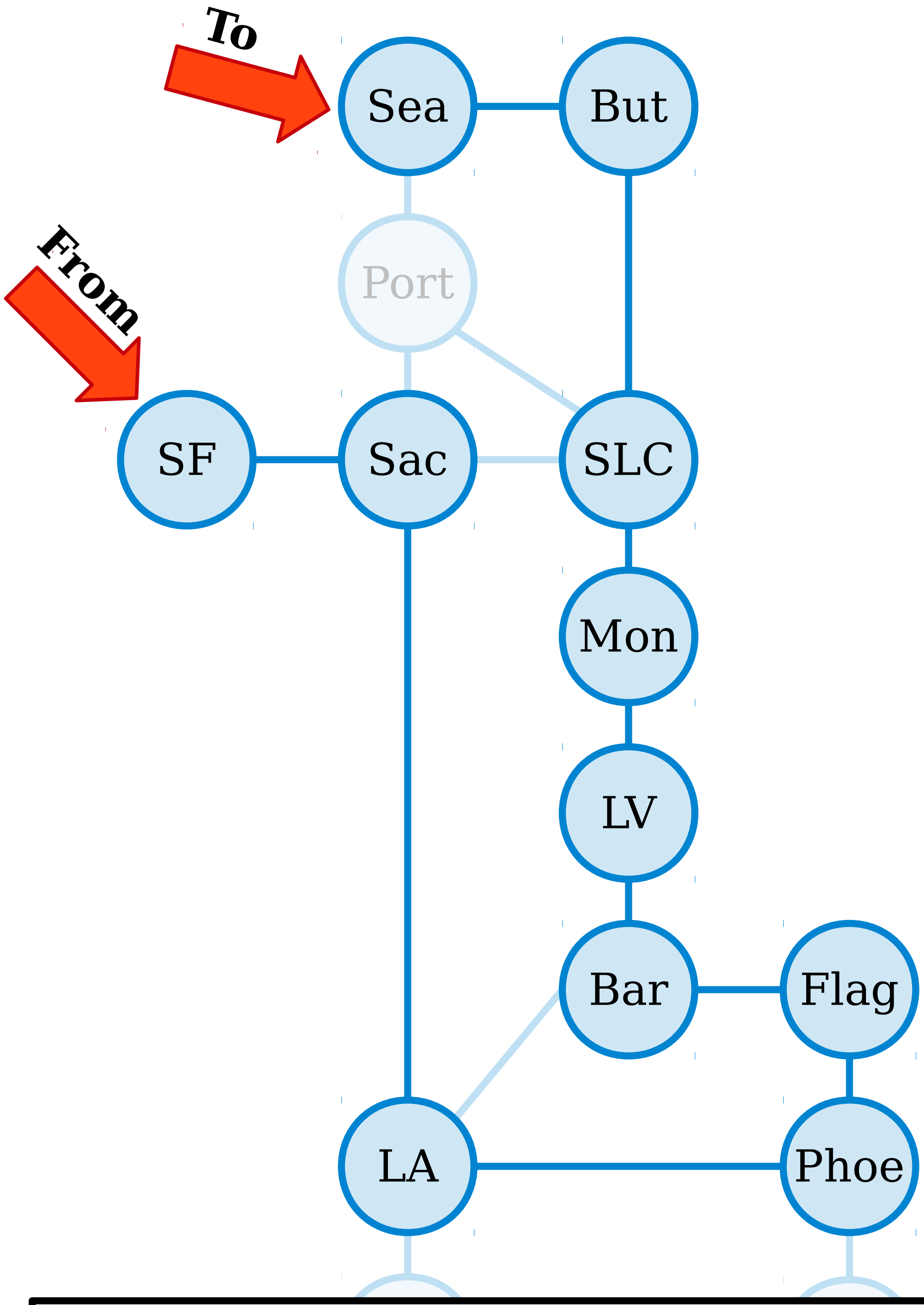
http://strangemaps.files.wordpress.com/2007/02/fullinterstatemap-web.jpg


Sea
But
Port
SF
Sac
SLC
Mon
LV
Bar
Flag
LA
Phoe
SD
Nog

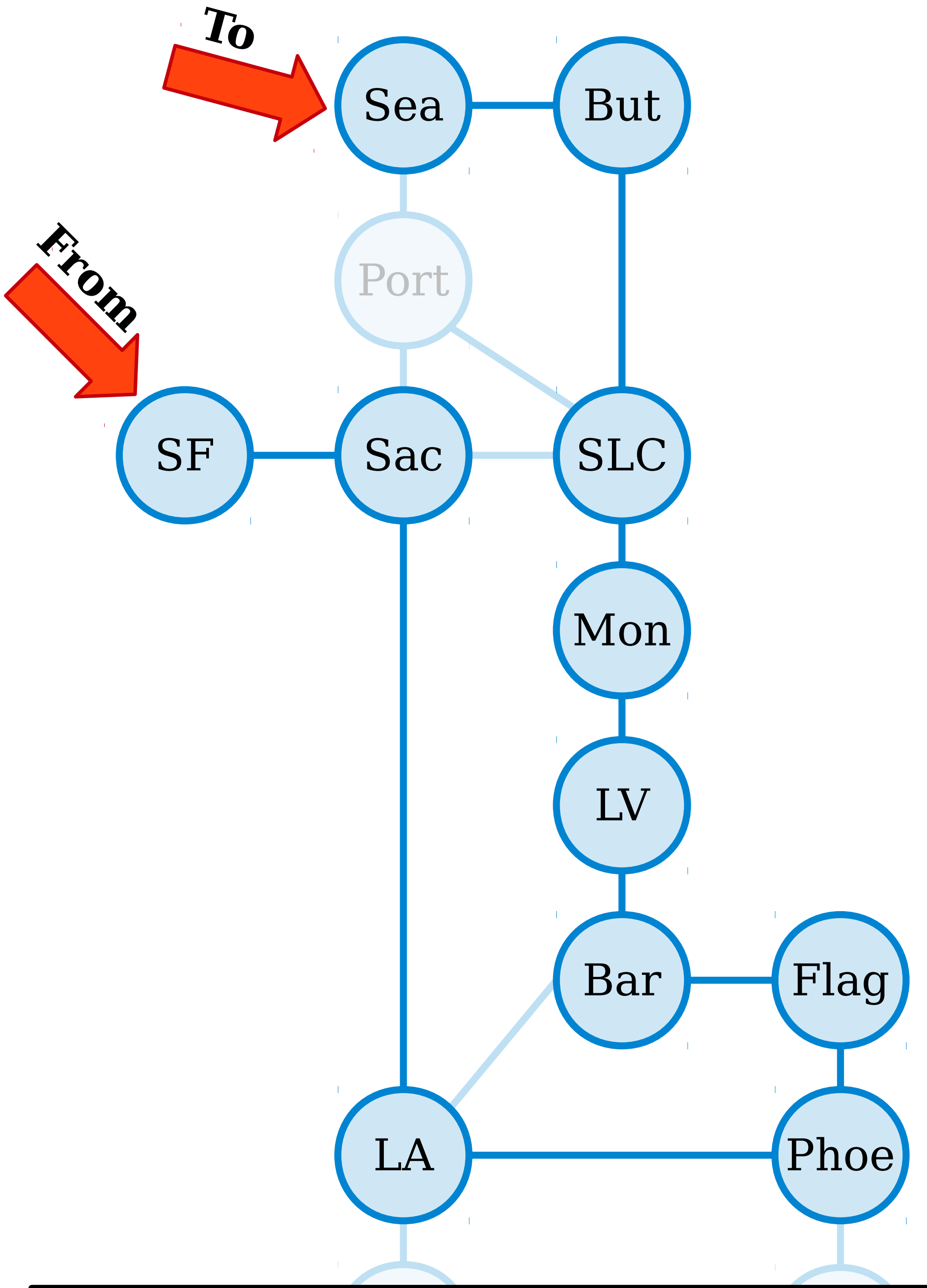

To
From
Sea
But
Port
SF
Sac
SLC
Mon
LV
Bar
Flag
LA
Phoe
SD
Nog


To
From
Sea
But
Port
SF
Sac
SLC
Mon
LV
Bar
Flag
LA
Phoe
SD
Nog
SF, Sac, Port, Sea


To
From
Sea
But
Port
SF
Sac
SLC
SF, Sac, SLC, Port, Sea
LV
Bar
Flag
LA
Phoe
SD
Nog

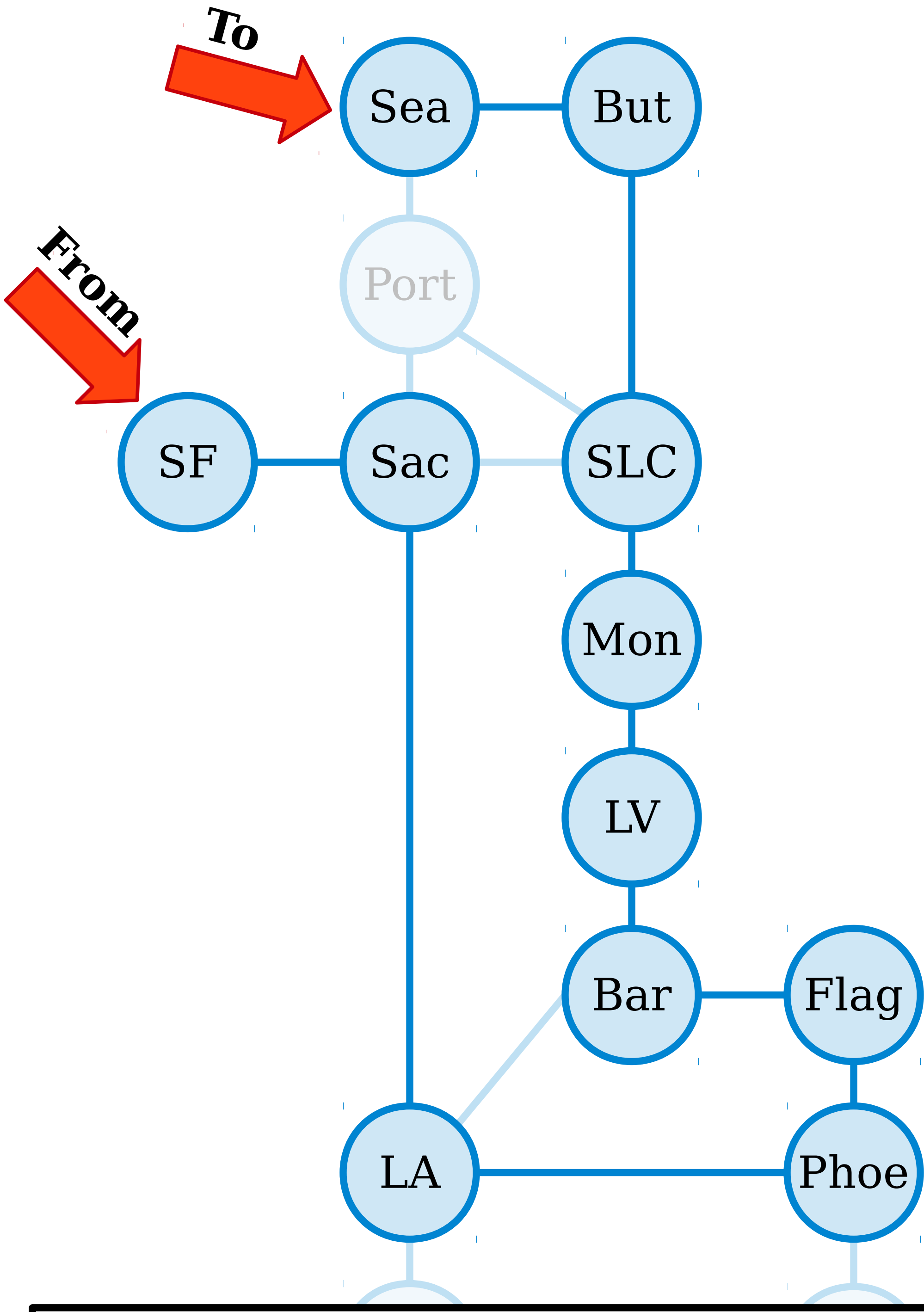


SF, Sac, LA, Phoe, Flag, Bar, LV, Mon, SLC, But, Sea



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

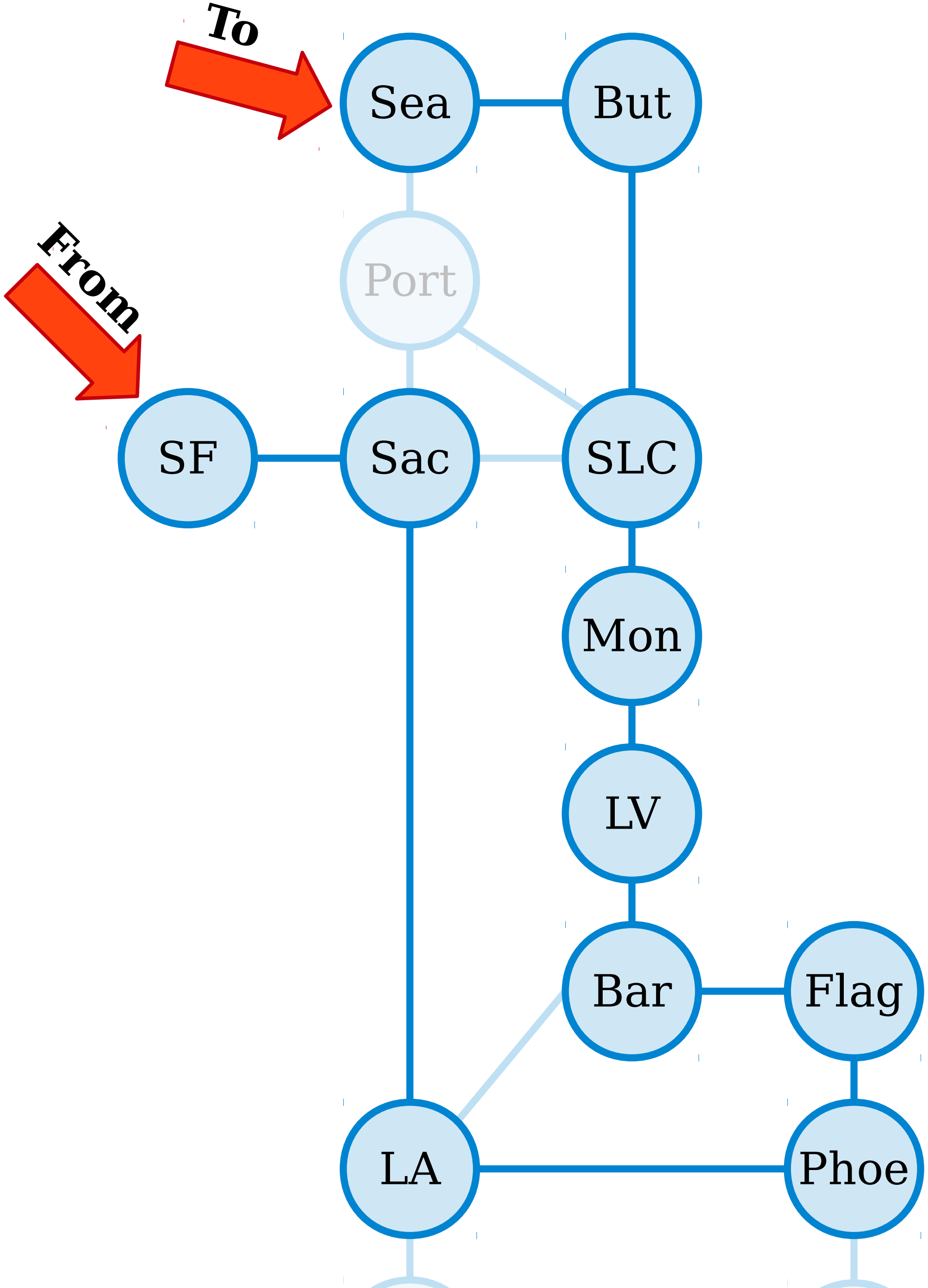
SF, Sac, LA, Phoe, Flag, Bar, LV, Mon, SLC, But, Sea



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

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

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

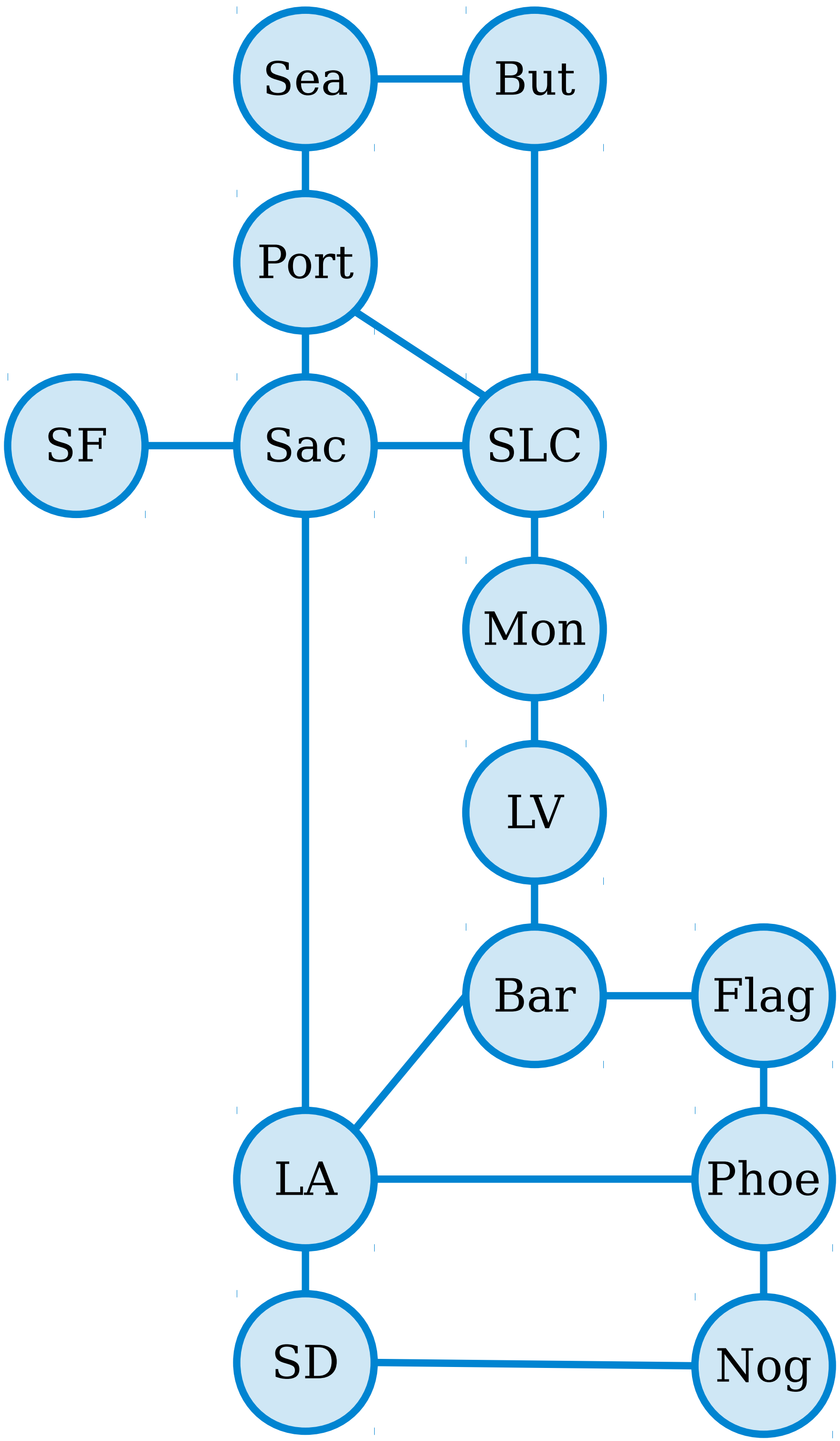


A ***walk*** in a graph $G = (V, E)$ is a sequence of one or more nodes $v_1, v_2, v_3, \ldots, v_n$ such that any two consecutive nodes in the sequence are adjacent.

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

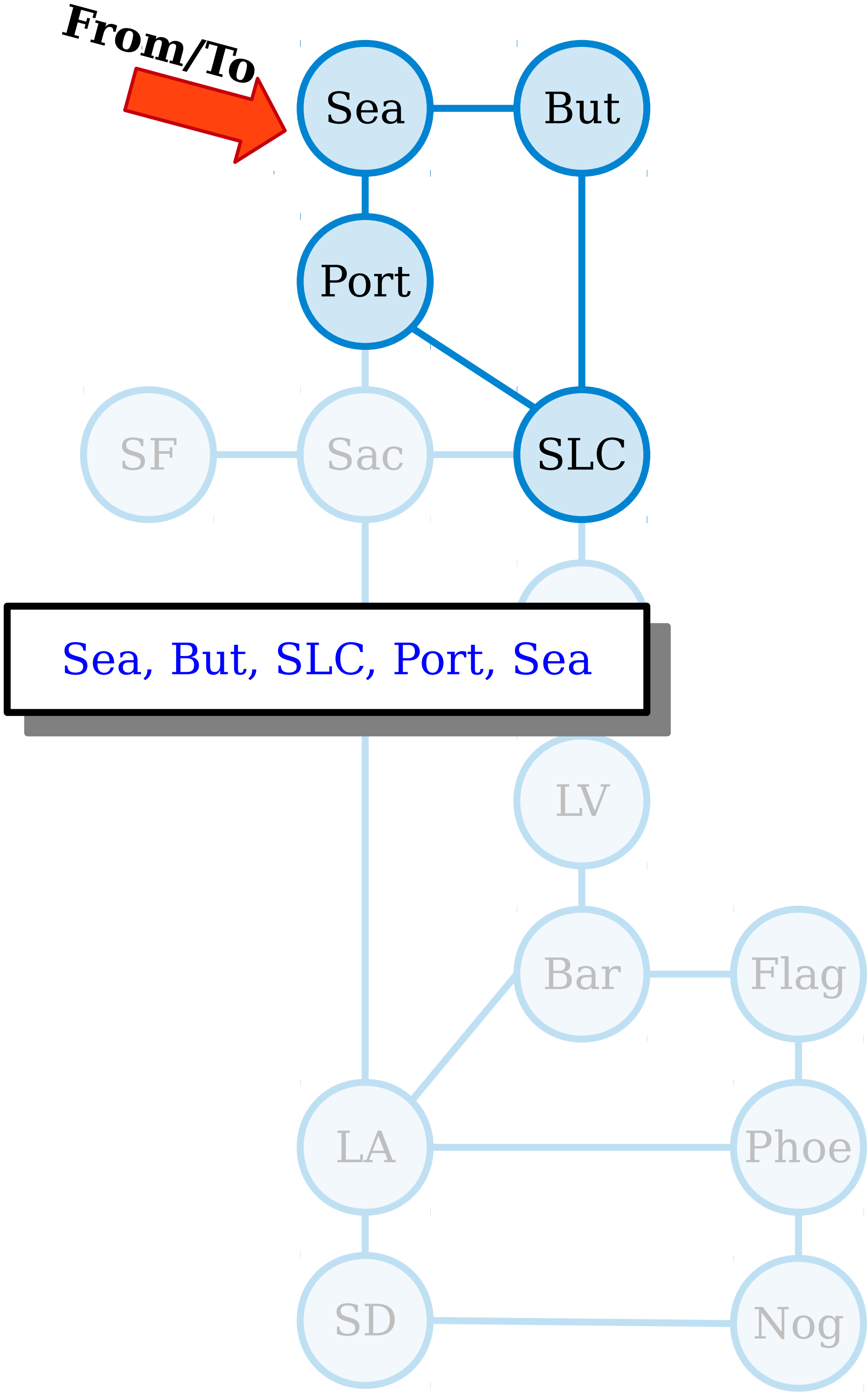
(This walk has length 10, but visits 11 cities.)

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



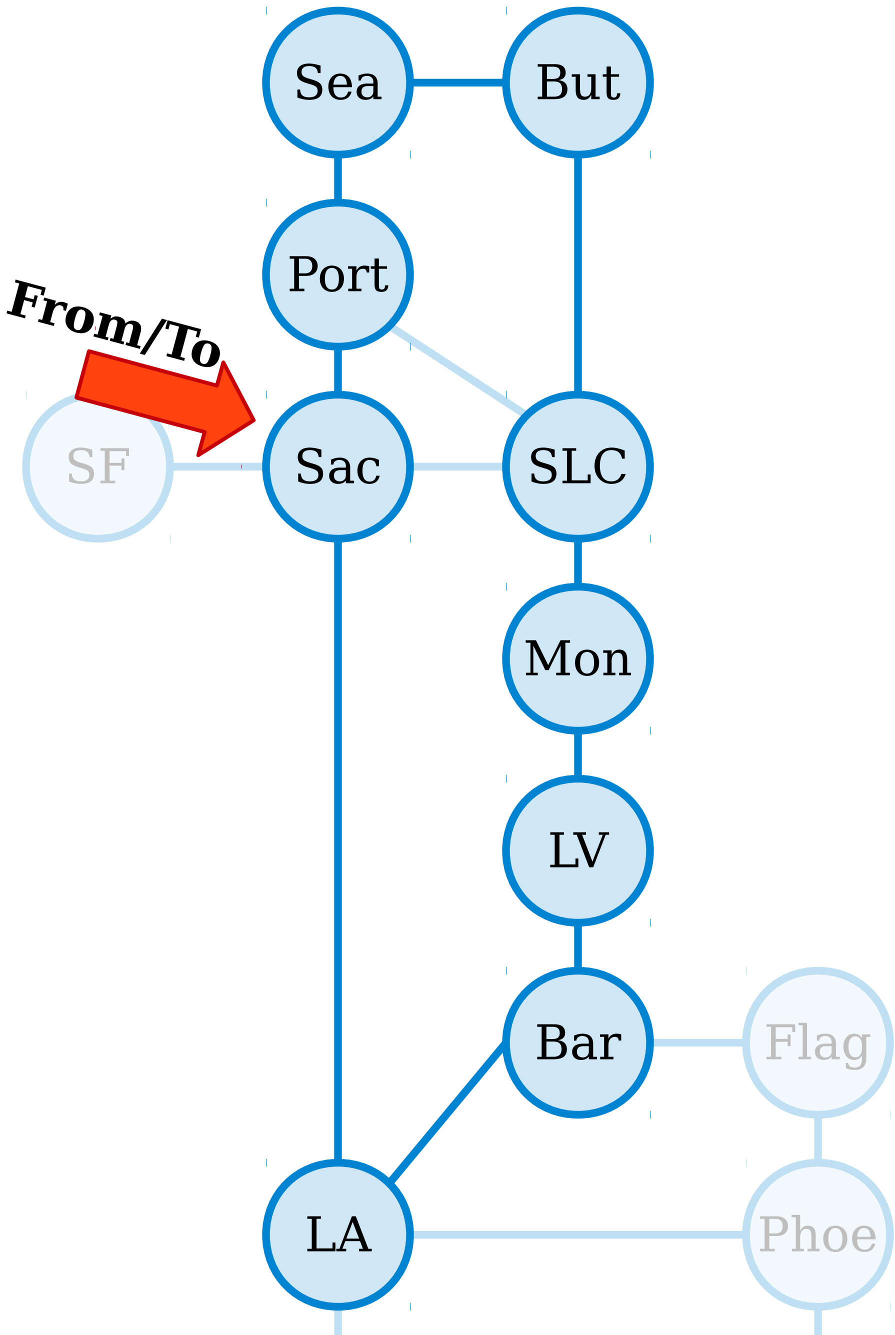
A ***walk*** in a graph $G = (V, E)$ is a sequence of one or more nodes $v_1, v_2, v_3, \ldots, v_n$ such that any two consecutive nodes in the sequence are adjacent.

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



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

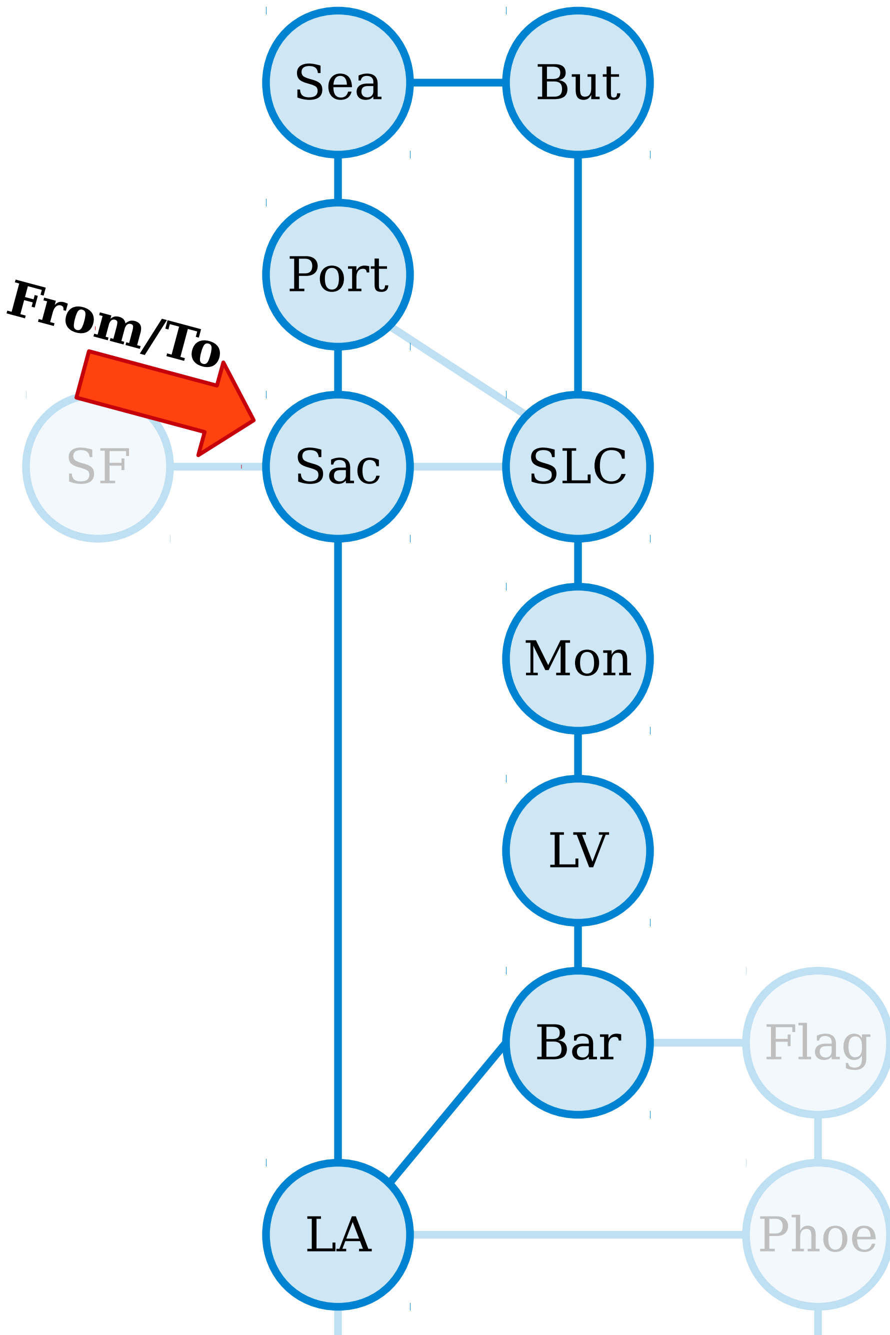
The ***length*** of the walk $v_1, \ldots, v_n$ is $n - 1$.



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

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

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

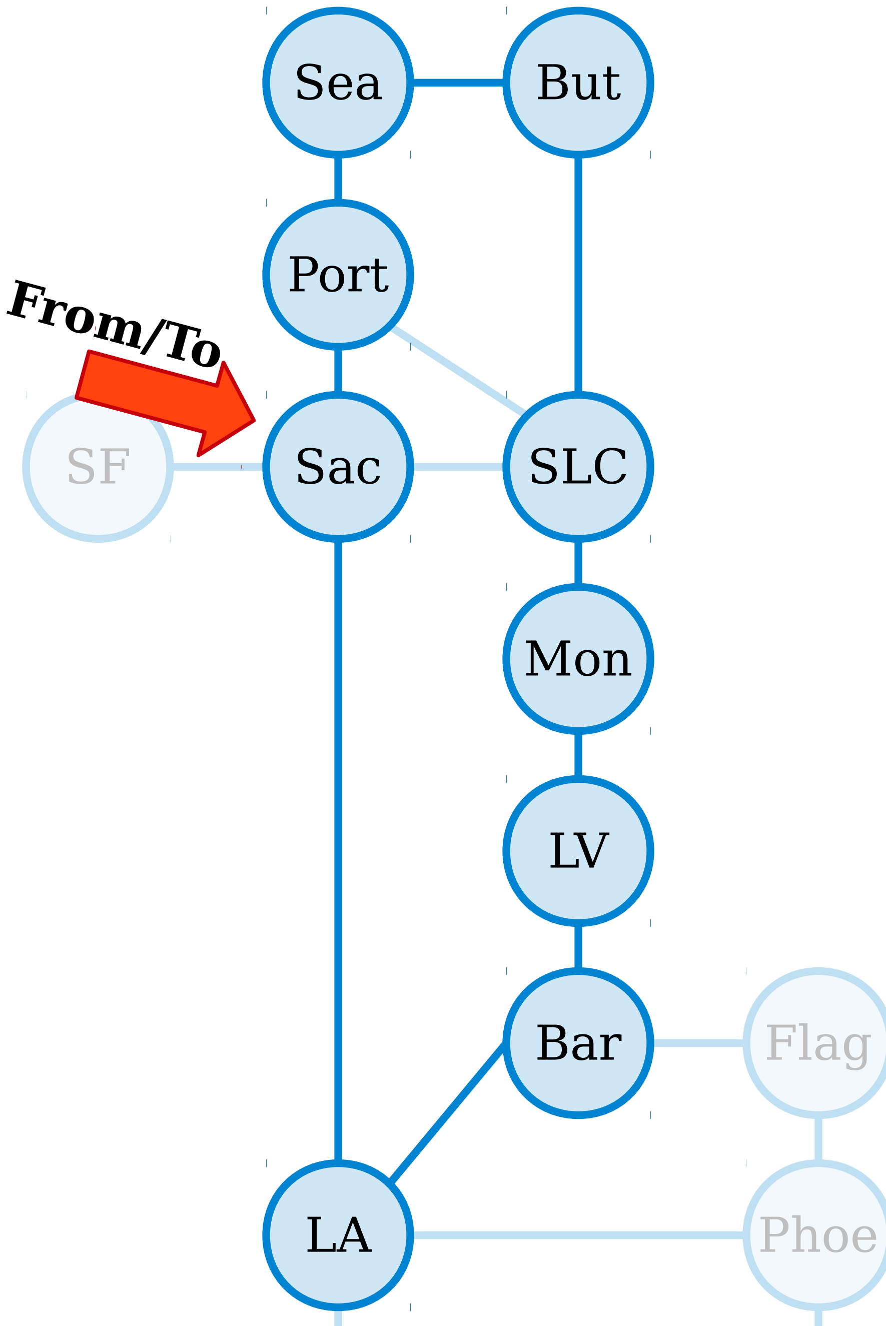


A ***walk*** in a graph $G = (V, E)$ is a sequence of one or more nodes $v_1, v_2, v_3, \ldots, v_n$ such that any two consecutive nodes in the sequence are adjacent.

The ***length*** of the walk $v_1, \ldots, v_n$ 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.)

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



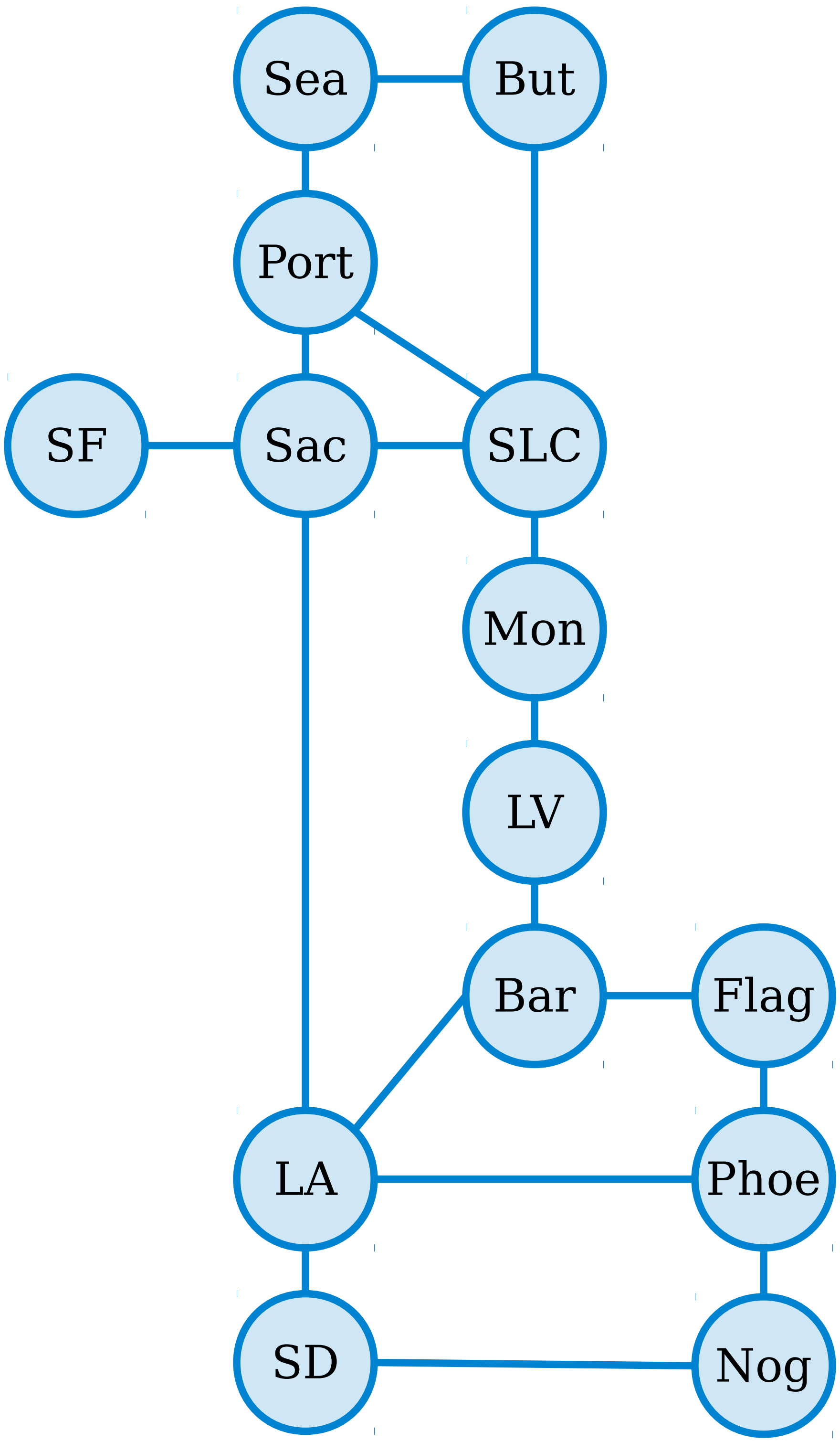
A ***walk*** in a graph $G = (V, E)$ is a sequence of one or more nodes $v_1, v_2, v_3, \ldots, v_n$ such that any two consecutive nodes in the sequence are adjacent.

The ***length*** of the walk $v_1, \ldots, v_n$ 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.)

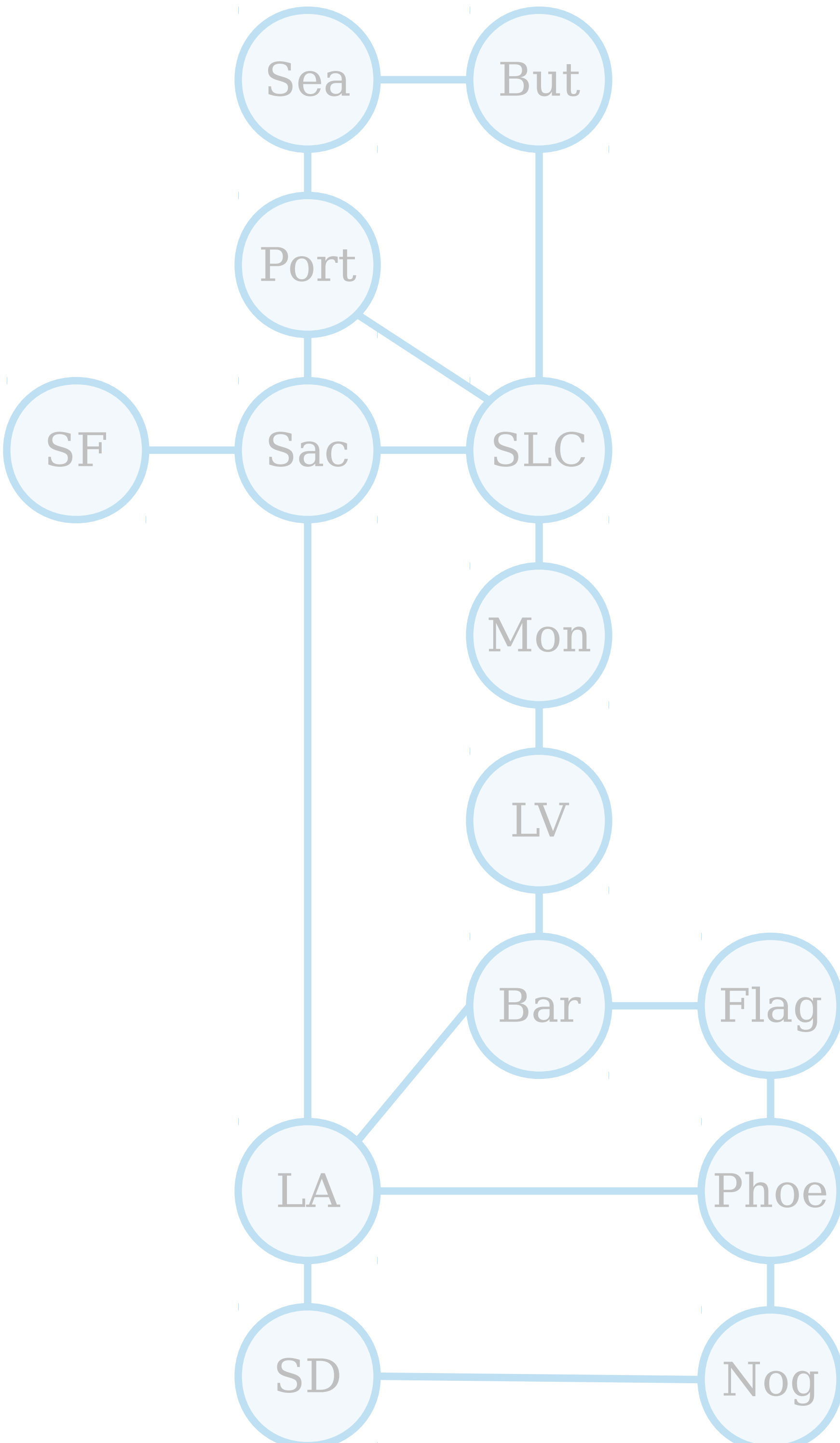
Sac, Port, Sea, But, SLC, Mon, LV, Bar, LA, Sac



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

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

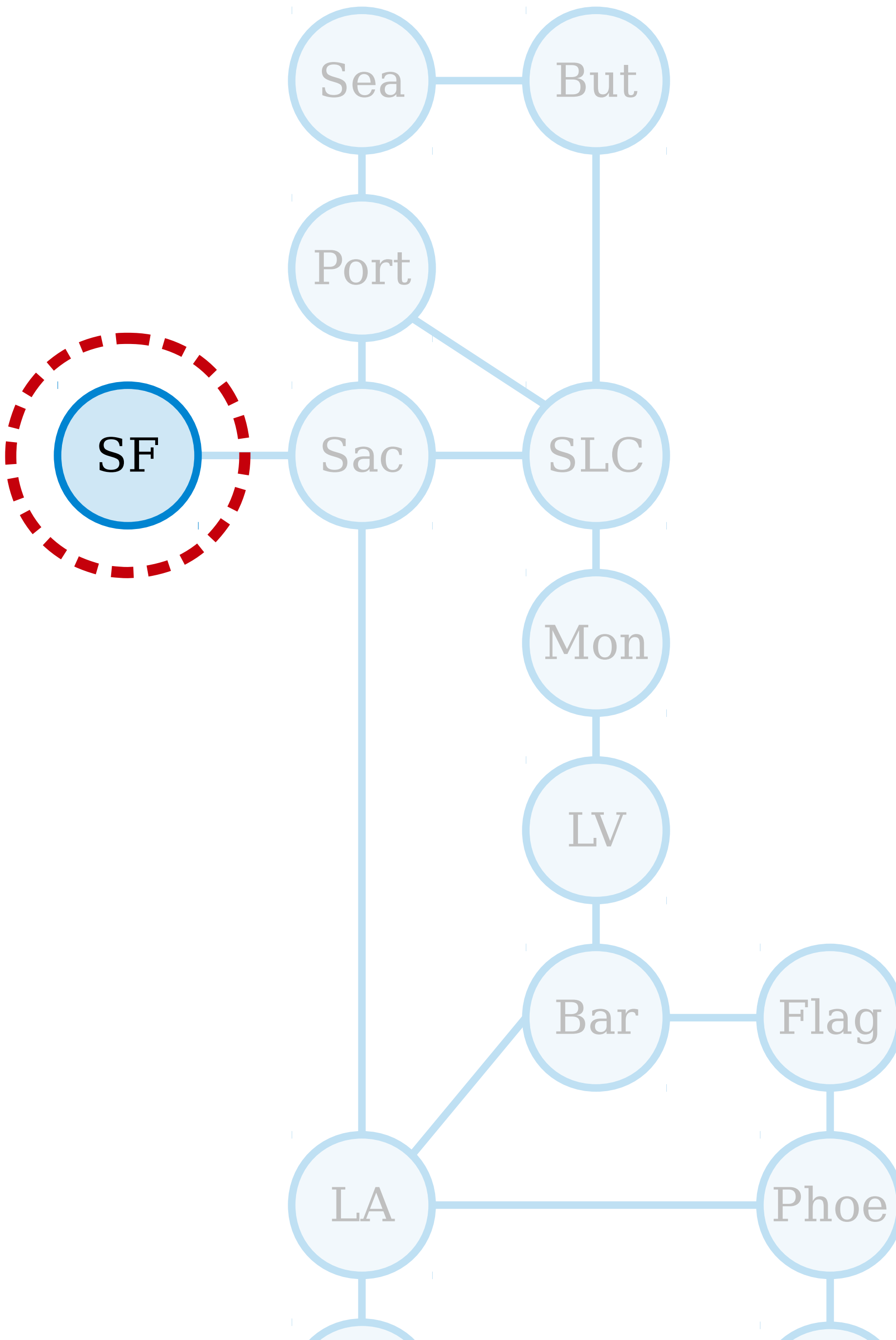
A ***closed walk*** in a graph is a walk from a node back to itself. (By convention, a closed walk cannot have length zero.)



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

The ***length*** of the walk $v_1, \ldots, v_n$ 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.)

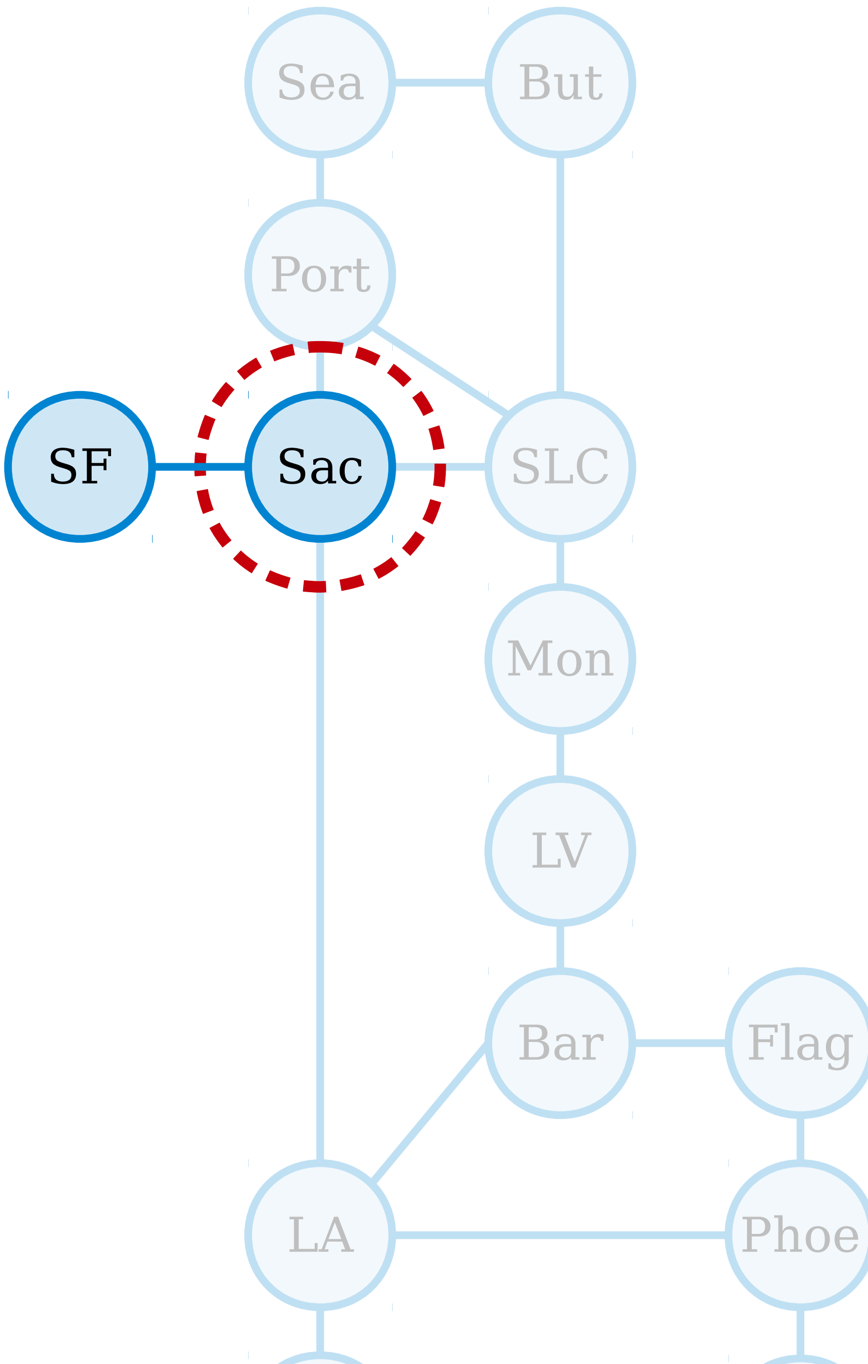


A ***walk*** in a graph $G = (V, E)$ is a sequence of one or more nodes $v_1, v_2, v_3, \ldots, v_n$ such that any two consecutive nodes in the sequence are adjacent.

The ***length*** of the walk $v_1, \ldots, v_n$ 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

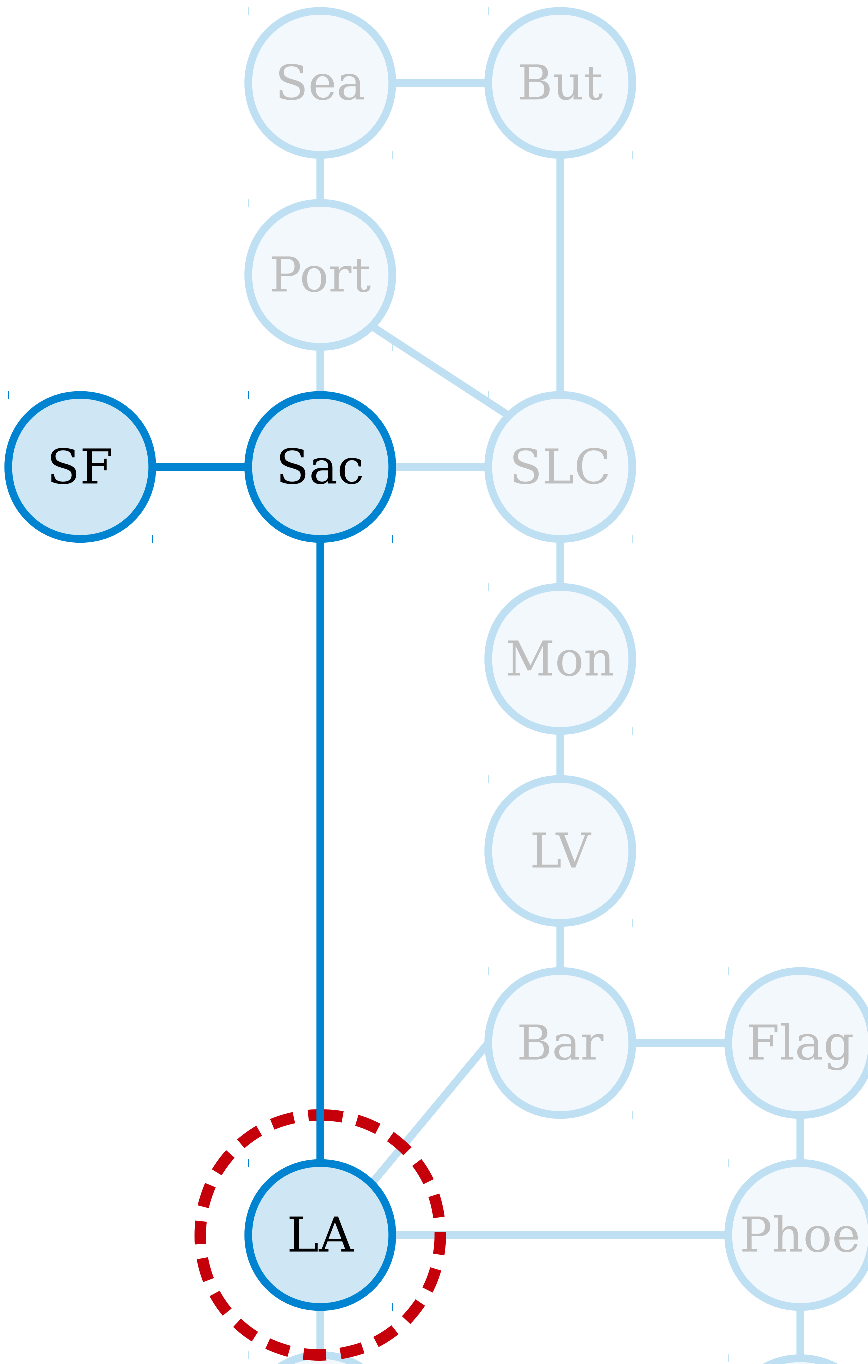


A ***walk*** in a graph $G = (V, E)$ is a sequence of one or more nodes $v_1, v_2, v_3, \ldots, v_n$ such that any two consecutive nodes in the sequence are adjacent.

The ***length*** of the walk $v_1, \ldots, v_n$ 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

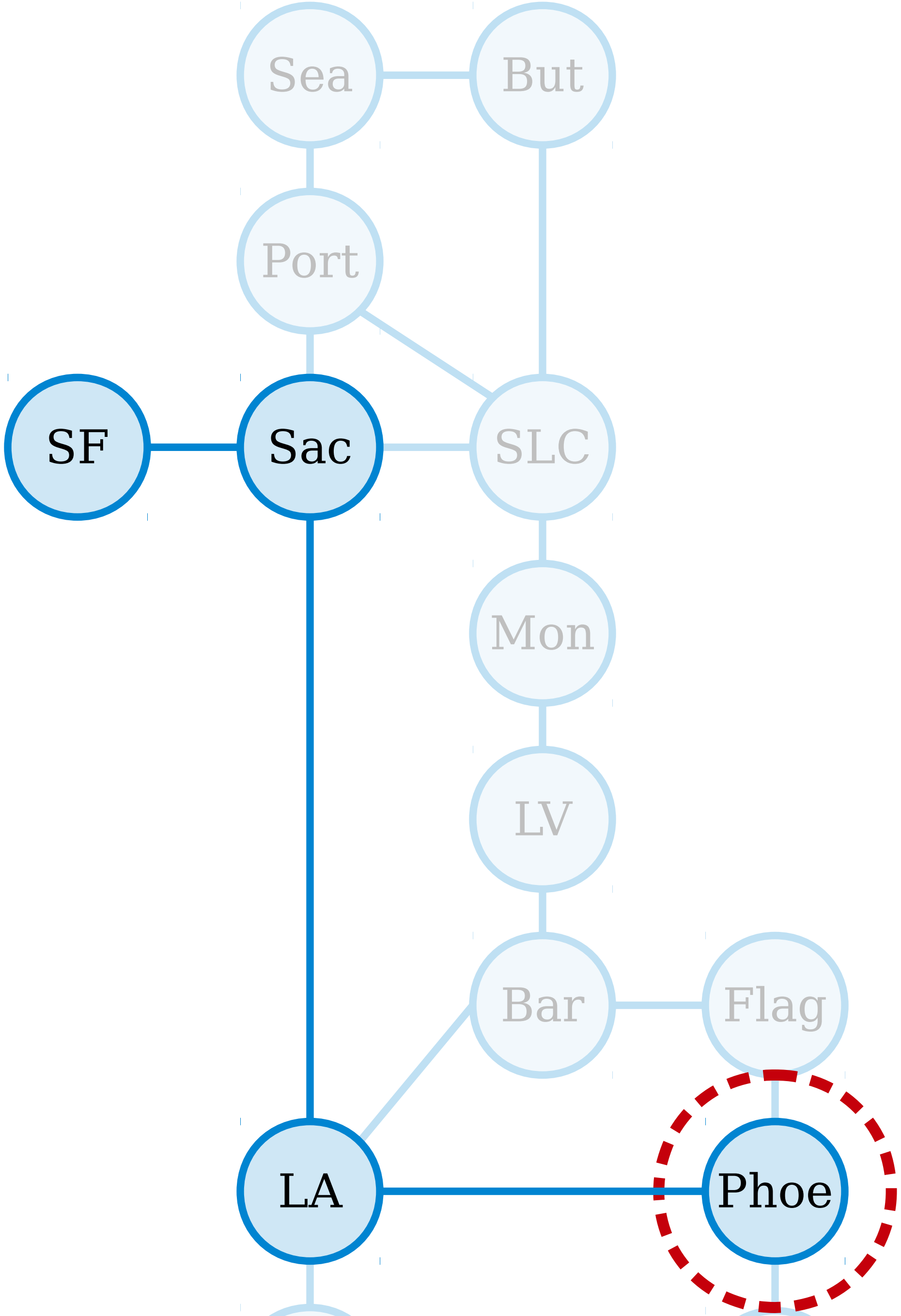


A ***walk*** in a graph $G = (V, E)$ is a sequence of one or more nodes $v_1, v_2, v_3, \ldots, v_n$ such that any two consecutive nodes in the sequence are adjacent.

The ***length*** of the walk $v_1, \ldots, v_n$ 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

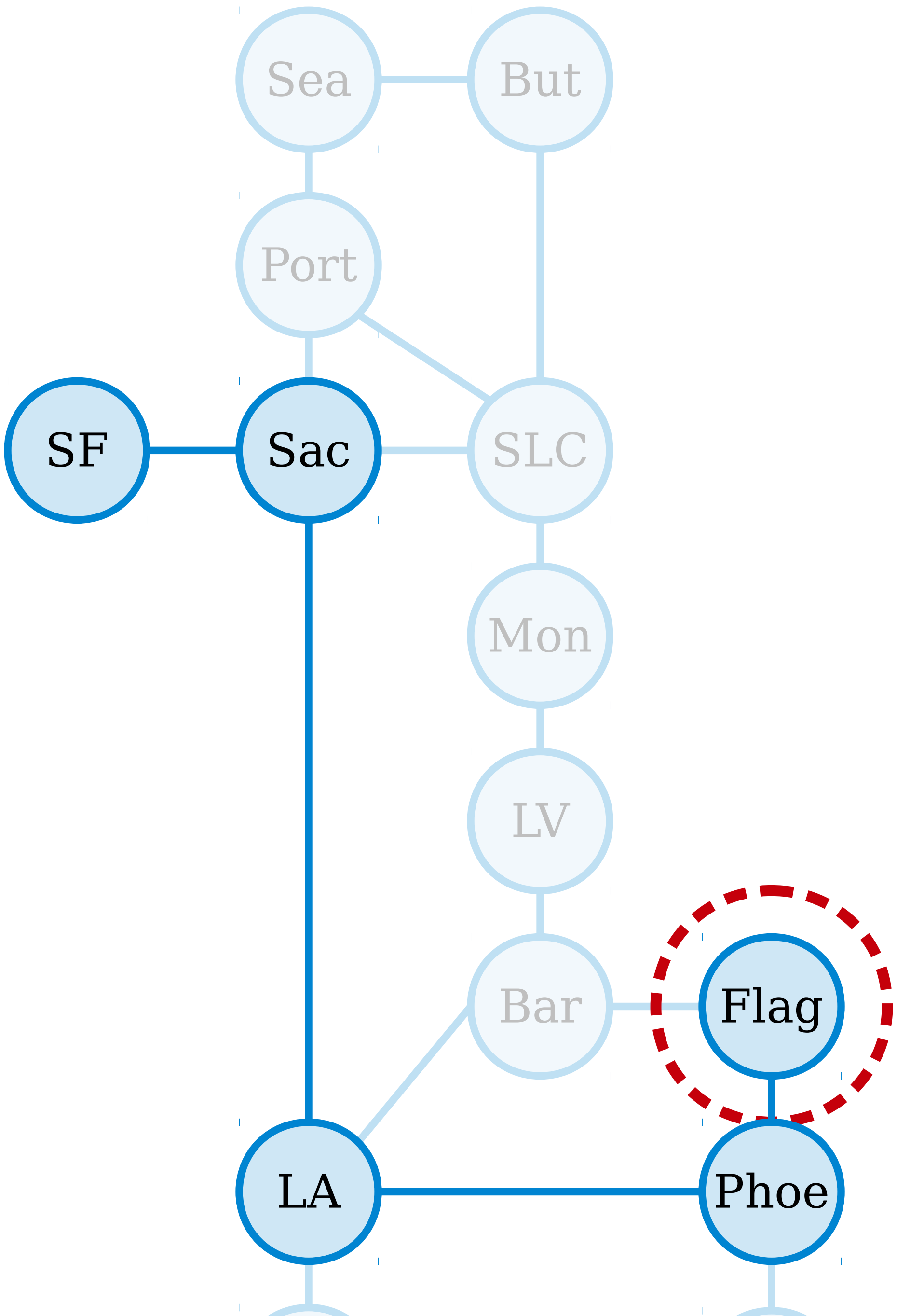


A ***walk*** in a graph $G = (V, E)$ is a sequence of one or more nodes $v_1, v_2, v_3, \ldots, v_n$ such that any two consecutive nodes in the sequence are adjacent.

The ***length*** of the walk $v_1, \ldots, v_n$ 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

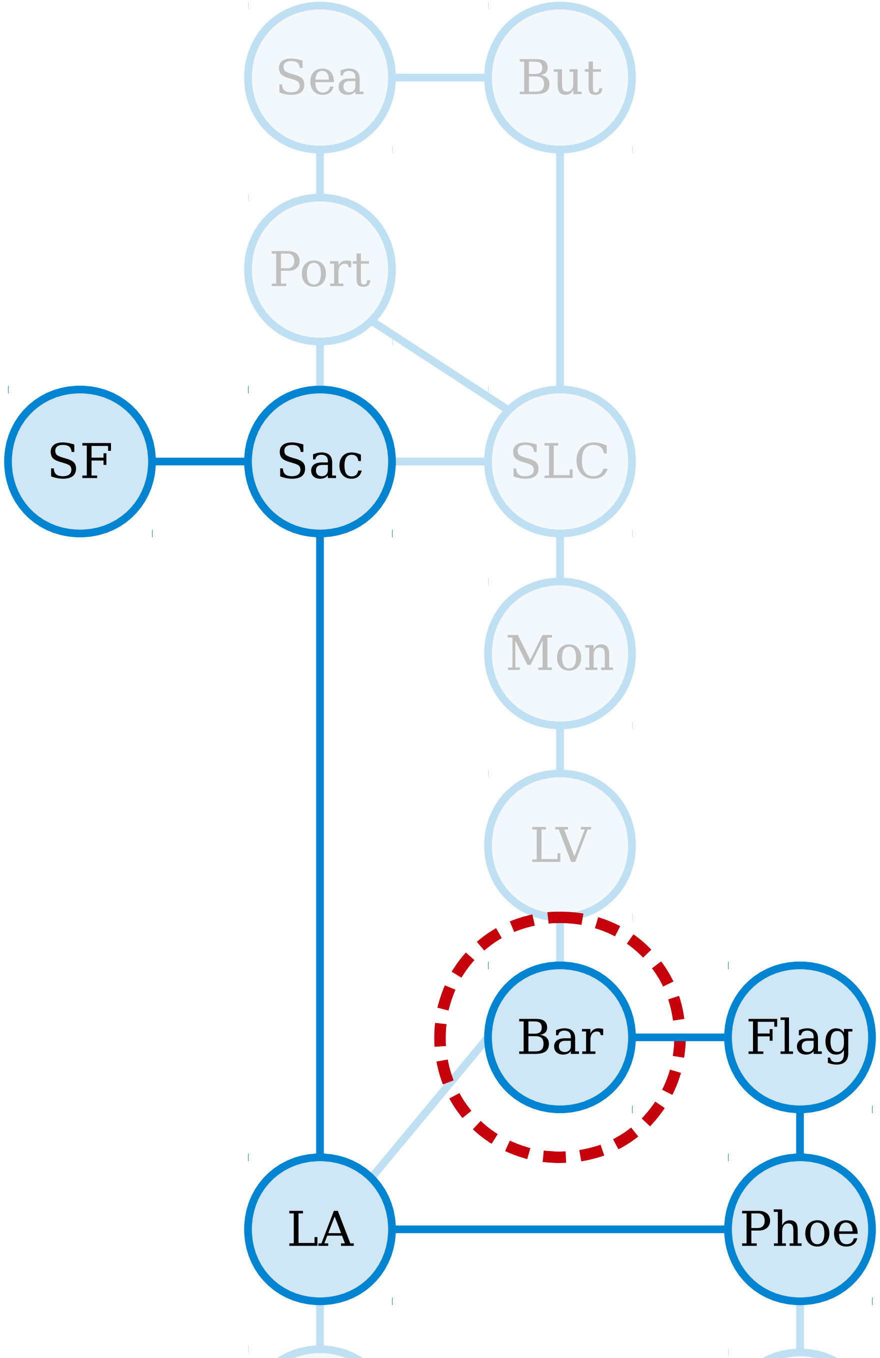


A ***walk*** in a graph $G = (V, E)$ is a sequence of one or more nodes $v_1, v_2, v_3, \ldots, v_n$ such that any two consecutive nodes in the sequence are adjacent.

The ***length*** of the walk $v_1, \ldots, v_n$ 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

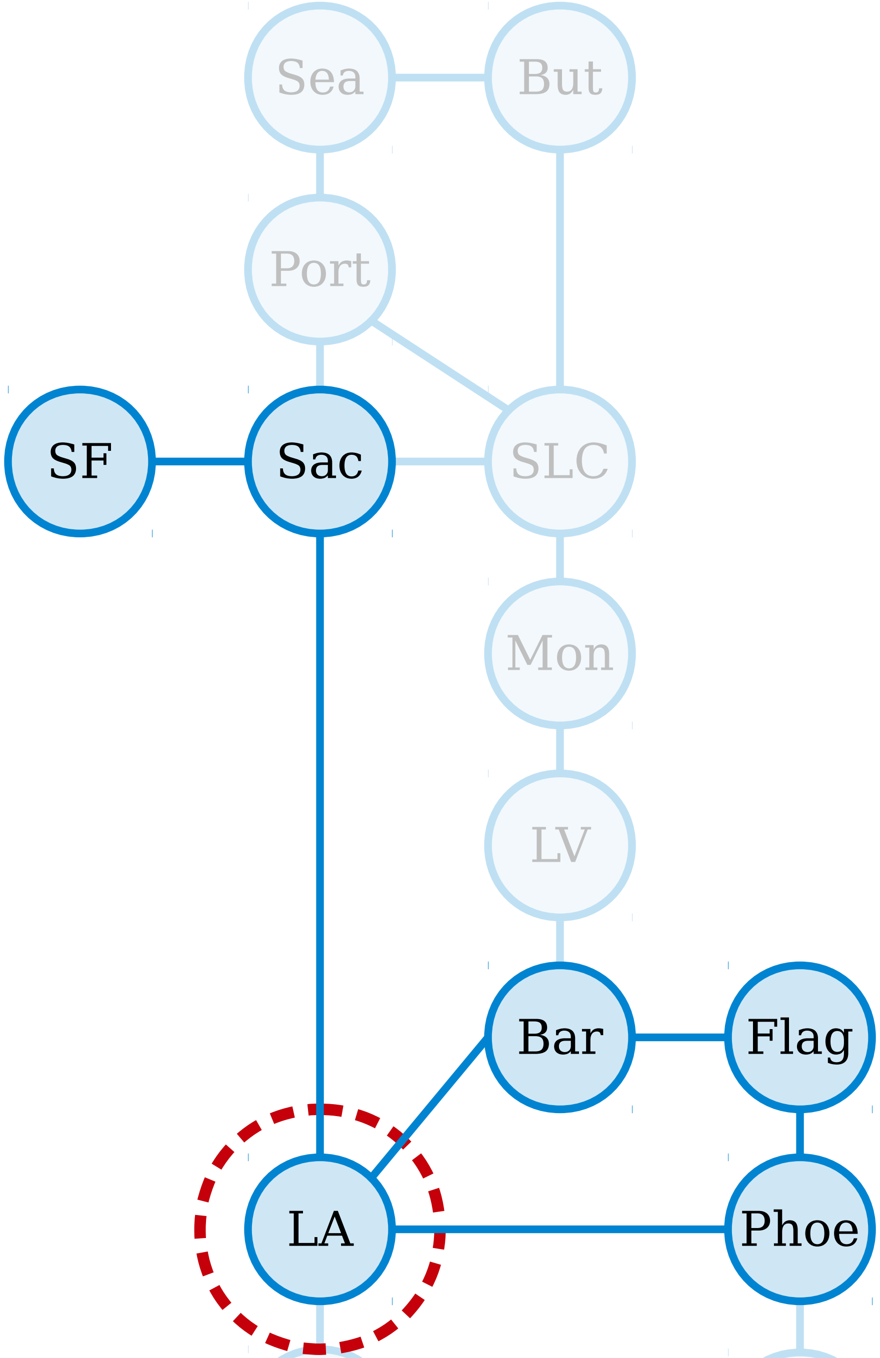


A ***walk*** in a graph $G = (V, E)$ is a sequence of one or more nodes $v_1, v_2, v_3, \ldots, v_n$ such that any two consecutive nodes in the sequence are adjacent.

The ***length*** of the walk $v_1, \ldots, v_n$ 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

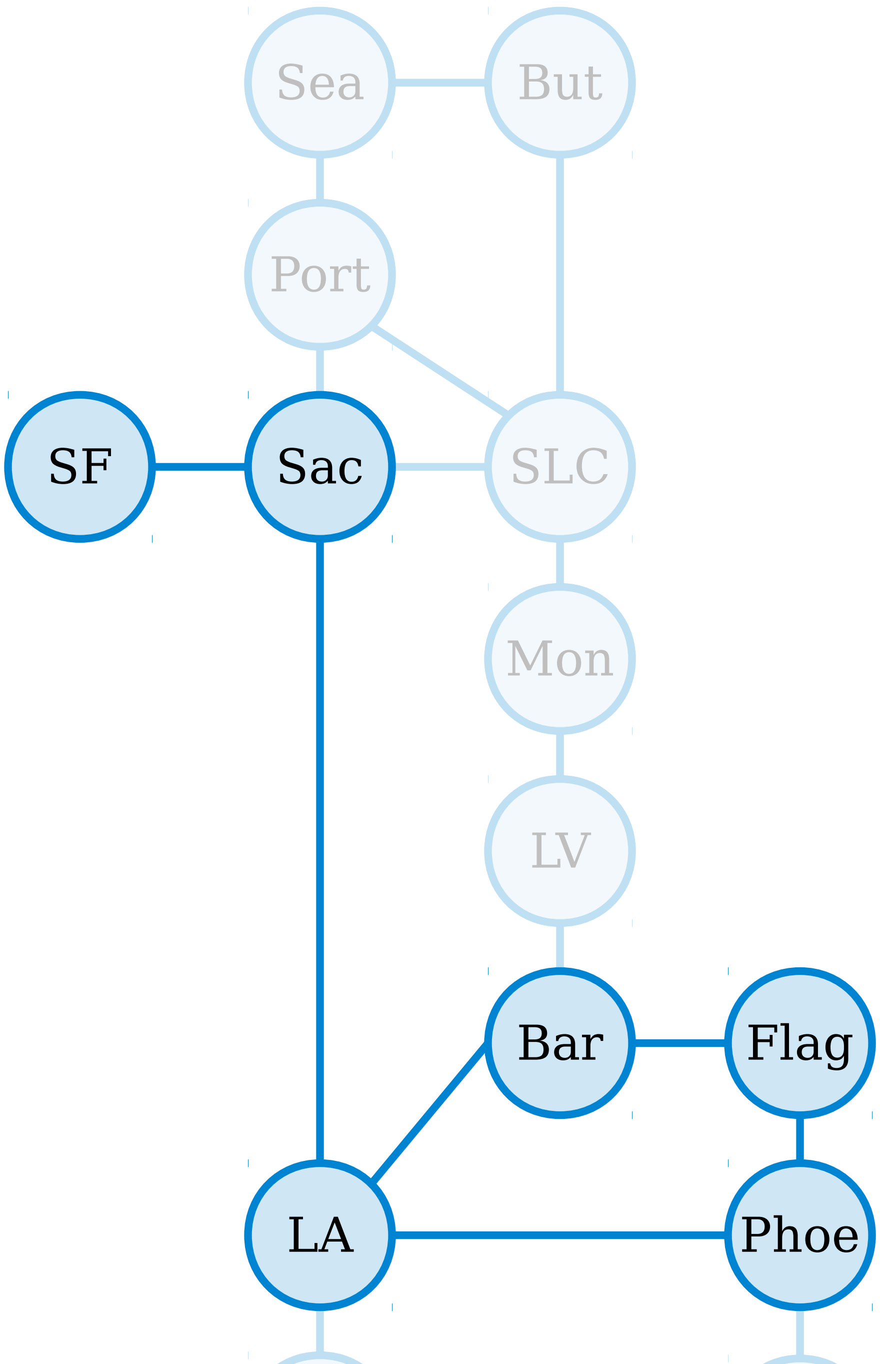


A ***walk*** in a graph $G = (V, E)$ is a sequence of one or more nodes $v_1, v_2, v_3, \ldots, v_n$ such that any two consecutive nodes in the sequence are adjacent.

The ***length*** of the walk $v_1, \ldots, v_n$ 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

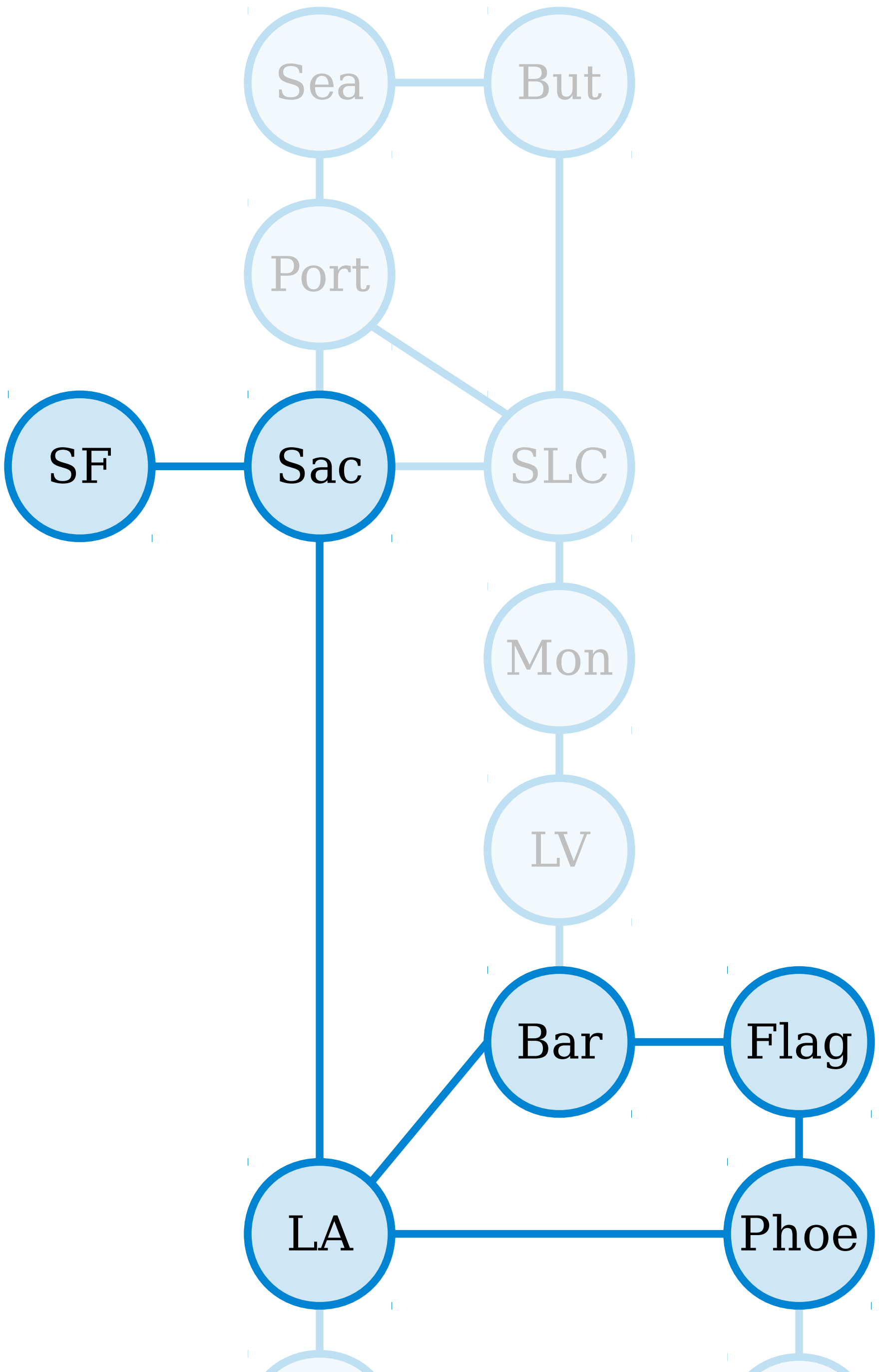


A ***walk*** in a graph $G = (V, E)$ is a sequence of one or more nodes $v_1, v_2, v_3, \ldots, v_n$ such that any two consecutive nodes in the sequence are adjacent.

The ***length*** of the walk $v_1, \ldots, v_n$ 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



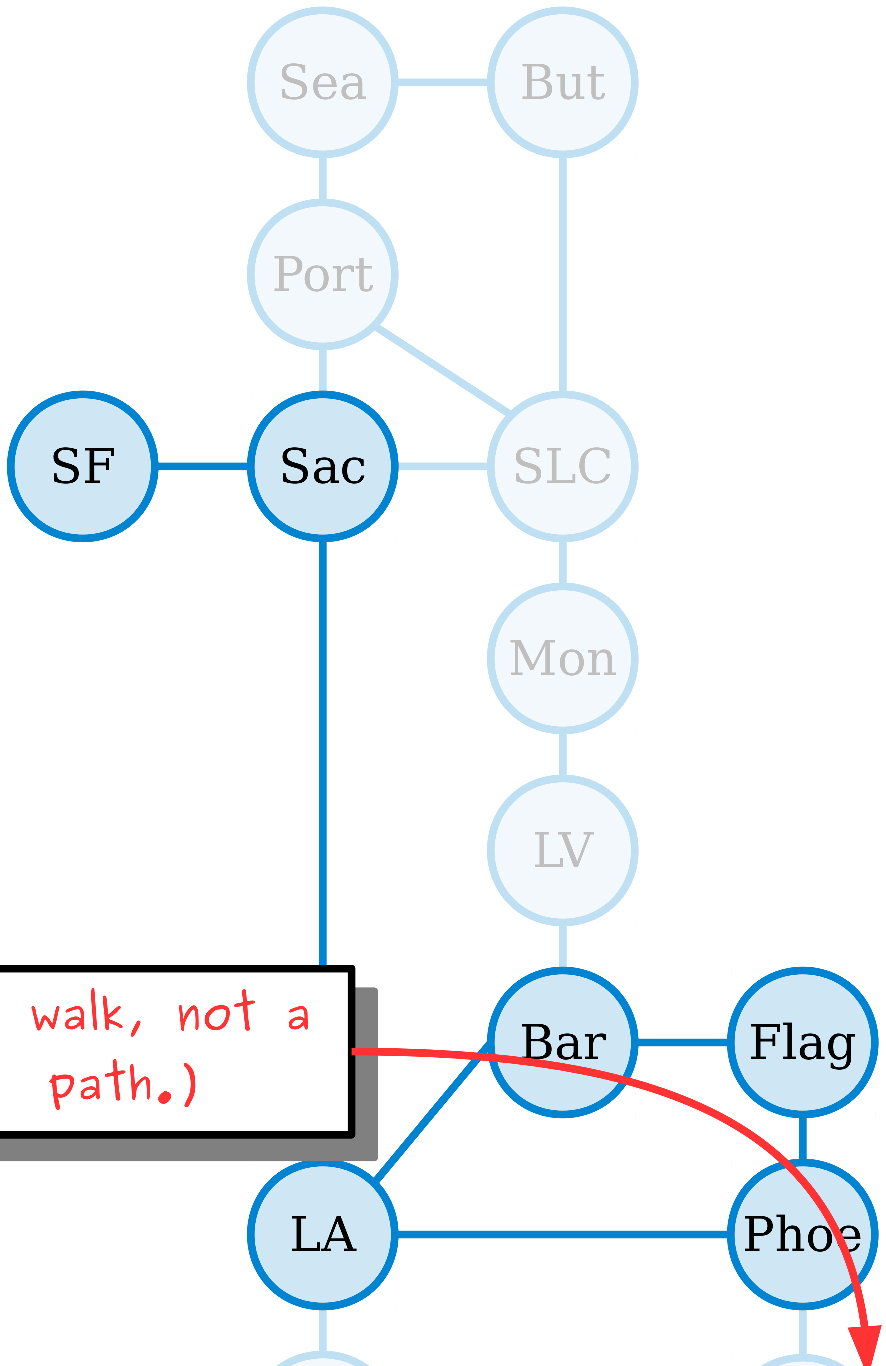
A ***walk*** in a graph $G = (V, E)$ is a sequence of one or more nodes $v_1, v_2, v_3, \ldots, v_n$ such that any two consecutive nodes in the sequence are adjacent.

The ***length*** of the walk $v_1, \ldots, v_n$ 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, 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 $v_1, v_2, v_3, \ldots, v_n$ such that any two consecutive nodes in the sequence are adjacent.

The ***length*** of the walk $v_1, \ldots, v_n$ 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.)

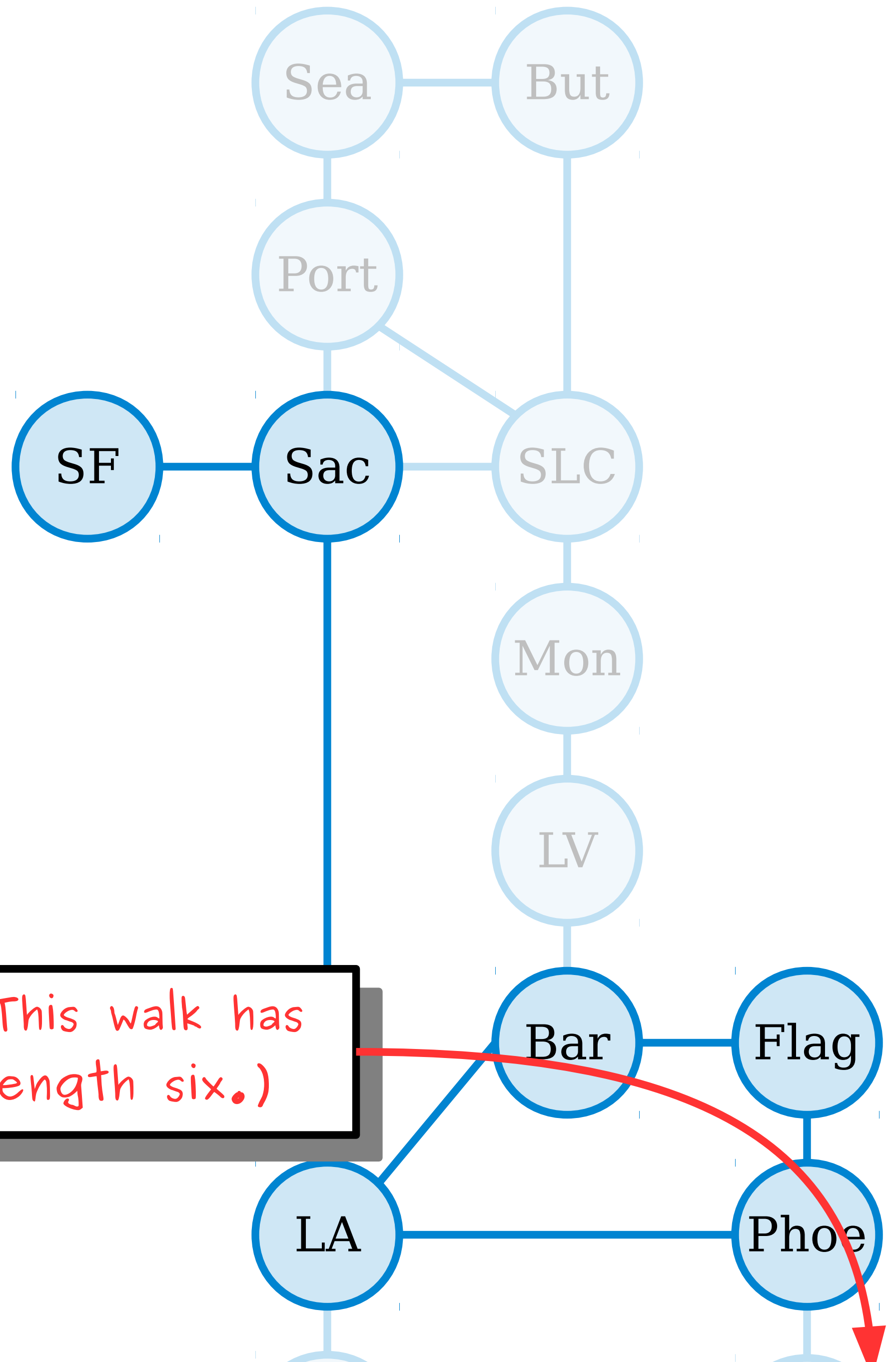
A ***path*** in a graph is walk that does not repeat any nodes.

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

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

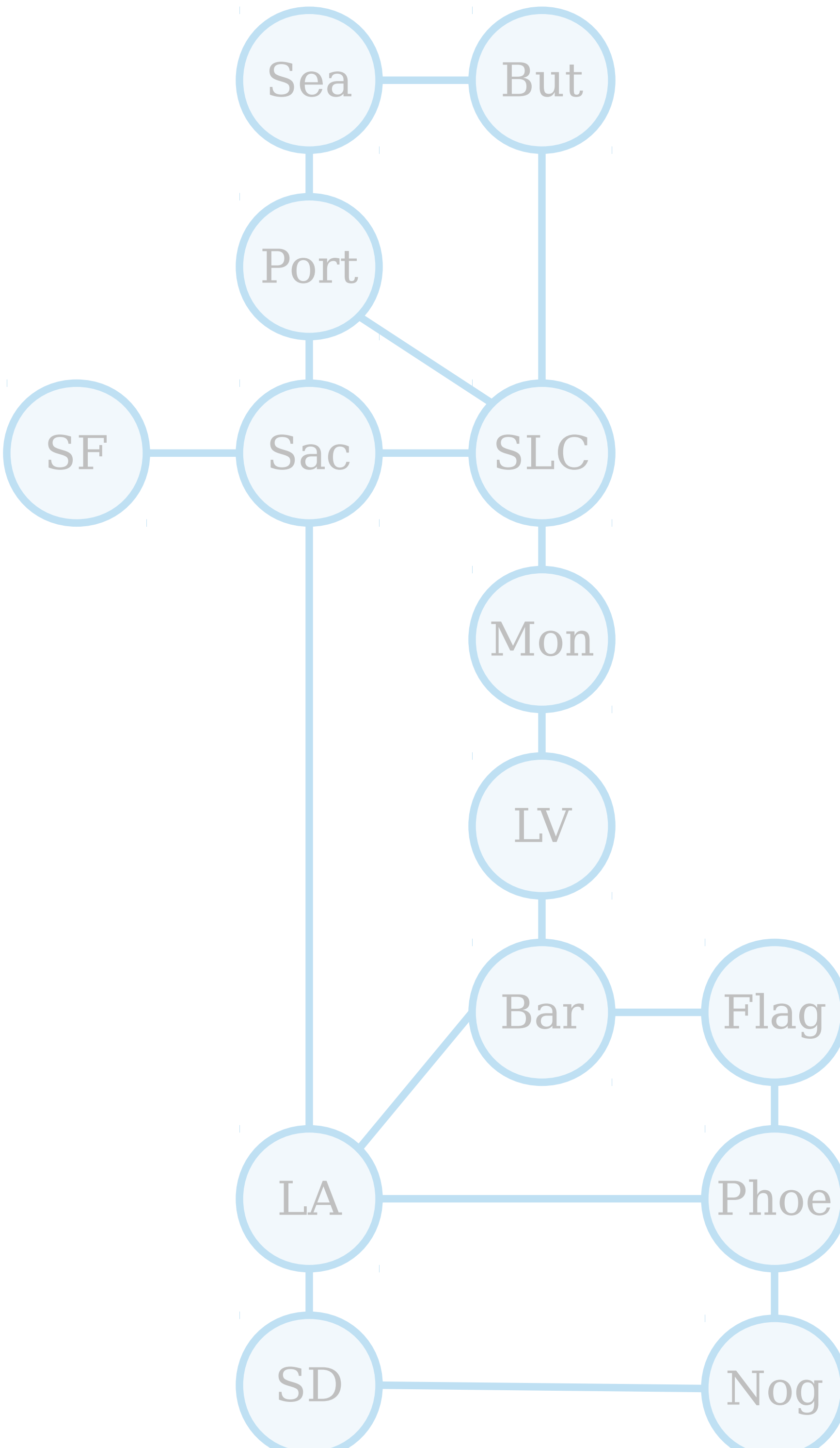
A ***closed walk*** in a graph is a walk from a node back to itself. (By convention, a closed walk cannot have length zero.)

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



(This walk has length six.)

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

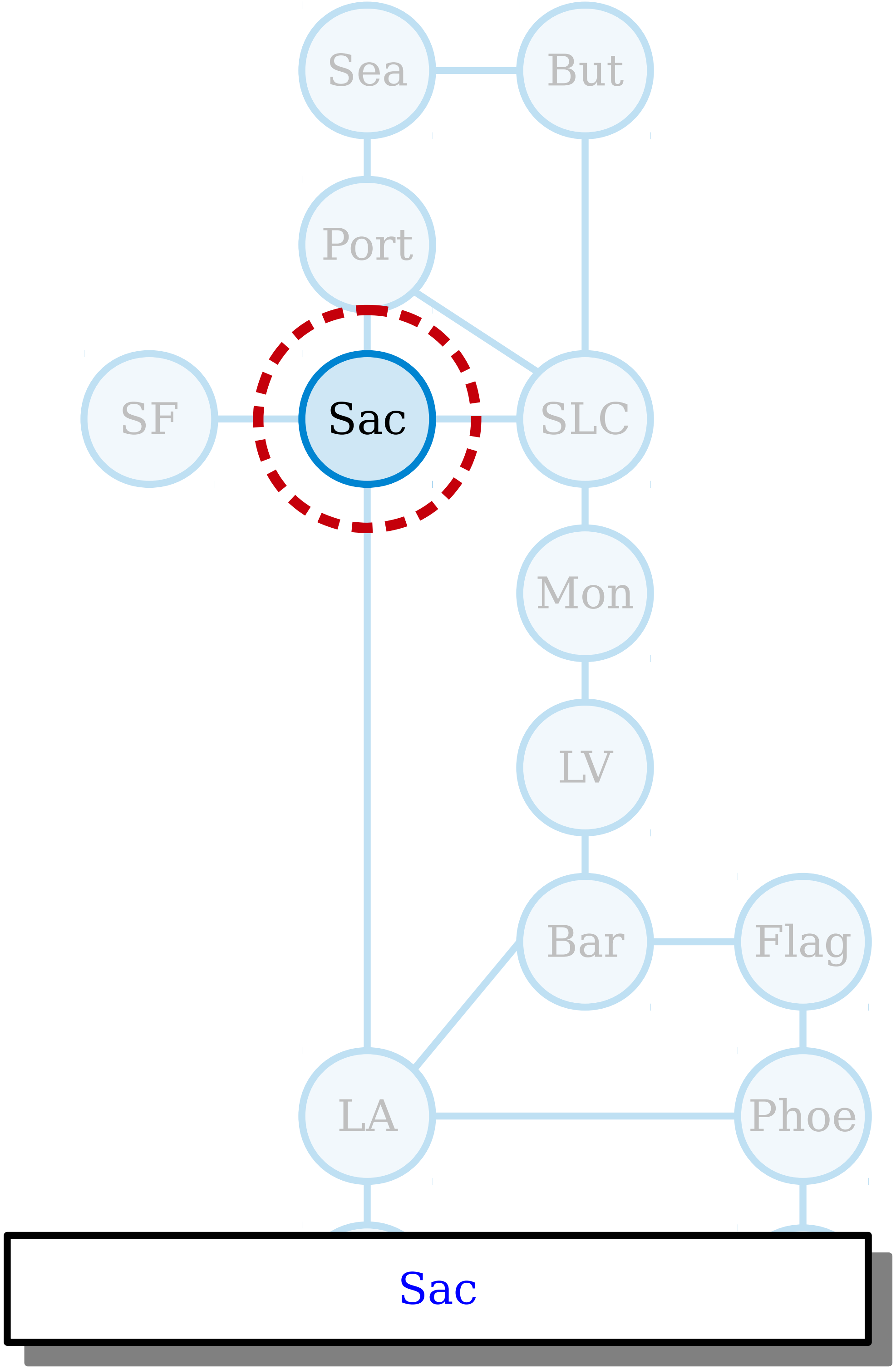


A ***walk*** in a graph $G = (V, E)$ is a sequence of one or more nodes $v_1, v_2, v_3, \ldots, v_n$ such that any two consecutive nodes in the sequence are adjacent.

The ***length*** of the walk $v_1, \ldots, v_n$ 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.

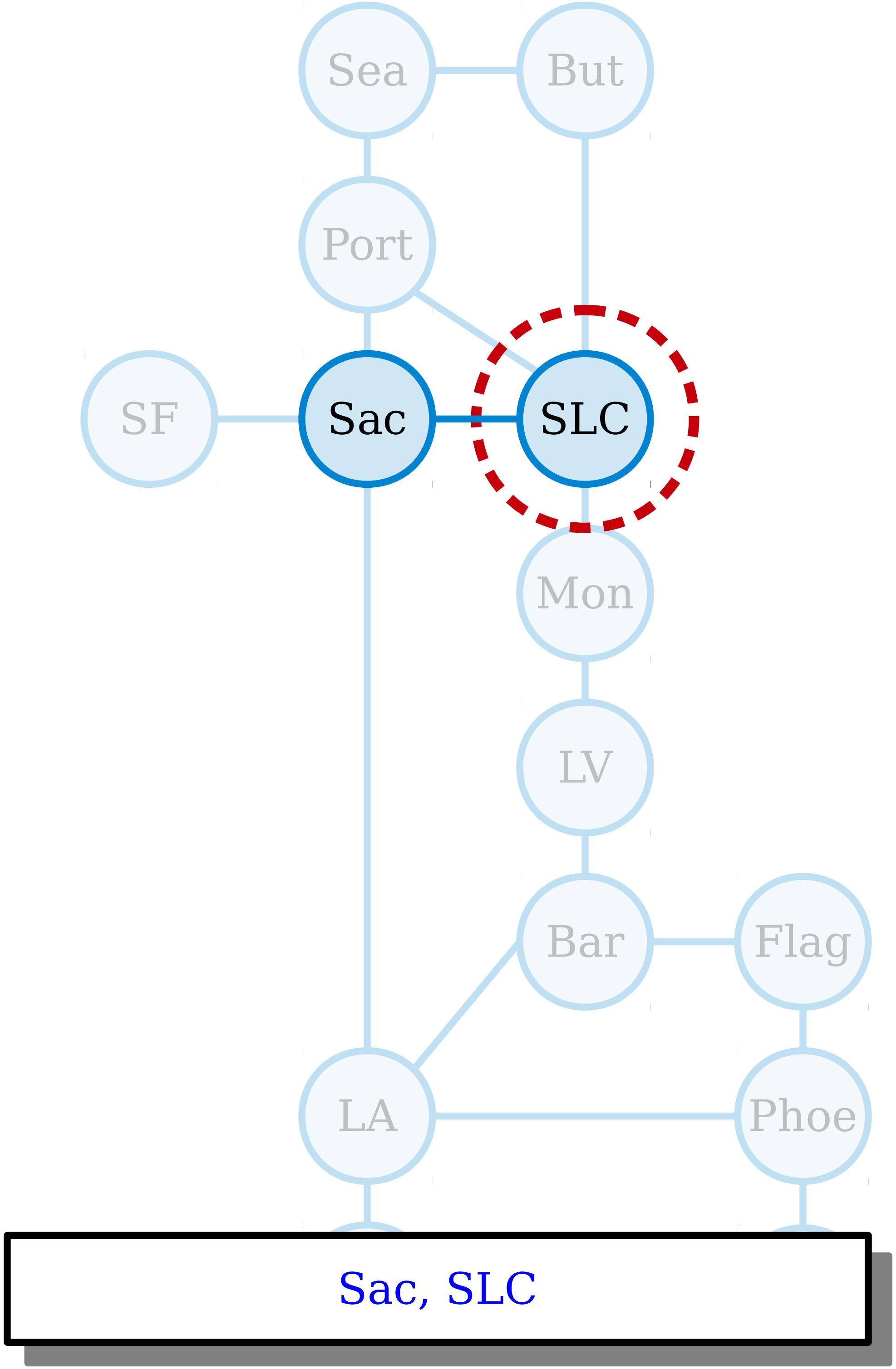


A ***walk*** in a graph $G = (V, E)$ is a sequence of one or more nodes $v_1, v_2, v_3, \ldots, v_n$ such that any two consecutive nodes in the sequence are adjacent.

The ***length*** of the walk $v_1, \ldots, v_n$ 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.

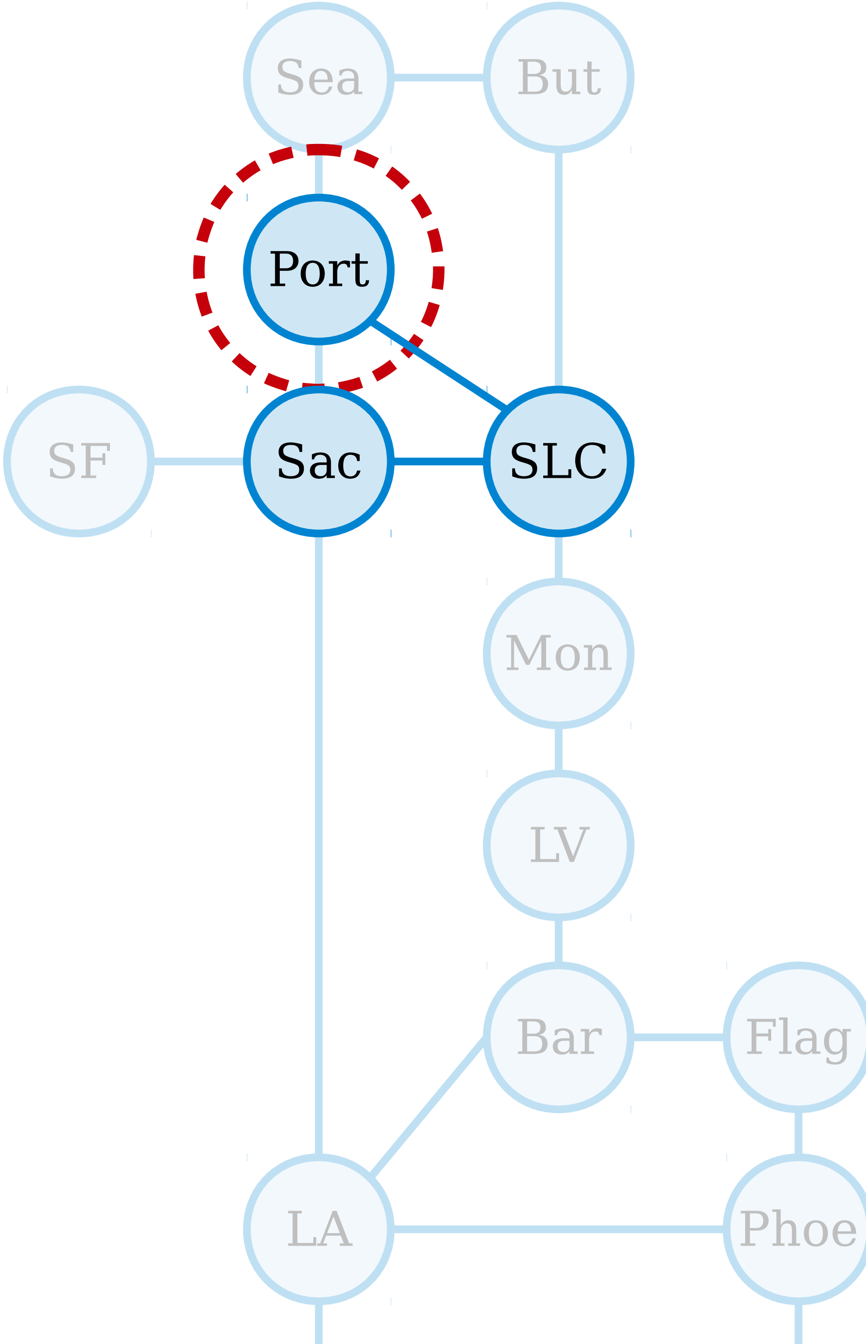


A ***walk*** in a graph $G = (V, E)$ is a sequence of one or more nodes $v_1, v_2, v_3, \ldots, v_n$ such that any two consecutive nodes in the sequence are adjacent.

The ***length*** of the walk $v_1, \ldots, v_n$ 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.



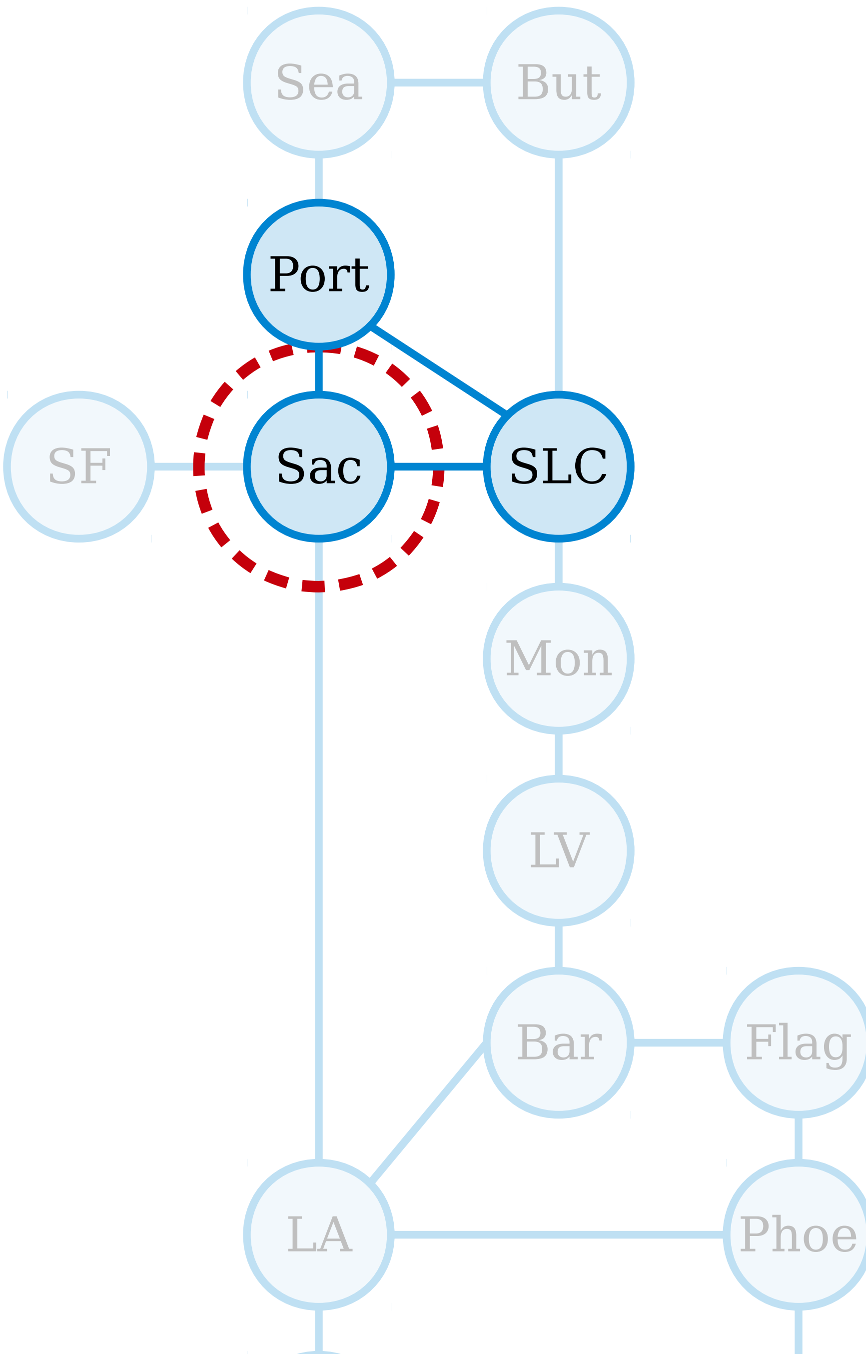
Sac, SLC, Port

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

The ***length*** of the walk $v_1, \ldots, v_n$ 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.



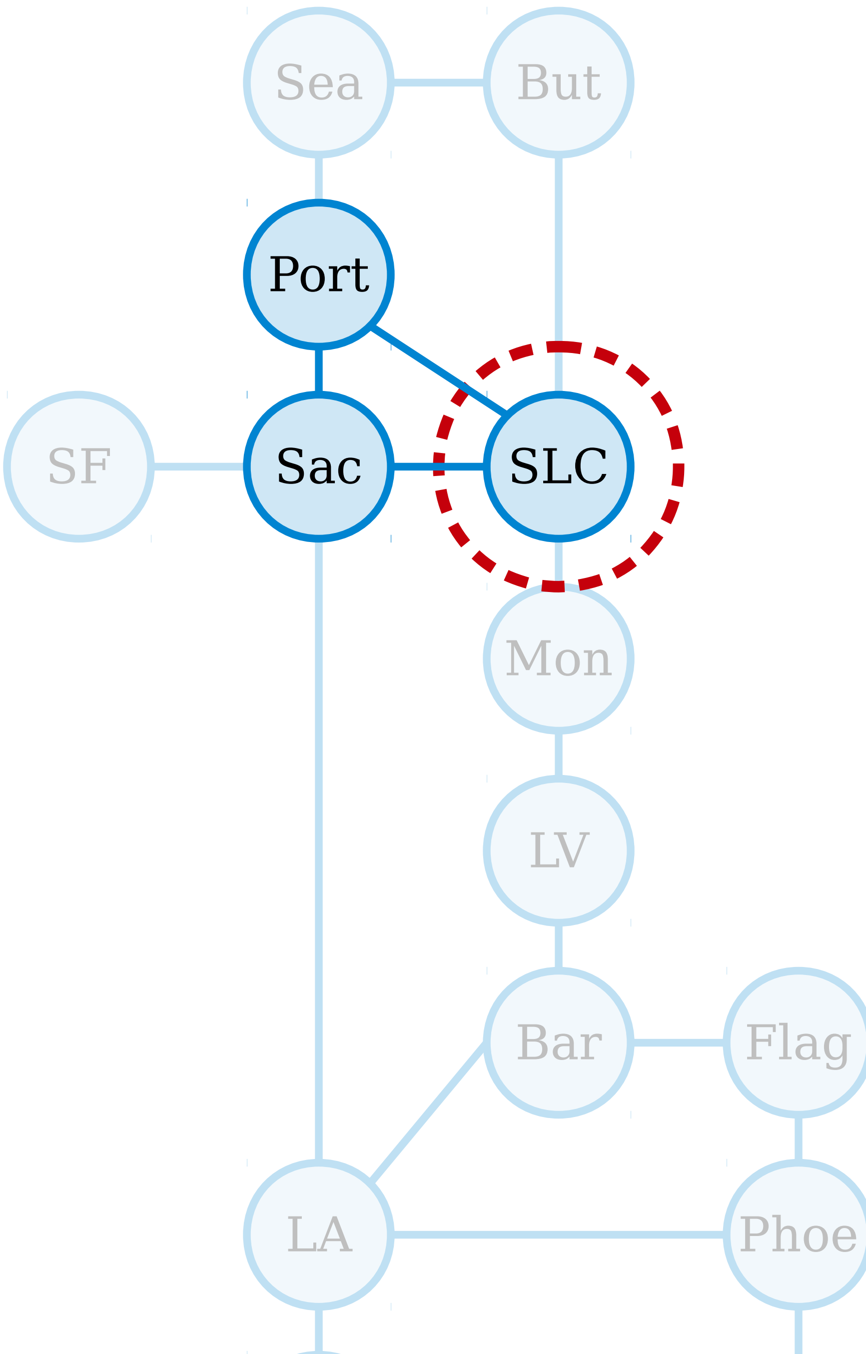
Sac, SLC, Port, Sac

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

The ***length*** of the walk $v_1, \ldots, v_n$ 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.



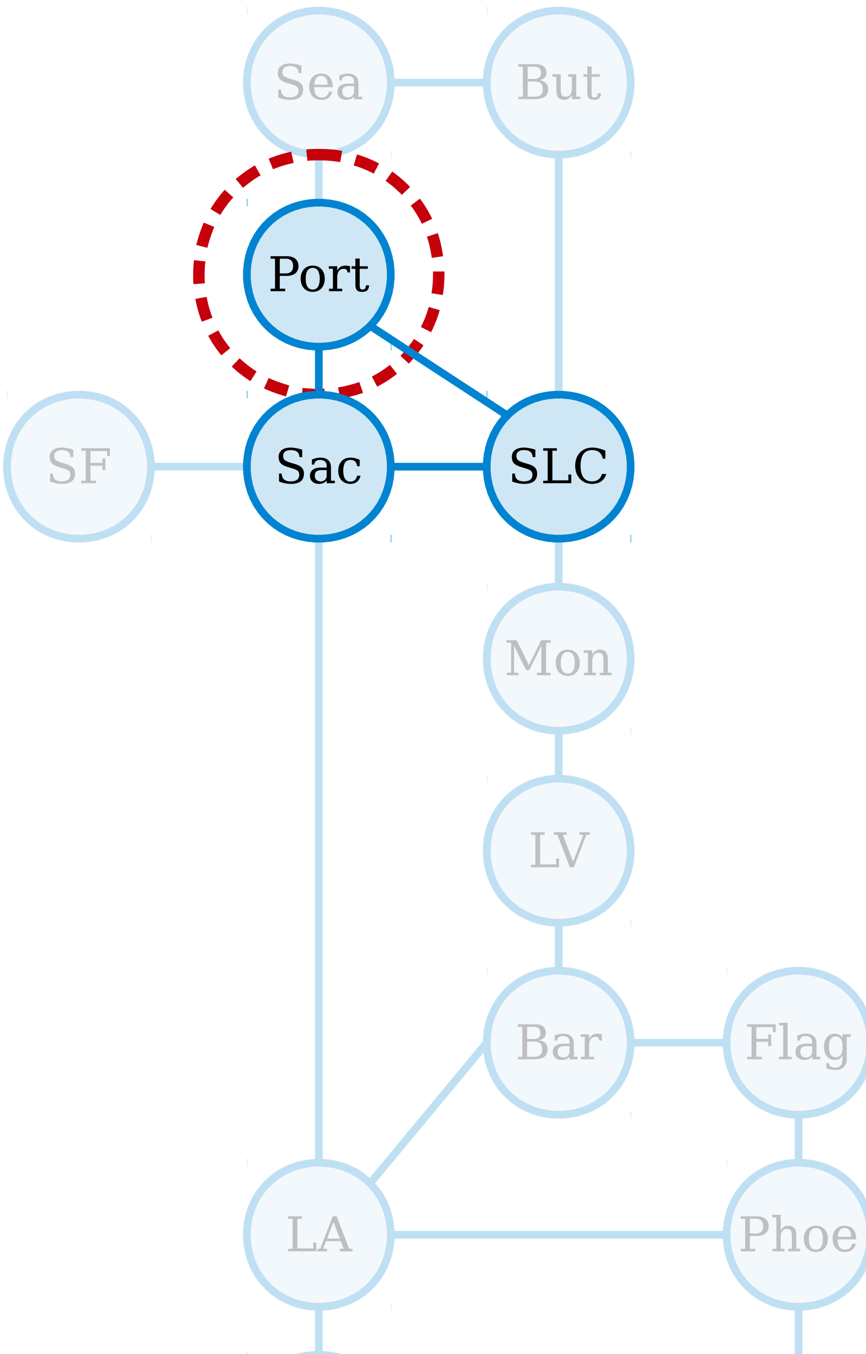
Sac, SLC, Port, Sac, SLC

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

The ***length*** of the walk $v_1, \ldots, v_n$ 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.



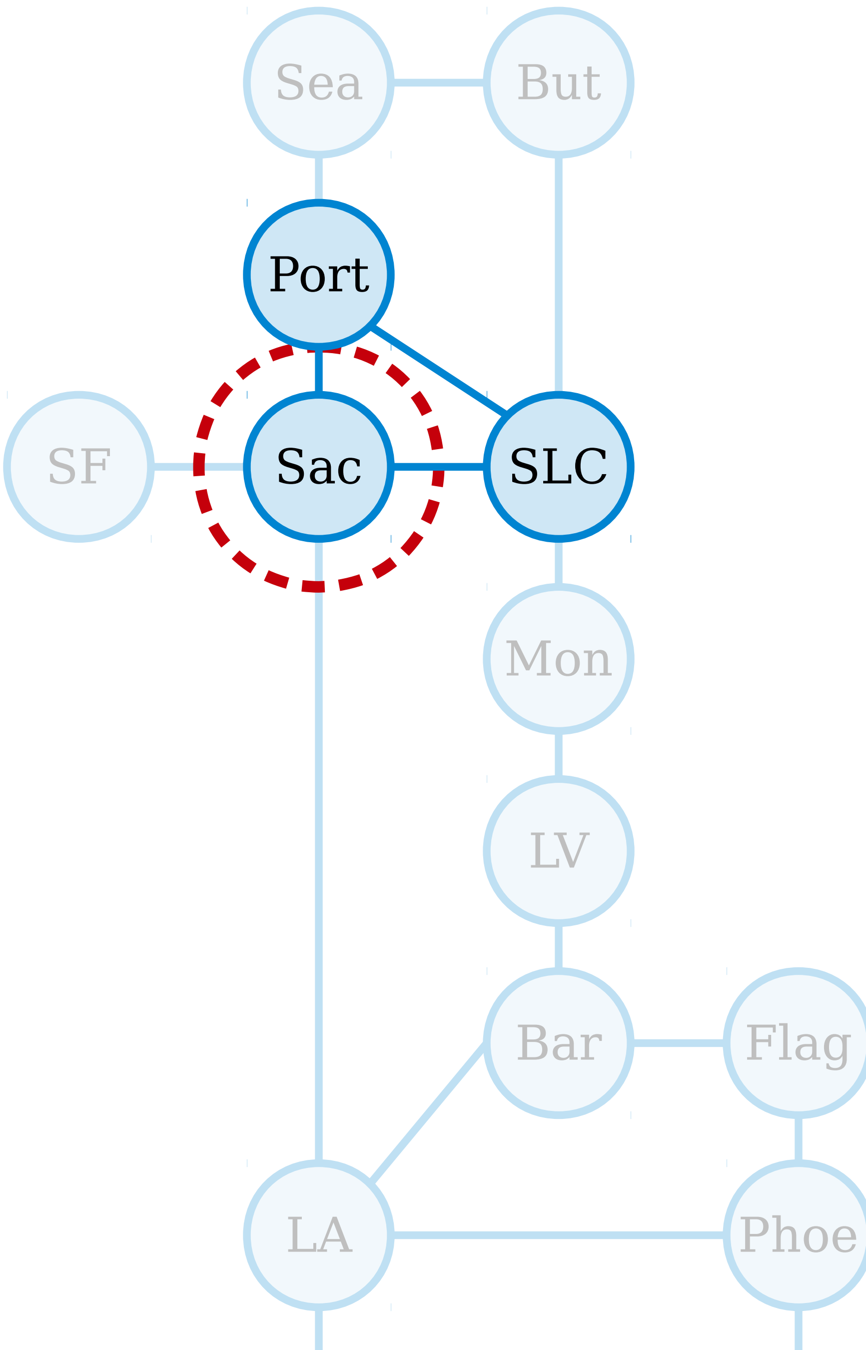
Sac, SLC, Port, Sac, SLC, Port

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

The ***length*** of the walk $v_1, \ldots, v_n$ 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.



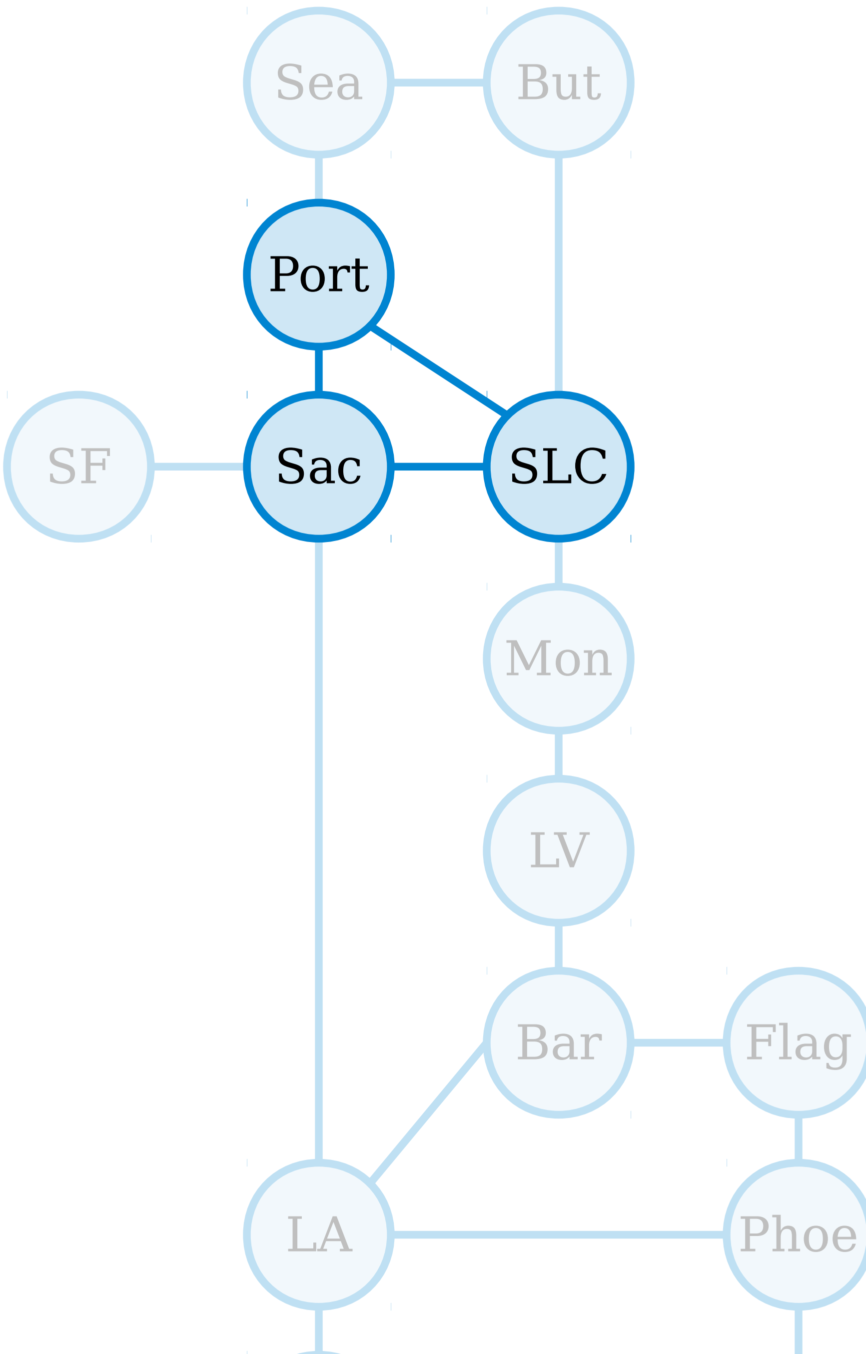
Sac, SLC, Port, Sac, SLC, Port, Sac

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

The ***length*** of the walk $v_1, \ldots, v_n$ 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

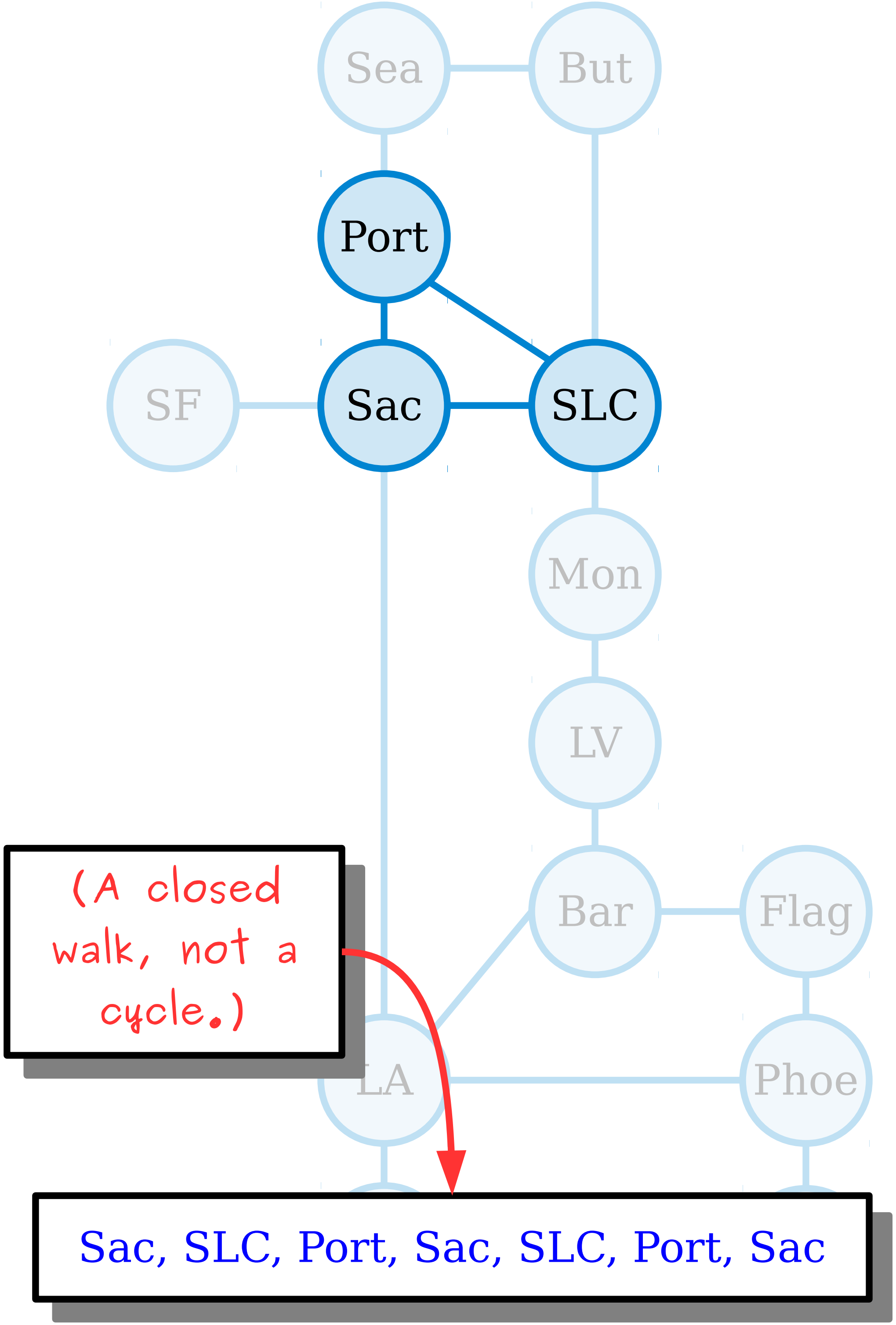
A ***walk*** in a graph $G = (V, E)$ is a sequence of one or more nodes $v_1, v_2, v_3, \ldots, v_n$ such that any two consecutive nodes in the sequence are adjacent.

The ***length*** of the walk $v_1, \ldots, v_n$ 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.



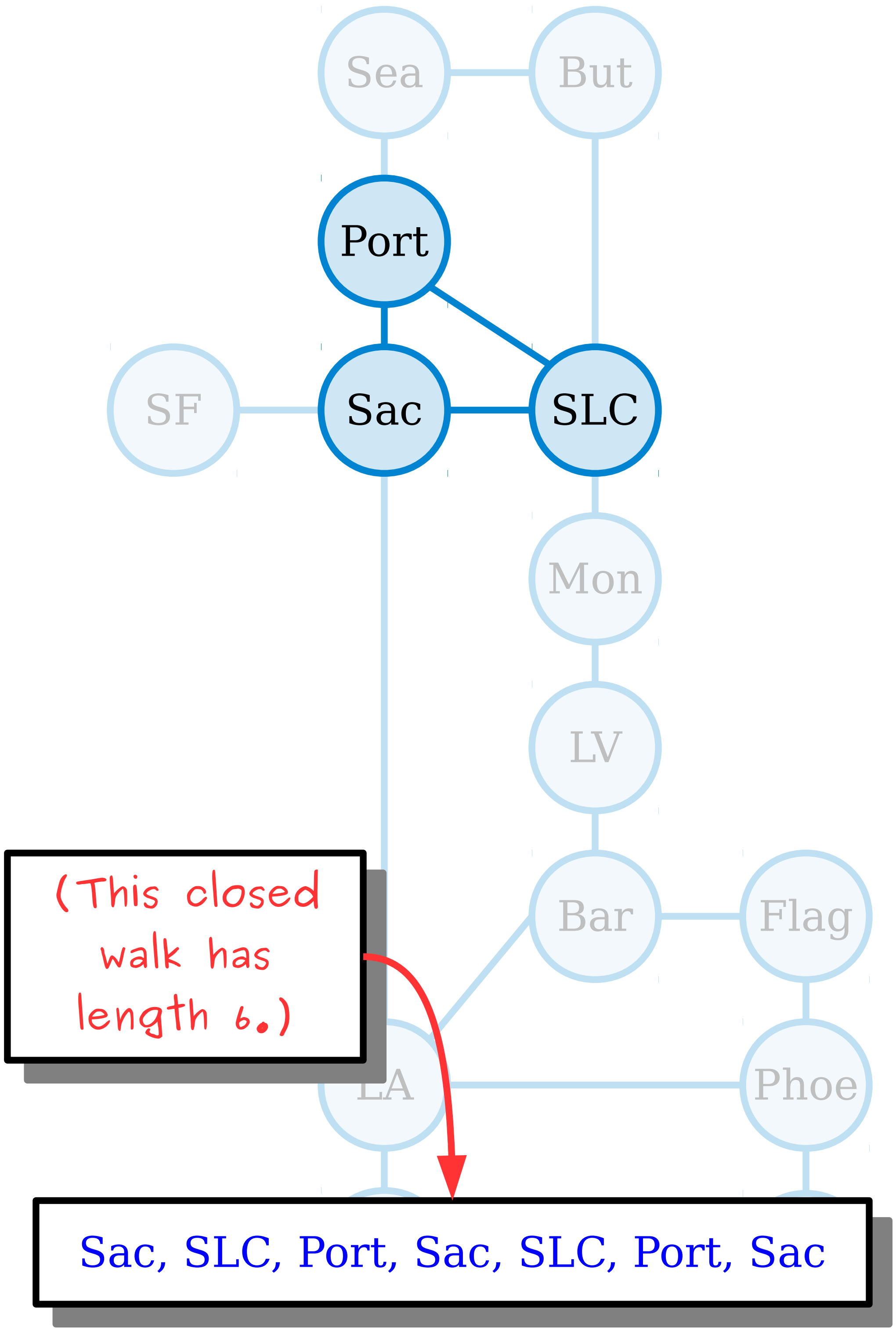
A ***walk*** in a graph $G = (V, E)$ is a sequence of one or more nodes $v_1, v_2, v_3, \ldots, v_n$ such that any two consecutive nodes in the sequence are adjacent.

The ***length*** of the walk $v_1, \ldots, v_n$ 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.



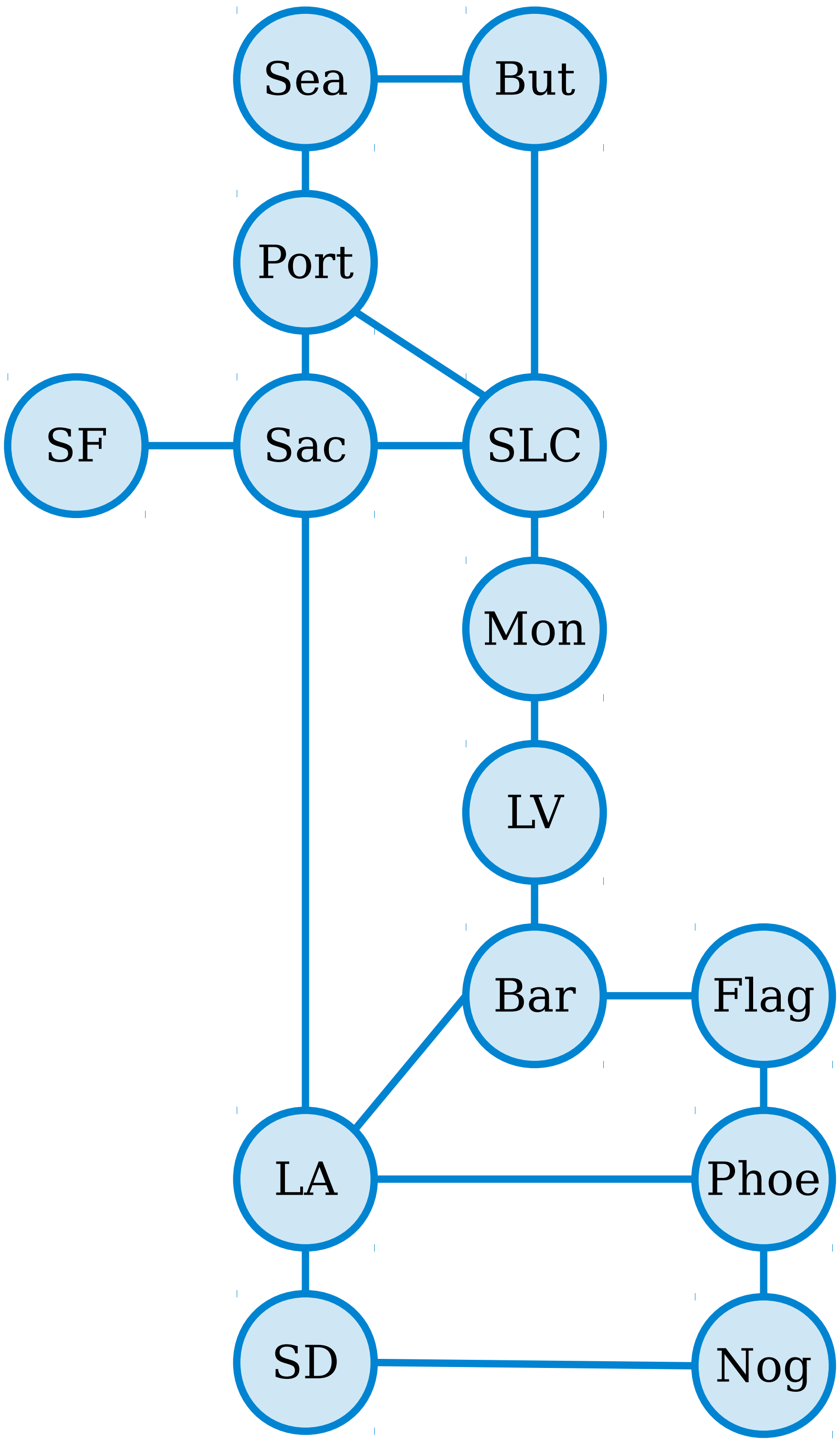
A ***walk*** in a graph $G = (V, E)$ is a sequence of one or more nodes $v_1, v_2, v_3, \ldots, v_n$ such that any two consecutive nodes in the sequence are adjacent.

The ***length*** of the walk $v_1, \ldots, v_n$ 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.



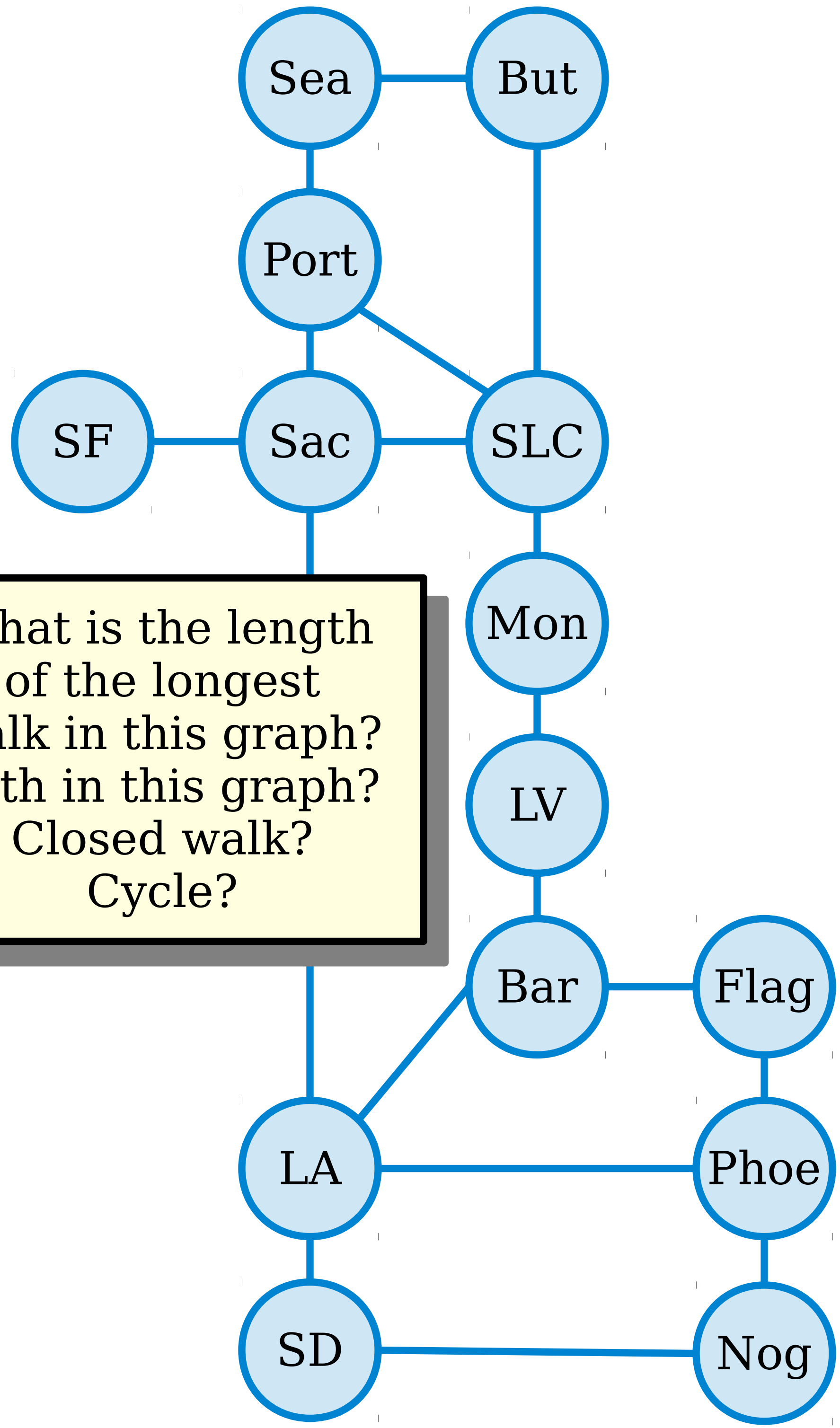
A ***walk*** in a graph $G = (V, E)$ is a sequence of one or more nodes $v_1, v_2, v_3, \ldots, v_n$ such that any two consecutive nodes in the sequence are adjacent.

The ***length*** of the walk $v_1, \ldots, v_n$ 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 in this graph? Closed walk? Cycle?

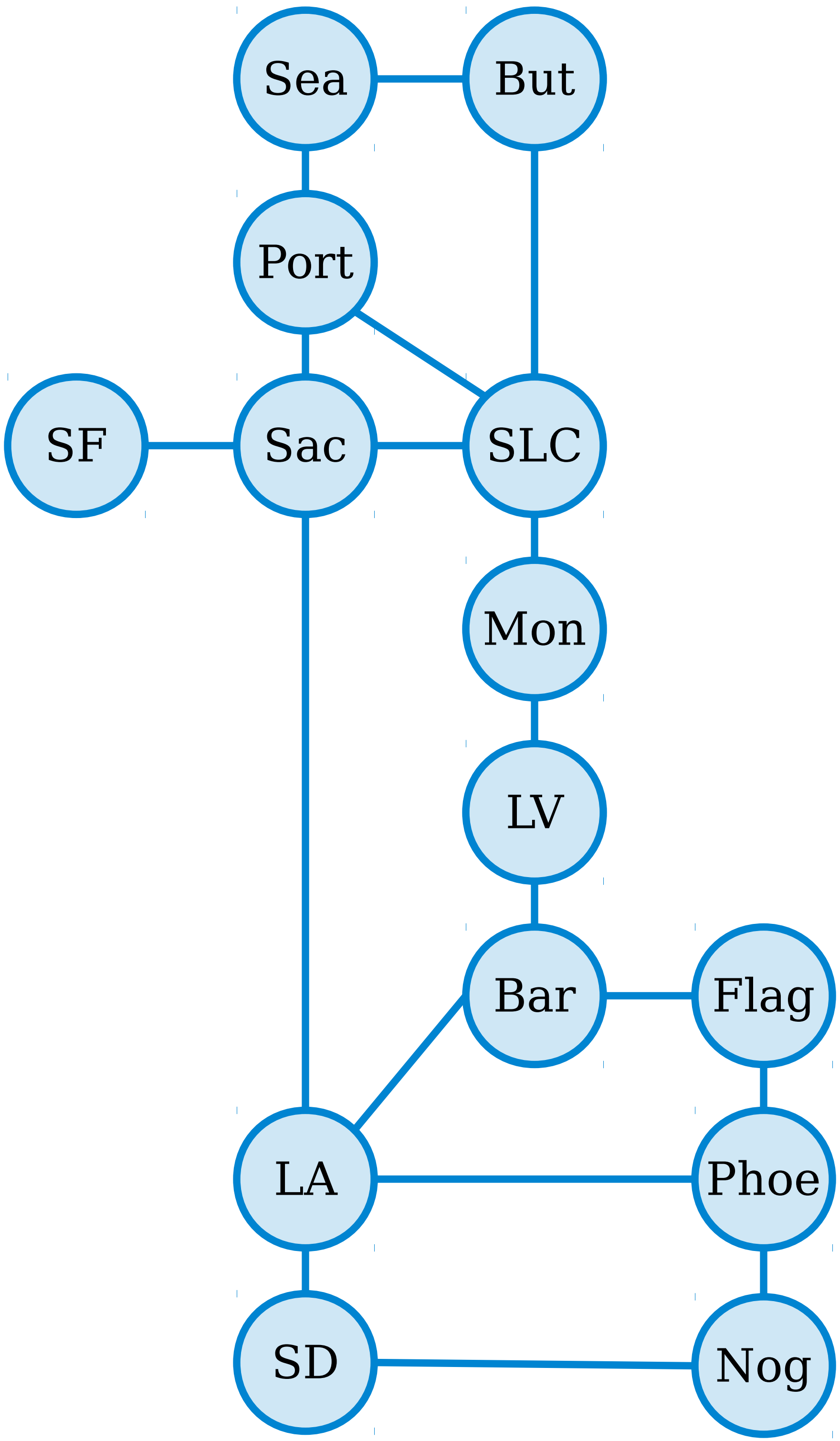
A ***walk*** in a graph $G = (V, E)$ is a sequence of one or more nodes $v_1, v_2, v_3, \ldots, v_n$ such that any two consecutive nodes in the sequence are adjacent.

The ***length*** of the walk $v_1, \ldots, v_n$ 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.)

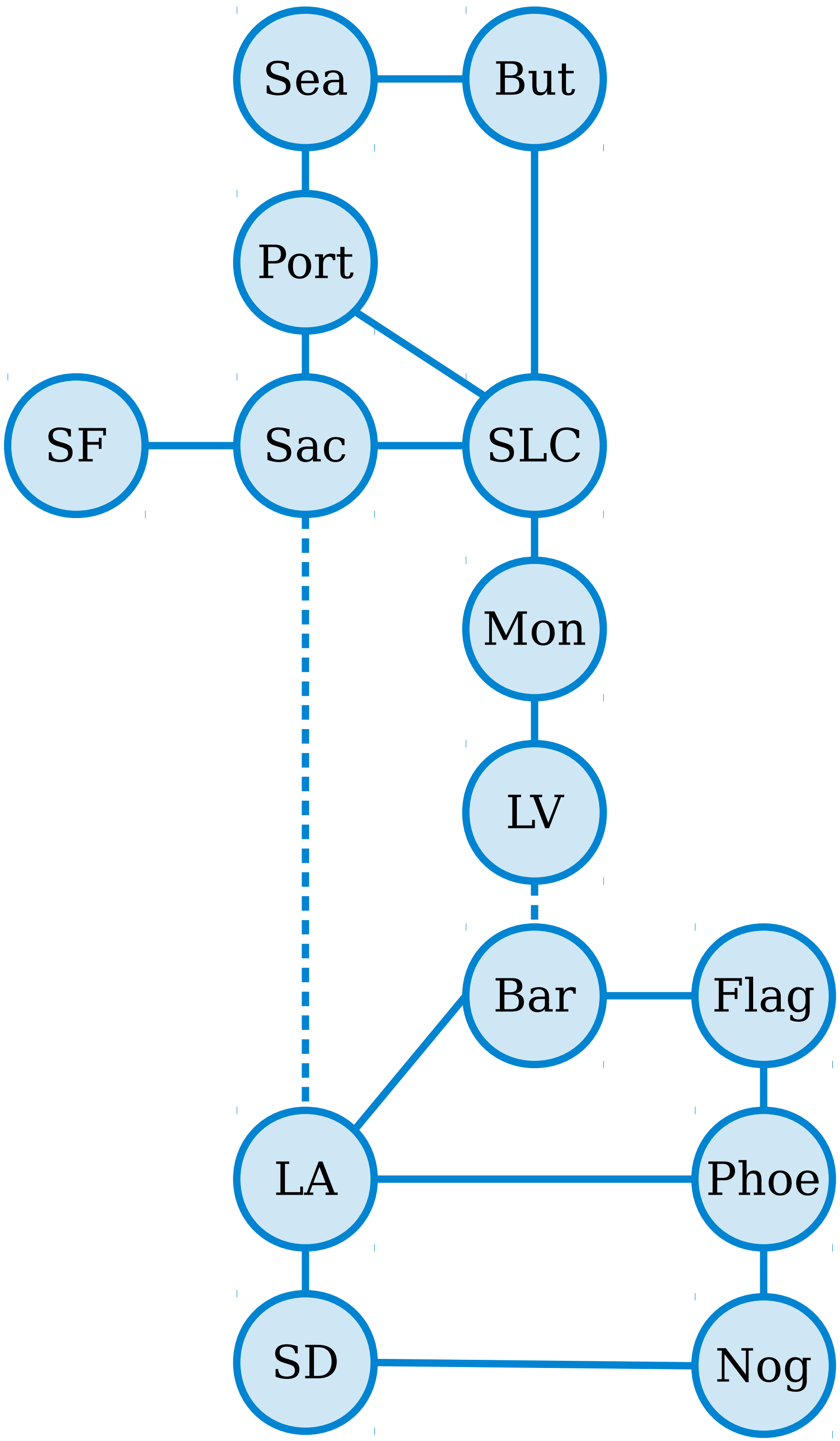
A ***path*** in a graph is walk that does not repeat any nodes.

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



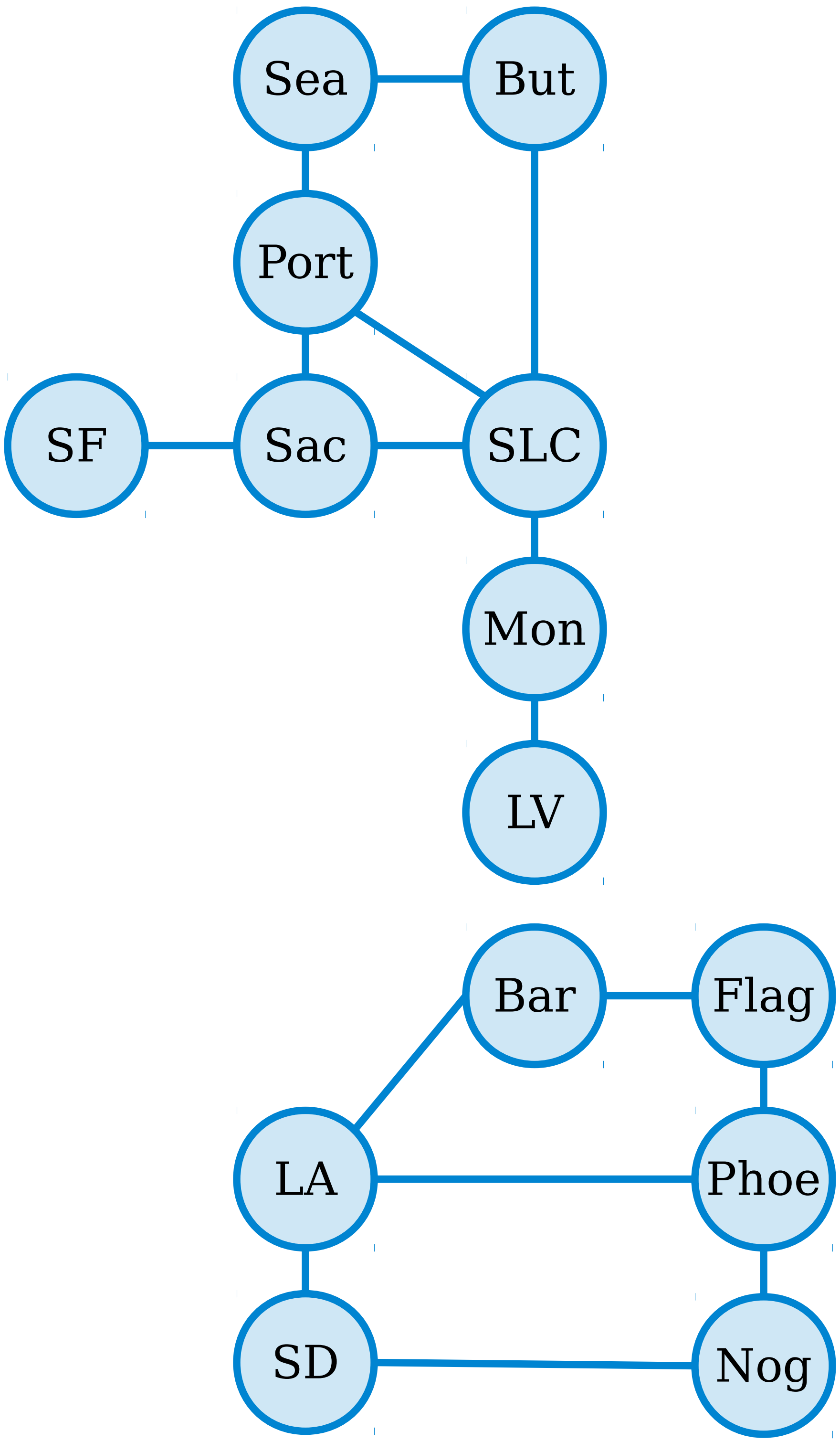
A ***walk*** in a graph $G = (V, E)$ is a sequence of one or more nodes $v_1, v_2, v_3, \ldots, v_n$ such that any two consecutive nodes in the sequence are adjacent.

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



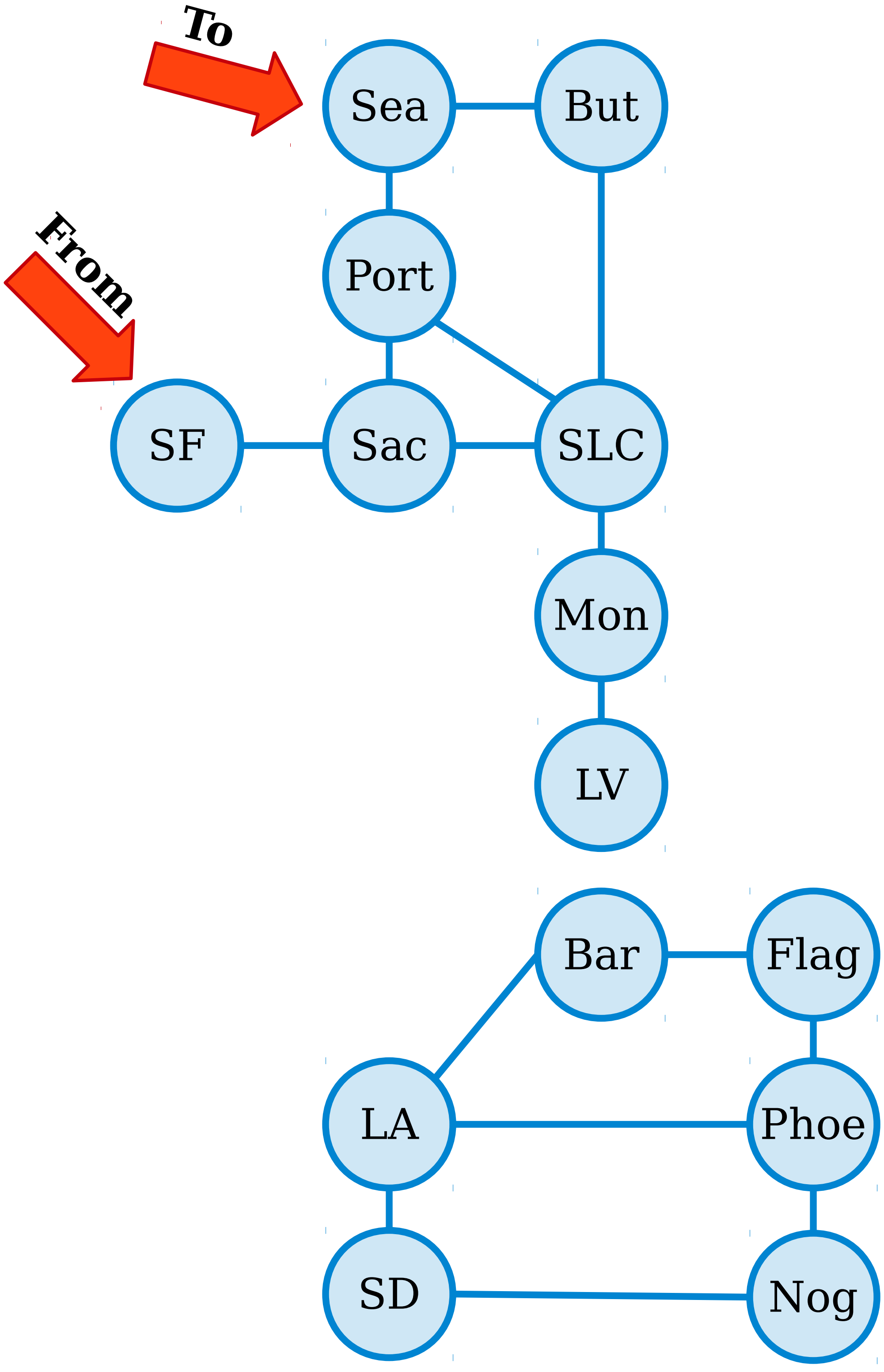
A ***walk*** in a graph $G = (V, E)$ is a sequence of one or more nodes $v_1, v_2, v_3, \ldots, v_n$ such that any two consecutive nodes in the sequence are adjacent.

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



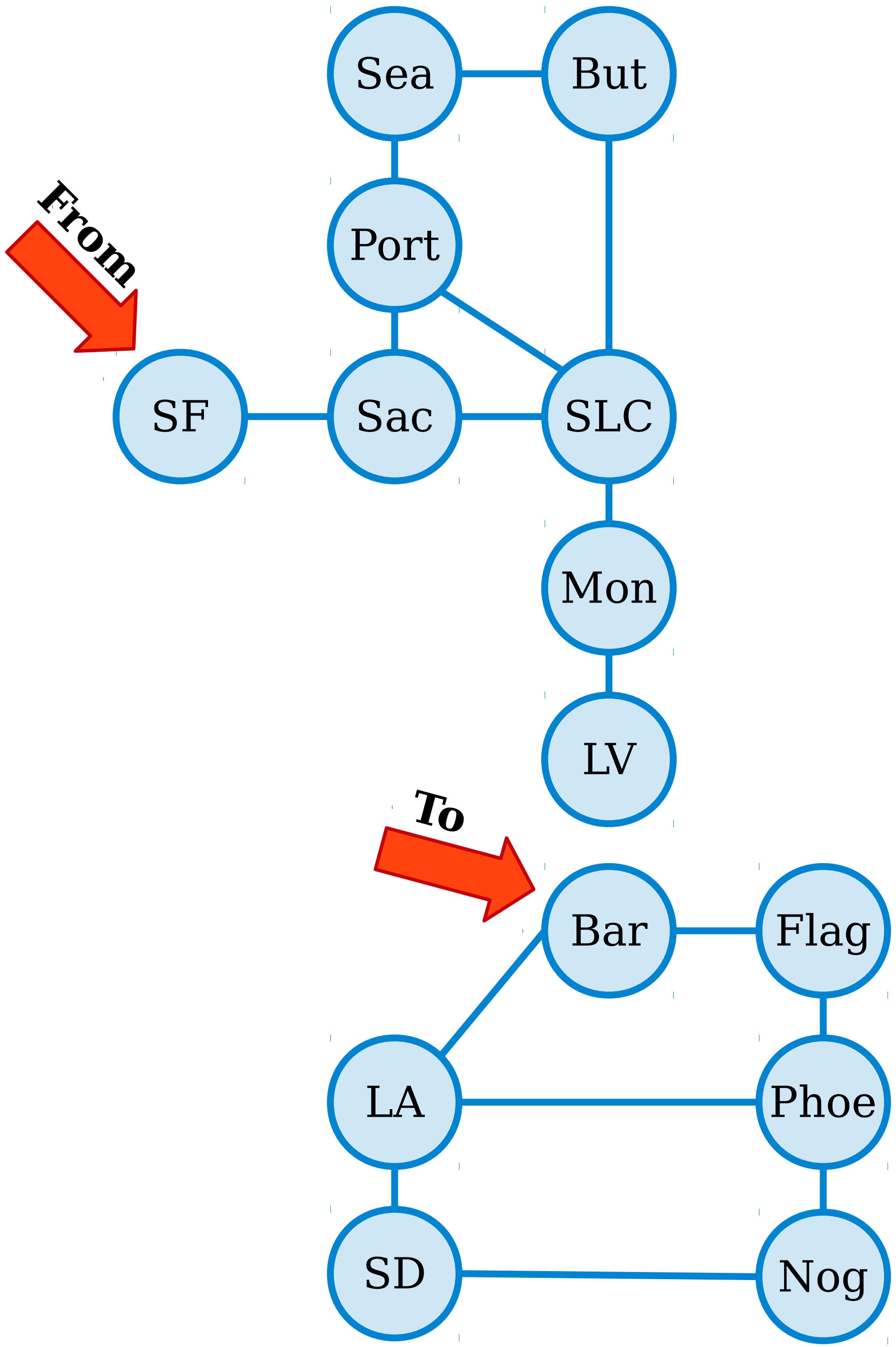
A ***walk*** in a graph $G = (V, E)$ is a sequence of one or more nodes $v_1, v_2, v_3, \ldots, v_n$ such that any two consecutive nodes in the sequence are adjacent.

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



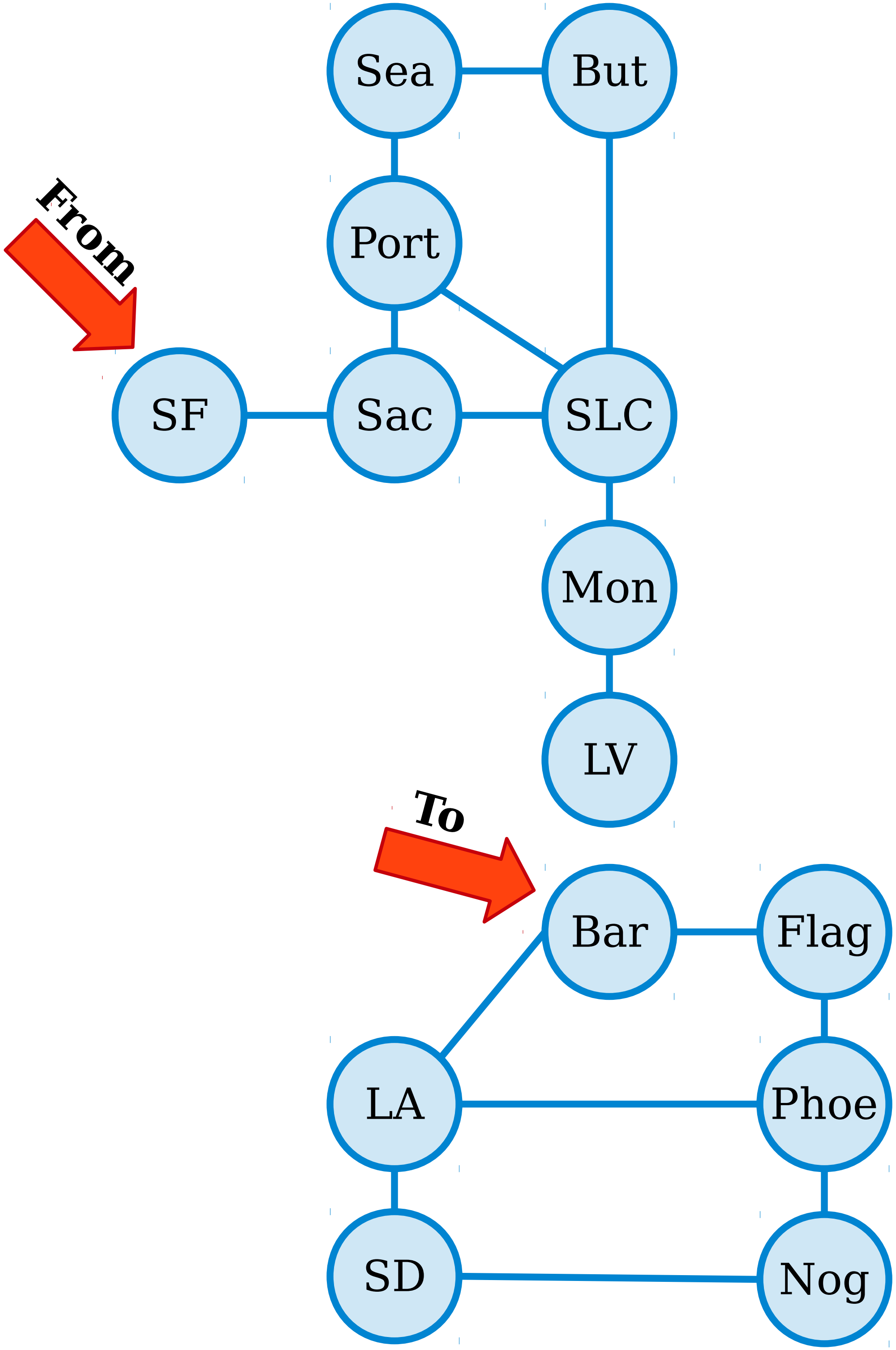
A ***walk*** in a graph $G = (V, E)$ is a sequence of one or more nodes $v_1, v_2, v_3, \ldots, v_n$ such that any two consecutive nodes in the sequence are adjacent.

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 $v_1, v_2, v_3, \ldots, v_n$ such that any two consecutive nodes in the sequence are adjacent.

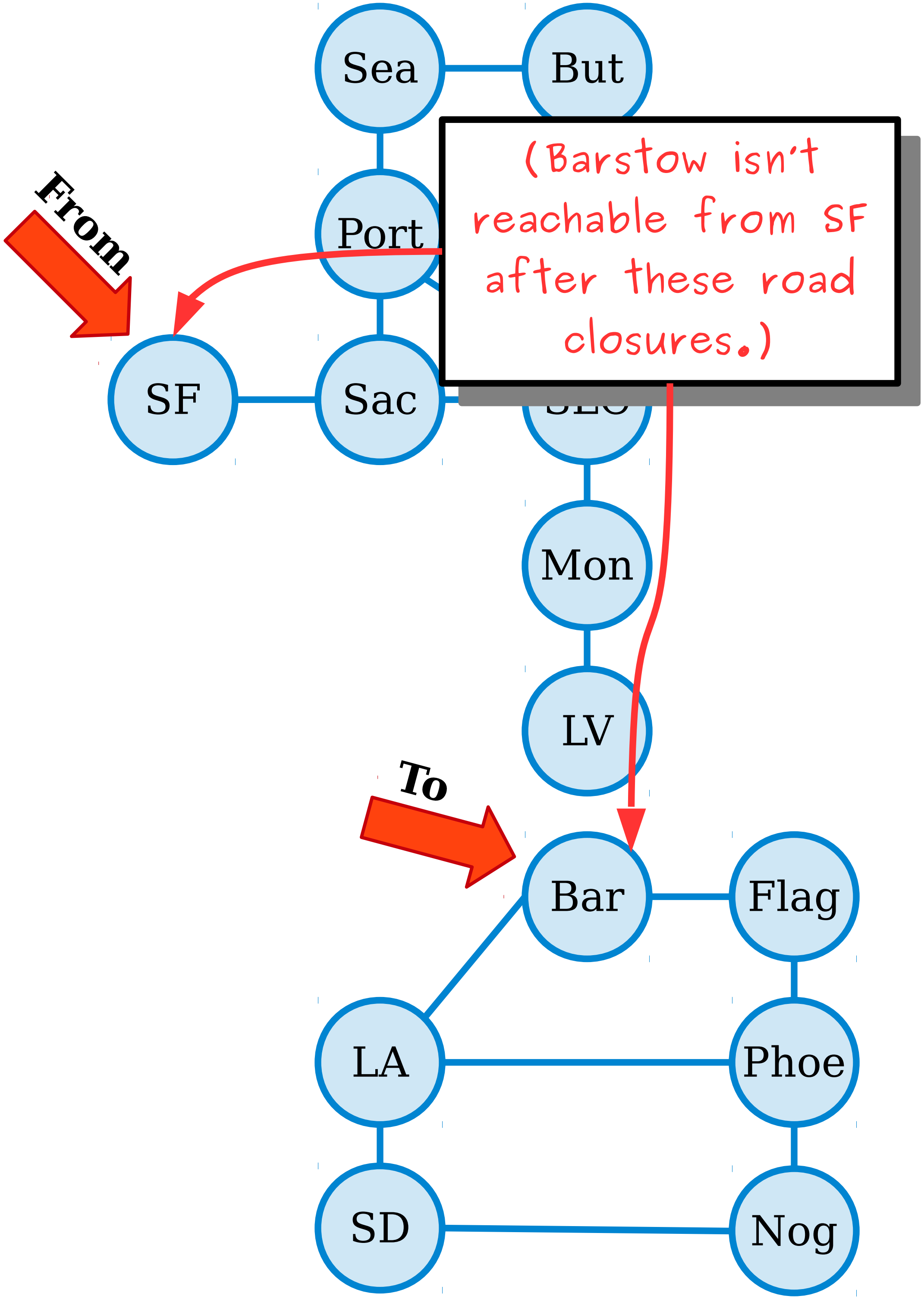
A ***path*** in a graph is walk that does not repeat any nodes.



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

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

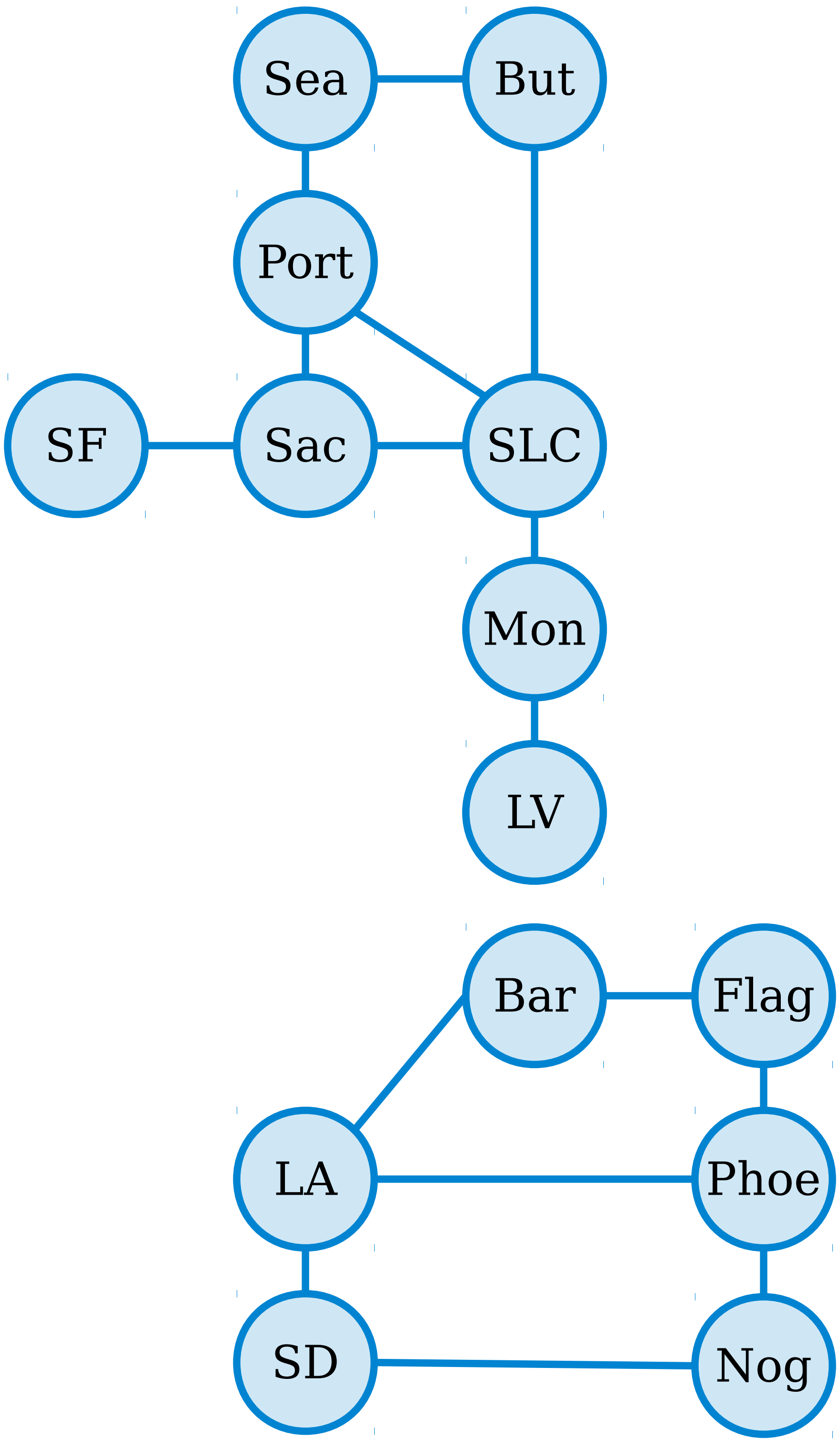
A node v is ***reachable*** from a node u if there is a path from u to v.



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

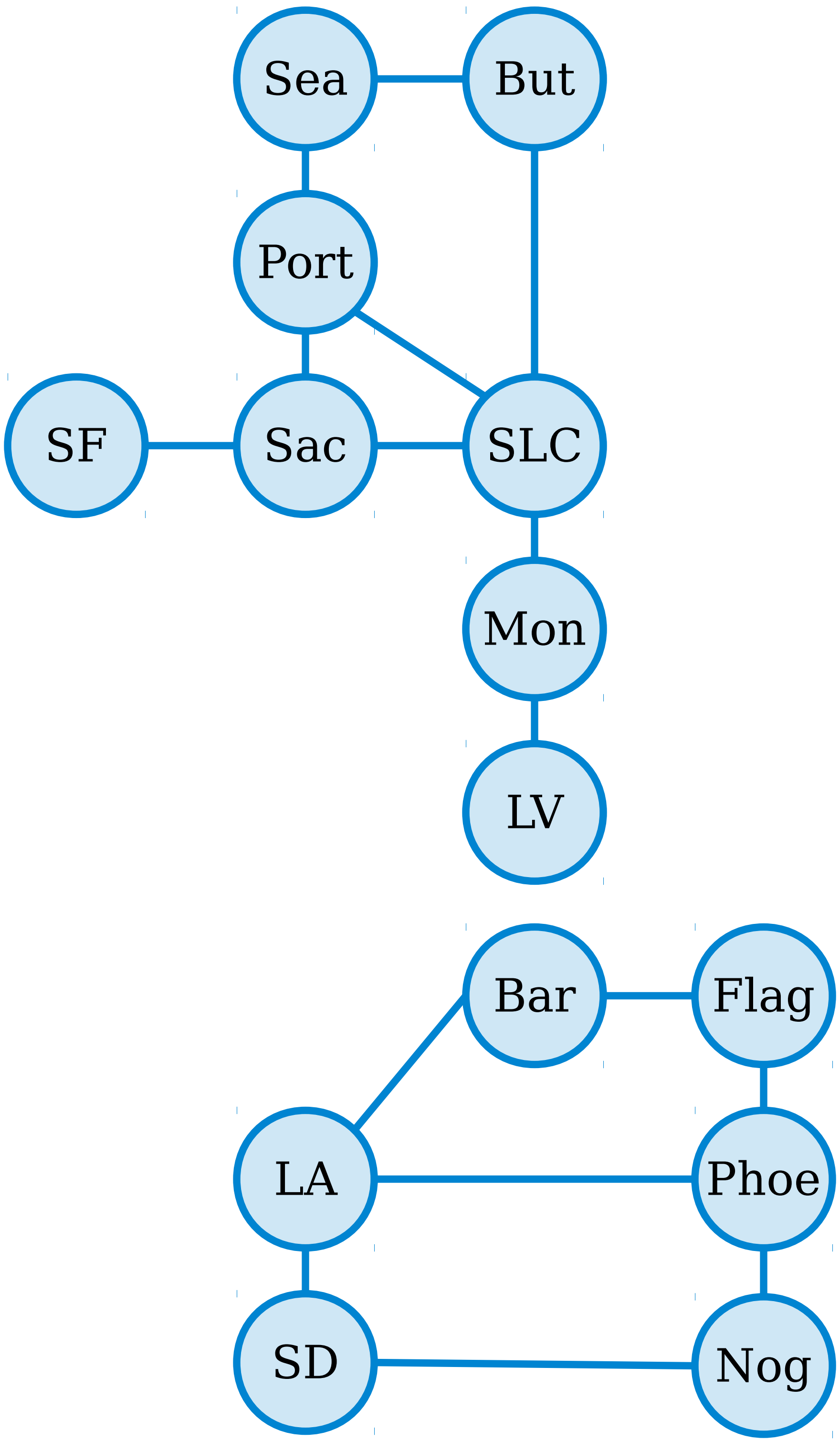


A ***walk*** in a graph $G = (V, E)$ is a sequence of one or more nodes $v_1, v_2, v_3, \ldots, v_n$ such that any two consecutive nodes in the sequence are adjacent.

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.



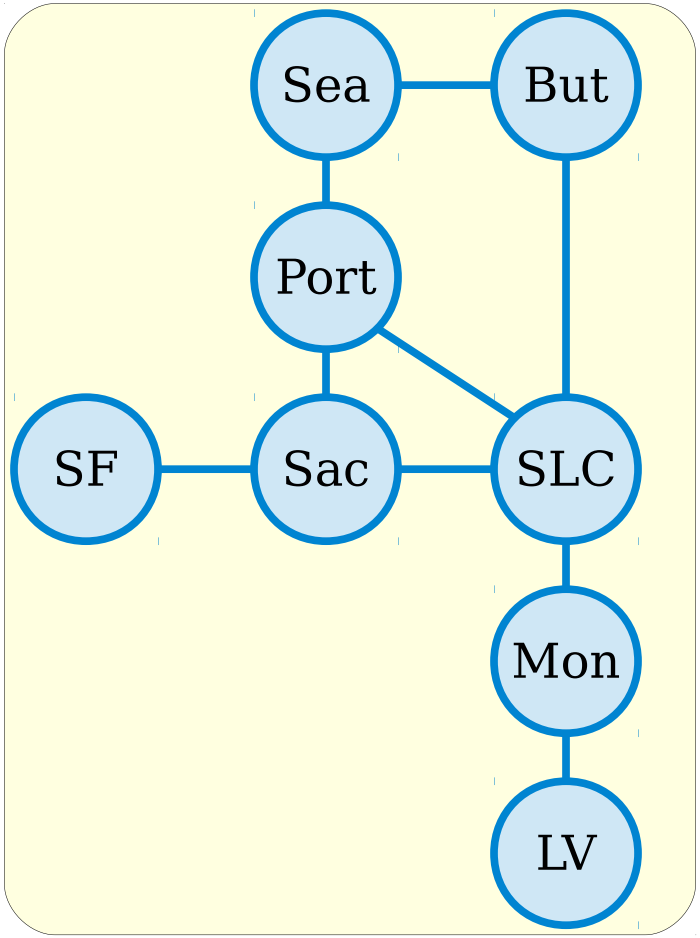
A ***walk*** in a graph $G = (V, E)$ is a sequence of one or more nodes $v_1, v_2, v_3, \ldots, v_n$ such that any two consecutive nodes in the sequence are adjacent.

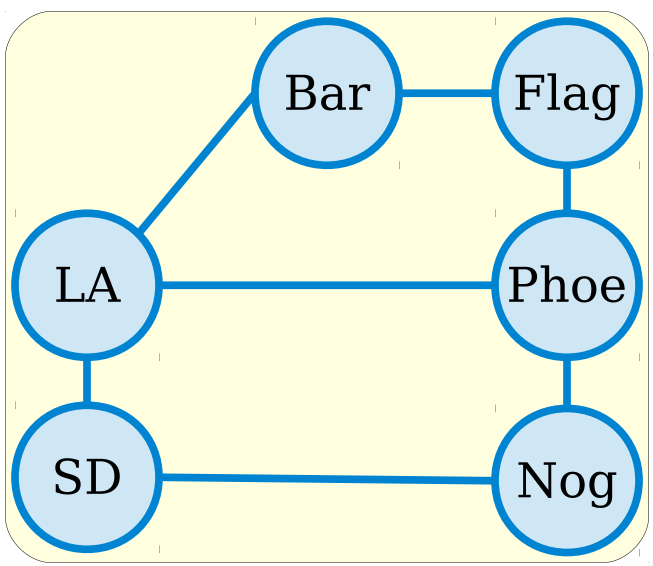
A ***path*** in a graph is walk that does not repeat any nodes.

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.)



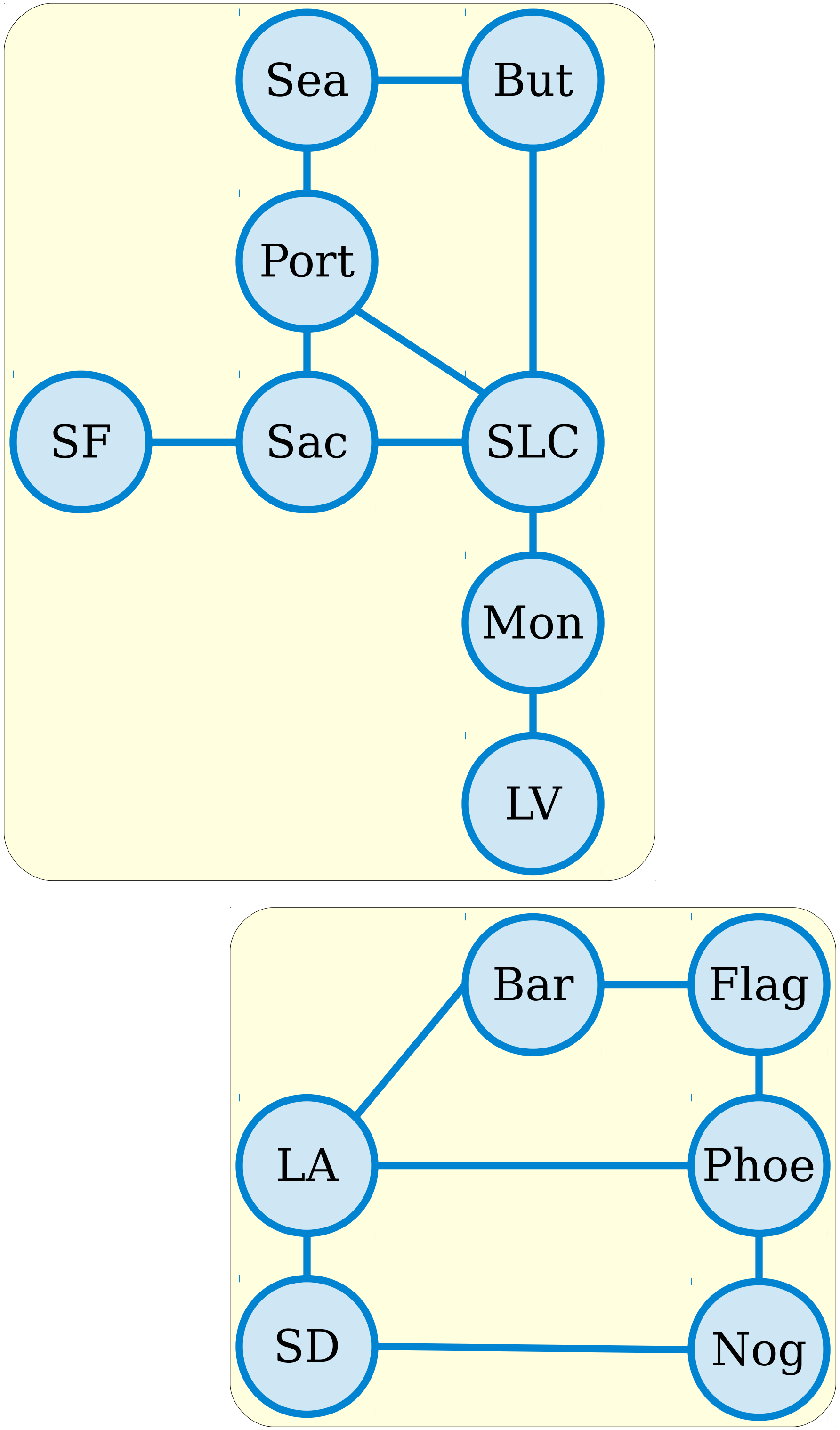


A ***walk*** in a graph $G = (V, E)$ is a sequence of one or more nodes $v_1, v_2, v_3, \ldots, v_n$ such that any two consecutive nodes in the sequence are adjacent.

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 $v_1, v_2, v_3, \ldots, v_n$ 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!

Time-Out for Announcements!

Problem Set Three

- Problem Set Two was due today at 4:00PM.
- Problem Set Three goes out today. It’s due next Friday at 4:00PM.
- As always, ping us if you need help working on this one: post on EdStem or stop by office hours.

Preparing for the Exam

- We’ve posted a “Preparing for the Exam” page on the course website with full details and logistics.
- It also includes advice from former CS103 students about how to do well here.
- Check it out – there are tons of goodies there!

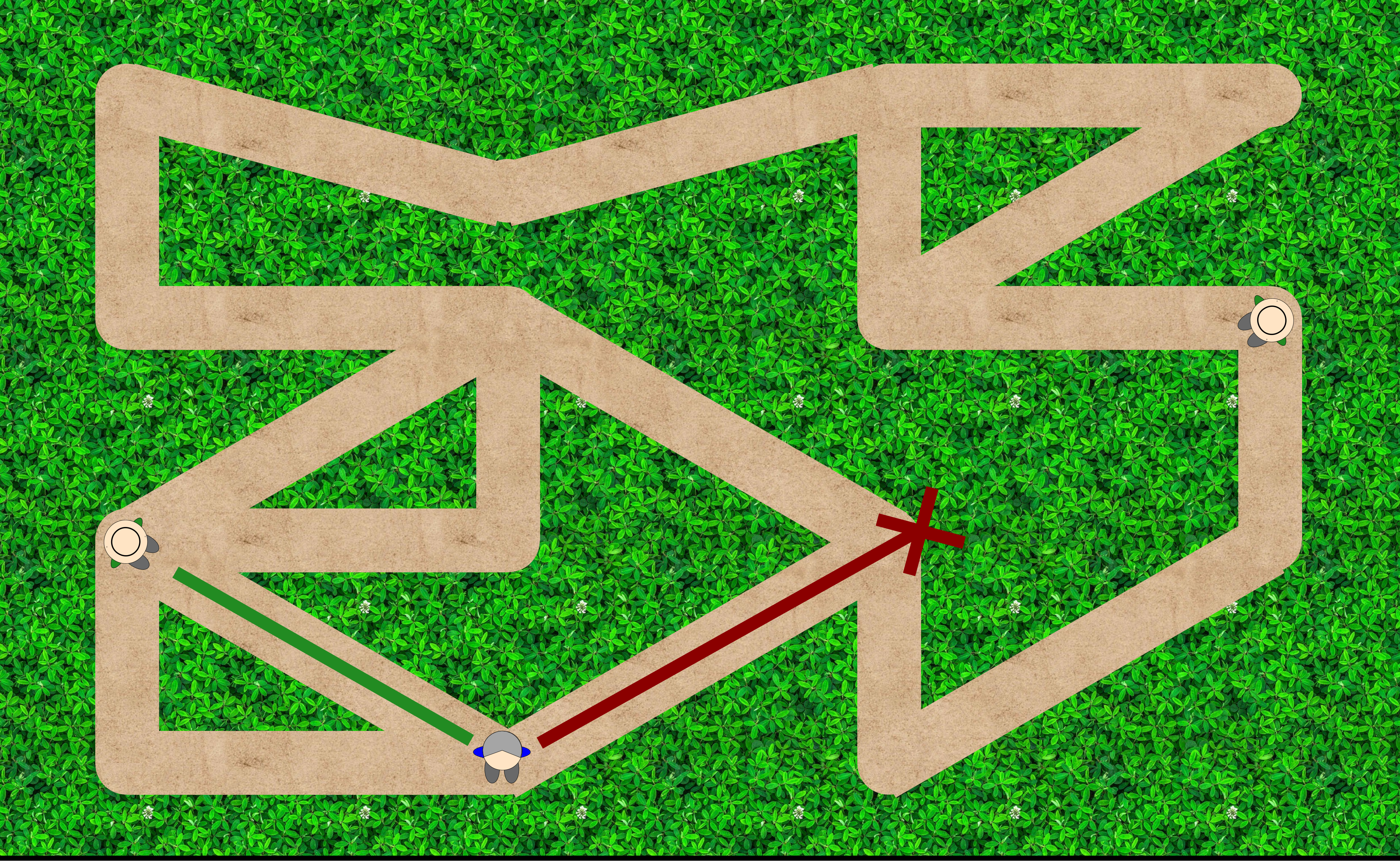
Practice Midterms

- We've also posted two practice midterms. Practice Midterm 1 is slightly easier than our exam will be. Practice Midterm 2 is fairly representative of the difficulty of the actual exam.
- Our recommendation:
  - Sometime during week 4, sit down and take Practice Midterm 1 as if it were the actual exam.
  - Identify any gaps in your understanding, and supplement with the extra practice problems as needed.
  - Sometime during week 5 (before the real exam), sit down and take Practice Midterm 2.
- Please do ***not*** read the solutions to a problem until you have worked through it.

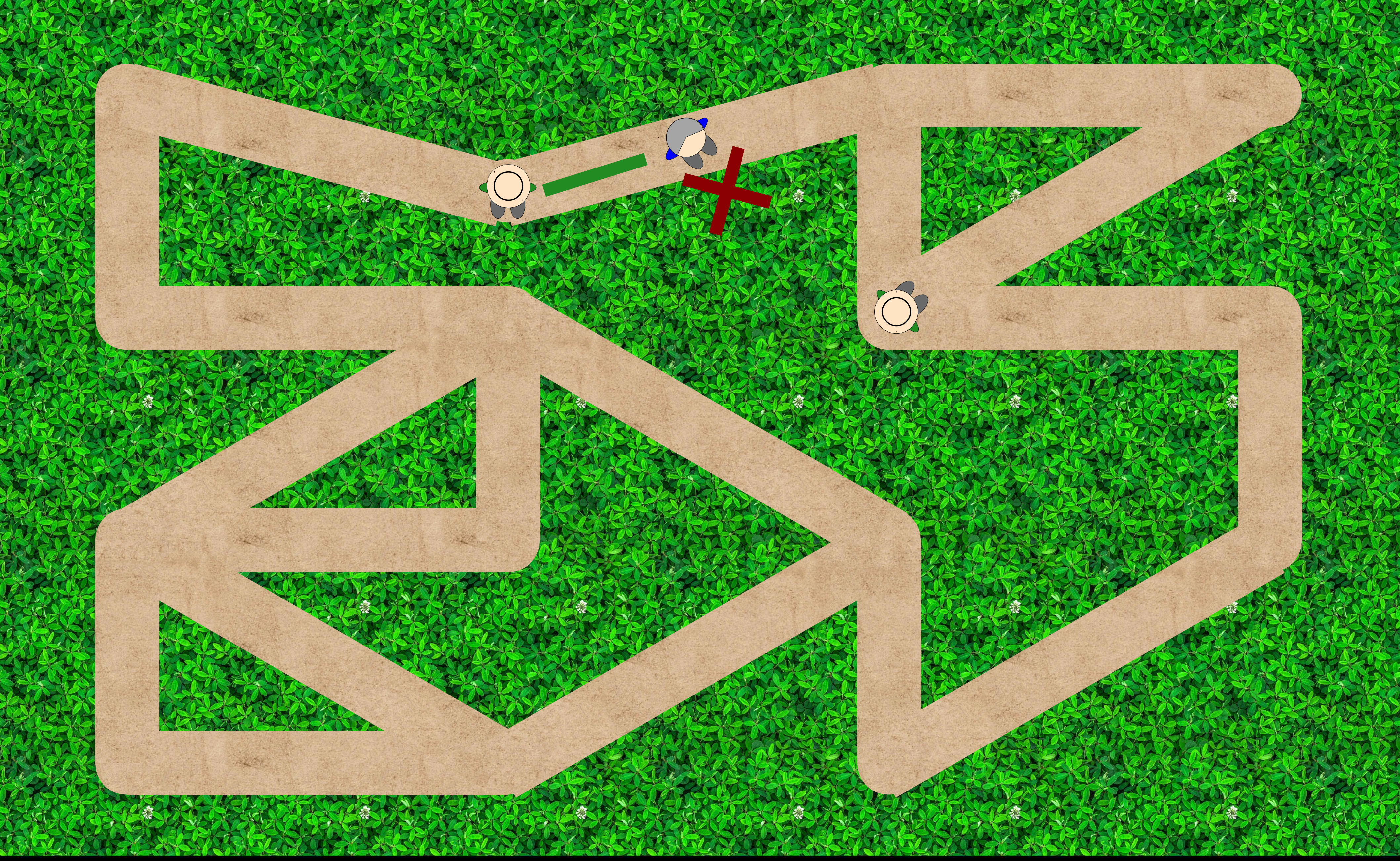
Back to CS103!

Independent Sets and Vertex Covers

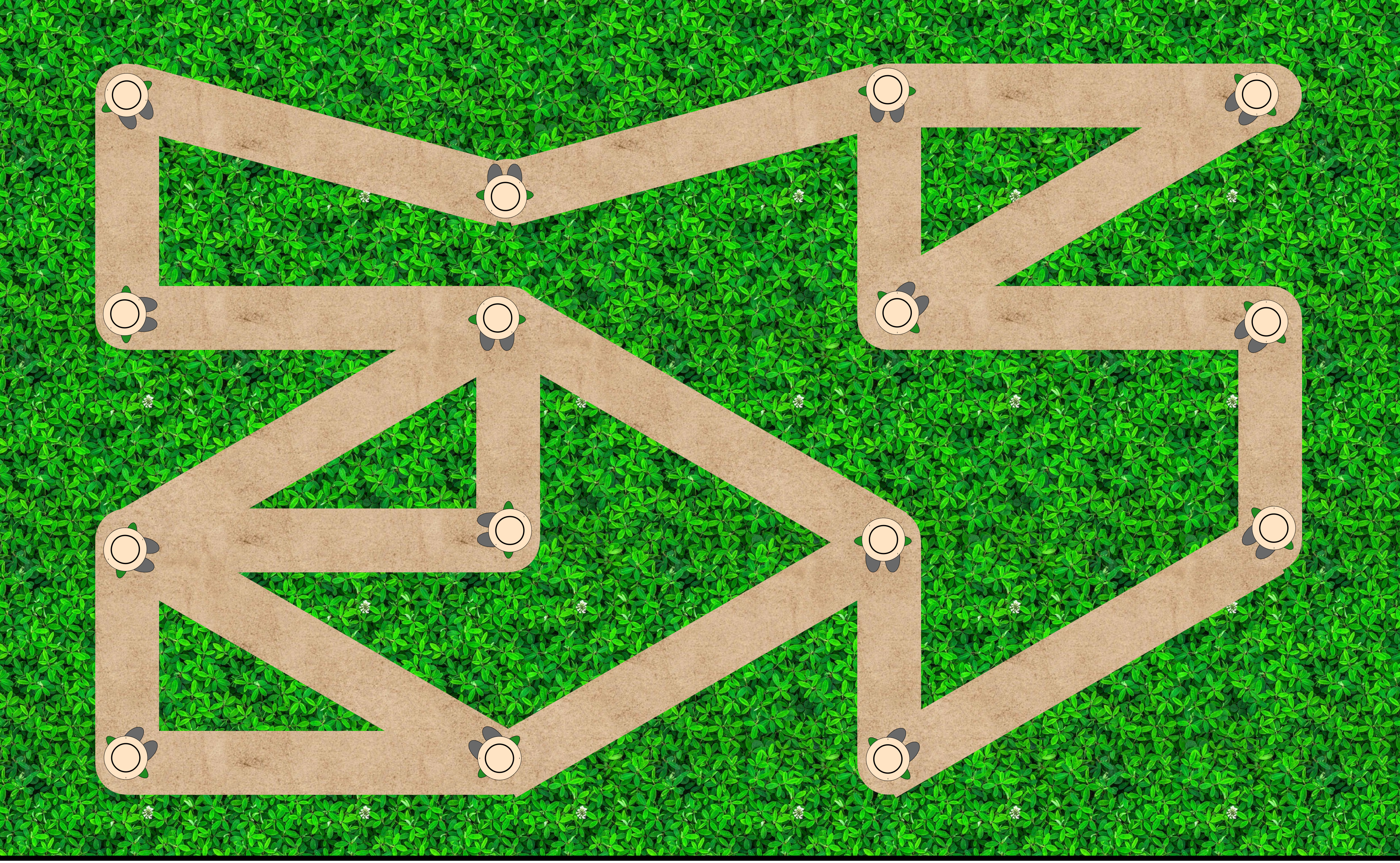
Two Motivating Problems



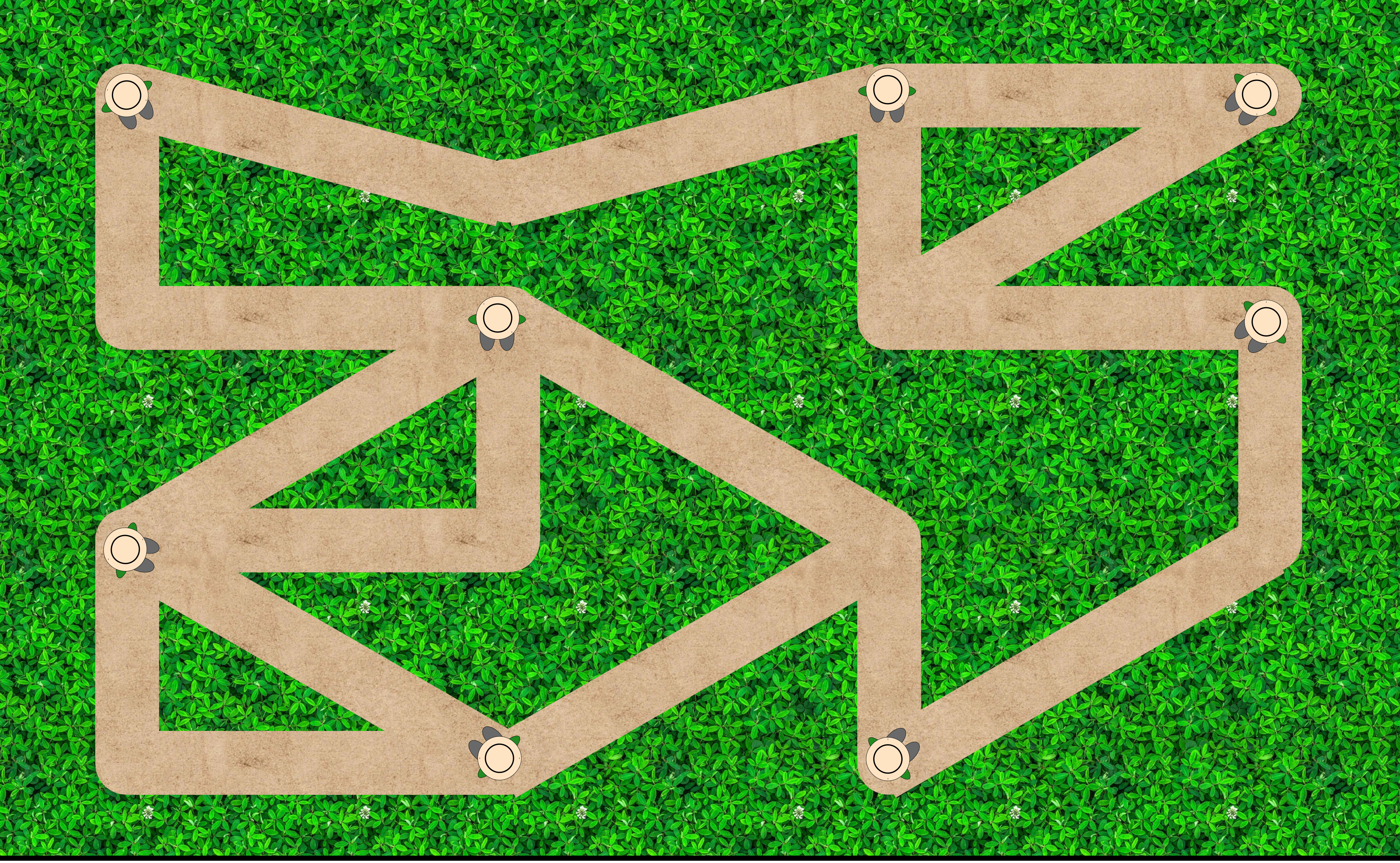
Place park rangers in these forest trails so that
a hiker anywhere on a trail can see a park ranger.



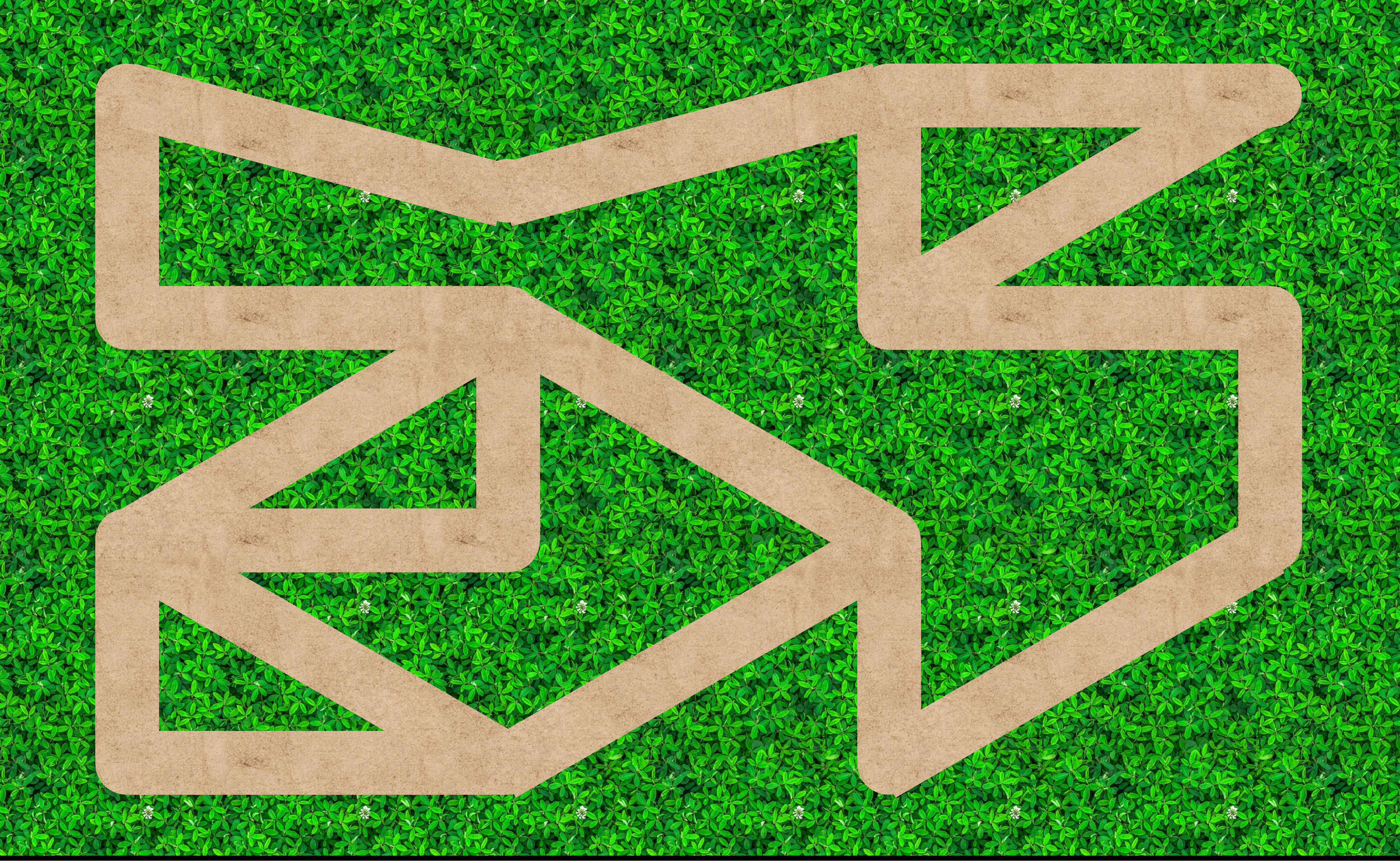
Place park rangers in these forest trails so that
a hiker anywhere on a trail can see a park ranger.



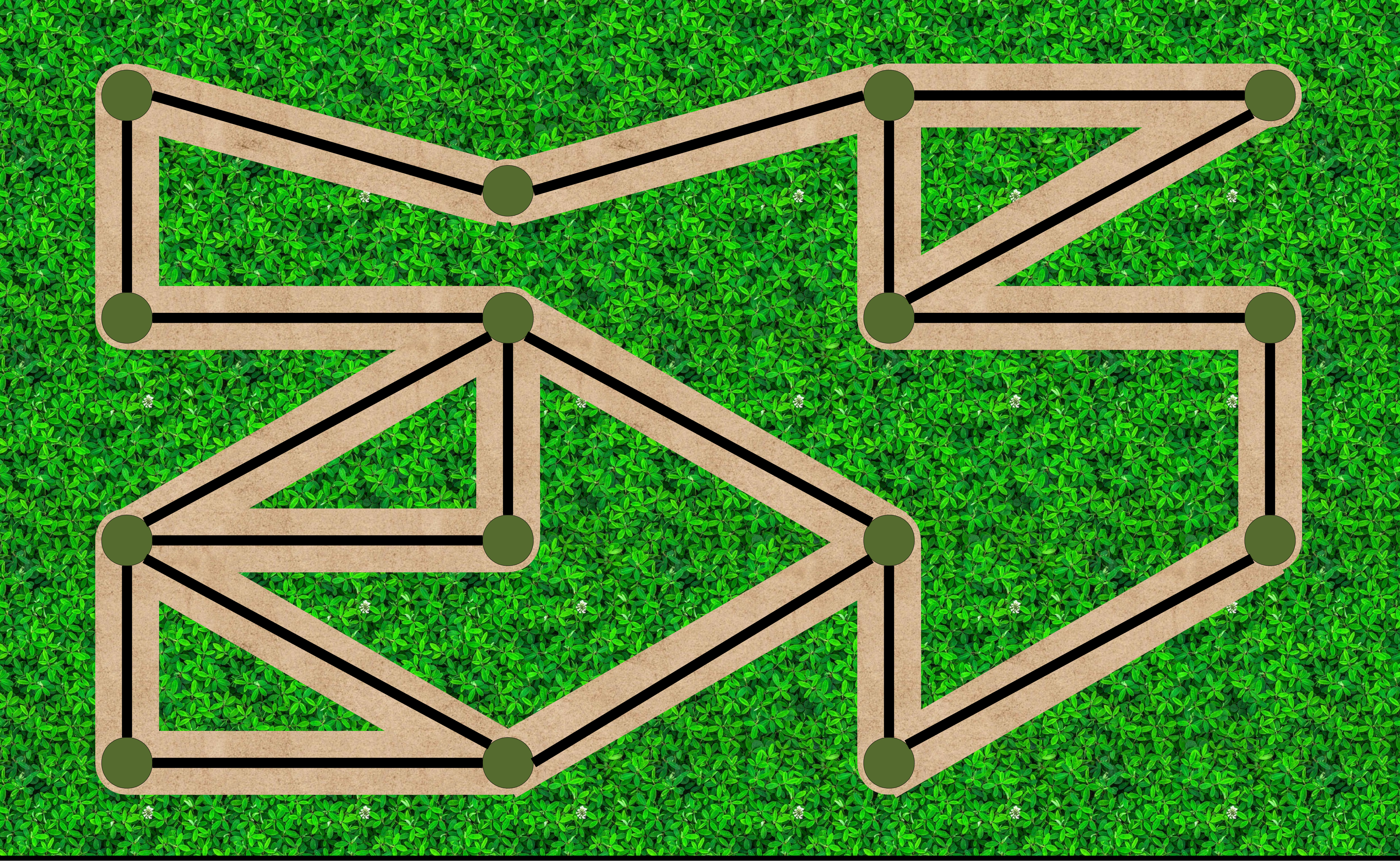
Place park rangers in these forest trails so that
a hiker anywhere on a trail can see a park ranger.



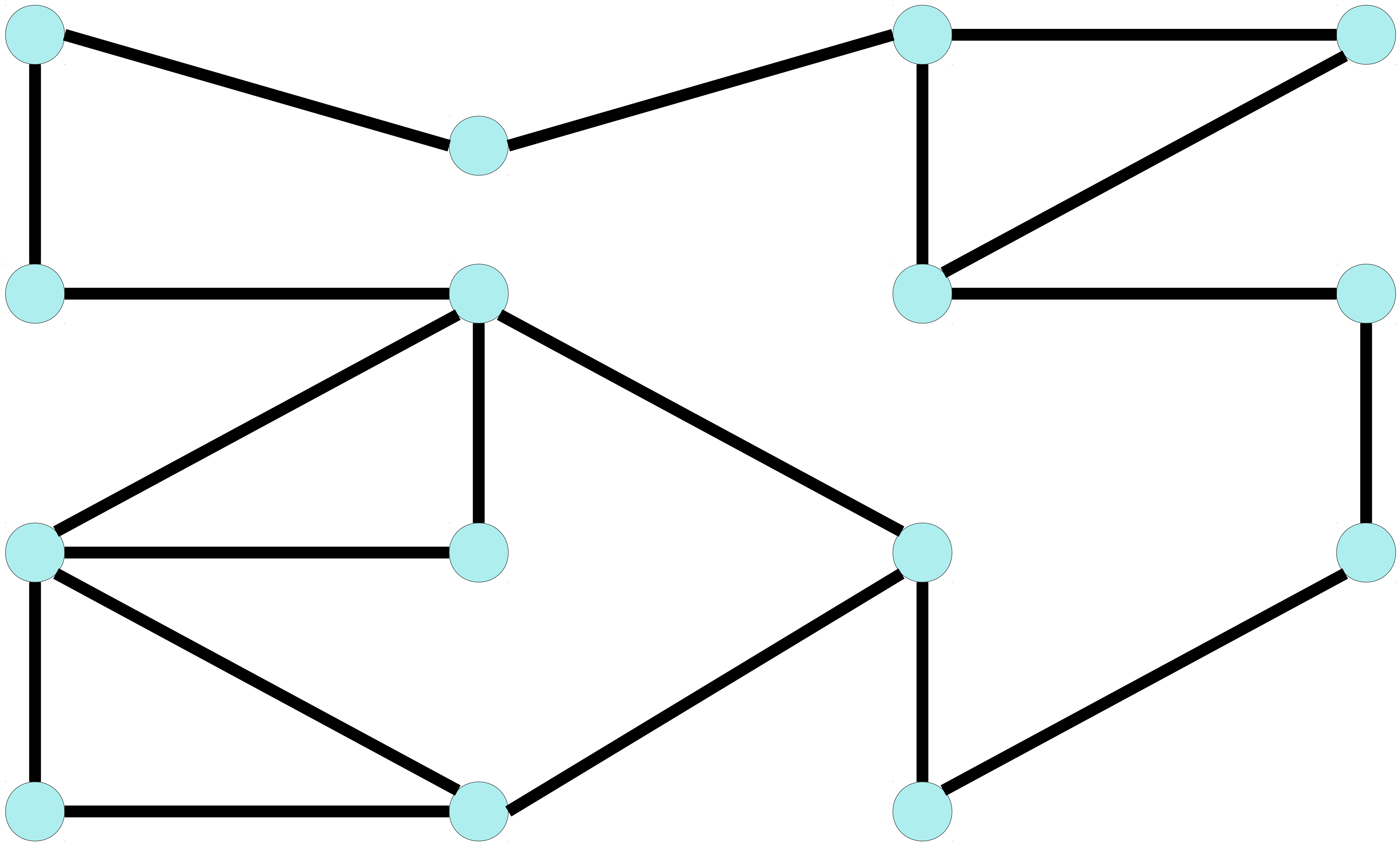
Place park rangers in these forest trails so that
a hiker anywhere on a trail can see a park ranger.



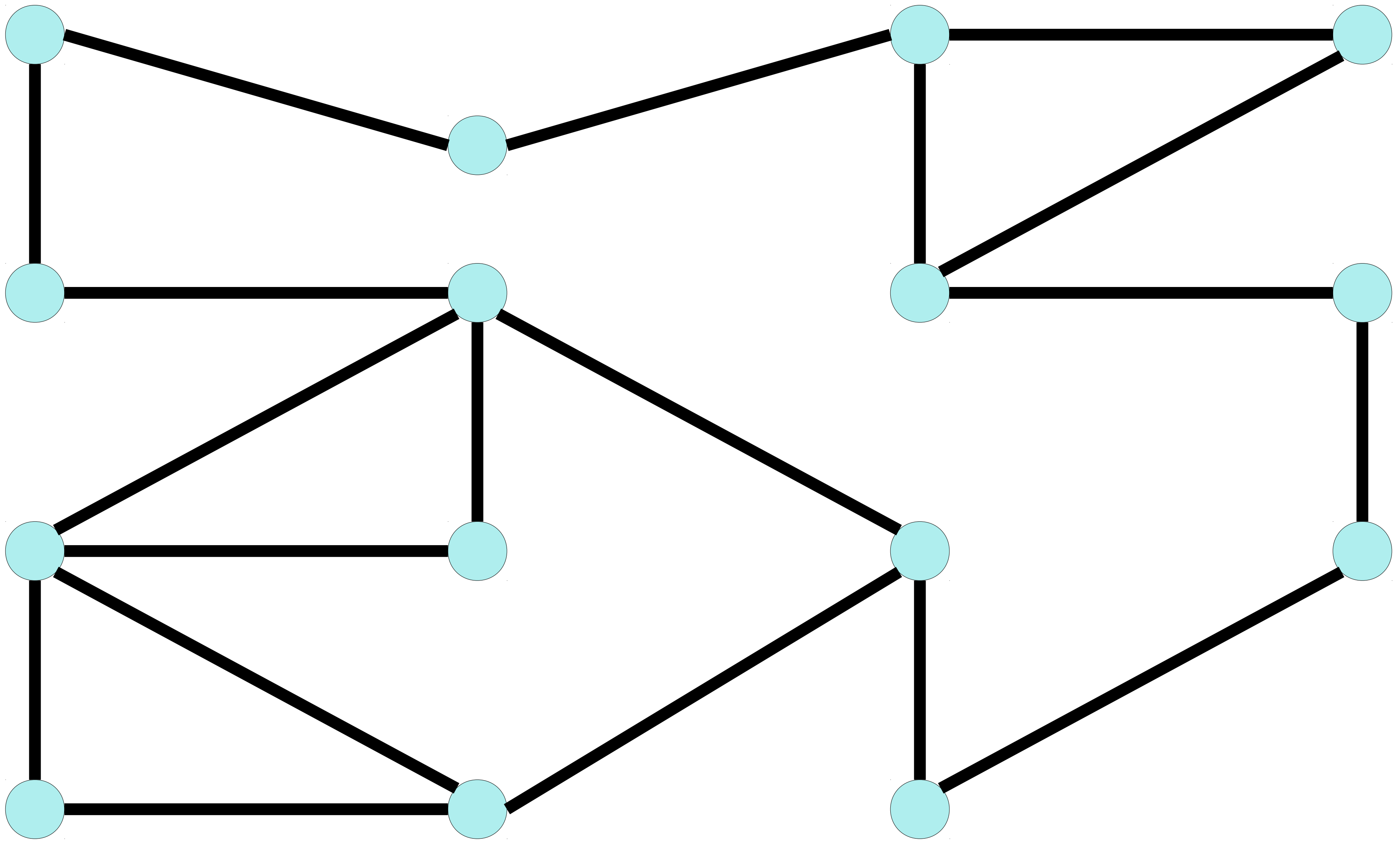
Place park rangers in these forest trails so that
a hiker anywhere on a trail can see a park ranger.



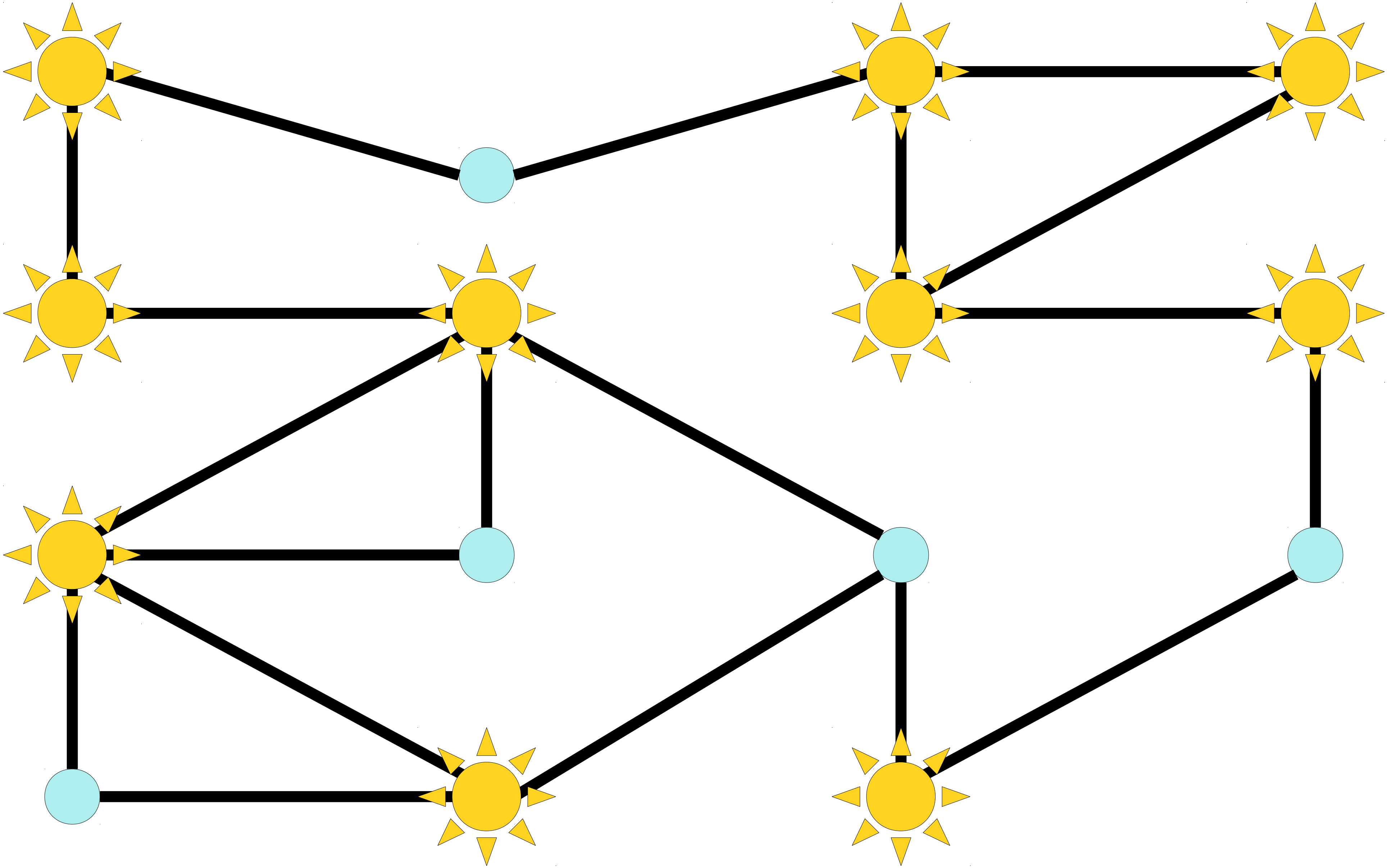
Place park rangers in these forest trails so that
a hiker anywhere on a trail can see a park ranger.



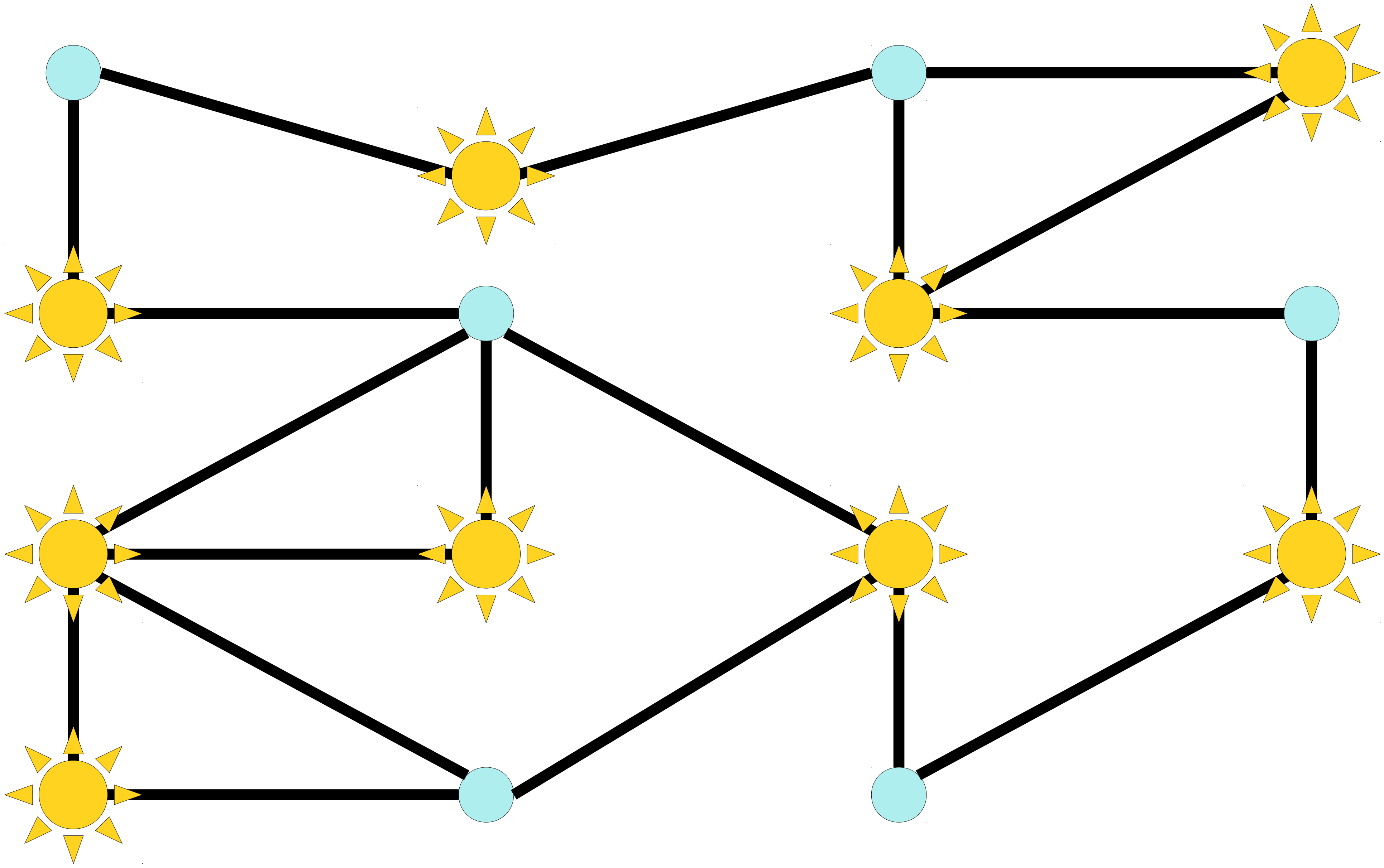
Place park rangers in these forest trails so that
a hiker anywhere on a trail can see a park ranger.



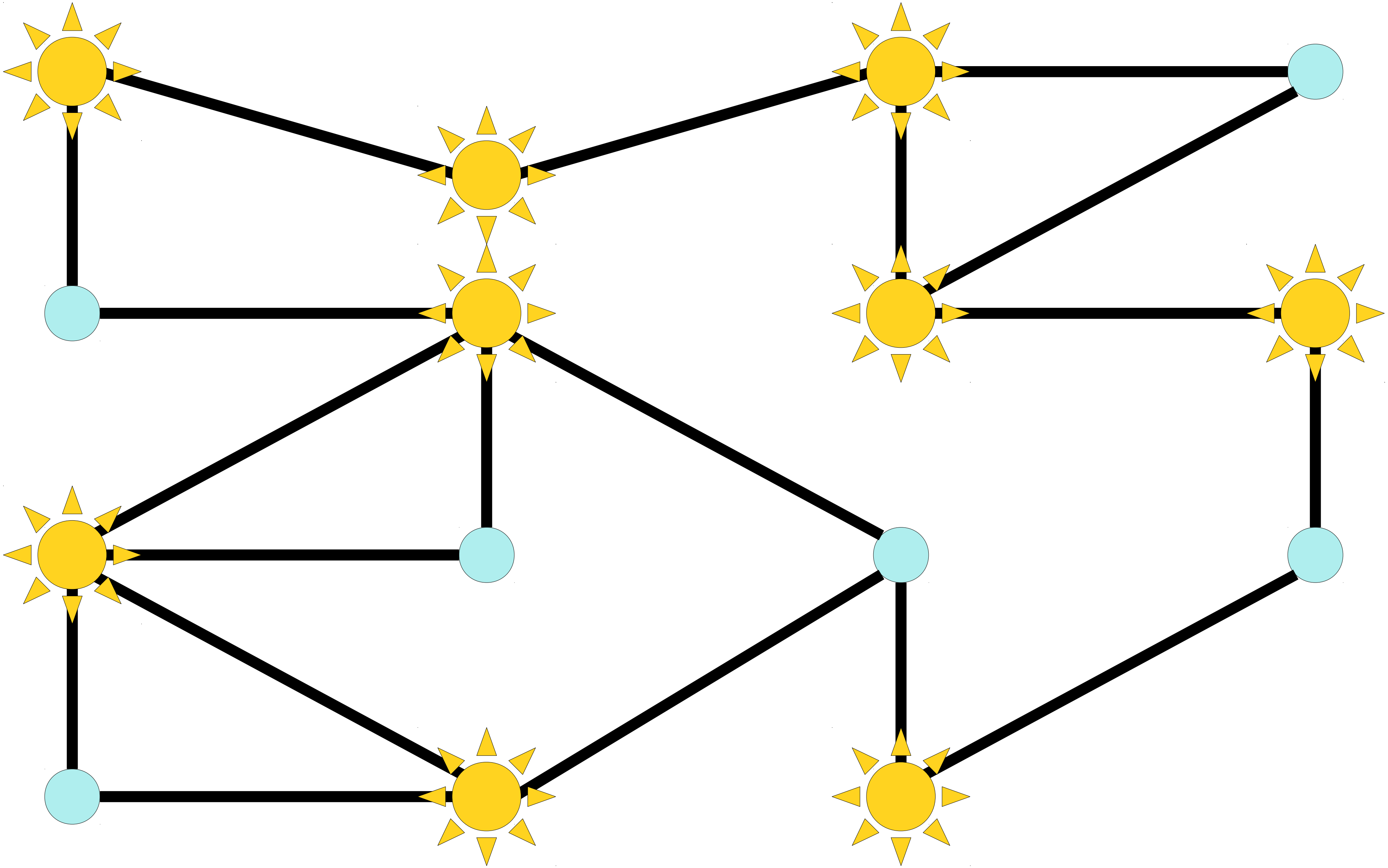
Choose at least one endpoint of each edge.



Choose at least one endpoint of each edge.



Choose at least one endpoint of each edge.



Choose at least one endpoint of each edge.

Vertex Covers

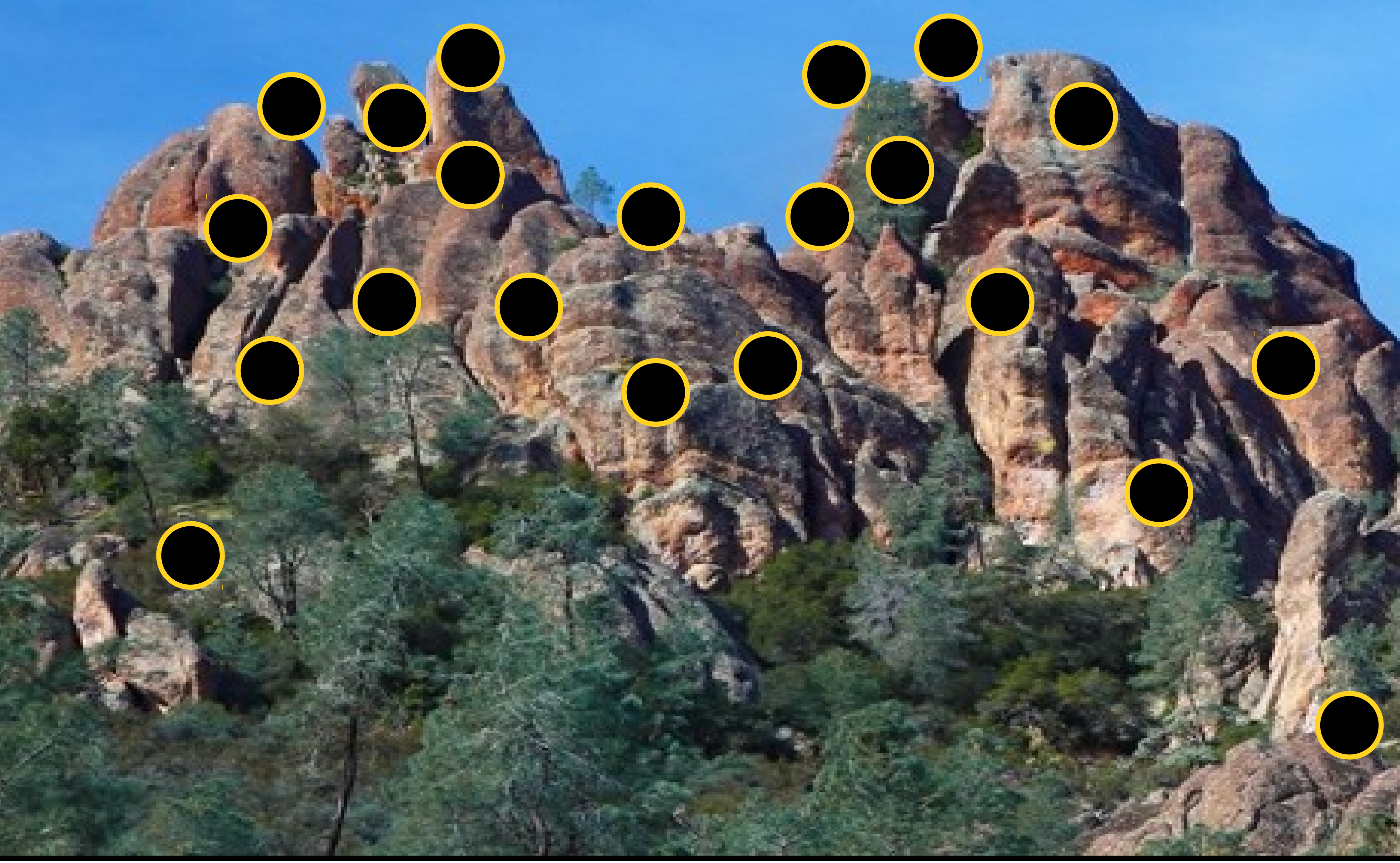
- Let $G = (V, E)$ be an undirected graph. A ***vertex cover*** of G is a set $C \subseteq V$ such that the following statement is true:

$$\forall x \in V.\ \forall y \in V.\ (\{x, y\} \in E \to (x \in C \lor y \in C))$$

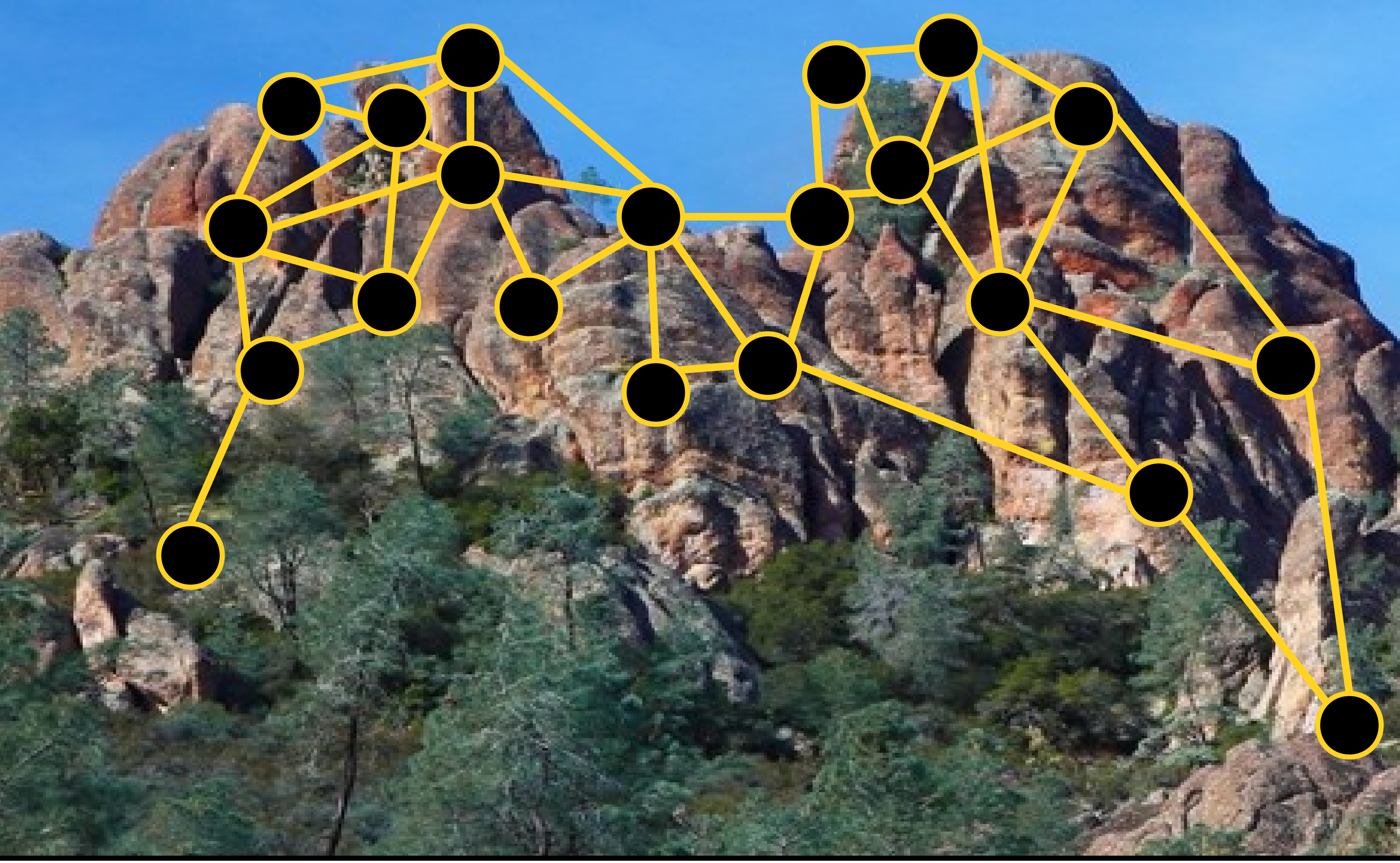
("Every edge has at least one endpoint in C.")

- Intuitively speaking, a vertex cover is a set formed by picking at least one endpoint of each edge in the graph.
- Vertex covers have applications to placing streetlights/benches/security guards, as well as in gene sequencing, machine learning, and combinatorics.

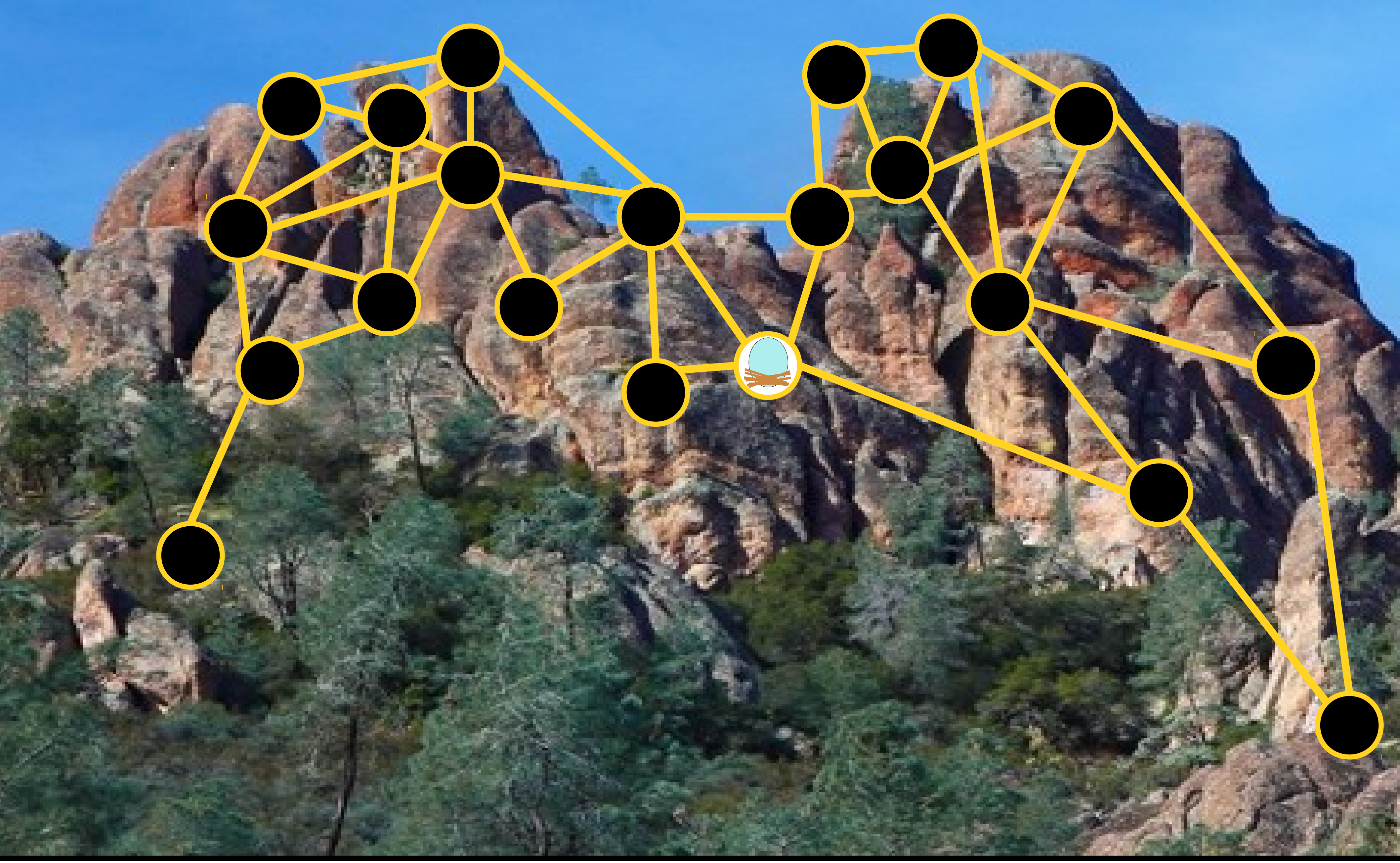
Set up nests for the California condor. Condors are territorial and won’t nest if they can see other condors.



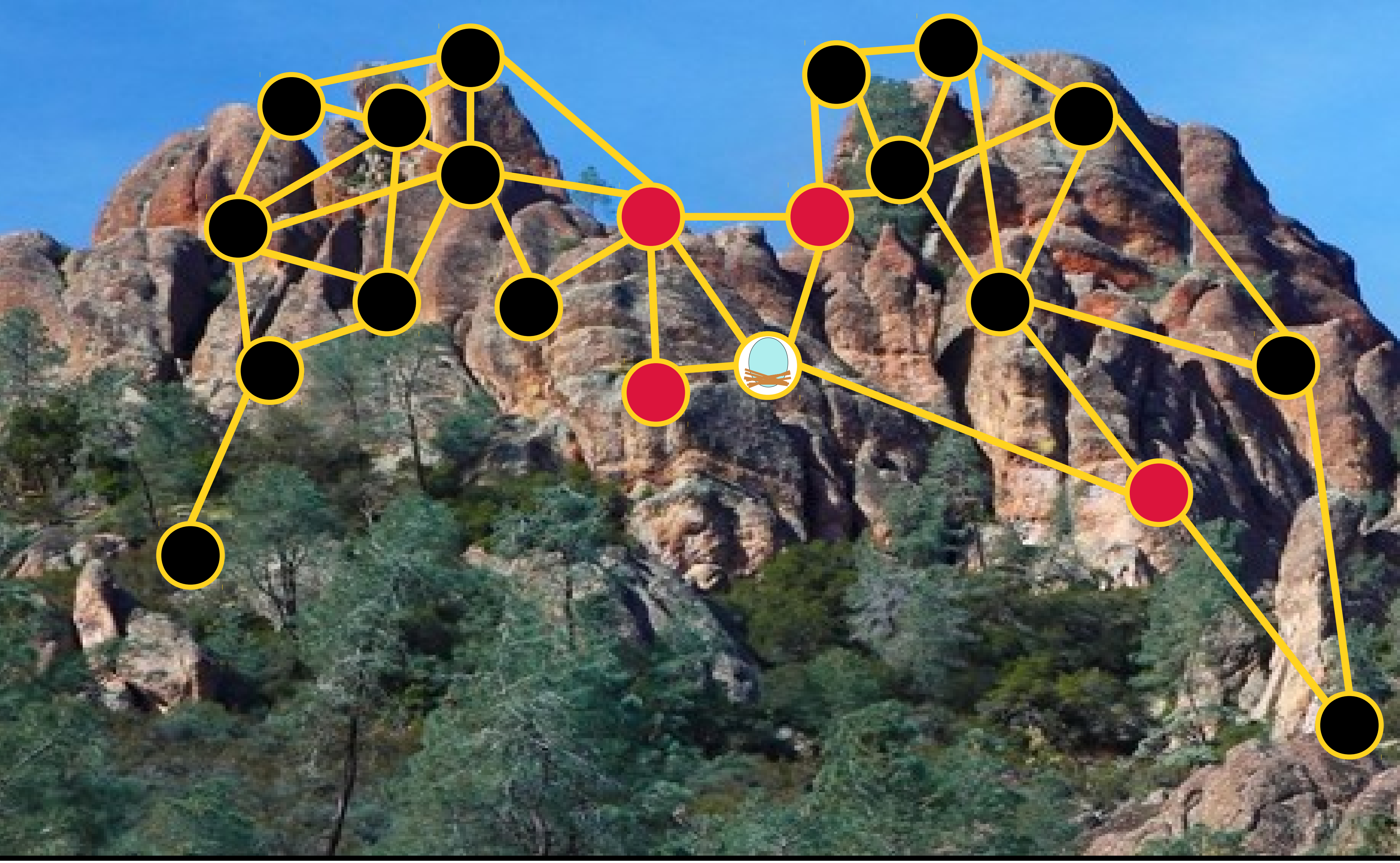
Set up nests for the California condor. Condors are territorial and won’t nest if they can see other condors.



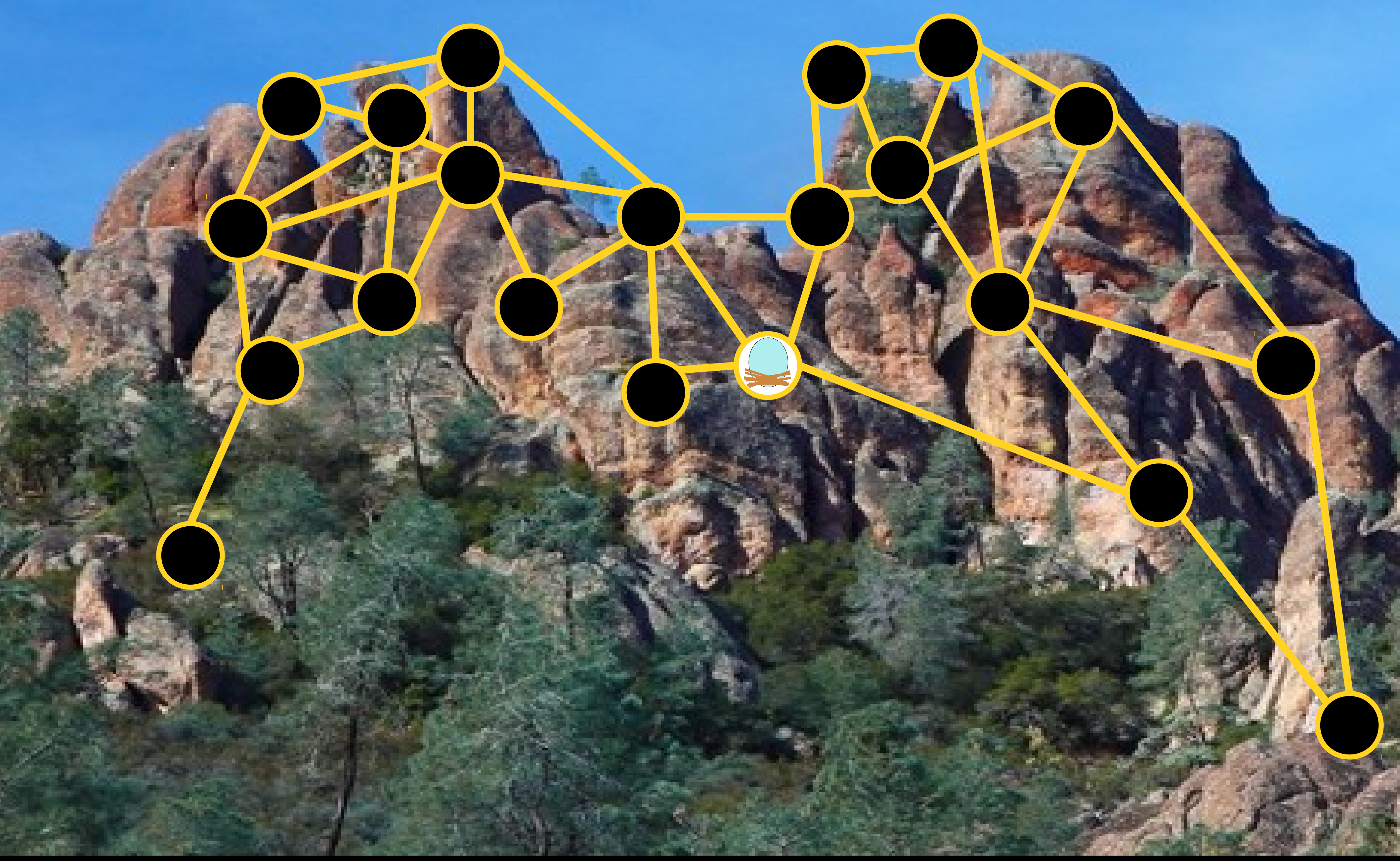
Set up nests for the California condor. Condors are territorial and won't nest if they can see other condors.



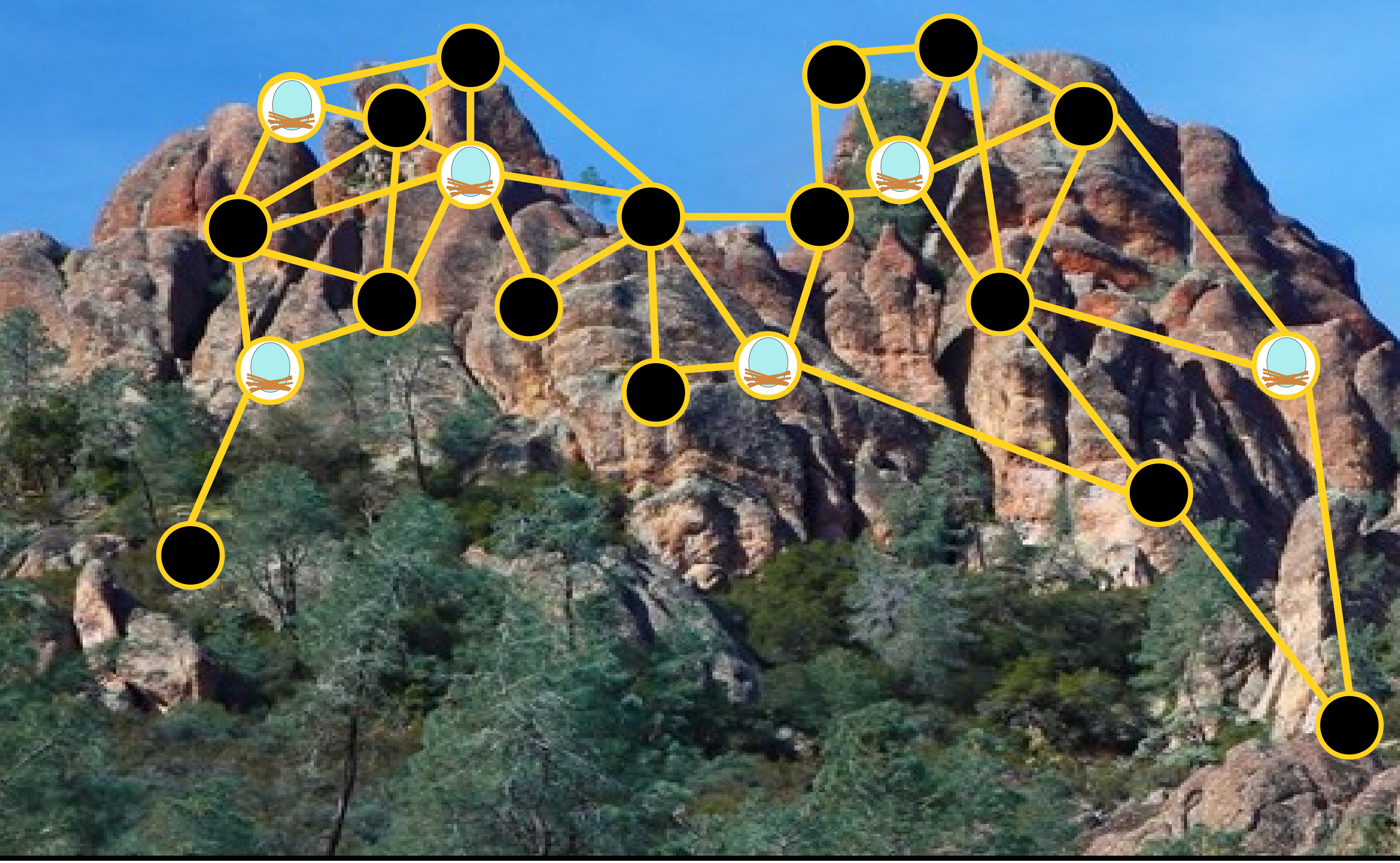
Set up nests for the California condor. Condors are territorial and won't nest if they can see other condors.



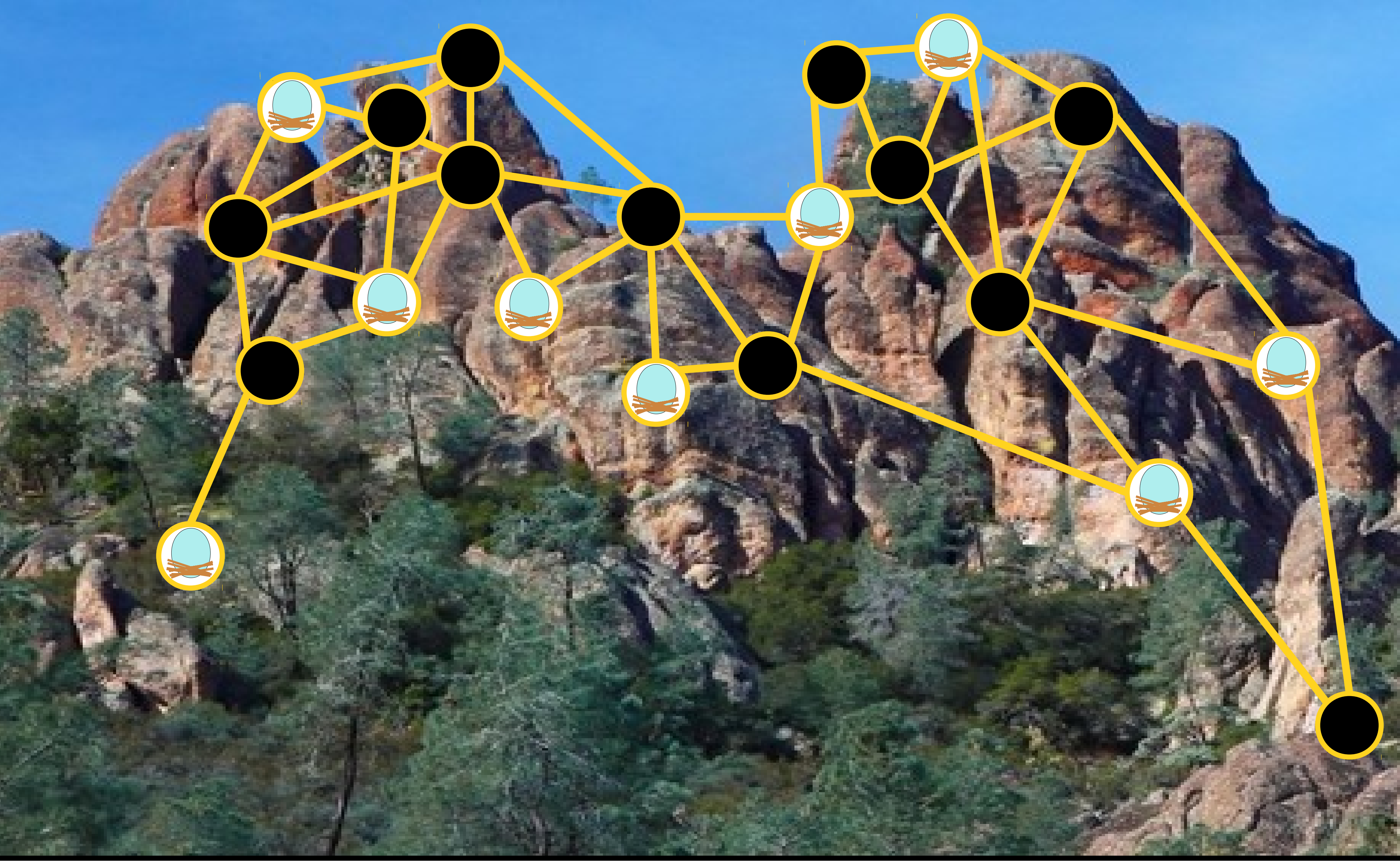
Set up nests for the California condor. Condors are territorial and won’t nest if they can see other condors.



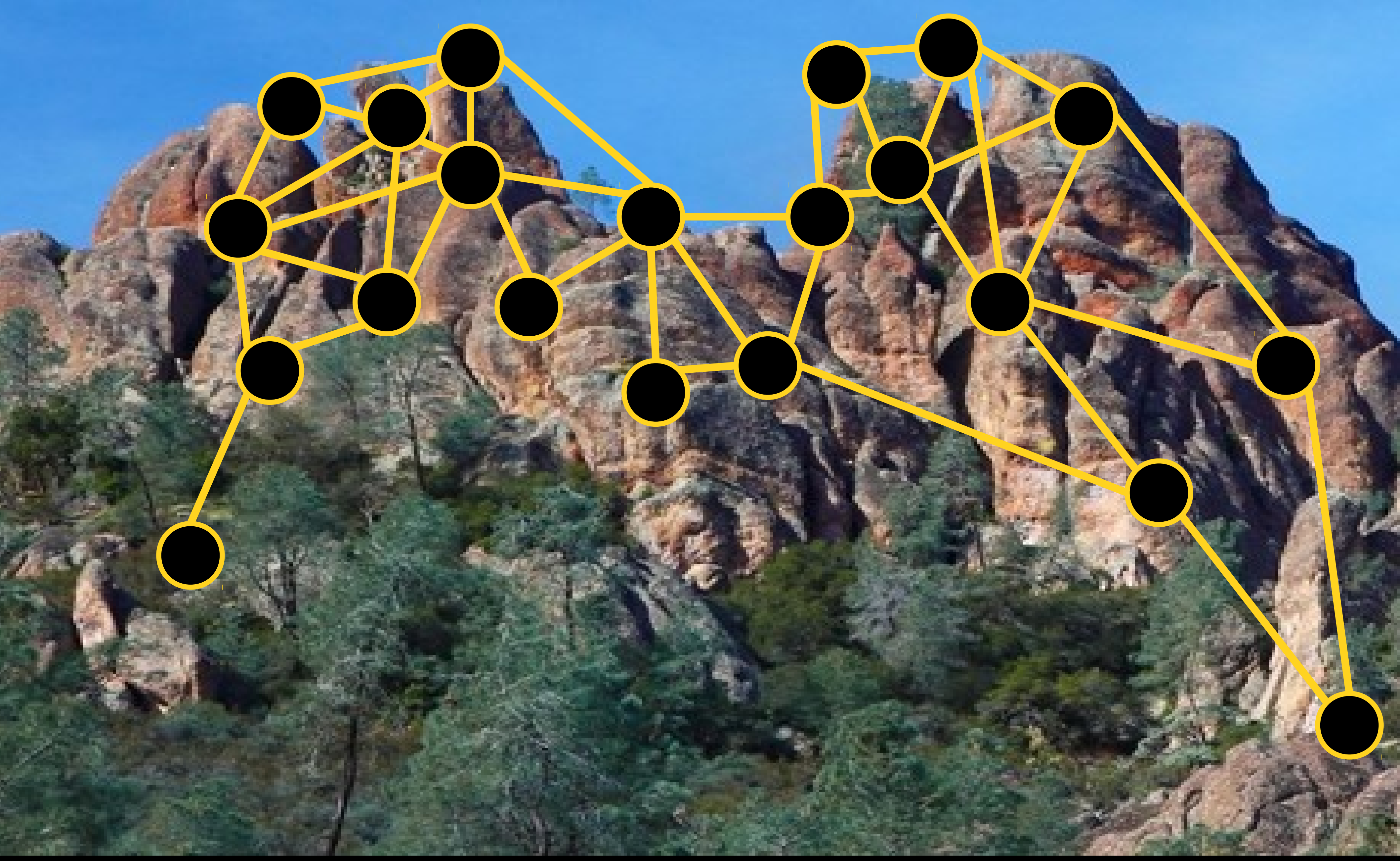
Set up nests for the California condor. Condors are territorial and won’t nest if they can see other condors.



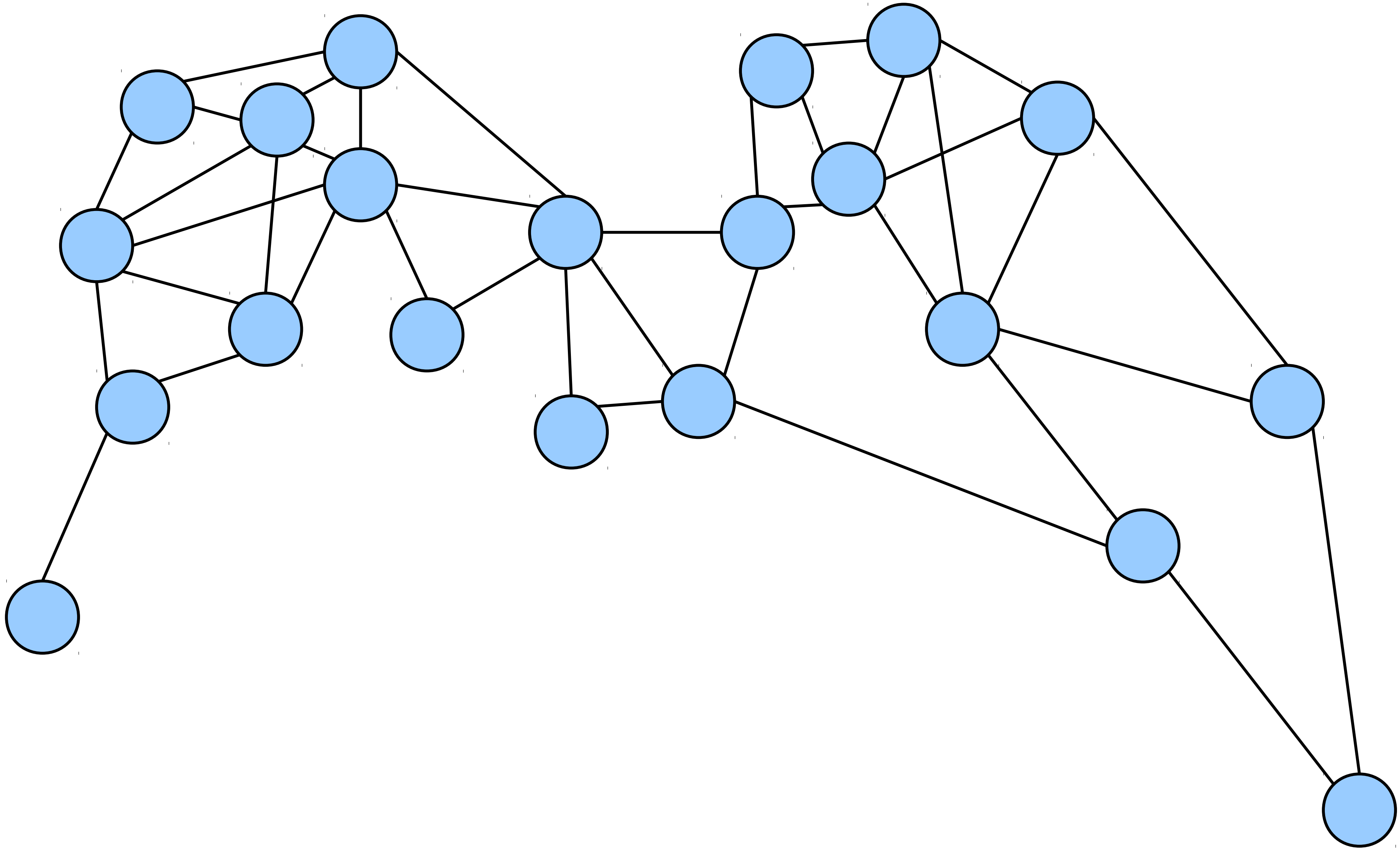
Set up nests for the California condor. Condors are territorial and won't nest if they can see other condors.



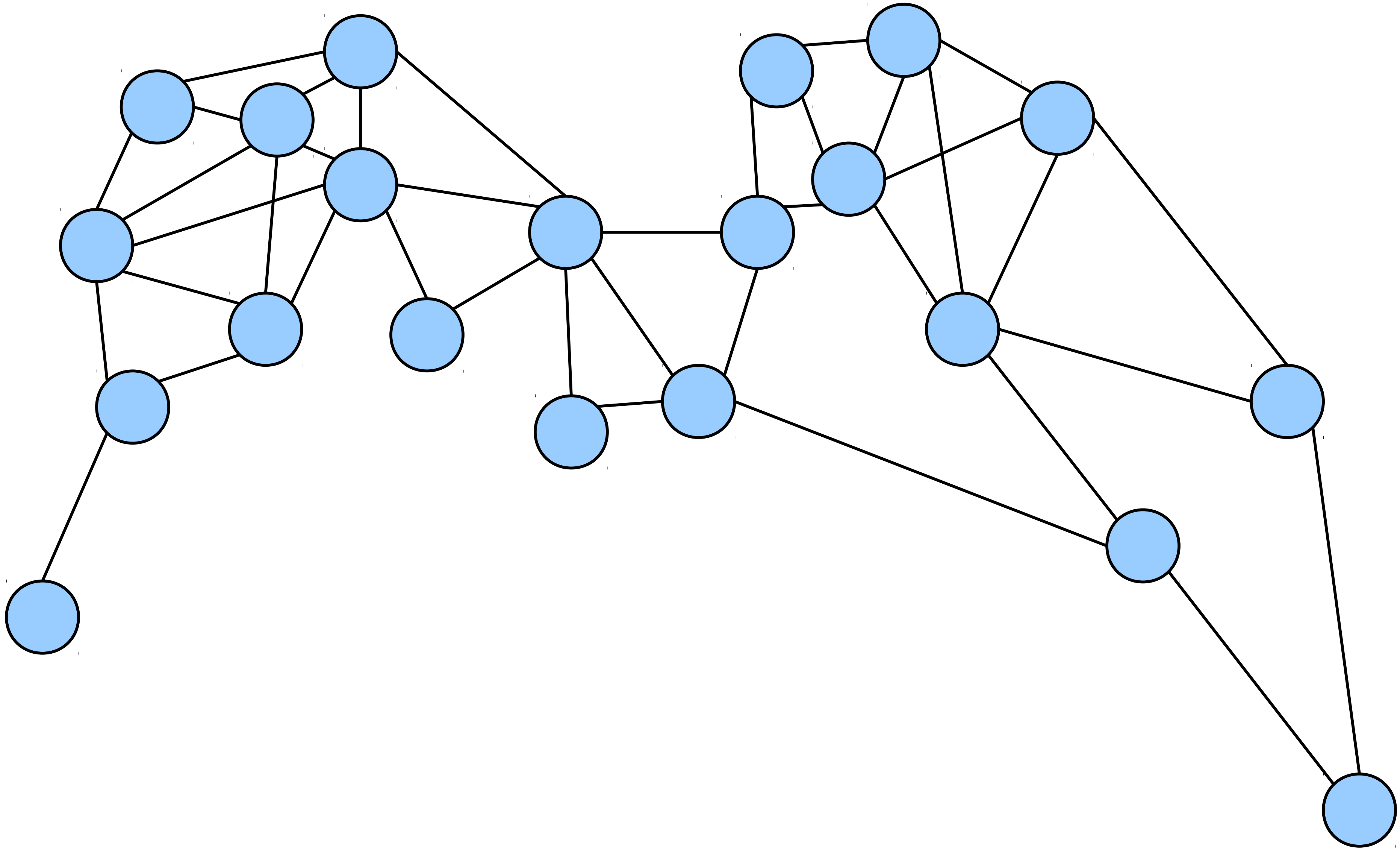
Set up nests for the California condor. Condors are territorial and won’t nest if they can see other condors.



Set up nests for the California condor. Condors are territorial and won't nest if they can see other condors.



Set up nests for the California condor. Condors are territorial and won’t nest if they can see other condors.



Choose a set of nodes, no two of which are adjacent.

Independent Sets

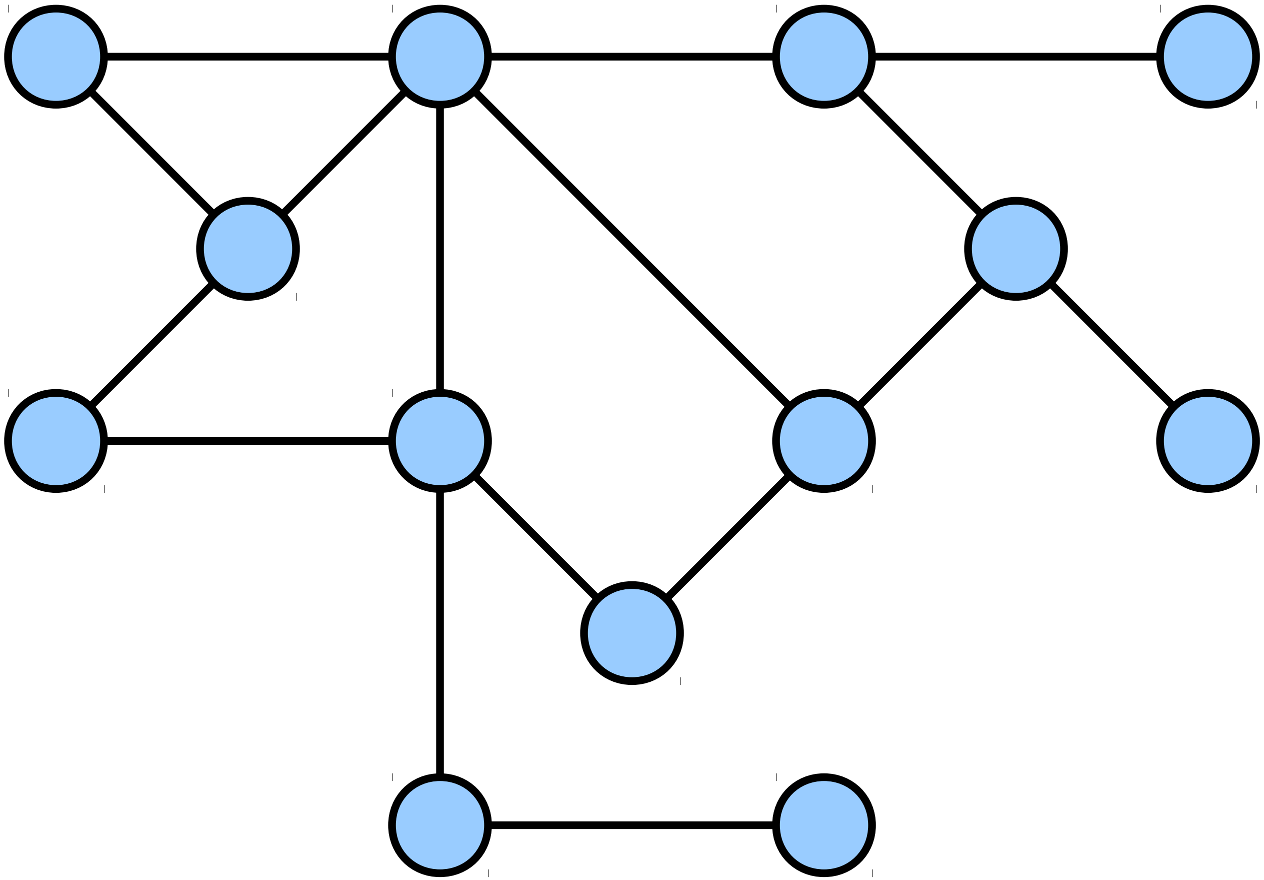
- If $G = (V, E)$ is an (undirected) graph, then an ***independent set*** in G is a set $I \subseteq V$ such that

$$\forall u \in I.\ \forall v \in I.\ \{u, v\} \notin E.$$

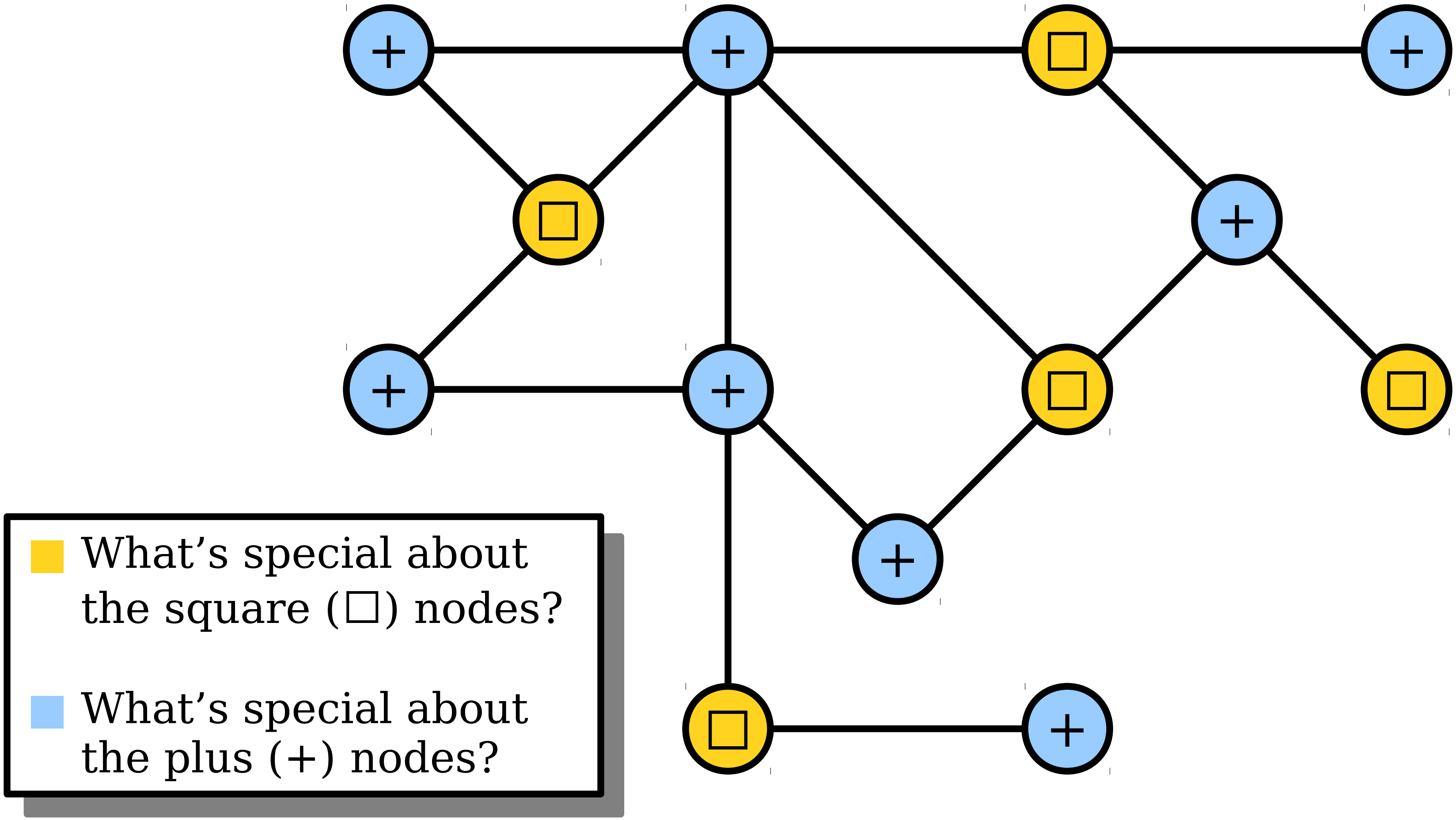
("No two nodes in I are adjacent.")

- Independent sets have applications to resource optimization, conflict minimization, error-correcting codes, cryptography, and more.

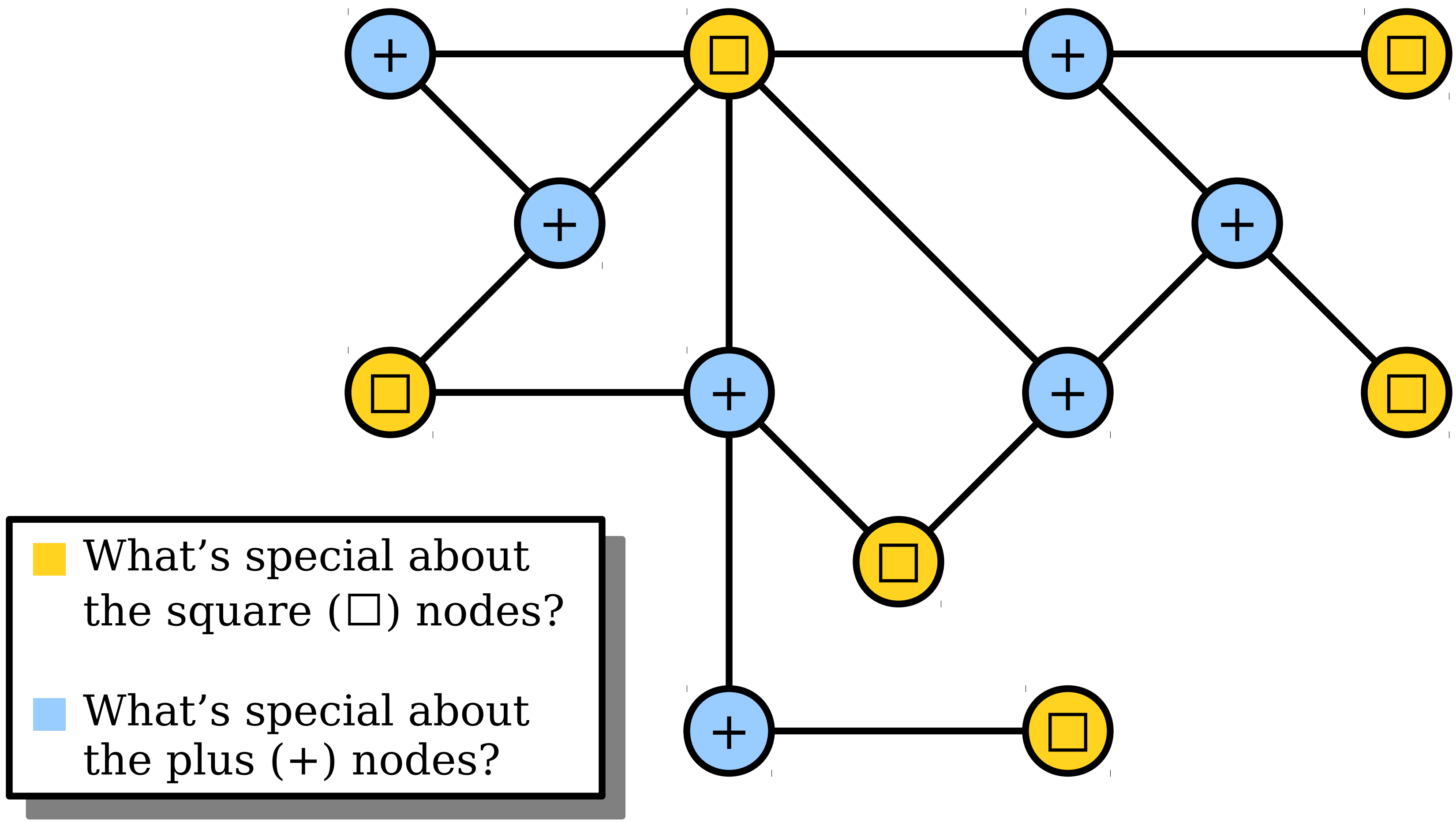
A Connection



Independent sets and vertex covers are related.



Independent sets and vertex covers are related.



Theorem: Let $G = (V, E)$ be a graph and let $C \subseteq V$ be a set. Then C is a vertex cover of G if and only if $V - C$ is an independent set in G.

Lemma 1: Let $G = (V, E)$ be a graph and let $C \subseteq V$ be a set. If C is a vertex cover of G, then $V - C$ is an independent set in G.

What We're Assuming	***What We Need To Show***
G is a graph.	$V - C$ is an independent set in G.
C is a vertex cover of G.	$\forall x \in V - C.$ $\forall y \in V - C.$ $\{x, y\} \notin E.$
$\forall u \in V. \forall v \in V. (\{u, v\} \in E \to u \in C \lor v \in C)$	

Lemma 1: Let $G = (V, E)$ be a graph and let $C \subseteq V$ be a set. If C is a vertex cover of G, then $V - C$ is an independent set in G.

What We're Assuming	***What We Need To Show***
G is a graph.	$V - C$ is an independent set in G.
C is a vertex cover of G.	$\forall x \in V - C.$ $\forall y \in V - C.$ $\{x, y\} \notin E.$
$\forall u \in V. \forall v \in V. (\{u, v\} \in E \to u \in C \lor v \in C)$	
We're assuming a universally-quantified statement. That means we *don't do anything right now* and instead wait for an edge to present itself.	We need to prove a universally-quantified statement. We'll ask the reader to pick arbitrary choices of x and y for us to work with.

Lemma 1: Let $G = (V, E)$ be a graph and let $C \subseteq V$ be a set. If C is a vertex cover of G, then $V - C$ is an independent set in G.

What We're Assuming

G is a graph.

C is a vertex cover of G.

$$\forall u \in V.\ \forall v \in V.\ (\{u, v\} \in E \to u \in C \lor v \in C)$$

What We Need To Show

$V - C$ is an independent set in G.

$$\forall x \in V - C.\ \forall y \in V - C.\ \{x, y\} \notin E.$$

We need to prove a universally-quantified statement. We'll ask the reader to pick arbitrary choices of x and y for us to work with.

Lemma 1: Let $G = (V, E)$ be a graph and let $C \subseteq V$ be a set. If C is a vertex cover of G, then $V - C$ is an independent set in G.

What We're Assuming

G is a graph.

C is a vertex cover of G.

$\forall u \in V.\ \forall v \in V.\ (\{u, v\} \in E \to$
$\quad u \in C \quad \lor \quad v \in C$
$)$

$x \in V - C$.

$y \in V - C$.

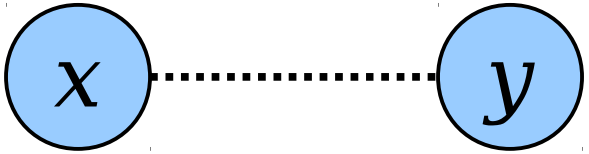
What We Need To Show

$V - C$ is an independent set in G.

$\forall x \in V - C.$

$\forall y \in V - C.$

$\{x, y\} \notin E.$



Lemma 1: Let $G = (V, E)$ be a graph and let $C \subseteq V$ be a set. If C is a vertex cover of G, then $V - C$ is an independent set of G.

Lemma 1: Let $G = (V, E)$ be a graph and let $C \subseteq V$ be a set. If C is a vertex cover of G, then $V - C$ is an independent set of G.

Proof:

Lemma 1: Let $G = (V, E)$ be a graph and let $C \subseteq V$ be a set. If C is a vertex cover of G, then $V - C$ is an independent set of G.

Proof: Assume C is a vertex cover of G.

> There's no need to introduce G or C here. That's done in the statement of the lemma itself.

Lemma 1: Let $G = (V, E)$ be a graph and let $C \subseteq V$ be a set. If C is a vertex cover of G, then $V - C$ is an independent set of G.

Proof: Assume C is a vertex cover of G. We need to show that $V - C$ is an independent set of G.

Lemma 1: Let $G = (V, E)$ be a graph and let $C \subseteq V$ be a set. If C is a vertex cover of G, then $V - C$ is an independent set of G.

Proof: Assume C is a vertex cover of G. We need to show that $V - C$ is an independent set of G. To do so, pick any nodes $x, y \in V - C$; we will show that $\{x, y\} \notin E$.

Lemma 1: Let $G = (V, E)$ be a graph and let $C \subseteq V$ be a set. If C is a vertex cover of G, then $V - C$ is an independent set of G.

Proof: Assume C is a vertex cover of G. We need to show that $V - C$ is an independent set of G. To do so, pick any nodes $x, y \in V - C$; we will show that $\{x, y\} \notin E$.

Suppose for the sake of contradiction that $\{x, y\} \in E$.

Lemma 1: Let $G = (V, E)$ be a graph and let $C \subseteq V$ be a set. If C is a vertex cover of G, then $V - C$ is an independent set of G.

Proof: Assume C is a vertex cover of G. We need to show that $V - C$ is an independent set of G. To do so, pick any nodes $x, y \in V - C$; we will show that $\{x, y\} \notin E$.

Suppose for the sake of contradiction that $\{x, y\} \in E$. Because $x, y \in V - C$, we know that $x \notin C$ and $y \notin C$.

Lemma 1: Let $G = (V, E)$ be a graph and let $C \subseteq V$ be a set. If C is a vertex cover of G, then $V - C$ is an independent set of G.

Proof: Assume C is a vertex cover of G. We need to show that $V - C$ is an independent set of G. To do so, pick any nodes $x, y \in V - C$; we will show that $\{x, y\} \notin E$.

Suppose for the sake of contradiction that $\{x, y\} \in E$. Because $x, y \in V - C$, we know that $x \notin C$ and $y \notin C$. However, since C is a vertex cover of G and $\{x, y\} \in E$, we also see that $x \in C$ or $y \in C$, contradicting our previous statement.

Lemma 1: Let $G = (V, E)$ be a graph and let $C \subseteq V$ be a set. If C is a vertex cover of G, then $V - C$ is an independent set of G.

Proof: Assume C is a vertex cover of G. We need to show that $V - C$ is an independent set of G. To do so, pick any nodes $x, y \in V - C$; we will show that $\{x, y\} \notin E$.

Suppose for the sake of contradiction that $\{x, y\} \in E$. Because $x, y \in V - C$, we know that $x \notin C$ and $y \notin C$. However, since C is a vertex cover of G and $\{x, y\} \in E$, we also see that $x \in C$ or $y \in C$, contradicting our previous statement.

We've reached a contradiction, so our assumption was wrong.

Lemma 1: Let $G = (V, E)$ be a graph and let $C \subseteq V$ be a set. If C is a vertex cover of G, then $V - C$ is an independent set of G.

Proof: Assume C is a vertex cover of G. We need to show that $V - C$ is an independent set of G. To do so, pick any nodes $x, y \in V - C$; we will show that $\{x, y\} \notin E$.

Suppose for the sake of contradiction that $\{x, y\} \in E$. Because $x, y \in V - C$, we know that $x \notin C$ and $y \notin C$. However, since C is a vertex cover of G and $\{x, y\} \in E$, we also see that $x \in C$ or $y \in C$, contradicting our previous statement.

We've reached a contradiction, so our assumption was wrong. Therefore, we have $\{x, y\} \notin E$, as required.

Lemma 1: Let $G = (V, E)$ be a graph and let $C \subseteq V$ be a set. If C is a vertex cover of G, then $V - C$ is an independent set of G.

Proof: Assume C is a vertex cover of G. We need to show that $V - C$ is an independent set of G. To do so, pick any nodes $x, y \in V - C$; we will show that $\{x, y\} \notin E$.

Suppose for the sake of contradiction that $\{x, y\} \in E$. Because $x, y \in V - C$, we know that $x \notin C$ and $y \notin C$. However, since C is a vertex cover of G and $\{x, y\} \in E$, we also see that $x \in C$ or $y \in C$, contradicting our previous statement.

We've reached a contradiction, so our assumption was wrong. Therefore, we have $\{x, y\} \notin E$, as required. ■

Taking Negations

- What is the negation of this statement, which says "C is a vertex cover?"

$$\forall u \in C.\ \forall v \in C.\ (\{u, v\} \in E \to$$
$$u \in C \quad \lor \quad v \in C$$
$$)$$

Respond at pollev.com/cs103

Taking Negations

- What is the negation of this statement, which says "C is a vertex cover?"

$$\neg\forall u \in C.\ \forall v \in C.\ (\{u, v\} \in E \rightarrow$$
$$u \in C \quad \lor \quad v \in C$$
$$)$$

Taking Negations

- What is the negation of this statement, which says "*C* is a vertex cover?"

$$\exists u \in C.\ \neg \forall v \in C.\ (\{u, v\} \in E \to$$
$$u \in C \quad \vee \quad v \in C$$
$$)$$

Taking Negations

- What is the negation of this statement, which says "*C* is a vertex cover?"

$$\exists u \in C.\ \exists v \in C.\ \neg(\{u, v\} \in E \rightarrow$$
$$u \in C \quad \vee \quad v \in C$$
$$)$$

Taking Negations

- What is the negation of this statement, which says "C is a vertex cover?"

$$\exists u \in C.\ \exists v \in C.\ (\{u, v\} \in E \land$$
$$\neg(u \in C \quad \lor \quad v \in C)$$
$$)$$

Taking Negations

- What is the negation of this statement, which says "C is a vertex cover?"

$$
\begin{aligned}
&\exists u \in C.\ \exists v \in C.\ (\{u, v\} \in E \land \\
&\quad u \notin C \quad \land \quad v \notin C \\
&)
\end{aligned}
$$

Taking Negations

- What is the negation of this statement, which says "*C* is a vertex cover?"

$$\exists u \in C.\ \exists v \in C.\ (\{u, v\} \in E\ \land$$
$$u \notin C \quad \land \quad v \notin C$$
$$)$$

- This says "there is an edge where both endpoints aren't in *C*."

Lemma 2: Let $G = (V, E)$ be a graph and let $C \subseteq V$ be a set. If C is not a vertex cover of G, then $V - C$ is not an independent set in G.

What We're Assuming	What We Need To Show
G is a graph.	$V - C$ is not an ind. set in G.
C is a not a vertex cover of G.	$\exists x \in V - C.$
$\exists u \in V. \exists v \in V. (\{u, v\} \in E \land$	$\exists y \in V - C.$
$u \notin C \quad \land \quad v \notin C$	$\{x, y\} \in E.$
$)$	

Lemma 2: Let $G = (V, E)$ be a graph and let $C \subseteq V$ be a set. If C is not a vertex cover of G, then $V - C$ is not an independent set in G.

What We're Assuming	*What We Need To Show*
G is a graph.	$V - C$ is not an ind. set in G.
C is a not a vertex cover of G.	$\exists x \in V - C.$ $\exists y \in V - C.$ $\{x, y\} \in E.$
$\exists u \in V. \exists v \in V. (\{u, v\} \in E \land u \notin C \land v \notin C)$	

We're assuming an existentially-quantified statement, so we'll *immediately* introduce variables u and v.

We're proving an existentially-quantified statement, so we *don't* introduce variables x and y. We're on a scavenger hunt!

Lemma 2: Let $G = (V, E)$ be a graph and let $C \subseteq V$ be a set. If C is not a vertex cover of G, then $V - C$ is not an independent set in G.

What We're Assuming	What We Need To Show
G is a graph.	$V - C$ is not an ind. set in G.
C is a not a vertex cover of G.	$\exists x \in V - C$.
$u \in V - C$.	$\exists y \in V - C$.
$v \in V - C$.	$\{x, y\} \in E$.
$\{u, v\} \in E$.	

We're assuming an existentially-quantified statement, so we'll *immediately* introduce variables u and v.

Lemma 2: Let $G = (V, E)$ be a graph and let $C \subseteq V$ be a set. If C is not a vertex cover of G, then $V - C$ is not an independent set in G.

What We're Assuming	What We Need To Show
G is a graph.	$V - C$ is not an ind. set in G.
C is a not a vertex cover of G.	$\exists x \in V - C.$
$u \in V - C.$	$\exists y \in V - C.$
$v \in V - C.$	$\{x, y\} \in E.$
$\{u, v\} \in E.$	

Any ideas about what we should pick x and y to be?

Lemma 2: Let $G = (V, E)$ be a graph and let $C \subseteq V$ be a set. If C is not a vertex cover of G, then $V – C$ is not an independent set of G.

Lemma 2: Let $G = (V, E)$ be a graph and let $C \subseteq V$ be a set. If C is not a vertex cover of G, then $V - C$ is not an independent set of G.

Proof:

Lemma 2: Let $G = (V, E)$ be a graph and let $C \subseteq V$ be a set. If C is not a vertex cover of G, then $V - C$ is not an independent set of G.

Proof: Assume C is not a vertex cover of G.

Lemma 2: Let $G = (V, E)$ be a graph and let $C \subseteq V$ be a set. If C is not a vertex cover of G, then $V – C$ is not an independent set of G.

Proof: Assume C is not a vertex cover of G. We need to show that $V – C$ is not an independent set of G.

Lemma 2: Let $G = (V, E)$ be a graph and let $C \subseteq V$ be a set. If C is not a vertex cover of G, then $V - C$ is not an independent set of G.

Proof: Assume C is not a vertex cover of G. We need to show that $V - C$ is not an independent set of G.

Since C is not a vertex cover of G, we know that there exists nodes $x, y \in V$ where $\{x, y\} \in E$, where $x \notin C$, and where $y \notin C$.

Lemma 2: Let $G = (V, E)$ be a graph and let $C \subseteq V$ be a set. If C is not a vertex cover of G, then $V - C$ is not an independent set of G.

Proof: Assume C is not a vertex cover of G. We need to show that $V - C$ is not an independent set of G.

Since C is not a vertex cover of G, we know that there exists nodes $x, y \in V$ where $\{x, y\} \in E$, where $x \notin C$, and where $y \notin C$. Because $x \in V$ and $x \notin C$, we know that $x \in V - C$.

Lemma 2: Let $G = (V, E)$ be a graph and let $C \subseteq V$ be a set. If C is not a vertex cover of G, then $V - C$ is not an independent set of G.

Proof: Assume C is not a vertex cover of G. We need to show that $V - C$ is not an independent set of G.

Since C is not a vertex cover of G, we know that there exists nodes $x, y \in V$ where $\{x, y\} \in E$, where $x \notin C$, and where $y \notin C$. Because $x \in V$ and $x \notin C$, we know that $x \in V - C$. Similarly, we see that $y \in V - C$.

Lemma 2: Let $G = (V, E)$ be a graph and let $C \subseteq V$ be a set. If C is not a vertex cover of G, then $V - C$ is not an independent set of G.

Proof: Assume C is not a vertex cover of G. We need to show that $V - C$ is not an independent set of G.

Since C is not a vertex cover of G, we know that there exists nodes $x, y \in V$ where $\{x, y\} \in E$, where $x \notin C$, and where $y \notin C$. Because $x \in V$ and $x \notin C$, we know that $x \in V - C$. Similarly, we see that $y \in V - C$.

This means that $\{x, y\} \in E$, that $x \in V - C$, and that $y \in V - C$, and therefore that $V - C$ is not an independent set of G, as required.

Lemma 2: Let $G = (V, E)$ be a graph and let $C \subseteq V$ be a set. If C is not a vertex cover of G, then $V - C$ is not an independent set of G.

Proof: Assume C is not a vertex cover of G. We need to show that $V - C$ is not an independent set of G.

Since C is not a vertex cover of G, we know that there exists nodes $x, y \in V$ where $\{x, y\} \in E$, where $x \notin C$, and where $y \notin C$. Because $x \in V$ and $x \notin C$, we know that $x \in V - C$. Similarly, we see that $y \in V - C$.

This means that $\{x, y\} \in E$, that $x \in V - C$, and that $y \in V - C$, and therefore that $V - C$ is not an independent set of G, as required. ■

Recap for Today

- A ***graph*** is a structure for representing items that may be linked together. ***Digraphs*** represent that same idea, but with a directionality on the links.
- Graphs can't have ***self-loops***; digraphs can.
- ***Vertex covers*** and ***independent sets*** are useful tools for modeling problems with graphs.
- The complement of a vertex cover is an independent set, and vice-versa.

Next Time

- ***The Pigeonhole Principle***
  - A simple, powerful, versatile theorem.
- ***Graph Theory Party Tricks***
  - Applying math to graphs of people!