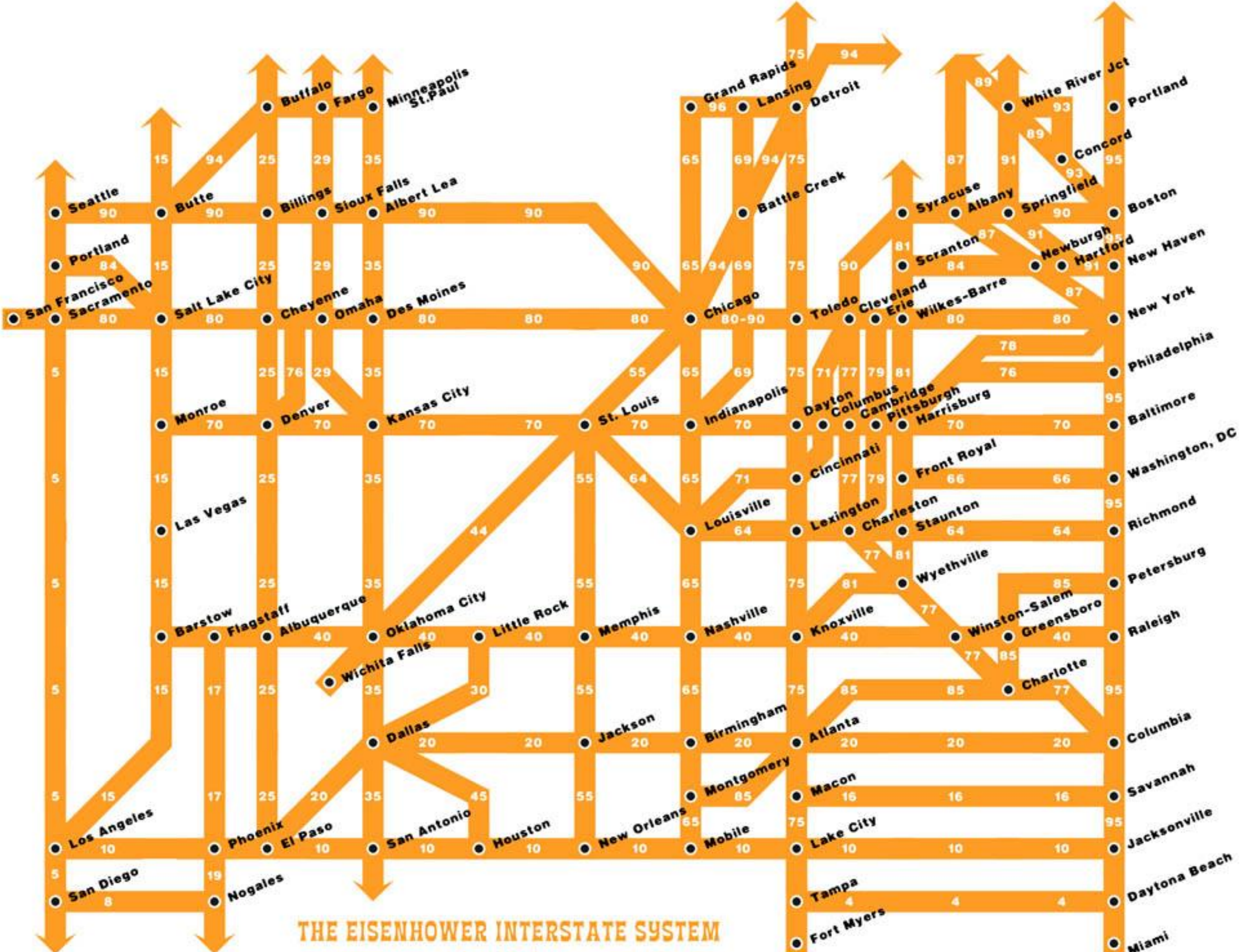
Graph Theory

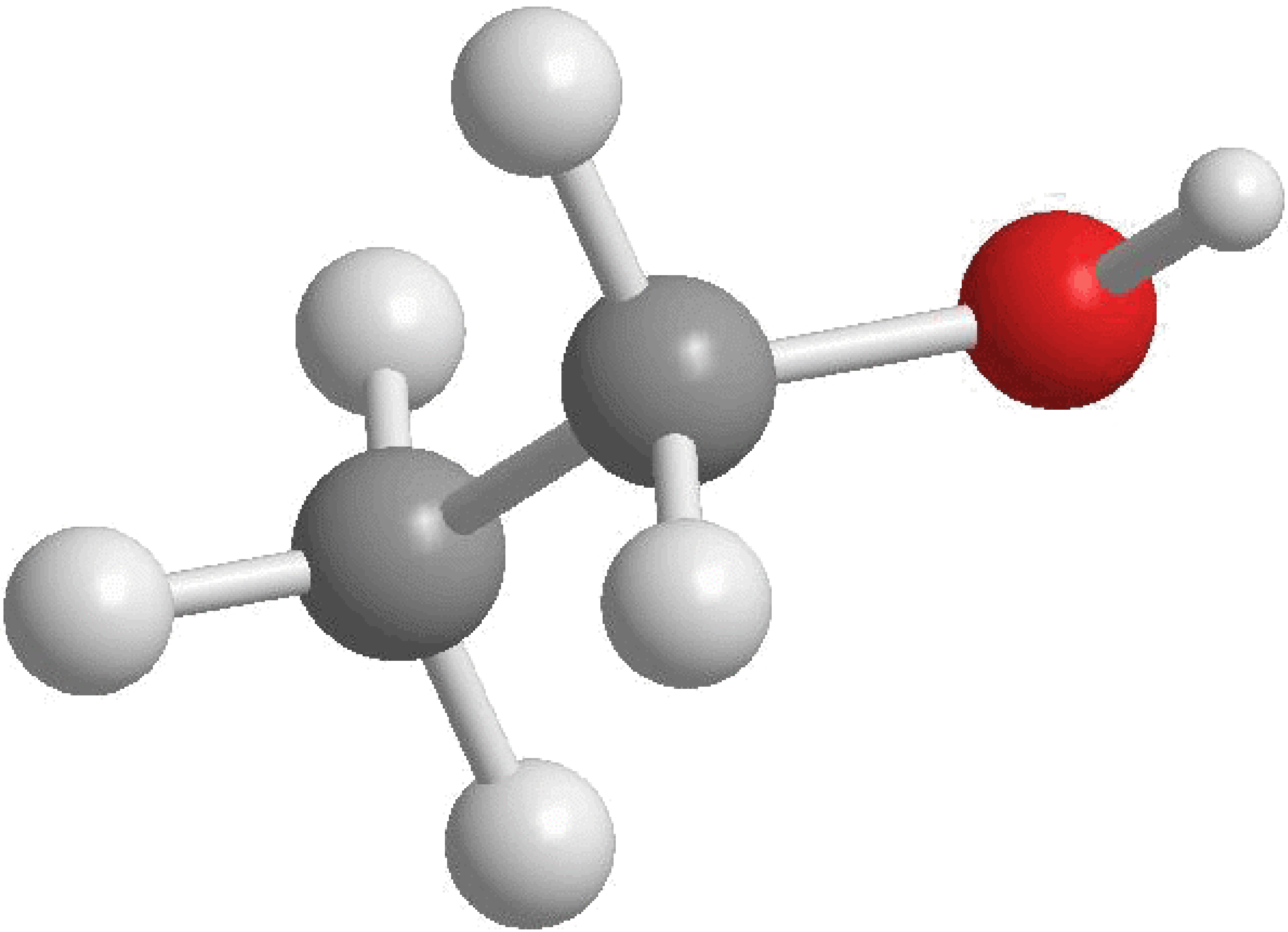


THE EISENHOWER INTERSTATE SYSTEM
(simplified)

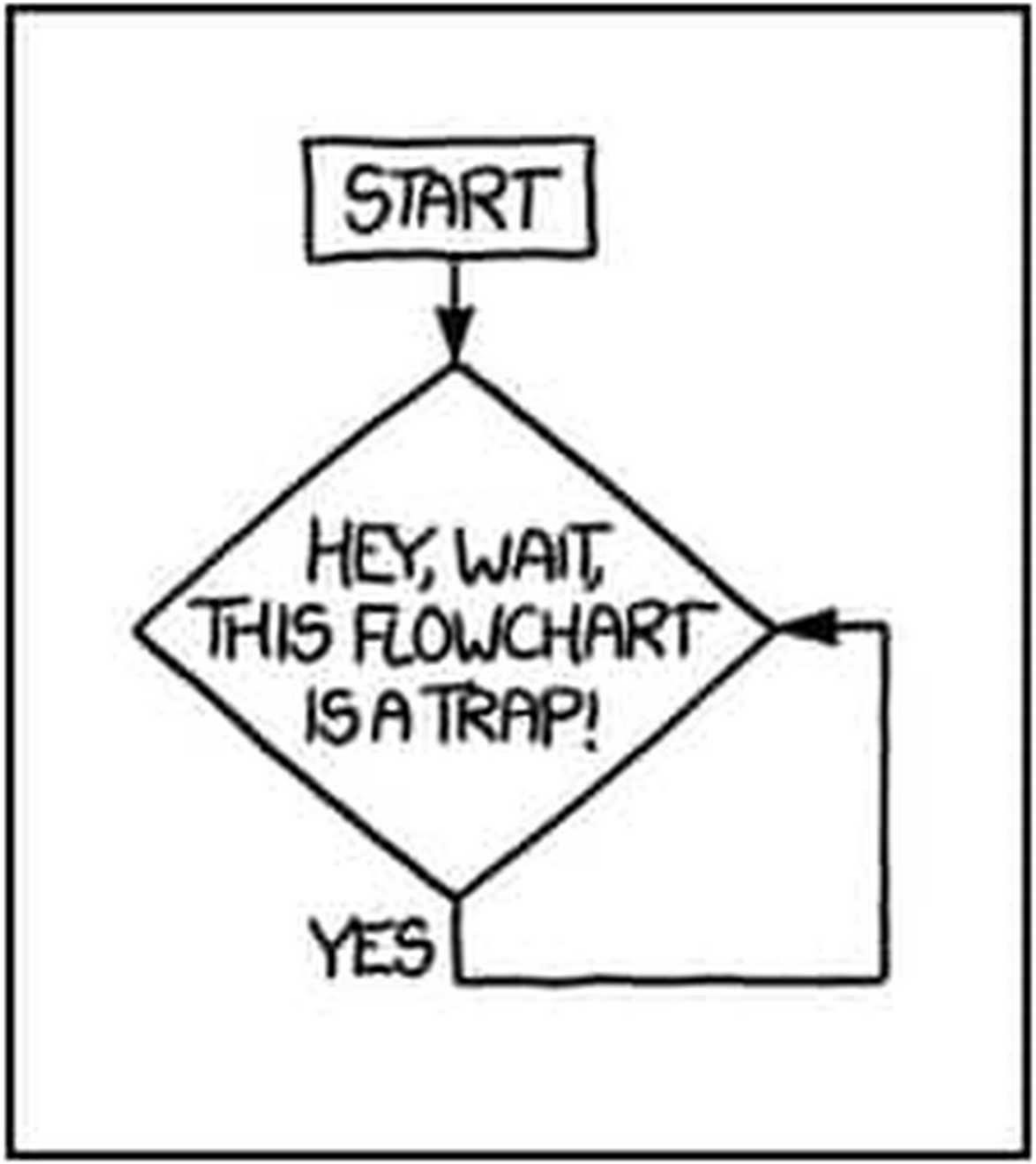
CHRIS YATES 2007

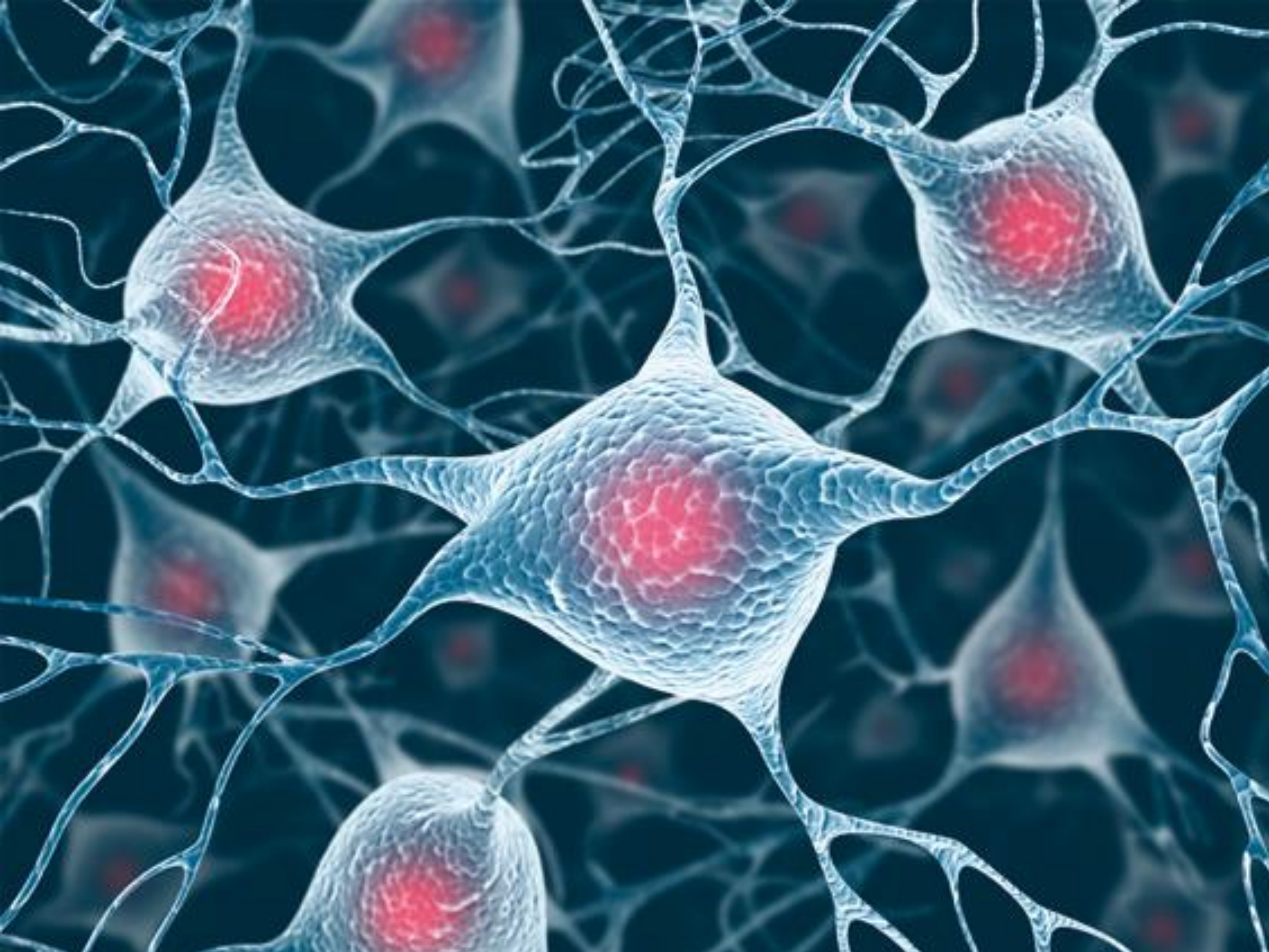
http://strangemaps.files.wordpress.com/2007/02/fullinterstatemap-web.jpg

Chemical Bonds



http://4.bp.blogspot.com/-xCtBJ8lKHqA/Tjm0BONWBRI/AAAAAAAAAK4/-mHrbAUOHHg/s1600/Ethanol2.gif


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



facebook®

What's in Common

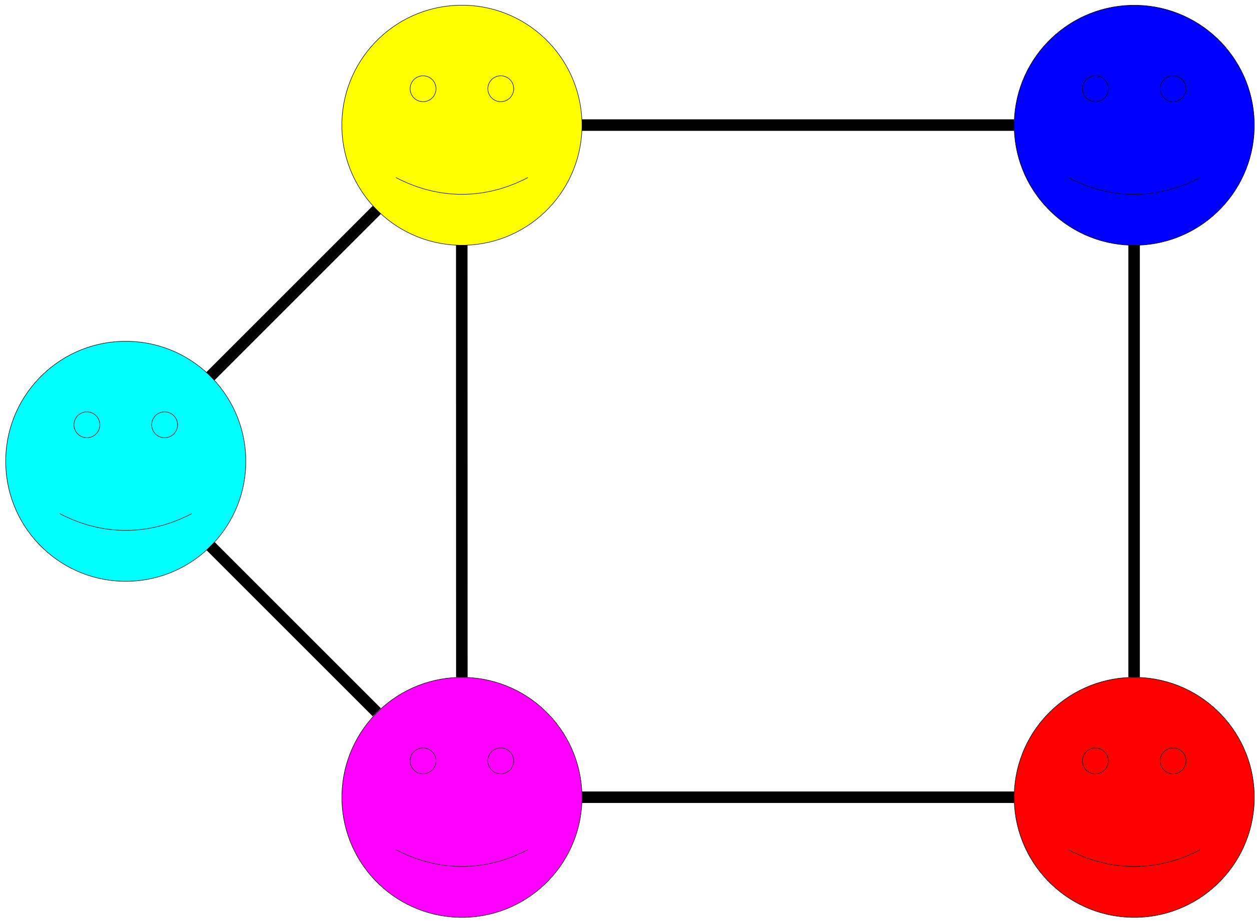
Each of these structures consists of

a collection of objects and

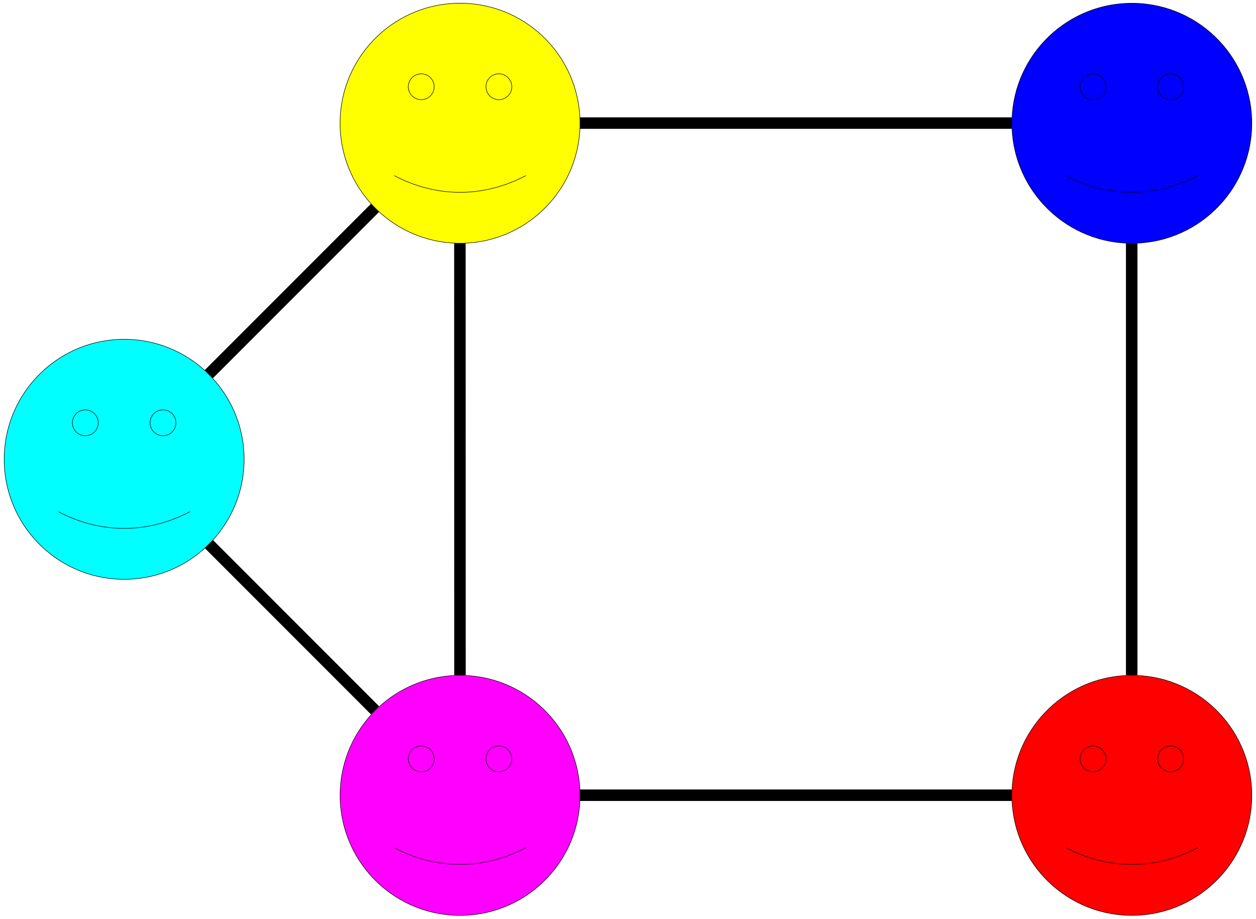
links between those objects.

Goal: find a general framework for describing these objects and their properties.

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

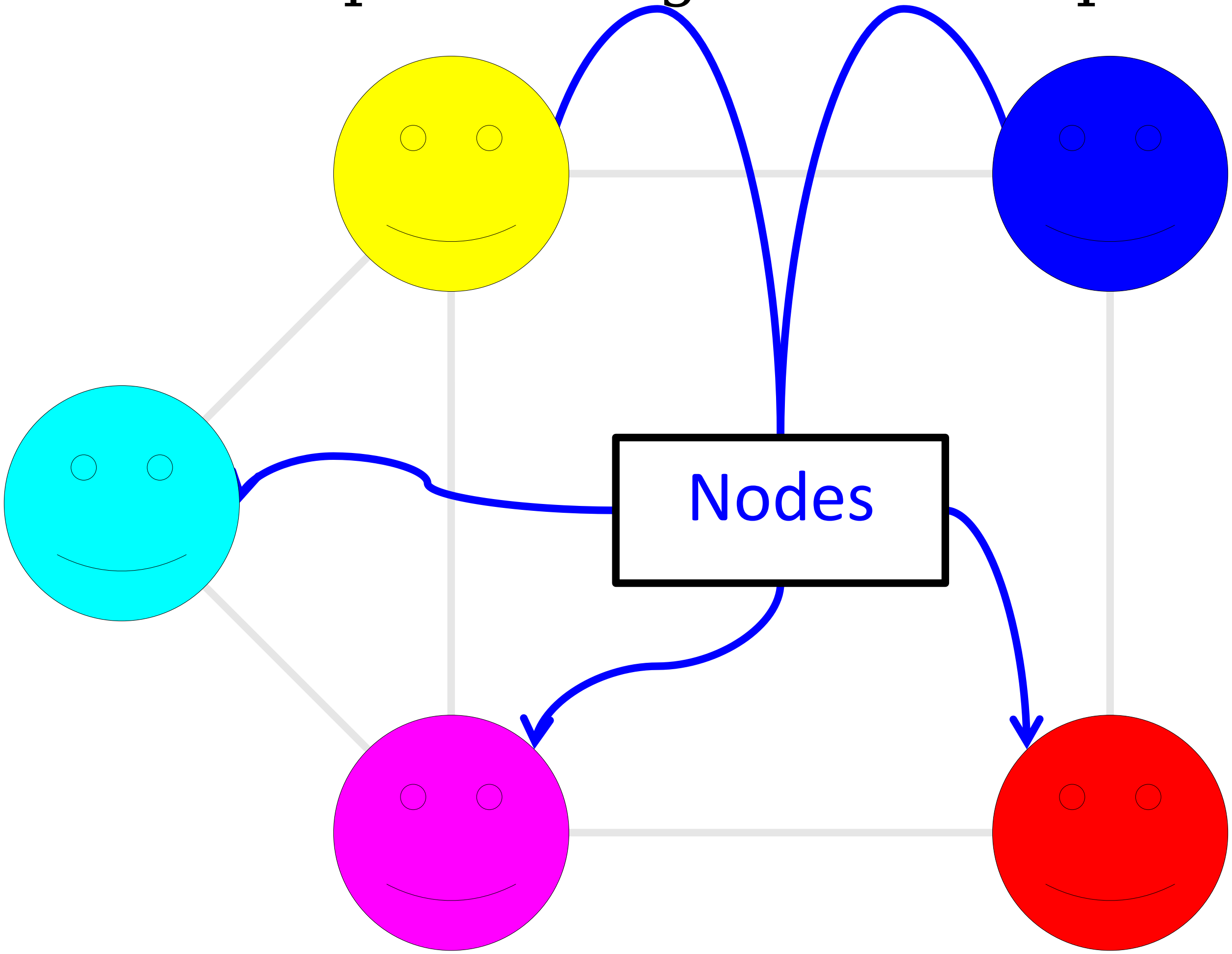


A ***graph*** is a mathematical structure
for representing relationships.



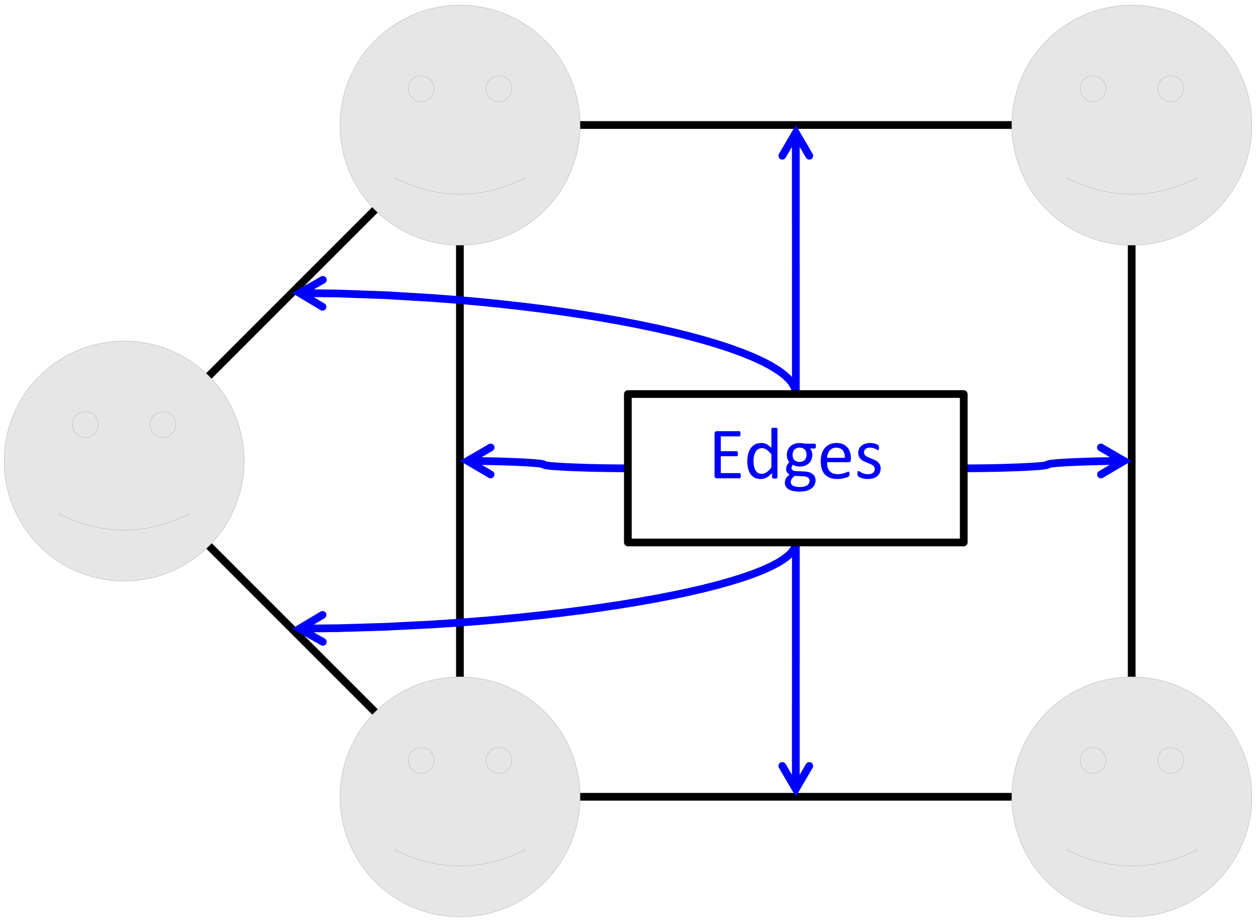
A graph consists of a set of ***nodes*** (or ***vertices***)
connected by ***edges*** (or ***arcs***)

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



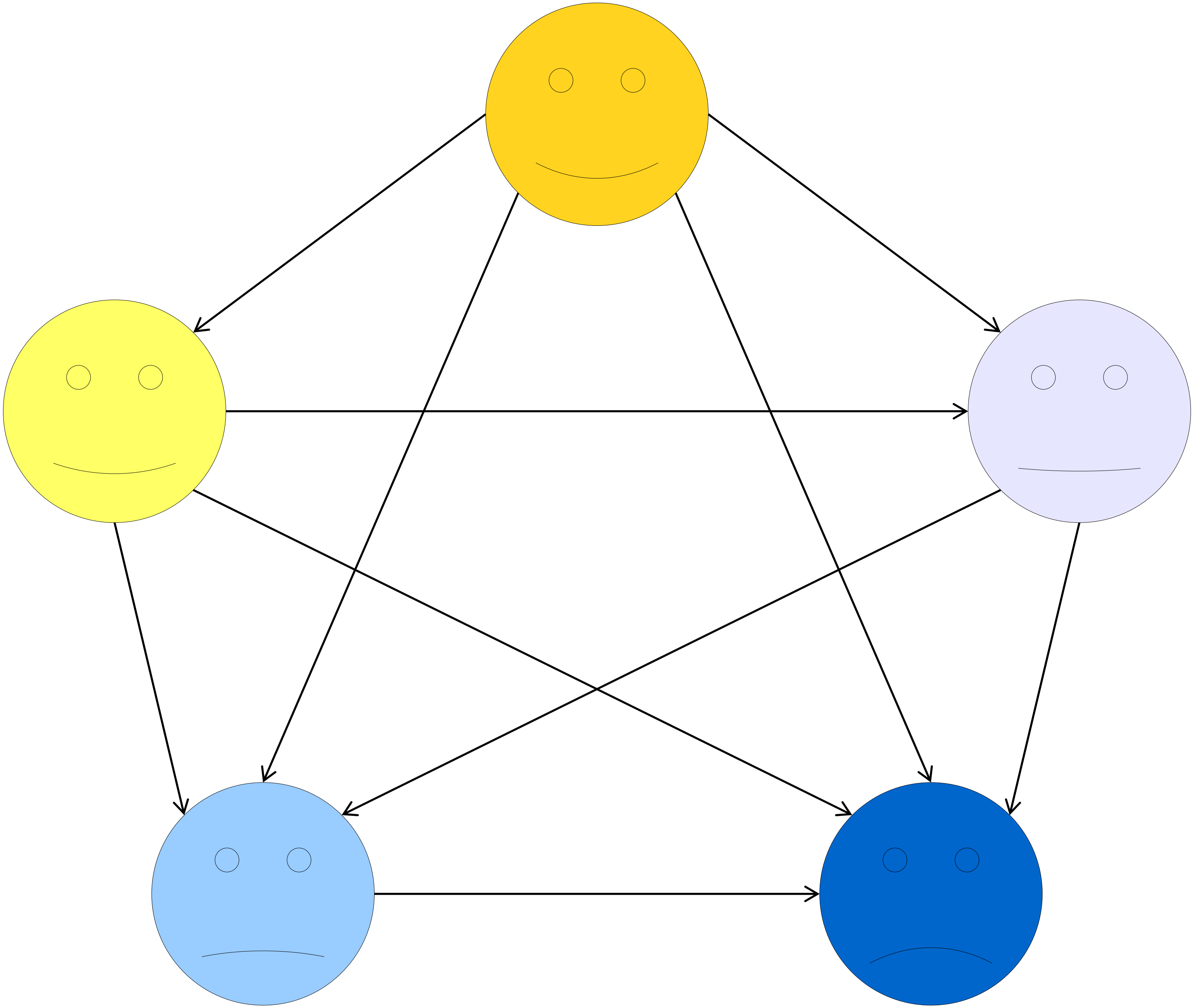
A graph consists of a set of ***nodes*** (or ***vertices***) connected by ***edges*** (or ***arcs***)

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

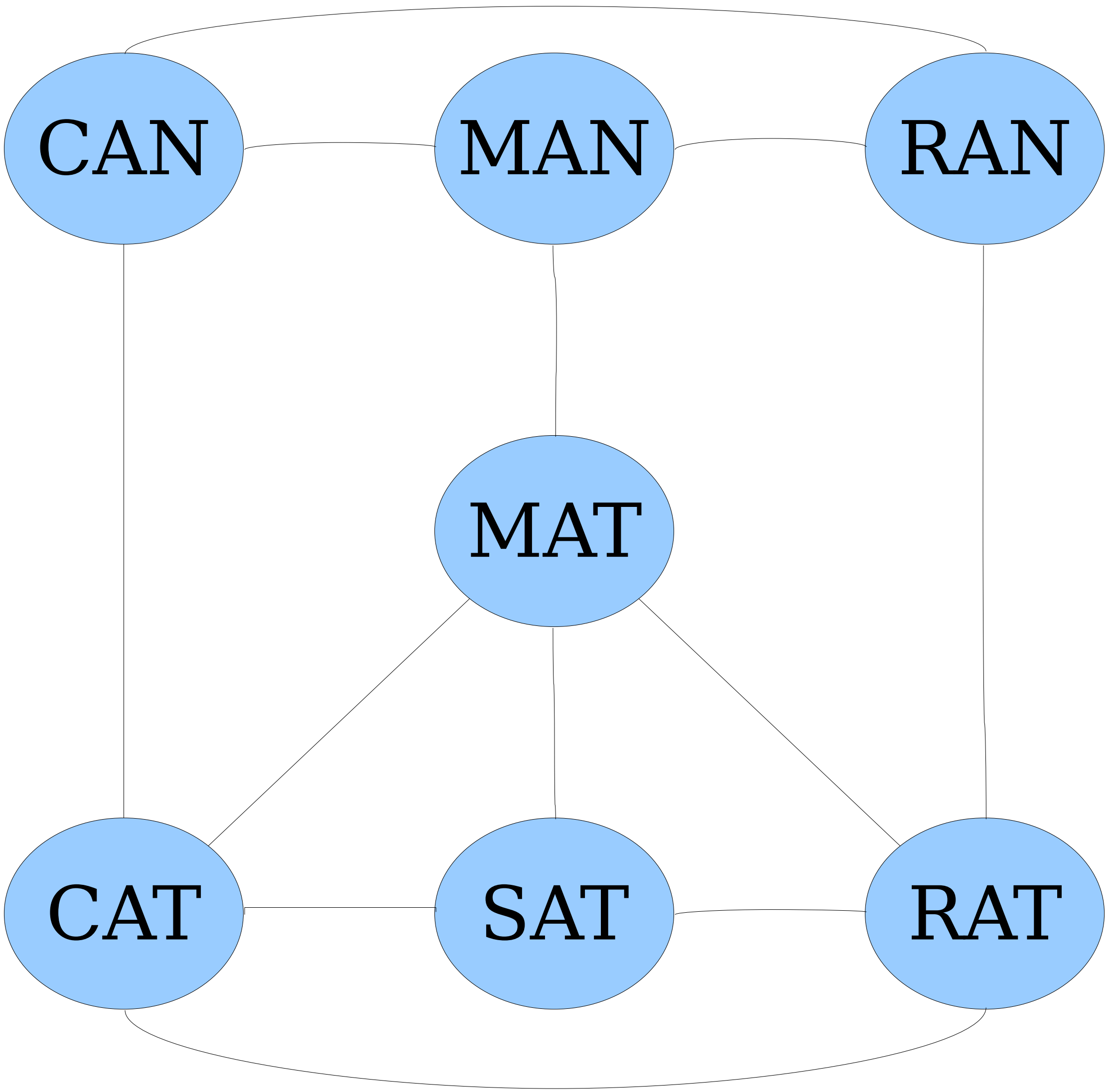


A graph consists of a set of ***nodes*** (or ***vertices***) connected by ***edges*** (or ***arcs***)

Some graphs are ***directed***.



Some graphs are ***undirected***.



Going forward, we're primarily going to focus on undirected graphs.

The term “graph” generally refers to undirected graphs with a finite number of nodes, unless specified otherwise.

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).

$\{0, 1\} = \{1, 0\}$

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*** 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.

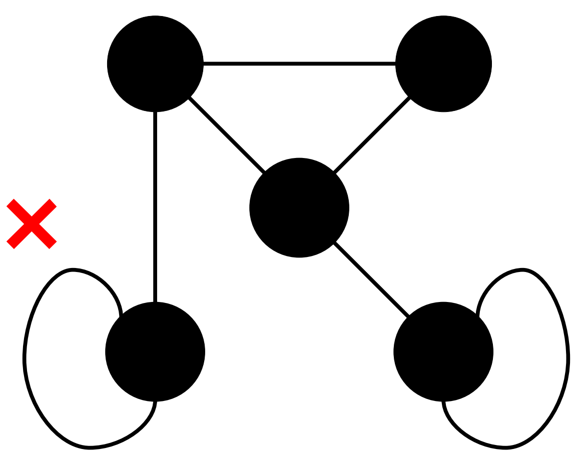
Self-Loops

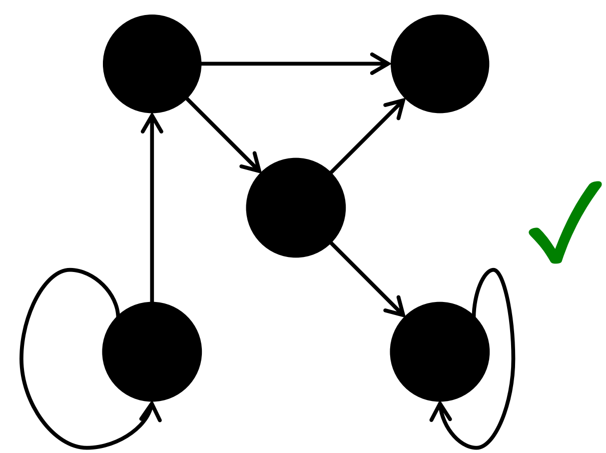
An edge from a node to itself is called a ***self-loop***.

In undirected graphs, self-loops are generally not allowed.

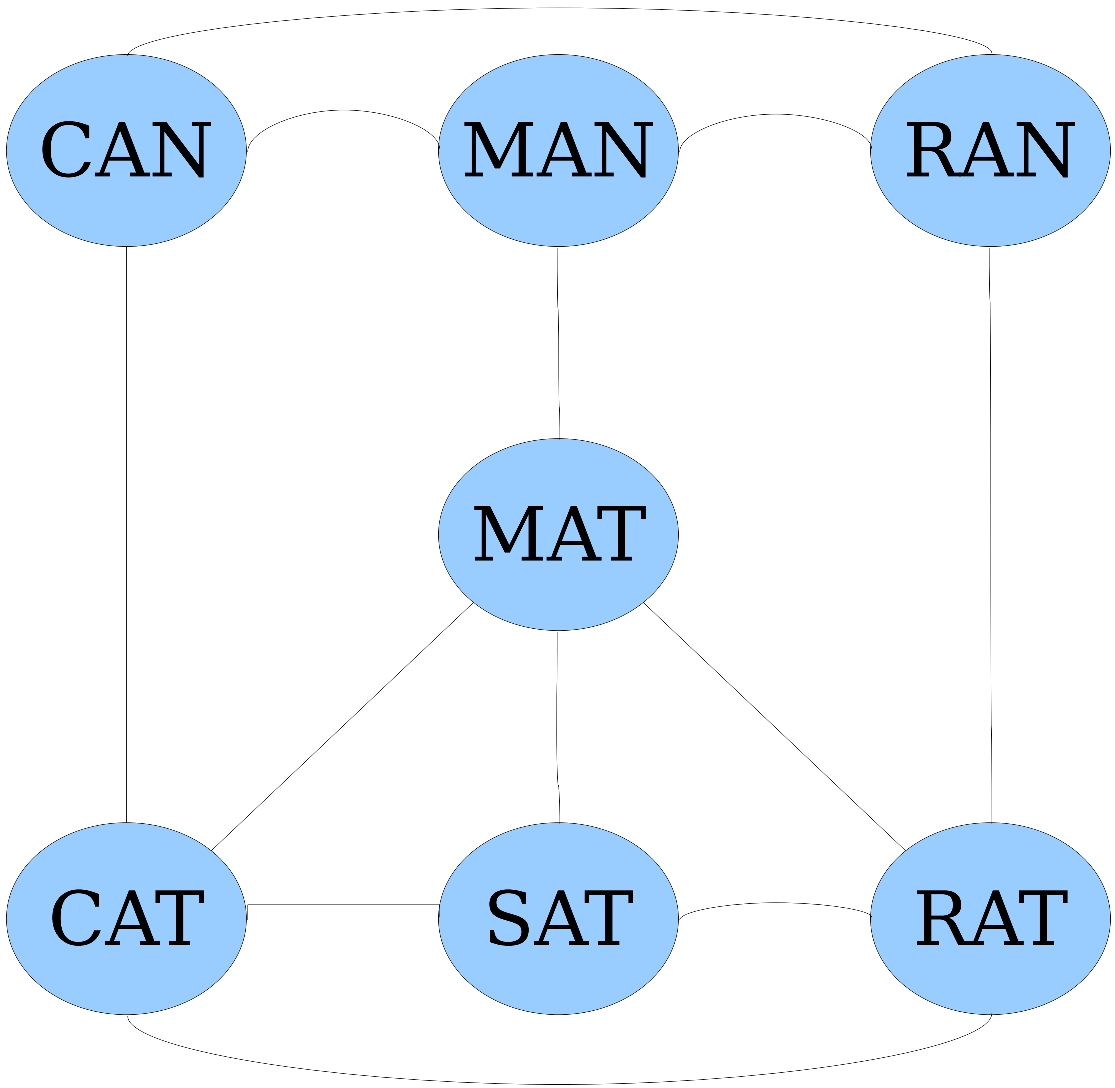
Can you see how this follows from the definition?

In directed graphs, self-loops are generally allowed unless specified otherwise.

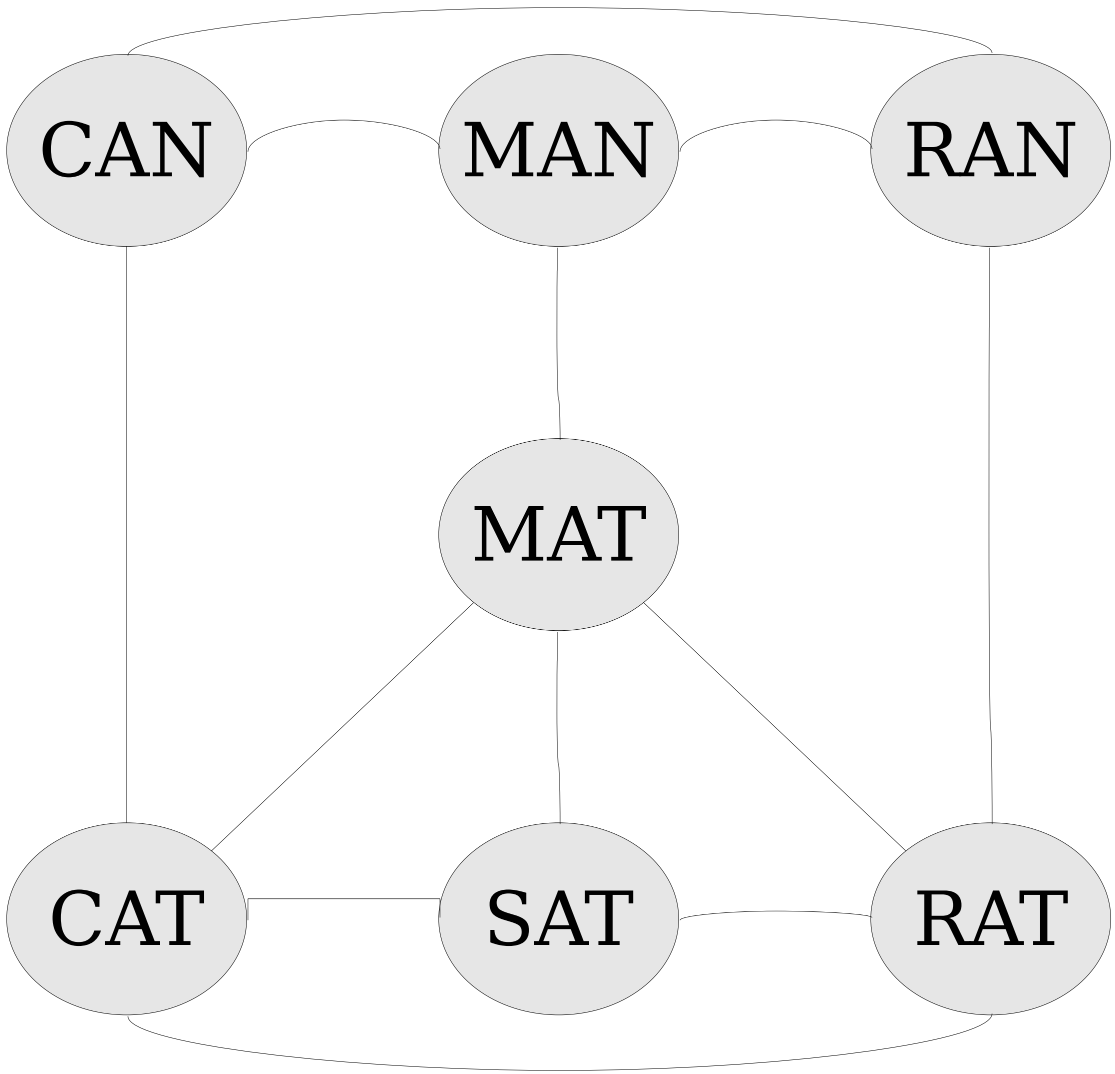




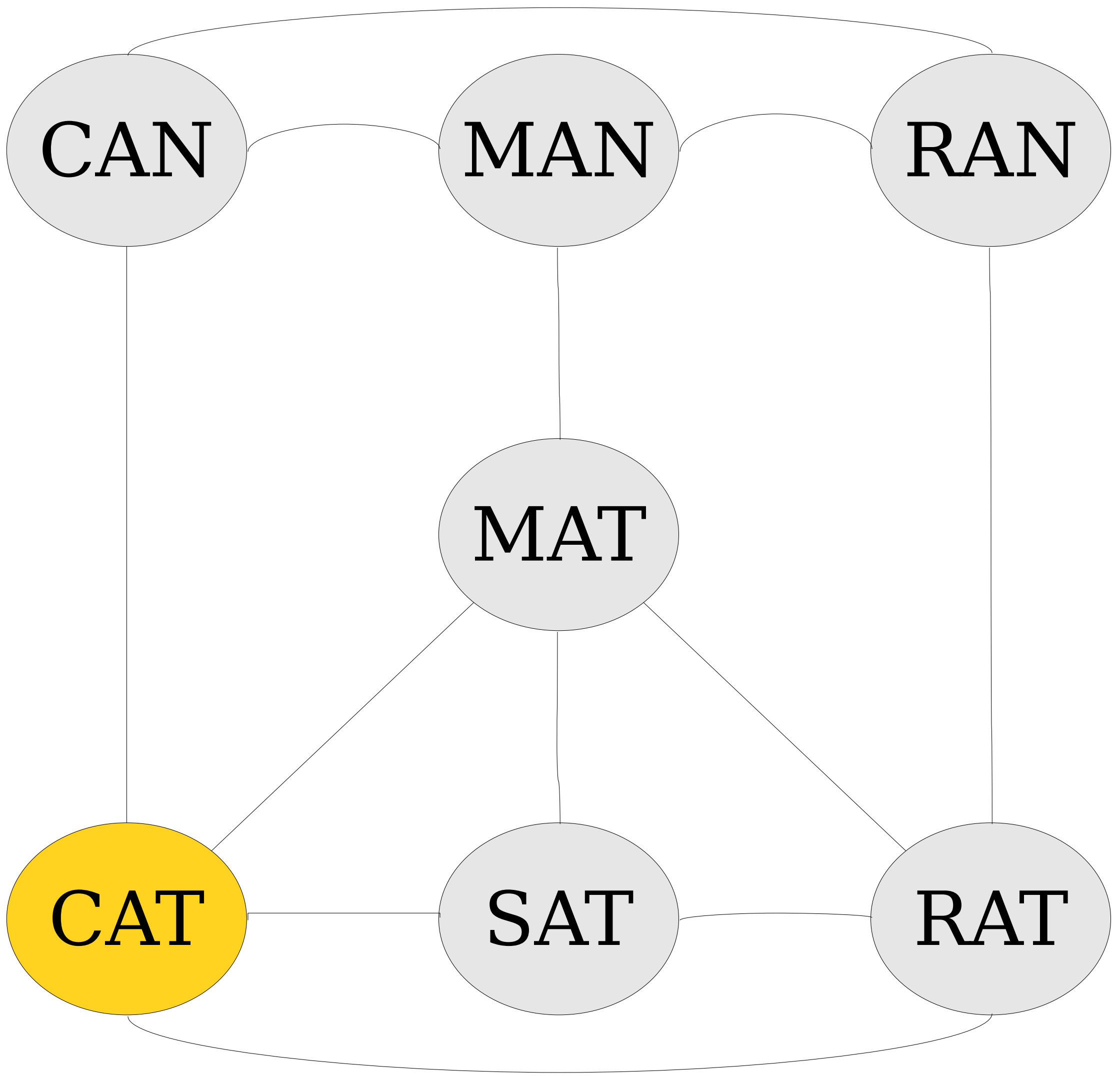
Standard Graph Terminology



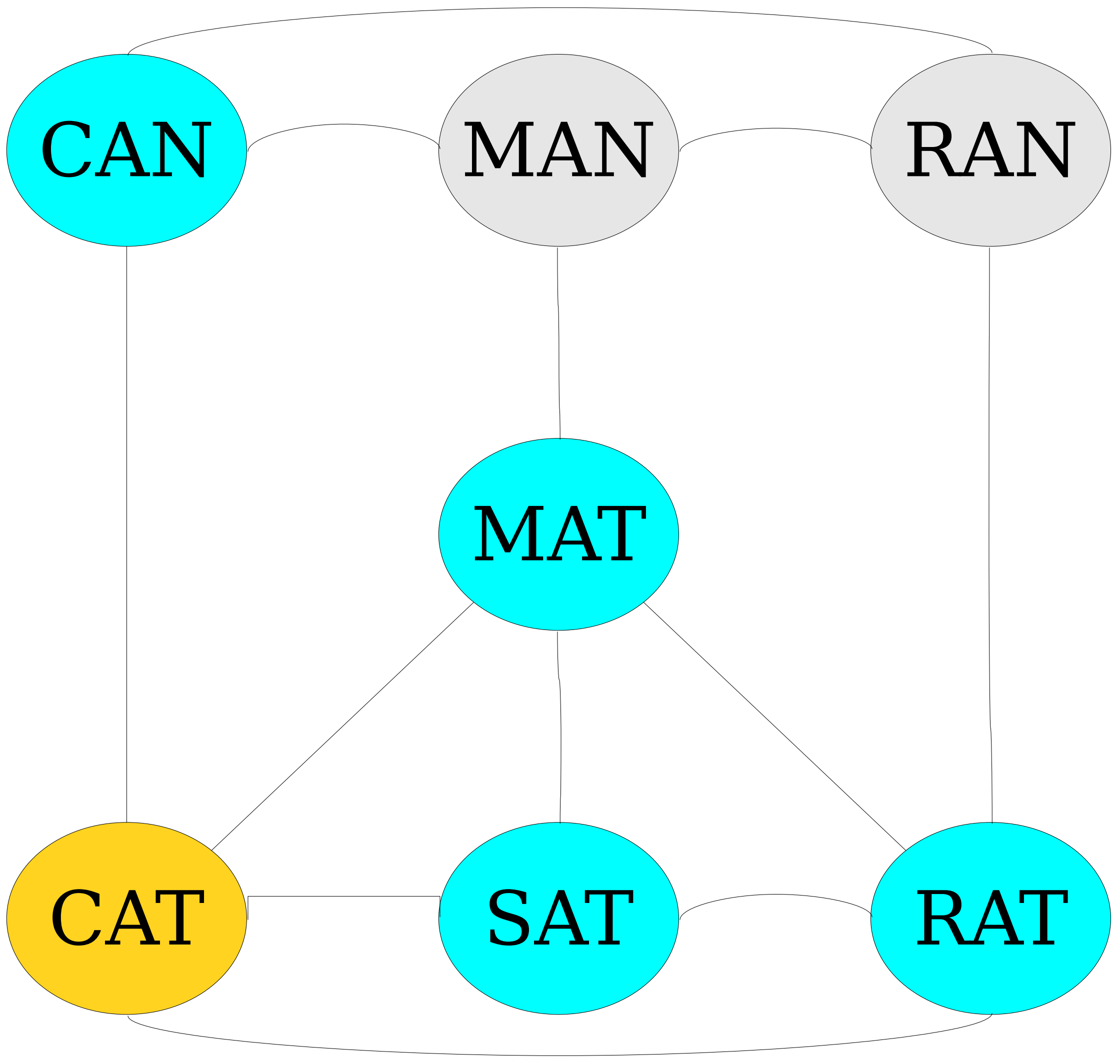
Two nodes are called ***adjacent*** if there is an edge between them.



nodes are called ***adjacent*** if there is an edge between t



nodes are called ***adjacent*** if there is an edge between th



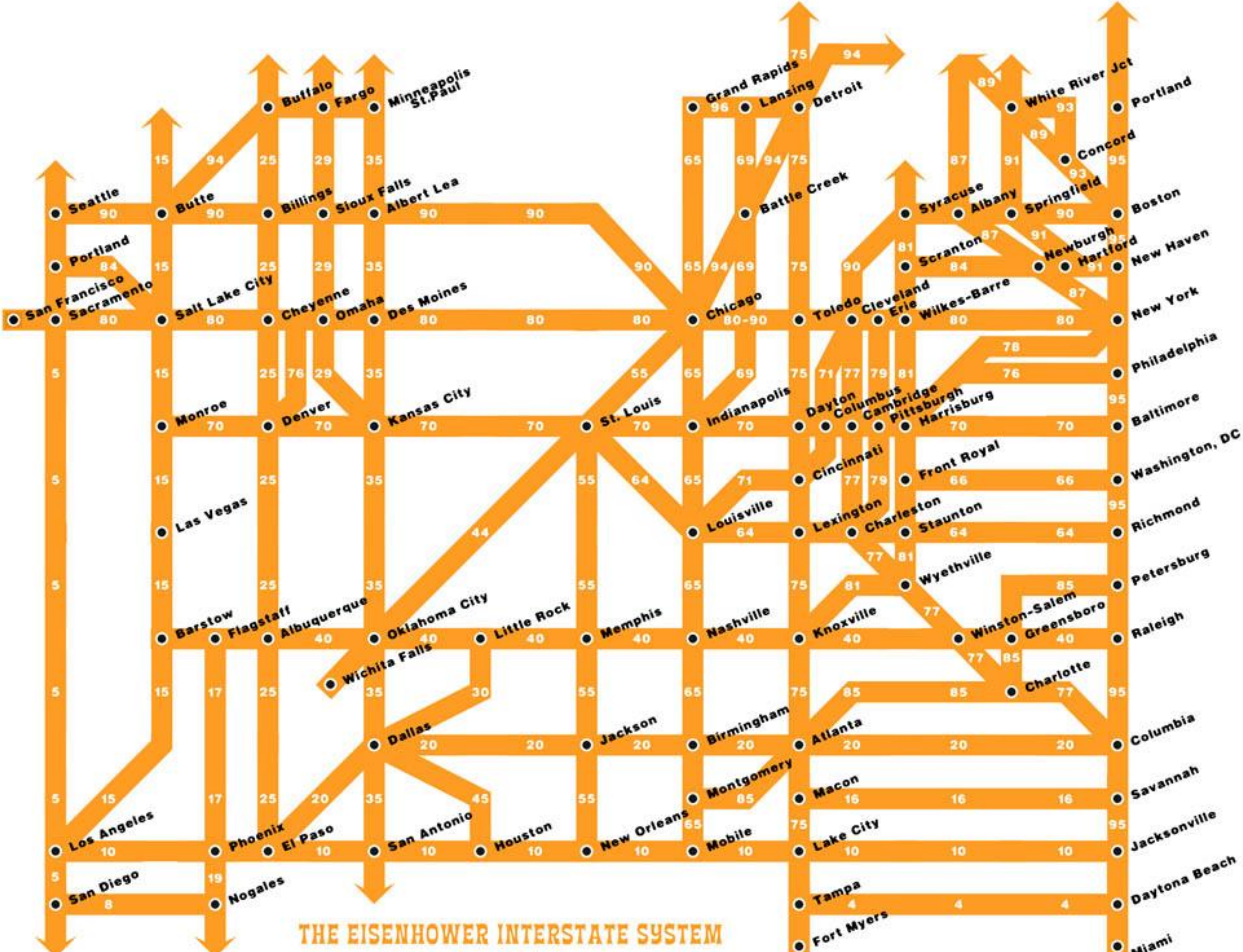
nodes are called ***adjacent*** if there is an edge between t

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 $\{u, v\} \in E$.

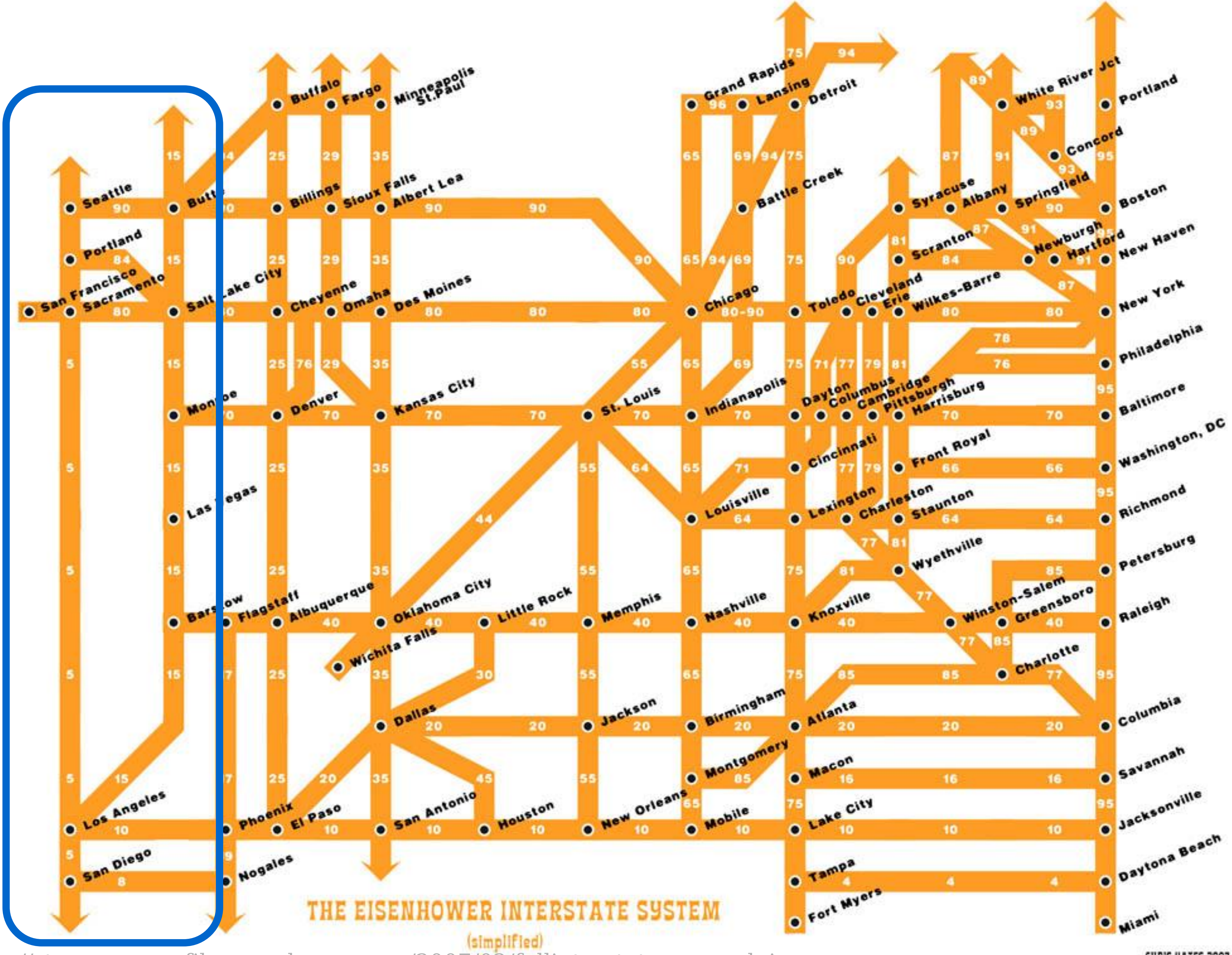


THE EISENHOWER INTERSTATE SYSTEM

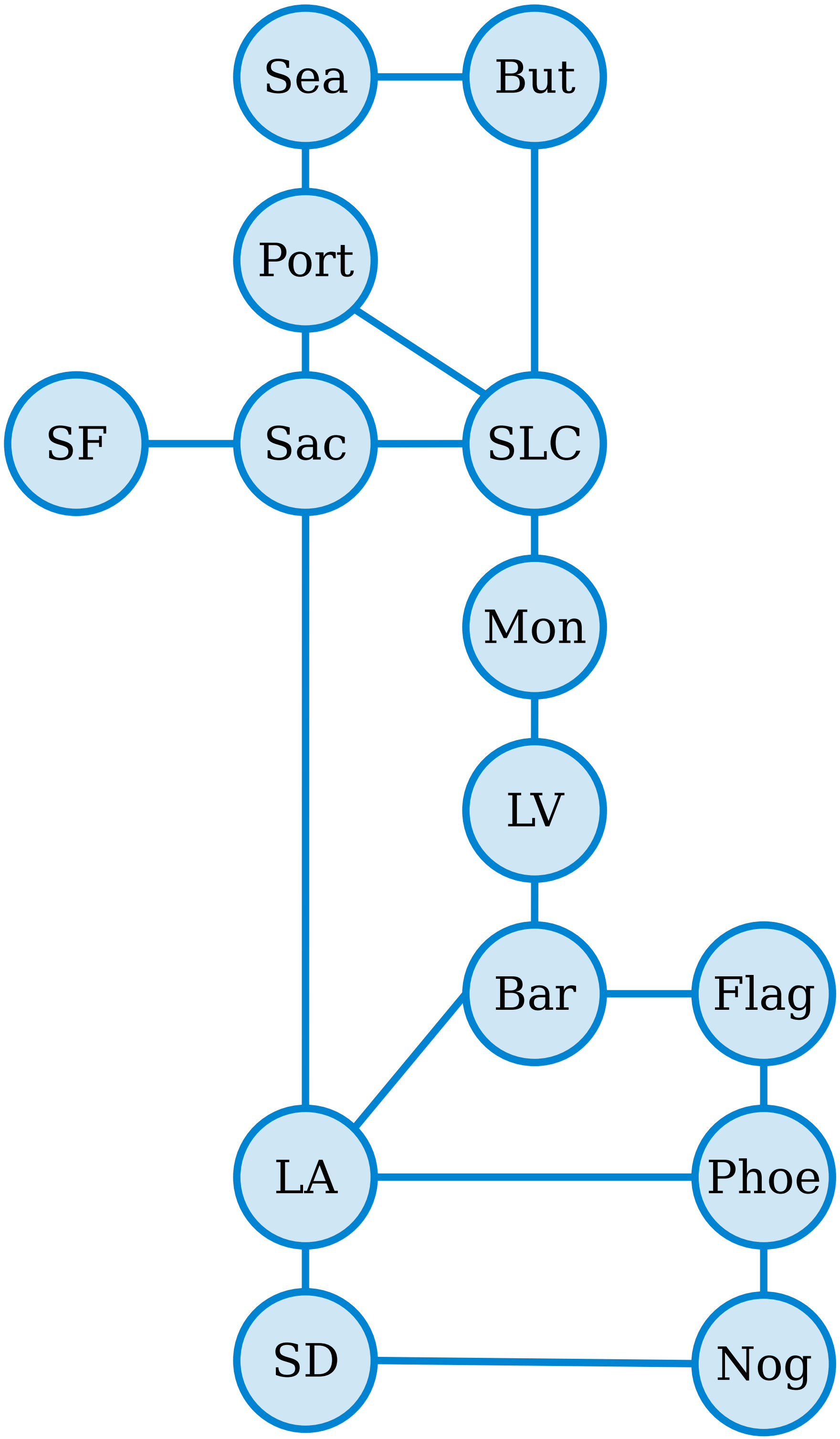
(simplified)

CHRIS YATES 2007

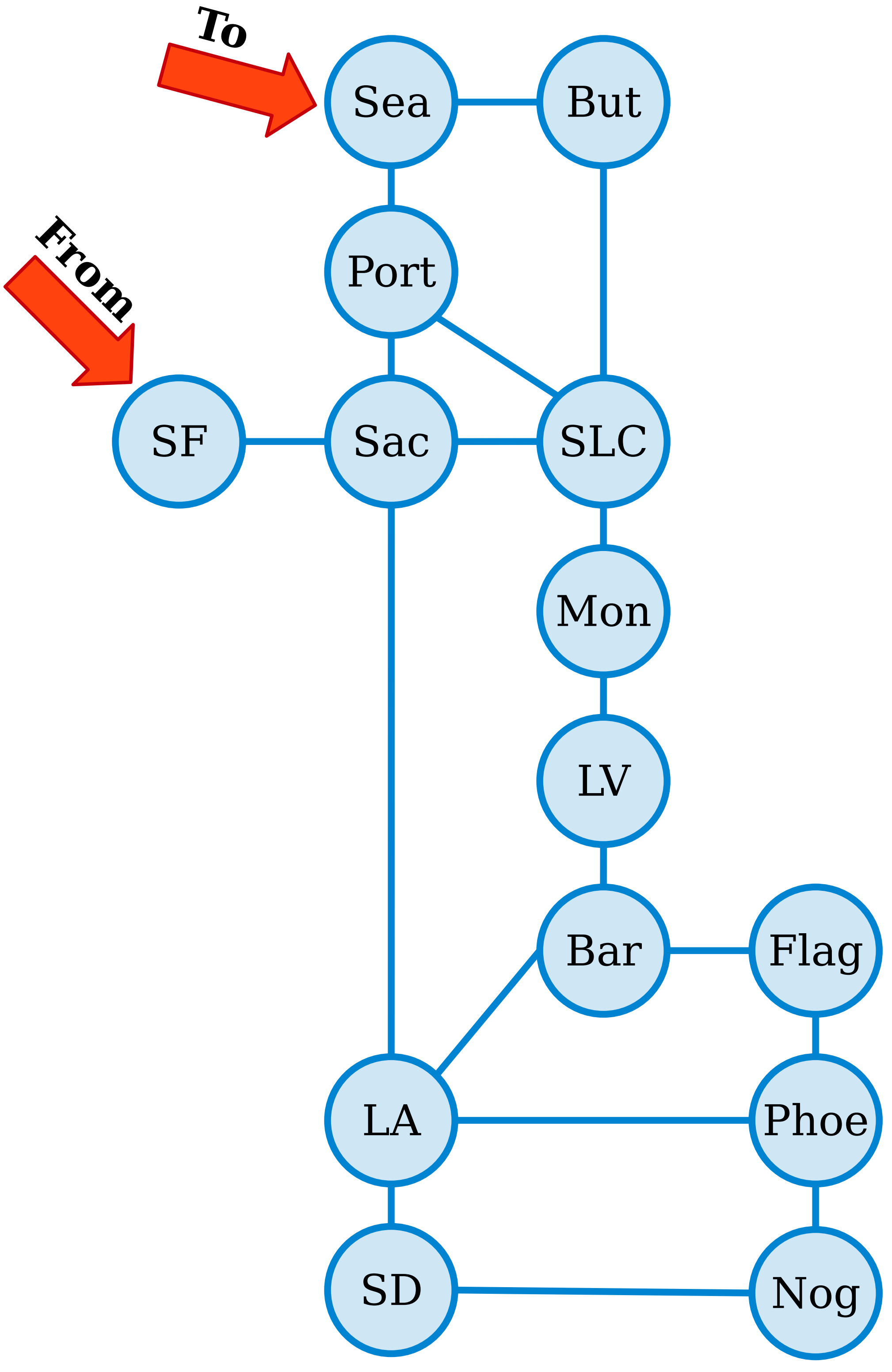
http://strangemaps.files.wordpress.com/2007/02/fullinterstatemap-web.jpg

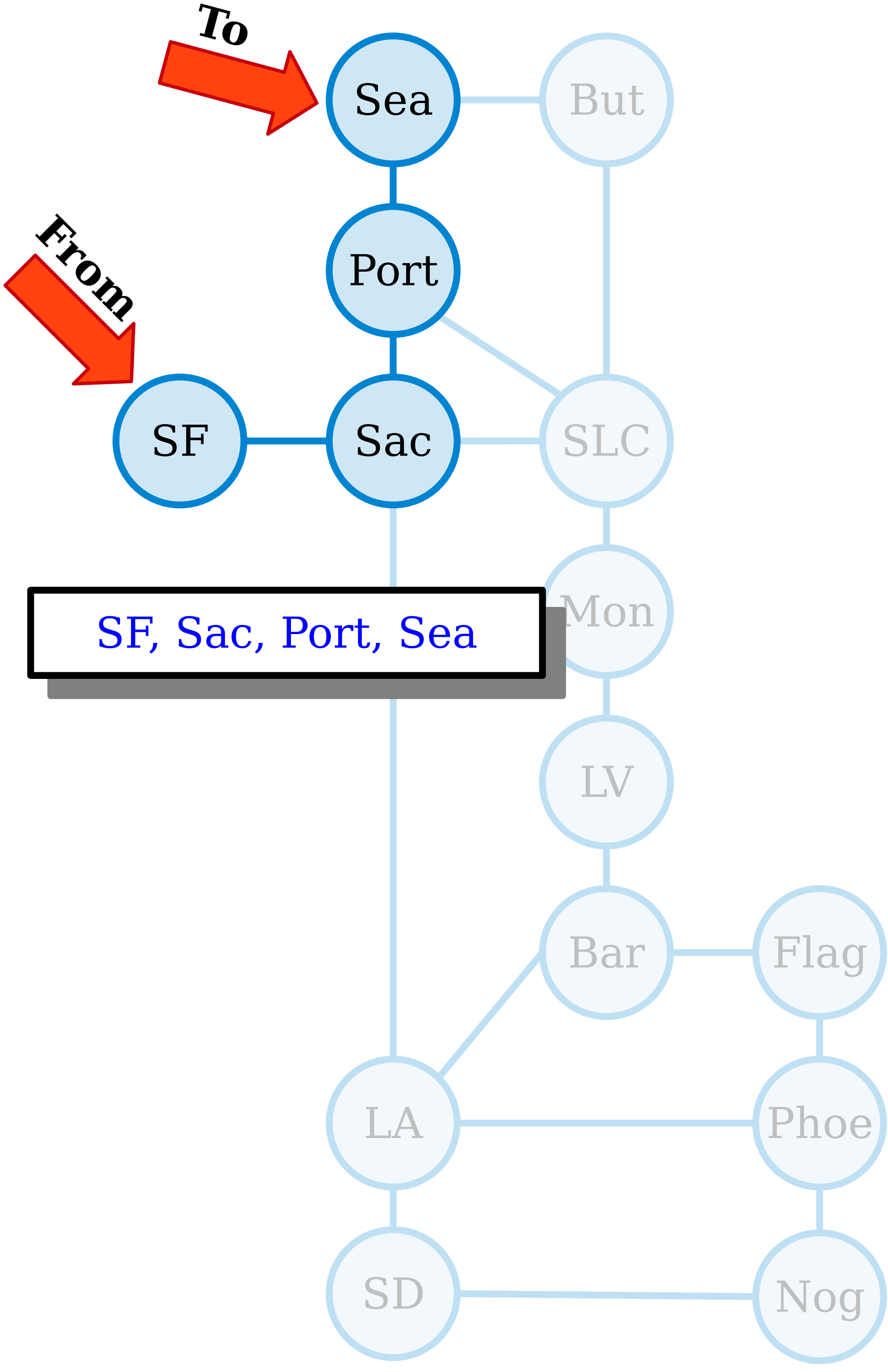


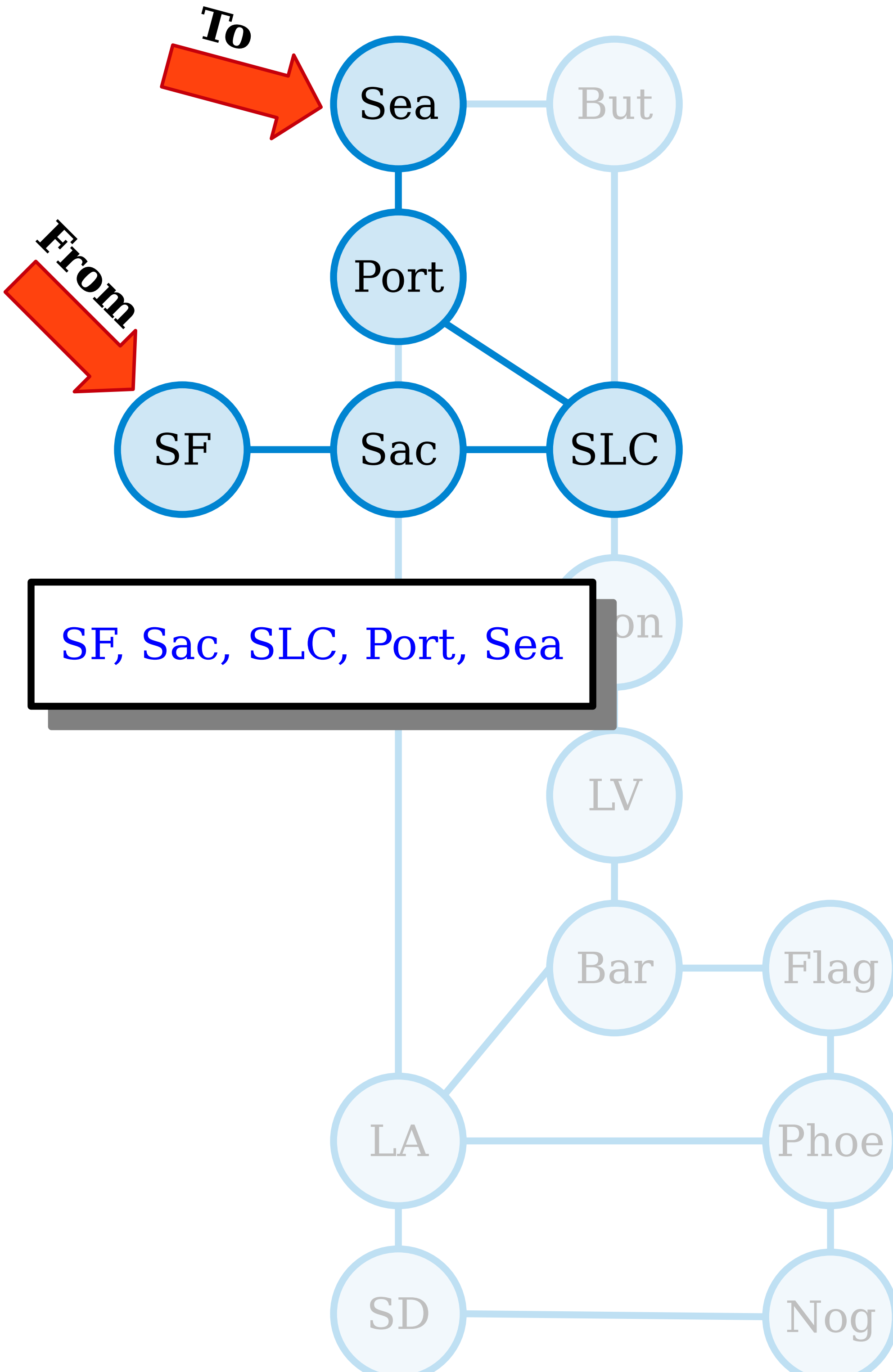
http://strangemaps.files.wordpress.com/2007/02/fullinterstatemap-web.jpg

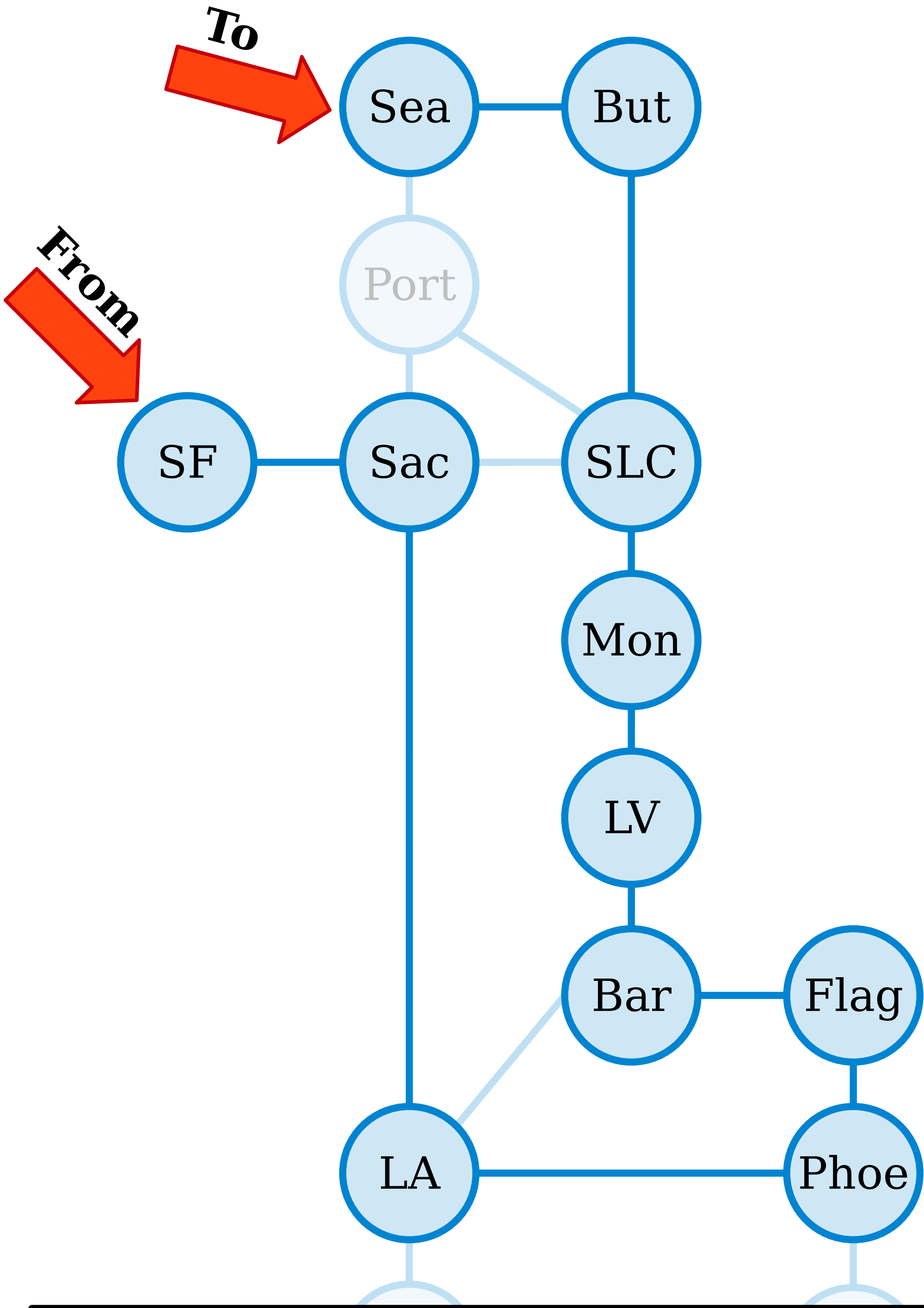


Sea
But
Port
SF
Sac
SLC
Mon
LV
Bar
Flag
LA
Phoe
SD
Nog

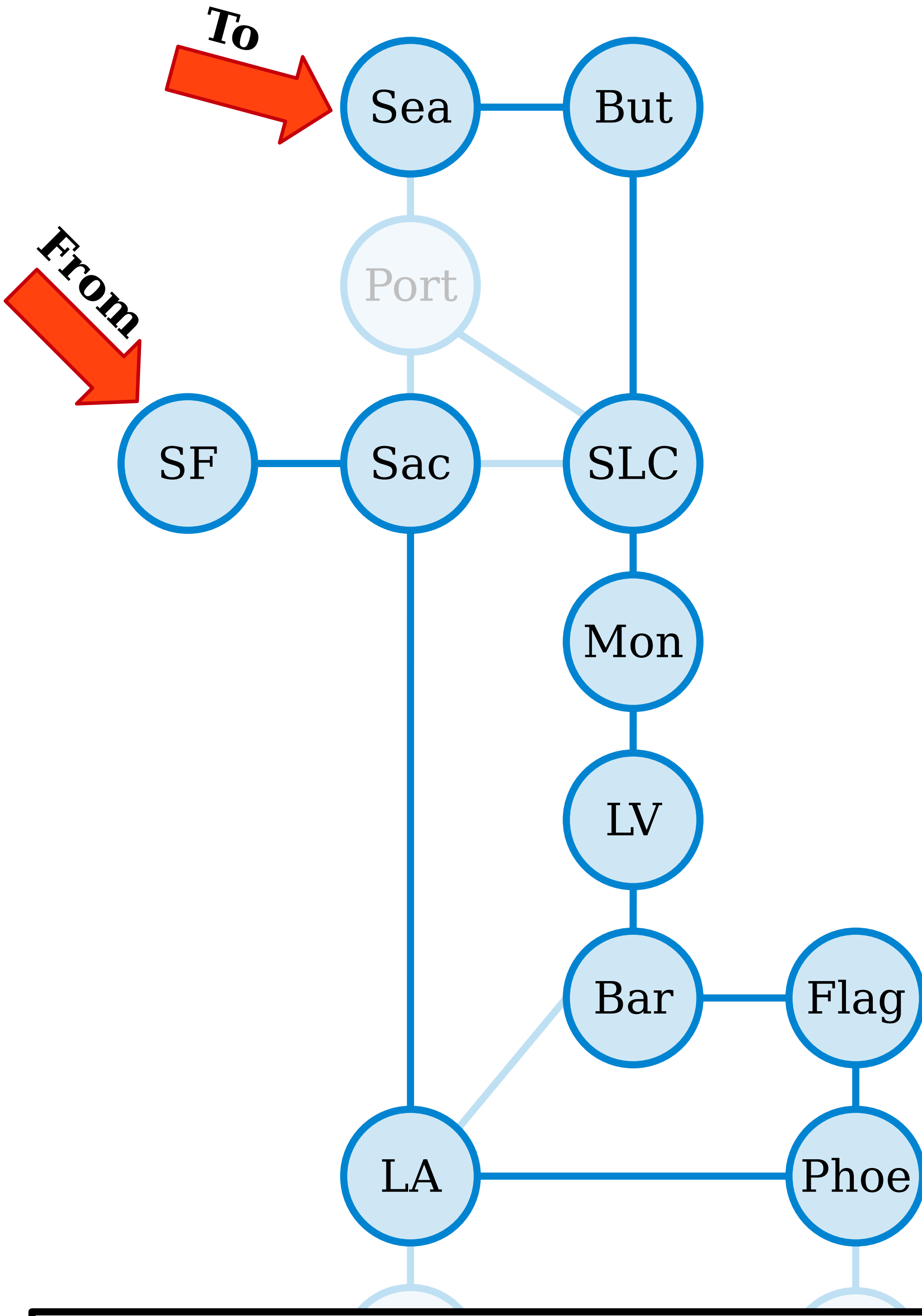

To
From
Sea
But
Port
SF
Sac
SLC
Mon
LV
Bar
Flag
LA
Phoe
SD
Nog


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

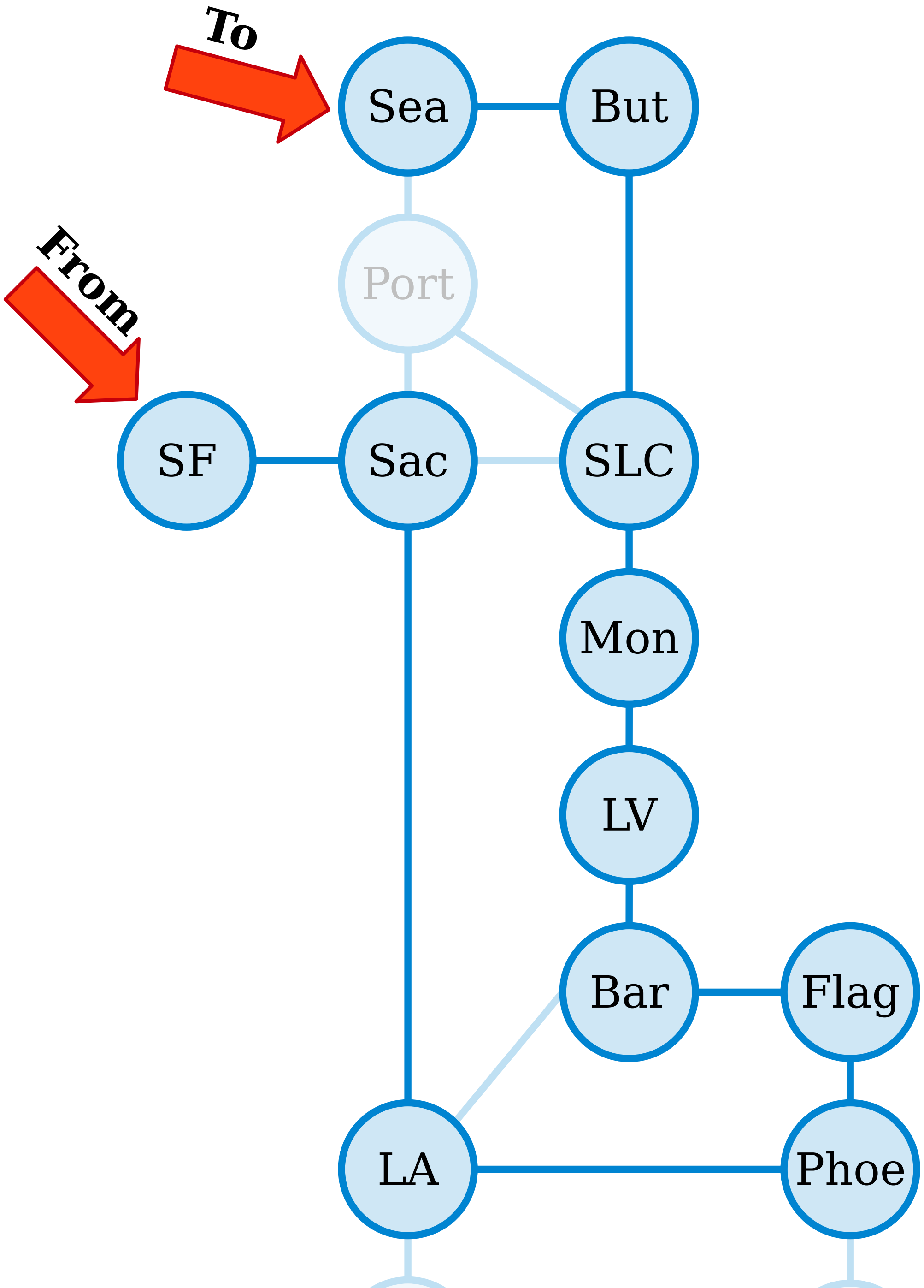


SF, Sac, LA, Phoe, Flag, Bar, LV, Mon, SLC, But, Sea



A ***path*** 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.

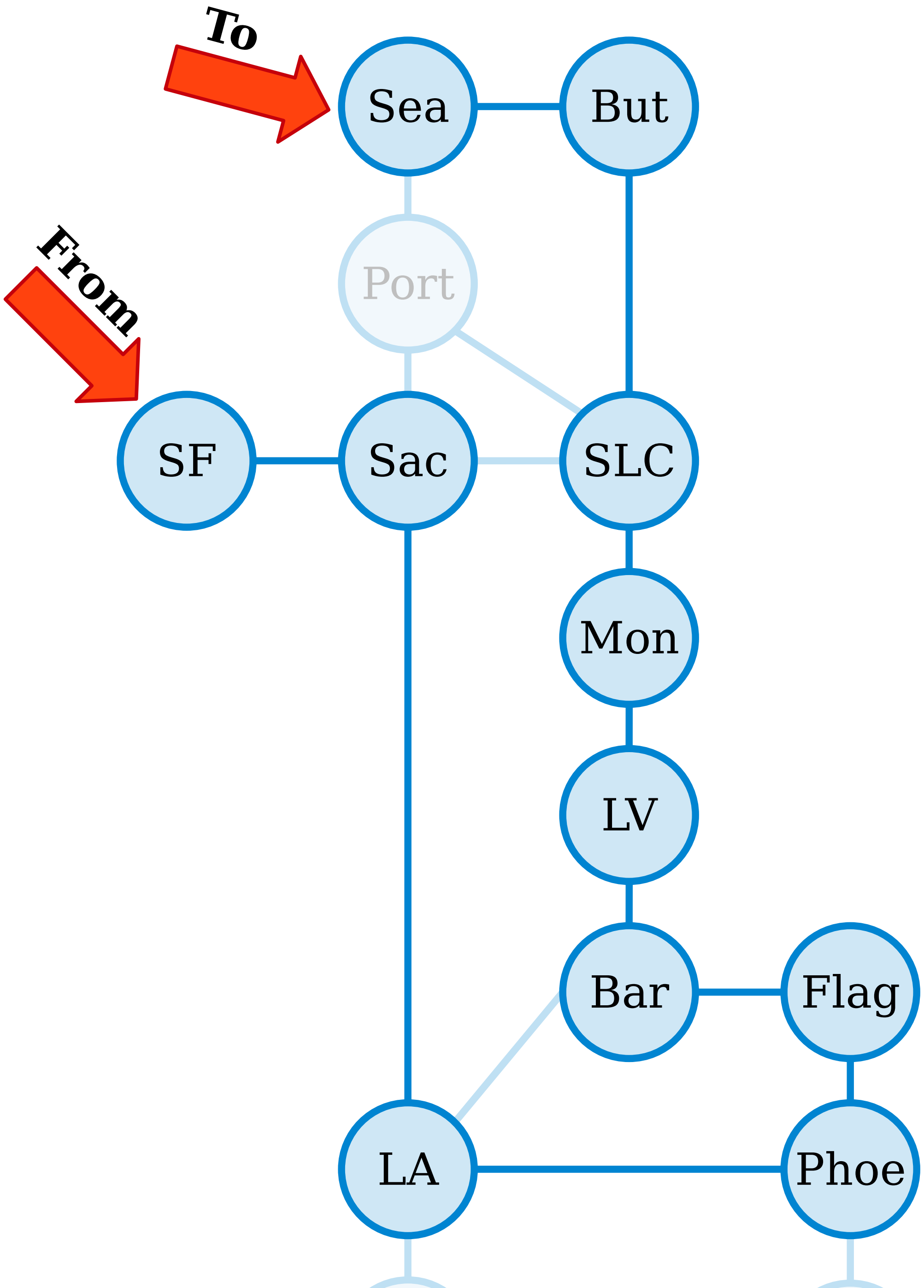
SF, Sac, LA, Phoe, Flag, Bar, LV, Mon, SLC, But, Sea



A ***path*** 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 path $v_1, \ldots, v_n$ is $n - 1$.

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

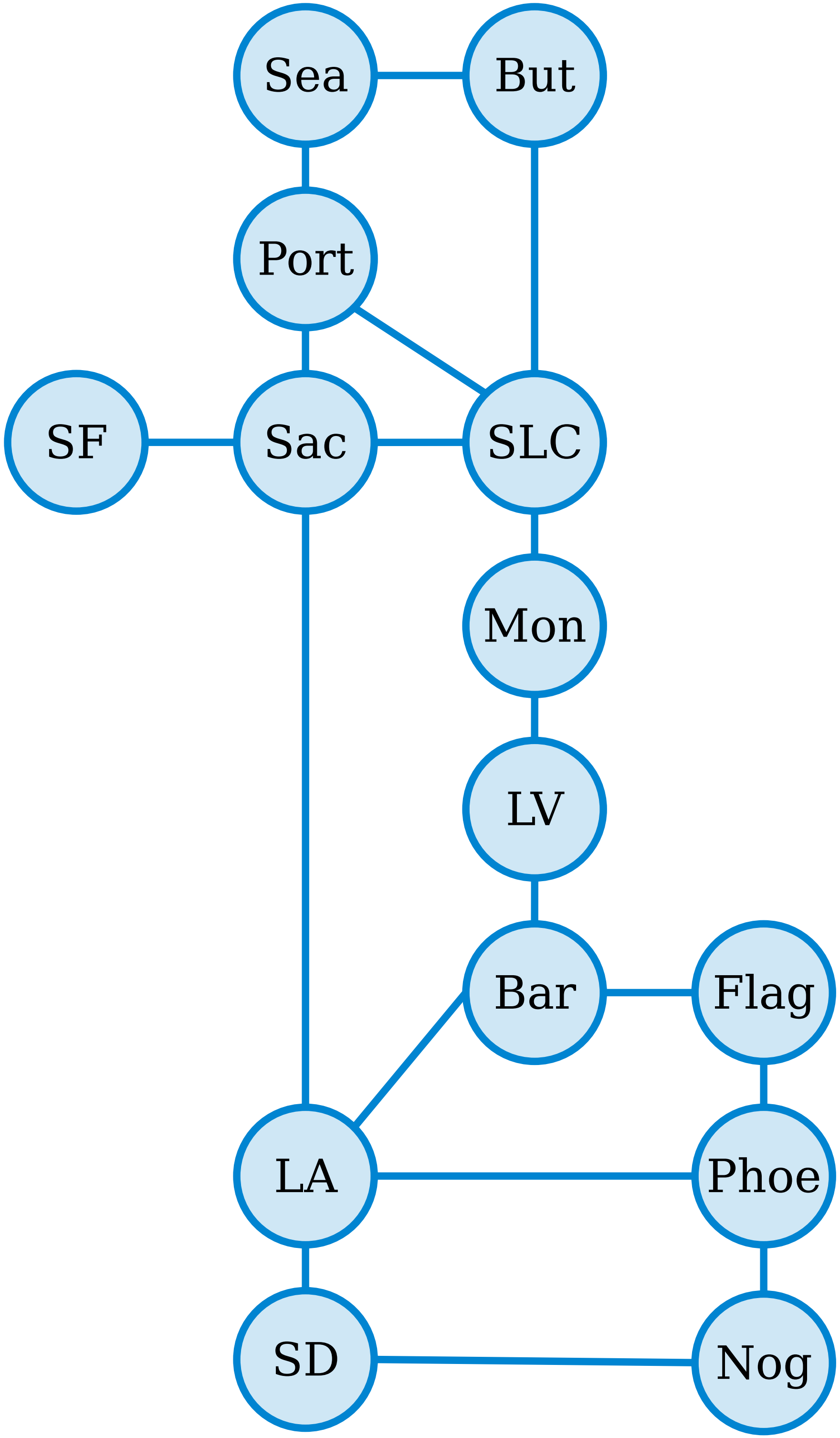


A ***path*** in a graph $G = (V, E)$ is a sequence of one or more nodes $v_1, v_2, v_3, \ldots, v_n$ such that any two consecutive nodes in the sequence are adjacent.

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

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

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

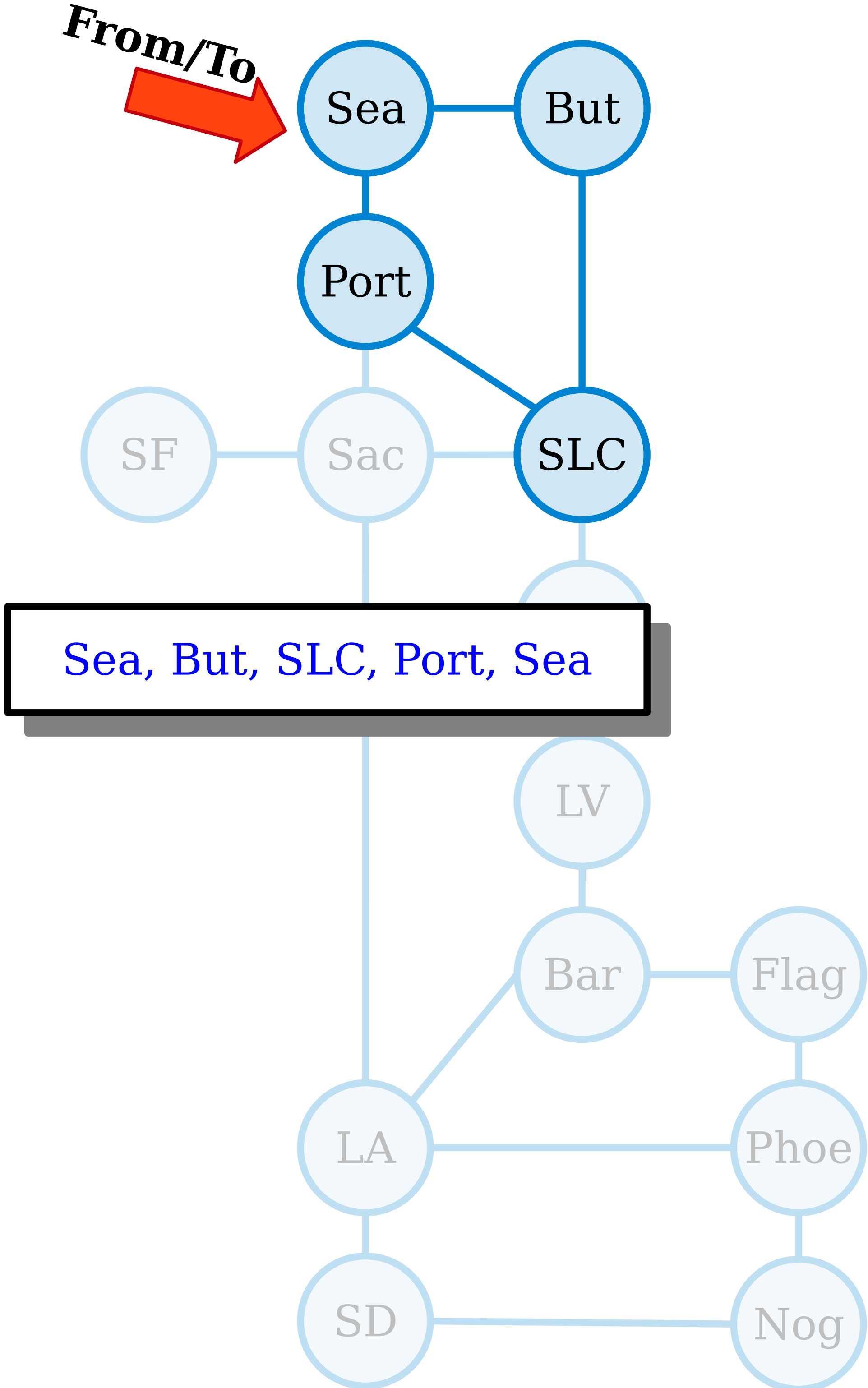


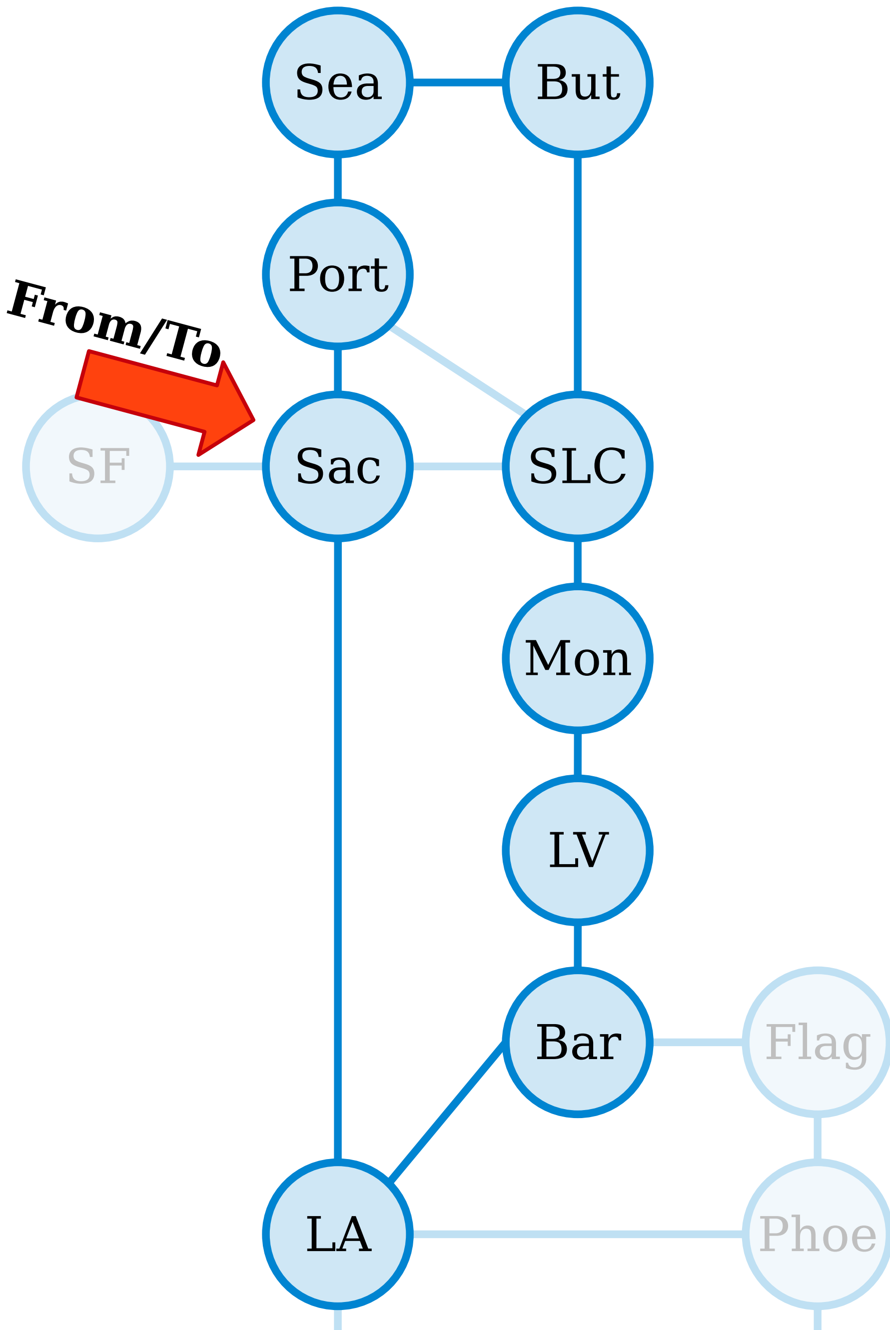
A ***path*** in a graph $G = (V, E)$ is a sequence of one or more nodes $v_1, v_2, v_3, \ldots, v_n$ such that any two consecutive nodes in the sequence are adjacent.

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

A ***path*** 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 path $v_1, \ldots, v_n$ is $n - 1$.

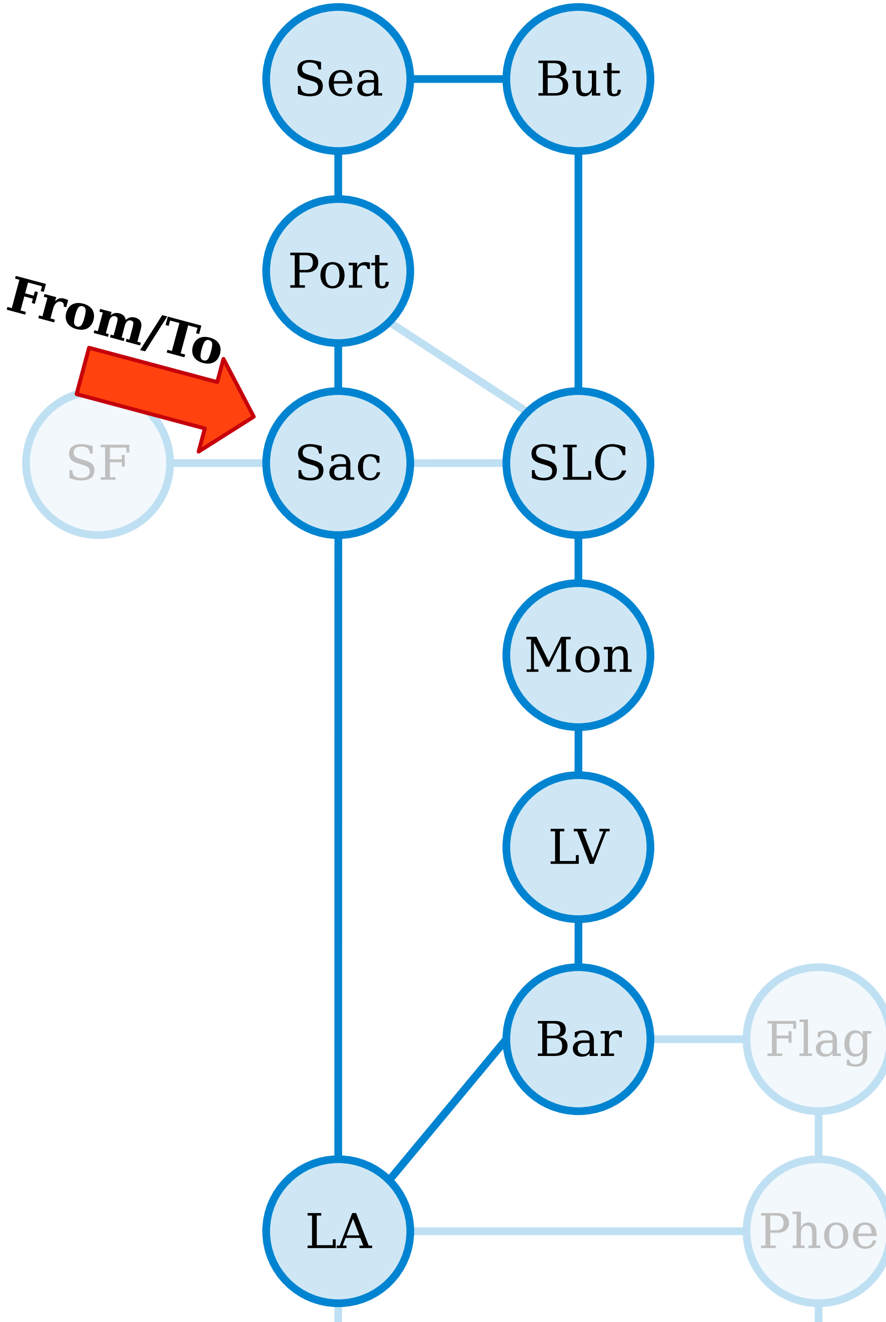




A ***path*** in a graph $G = (V, E)$ is a sequence of one or more nodes $v_1, v_2, v_3, \ldots, v_n$ such that any two consecutive nodes in the sequence are adjacent.

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

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

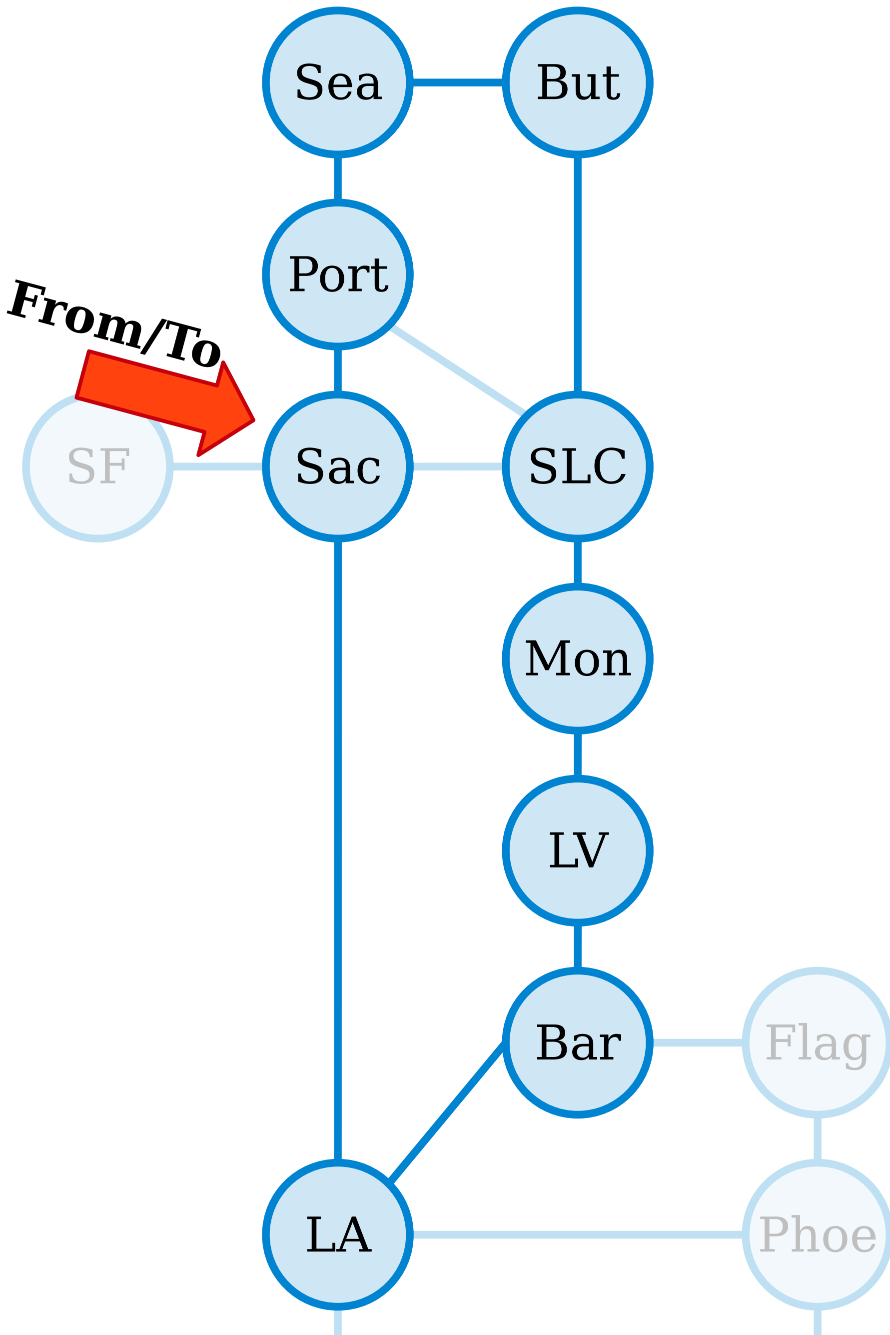


A ***path*** in a graph $G = (V, E)$ is a sequence of one or more nodes $v_1, v_2, v_3, \ldots, v_n$ such that any two consecutive nodes in the sequence are adjacent.

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

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

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



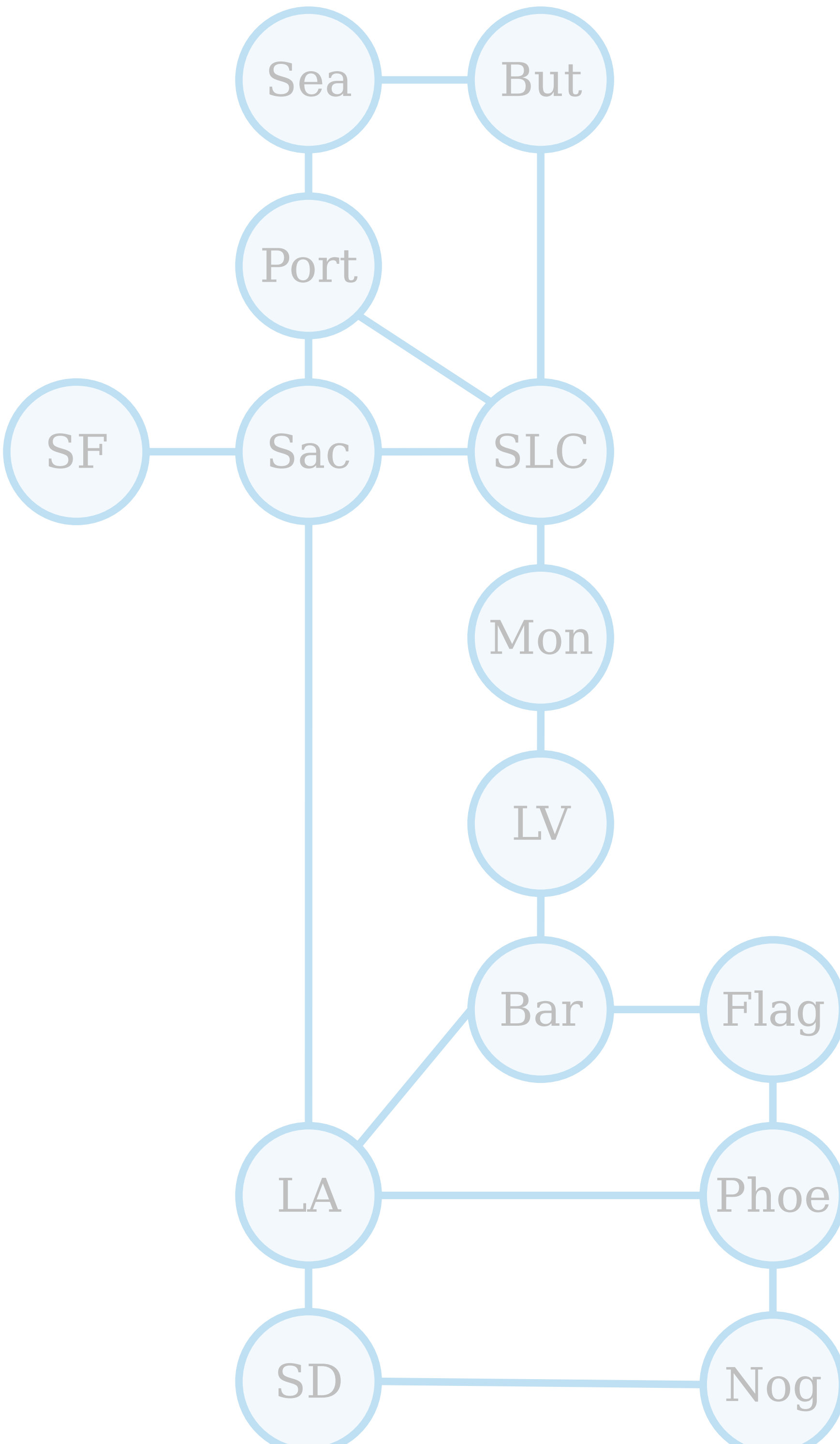
A ***path*** in a graph $G = (V, E)$ is a sequence of one or more nodes $v_1, v_2, v_3, \ldots, v_n$ such that any two consecutive nodes in the sequence are adjacent.

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

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

(This cycle has length nine and visits nine different cities.)

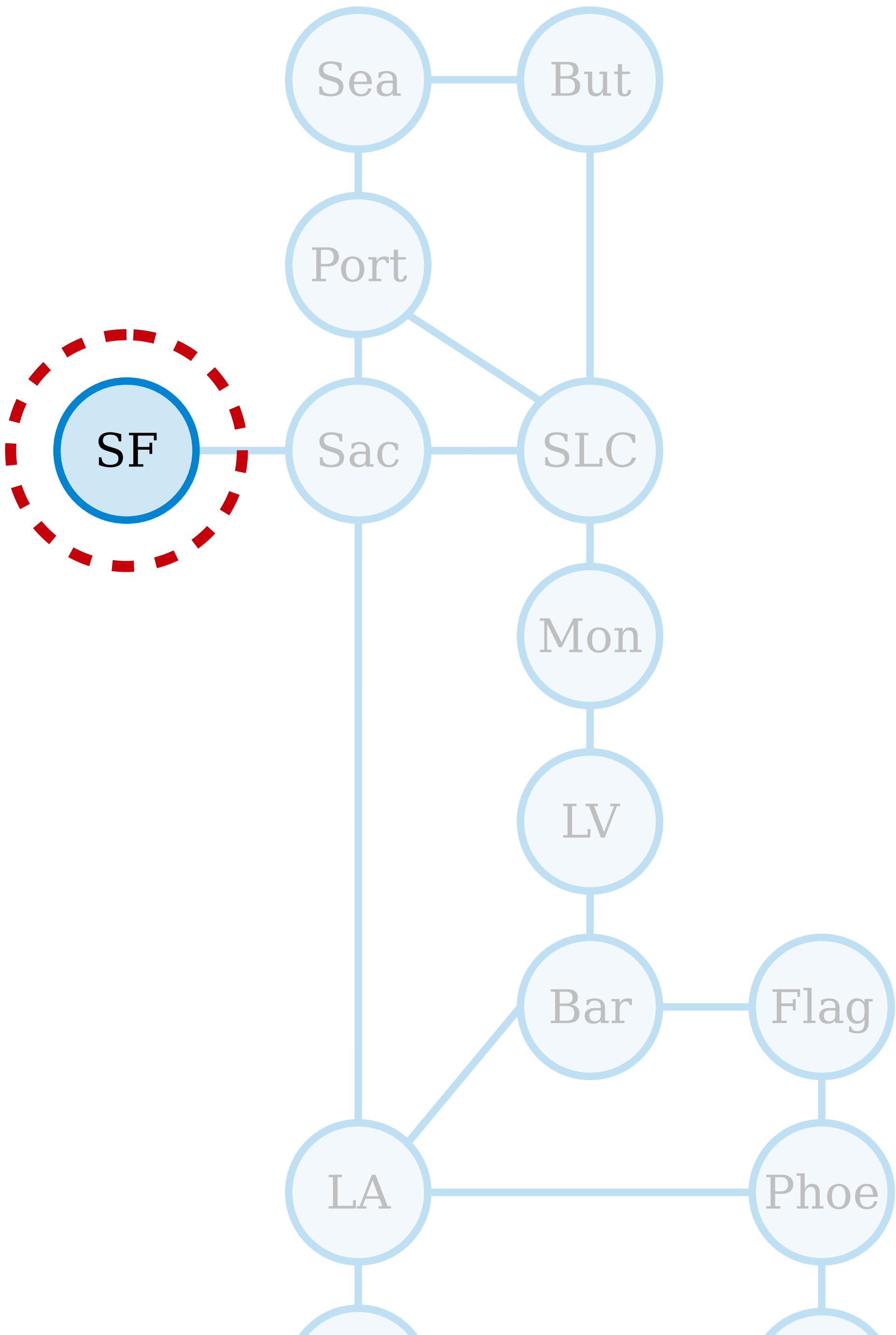
Sac, Port, Sea, But, SLC, Mon, LV, Bar, LA, Sac



A ***path*** 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 path $v_1, \ldots, v_n$ is $n - 1$.

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

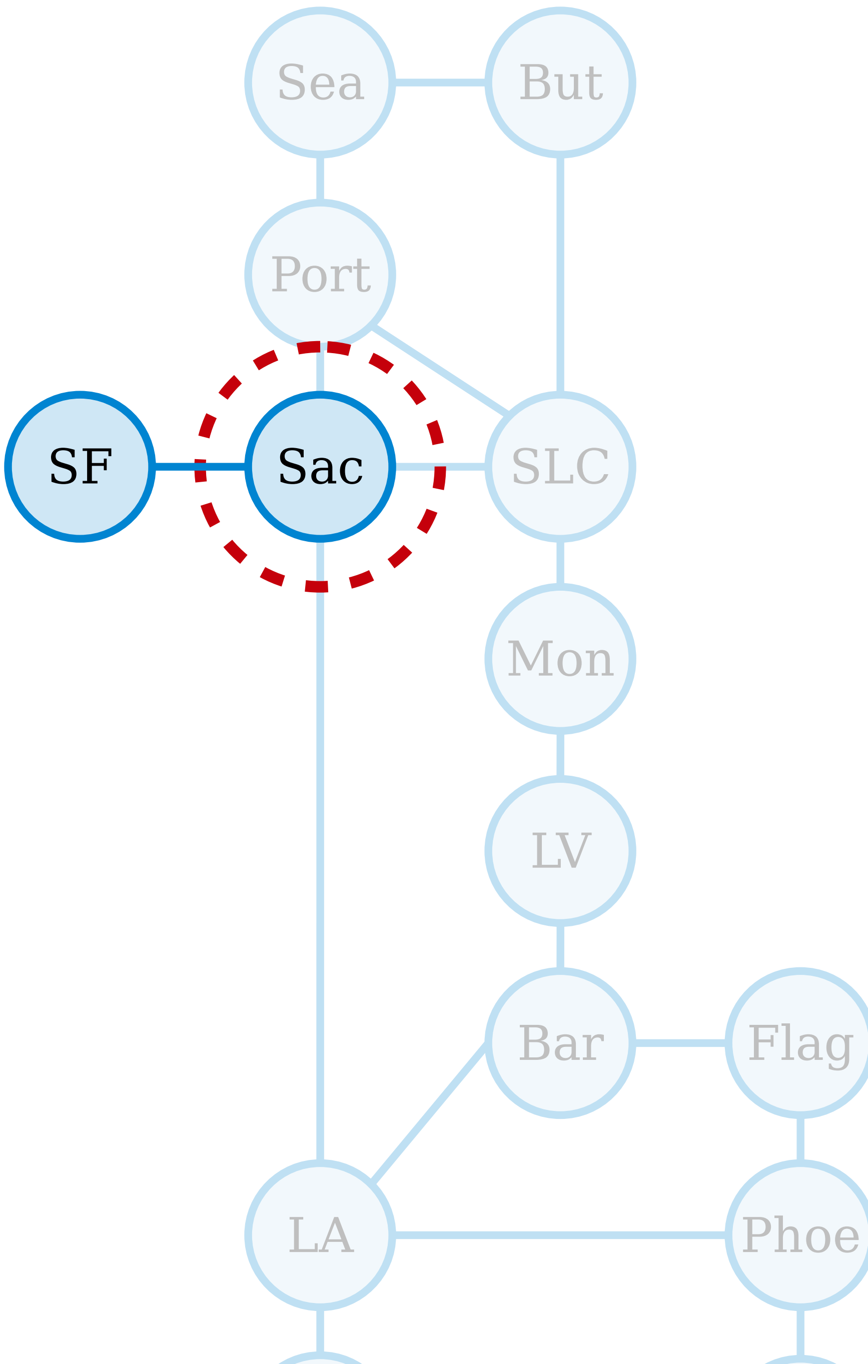


A ***path*** in a graph $G = (V, E)$ is a sequence of one or more nodes $v_1, v_2, v_3, \ldots, v_n$ such that any two consecutive nodes in the sequence are adjacent.

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

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

SF

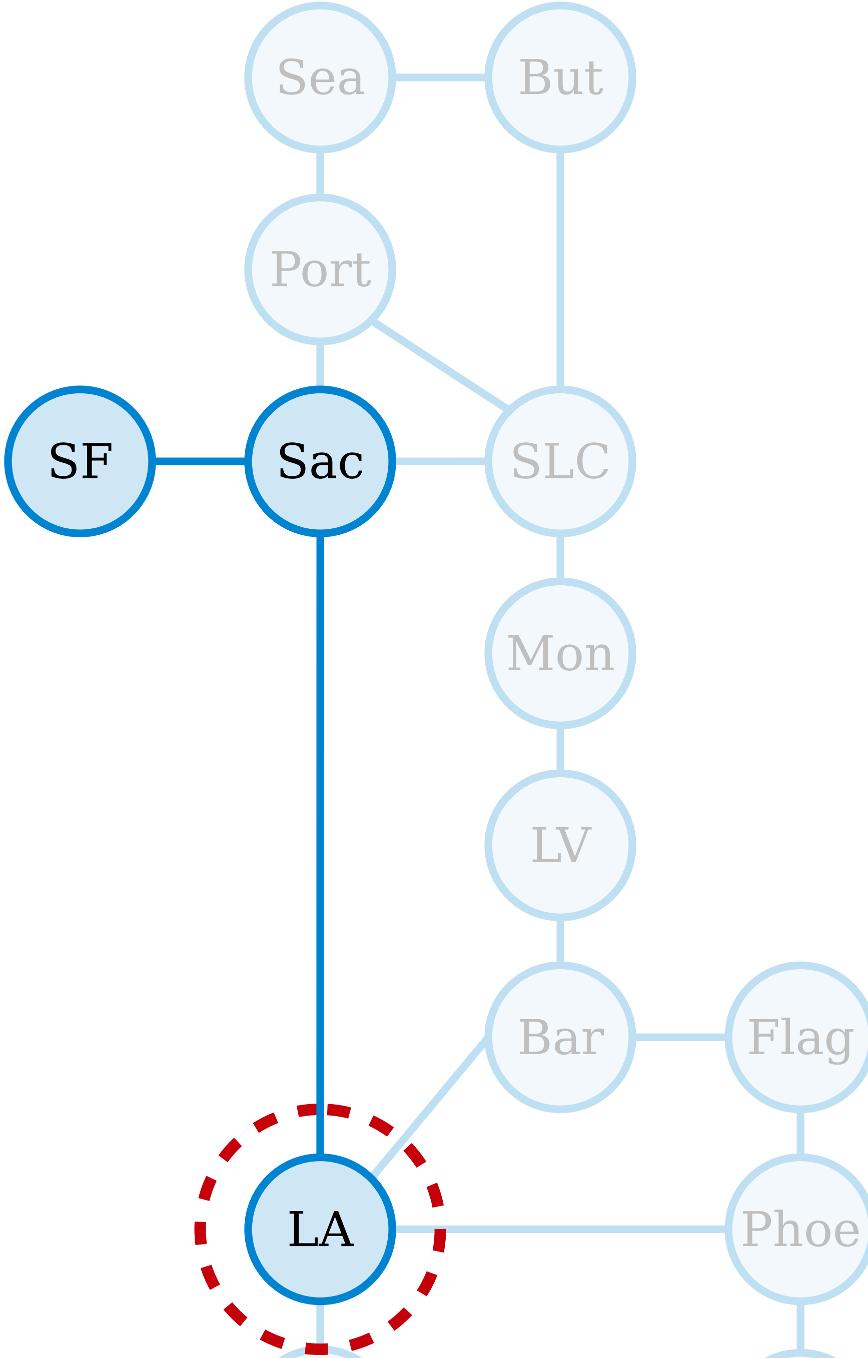


A ***path*** in a graph $G = (V, E)$ is a sequence of one or more nodes $v_1, v_2, v_3, \ldots, v_n$ such that any two consecutive nodes in the sequence are adjacent.

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

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

SF, Sac

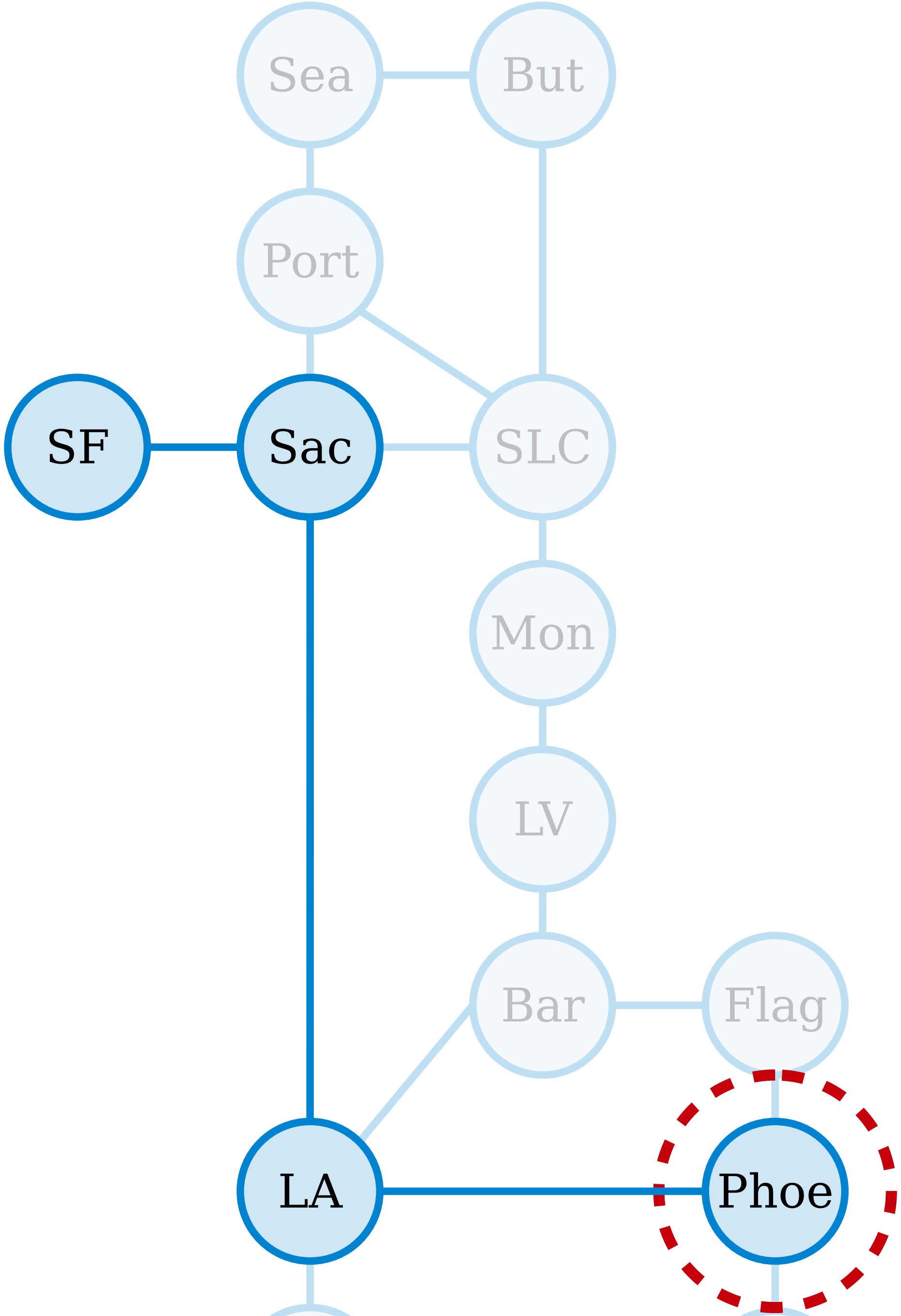


A ***path*** in a graph $G = (V, E)$ is a sequence of one or more nodes $v_1, v_2, v_3, \ldots, v_n$ such that any two consecutive nodes in the sequence are adjacent.

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

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

SF, Sac, LA

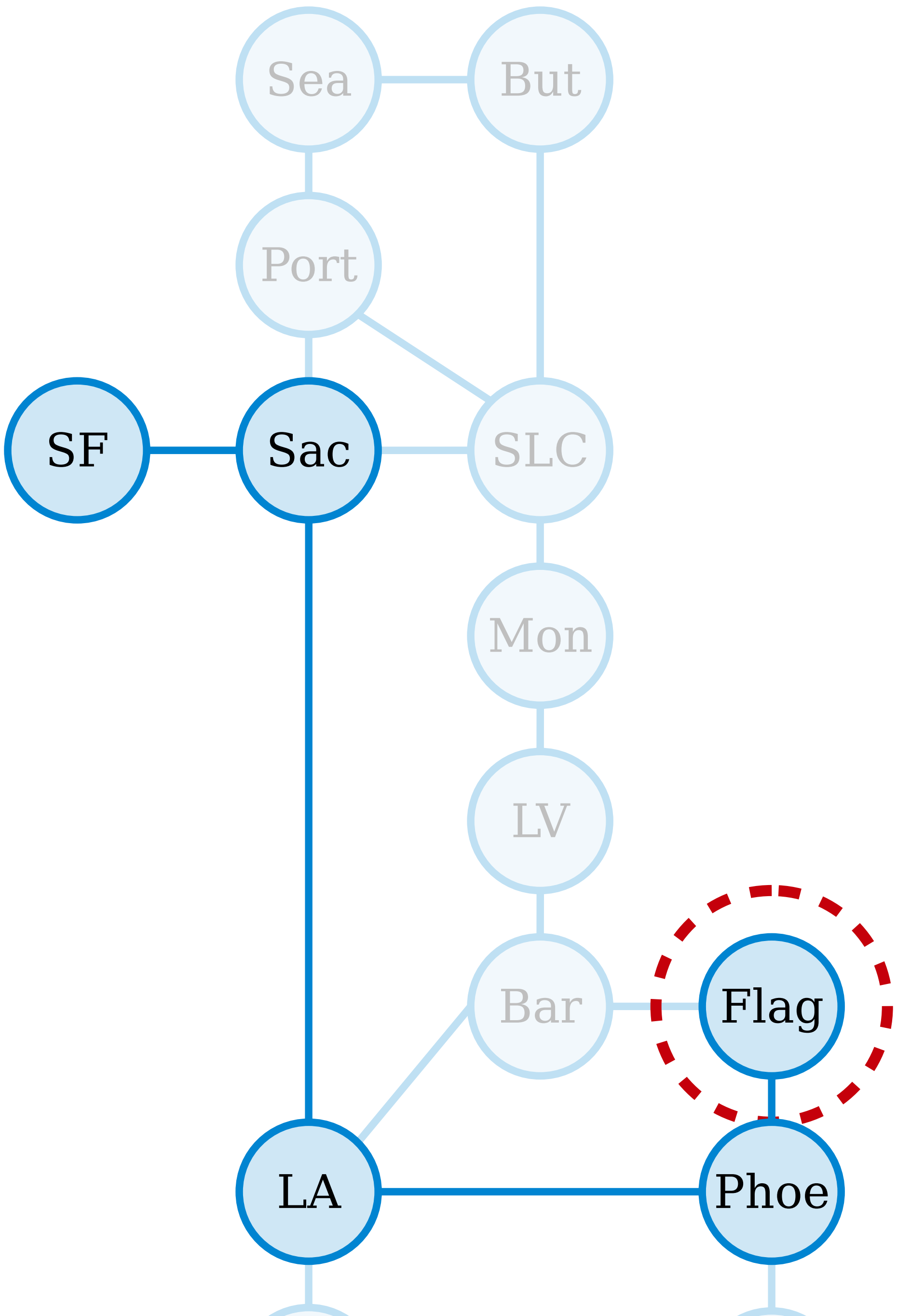


A ***path*** in a graph $G = (V, E)$ is a sequence of one or more nodes $v_1, v_2, v_3, \ldots, v_n$ such that any two consecutive nodes in the sequence are adjacent.

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

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

SF, Sac, LA, Phoe

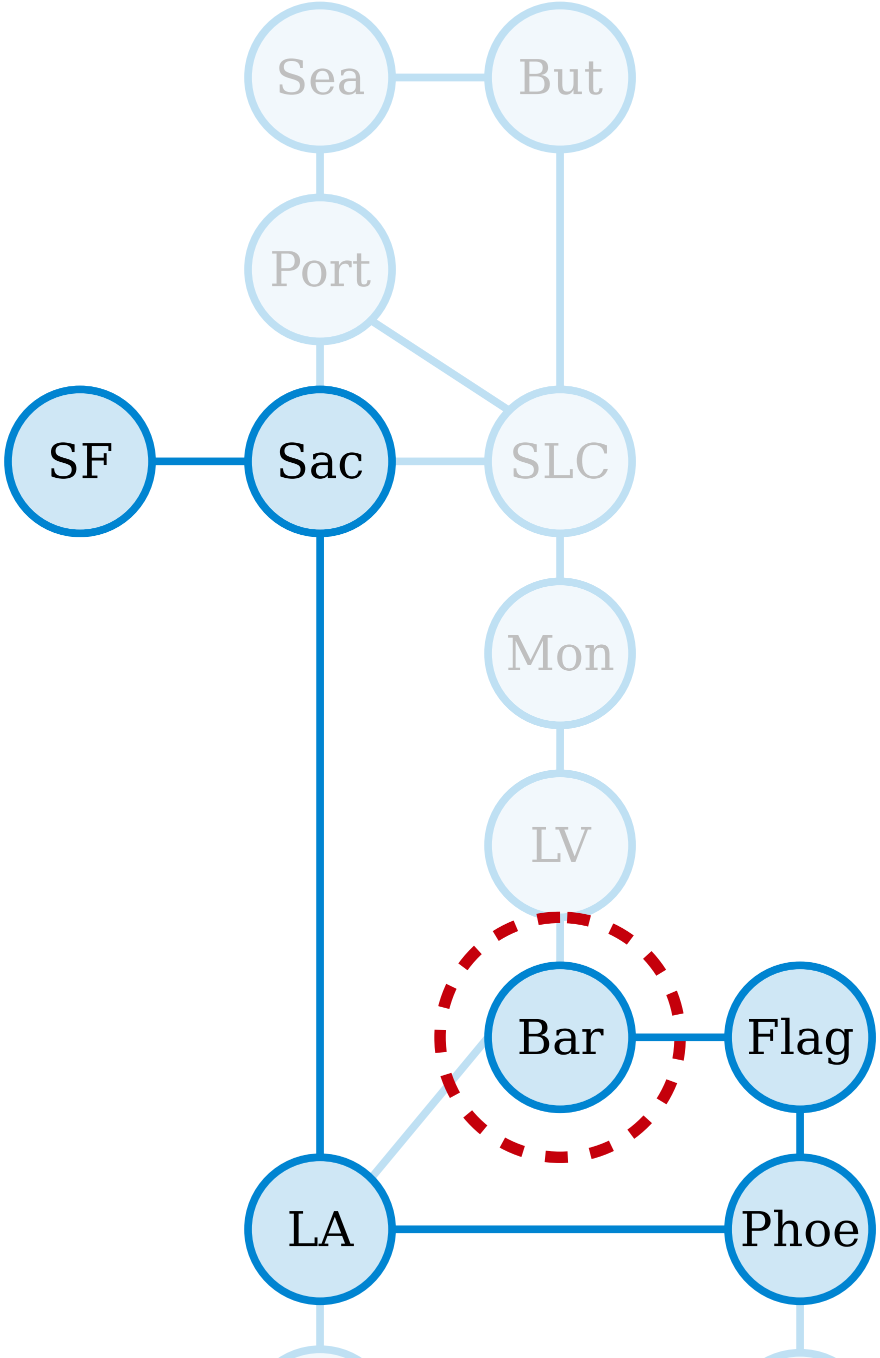


A ***path*** in a graph $G = (V, E)$ is a sequence of one or more nodes $v_1, v_2, v_3, \ldots, v_n$ such that any two consecutive nodes in the sequence are adjacent.

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

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

SF, Sac, LA, Phoe, Flag

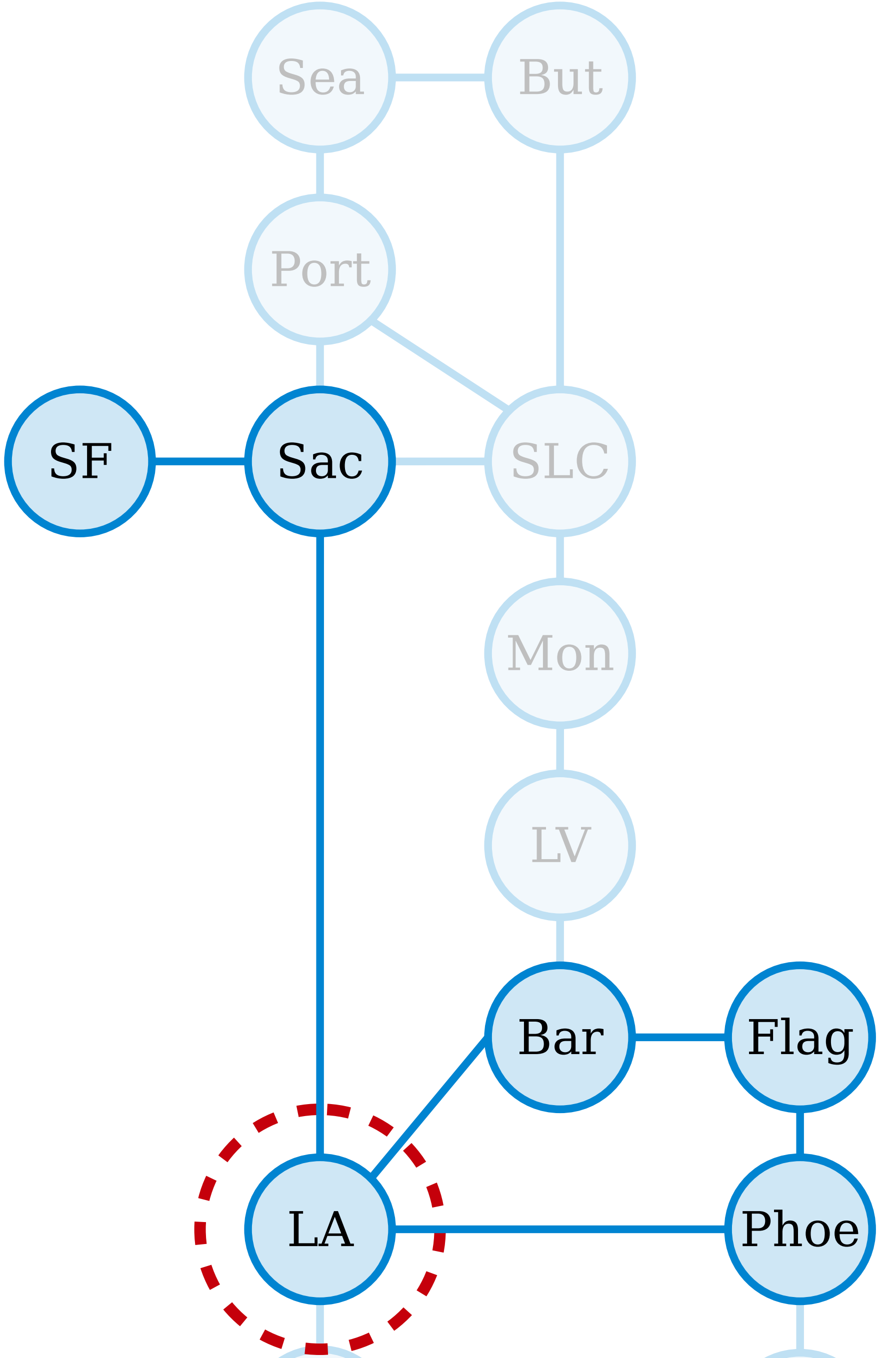


A ***path*** in a graph $G = (V, E)$ is a sequence of one or more nodes $v_1, v_2, v_3, \ldots, v_n$ such that any two consecutive nodes in the sequence are adjacent.

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

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

SF, Sac, LA, Phoe, Flag, Bar

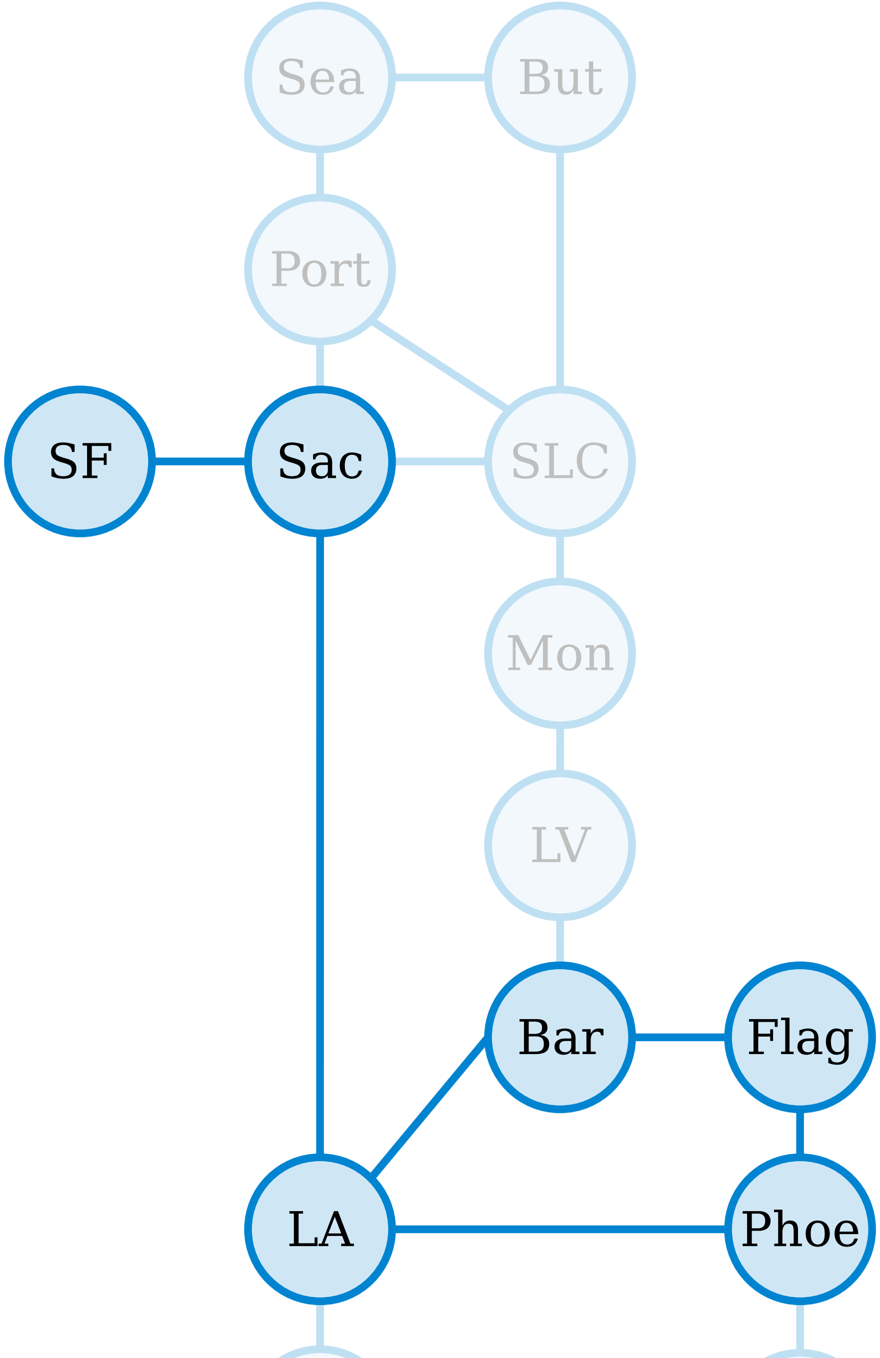


A ***path*** in a graph $G = (V, E)$ is a sequence of one or more nodes $v_1, v_2, v_3, \ldots, v_n$ such that any two consecutive nodes in the sequence are adjacent.

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

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

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

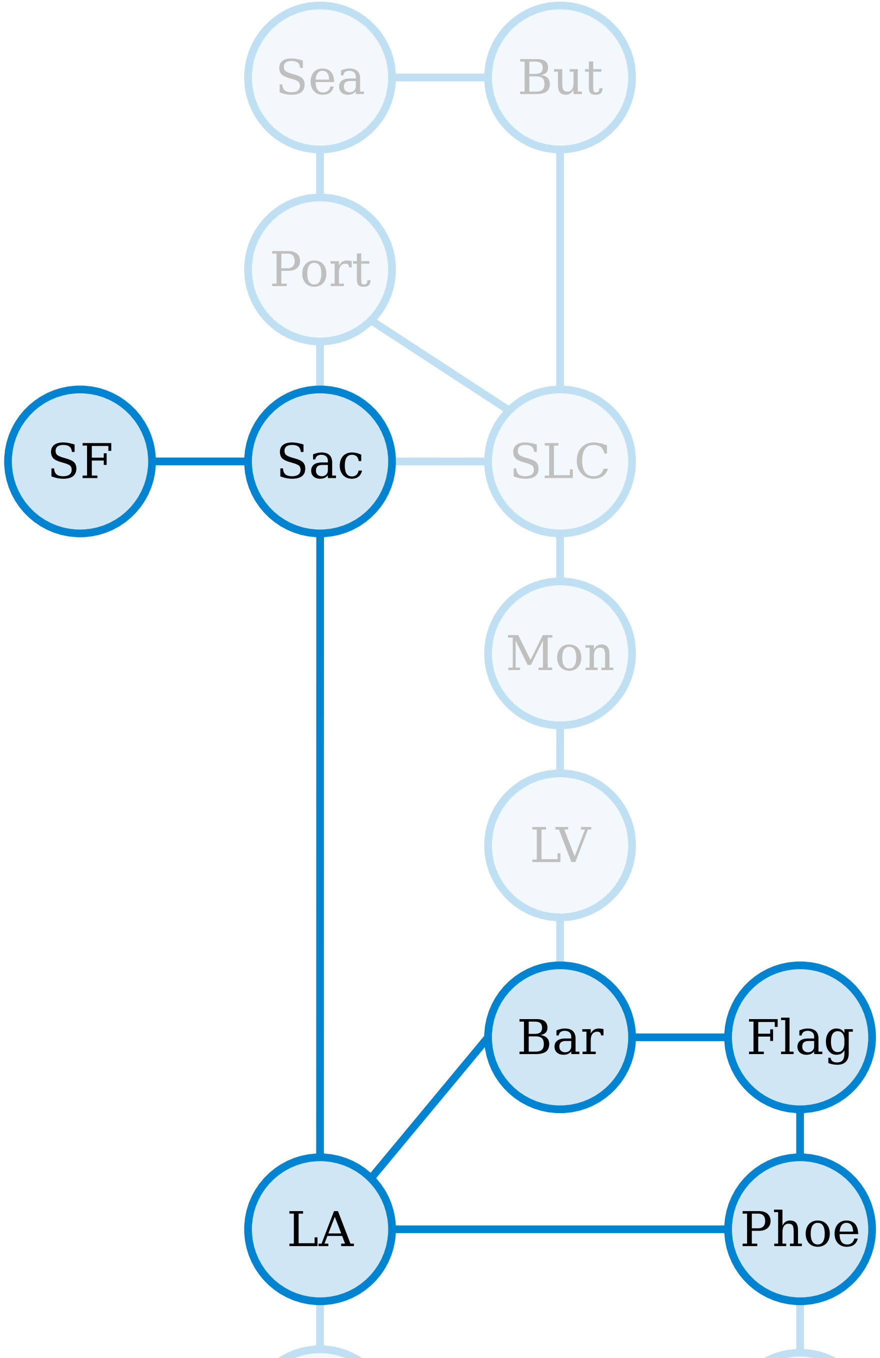


A ***path*** in a graph $G = (V, E)$ is a sequence of one or more nodes $v_1, v_2, v_3, \ldots, v_n$ such that any two consecutive nodes in the sequence are adjacent.

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

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

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



A ***path*** 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 path $v_1, \ldots, v_n$ is $n - 1$.

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

A ***simple path*** in a graph is path that does not repeat any nodes or edges.

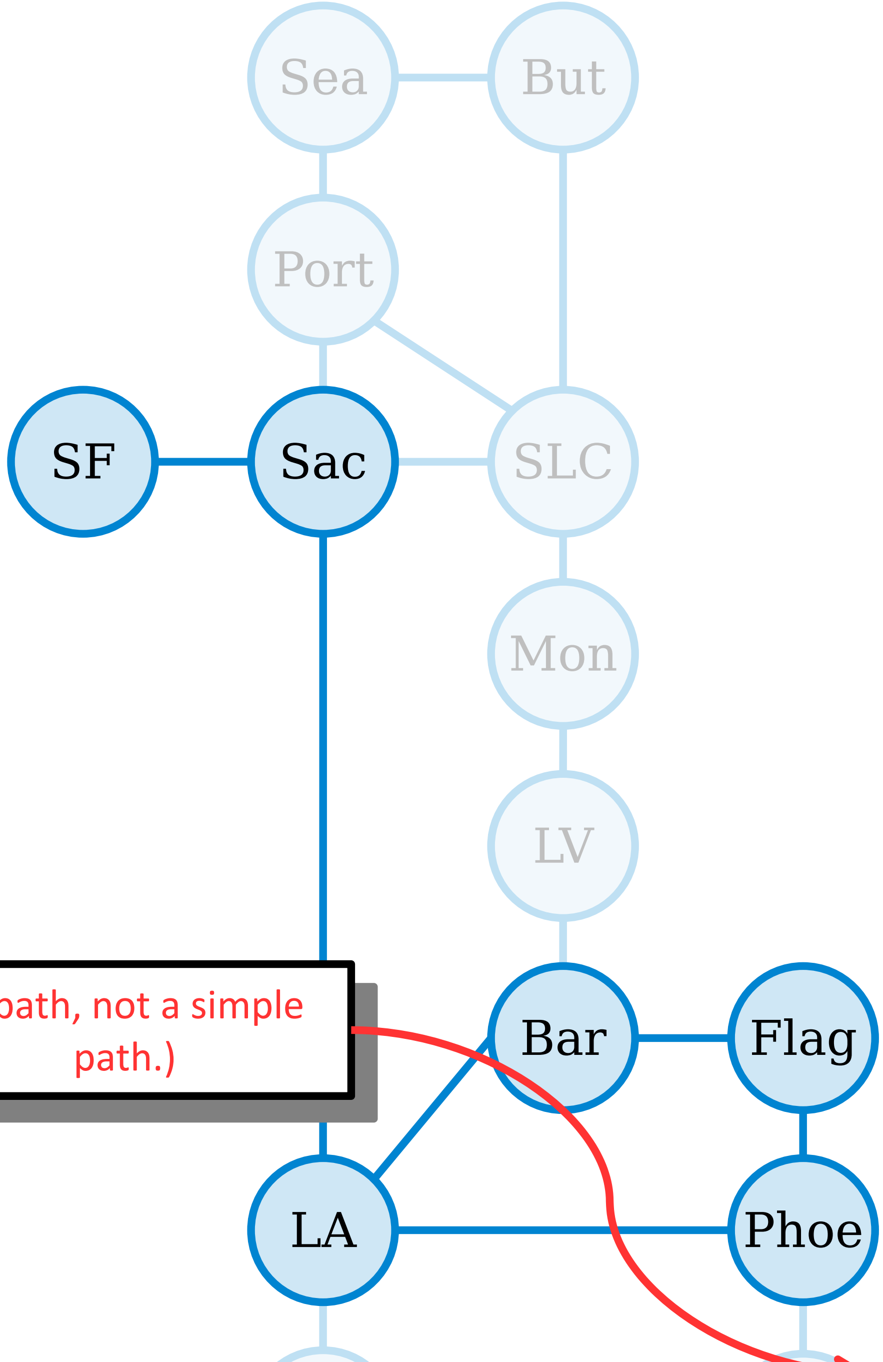
SF, Sac, LA, Phoe, Flag, Bar, LA

A ***path*** 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 path $v_1, \ldots, v_n$ is $n - 1$.

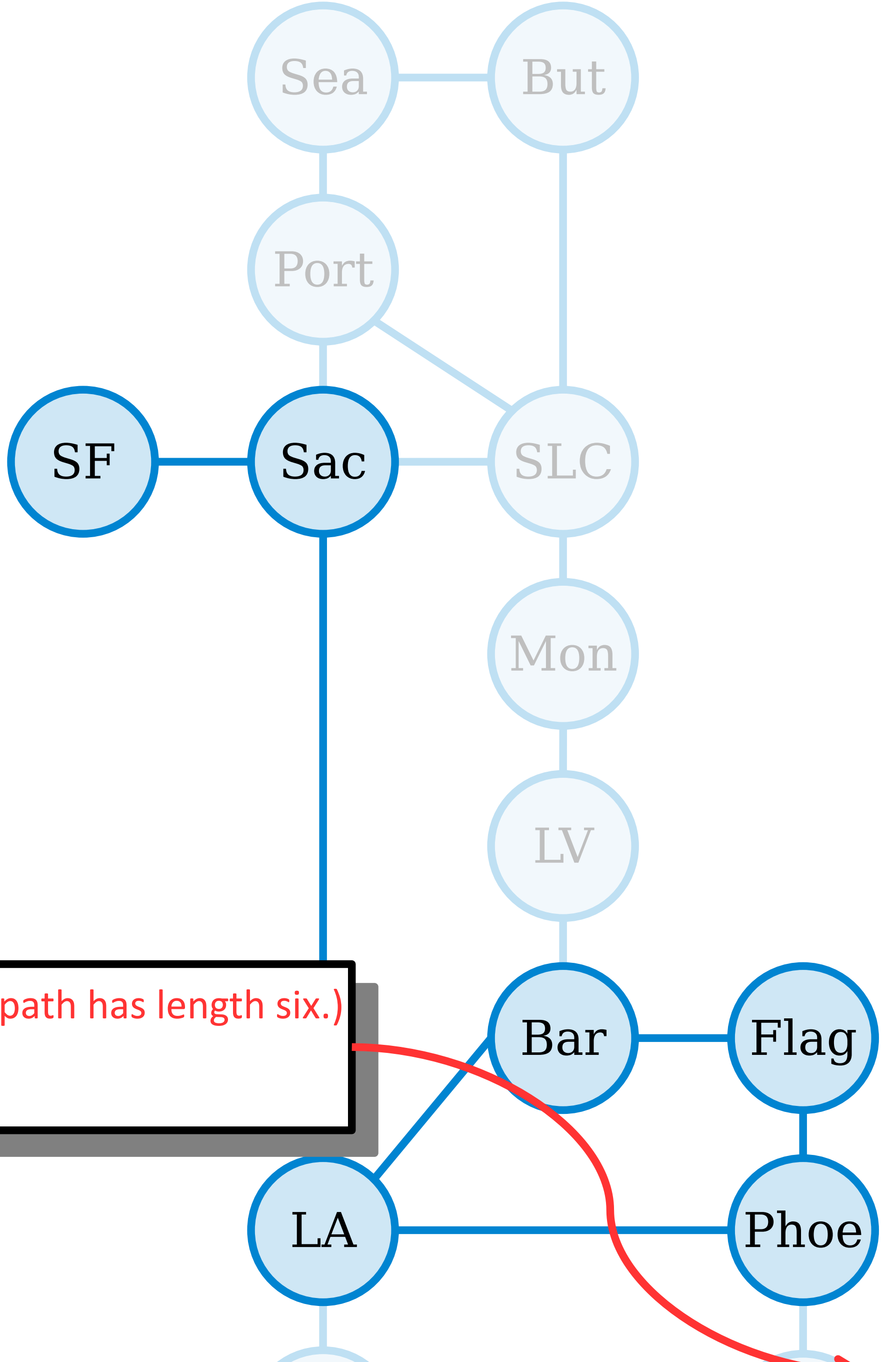
A ***cycle*** in a graph is a path from a node back to itself. (By convention, a cycle cannot have length zero.)

A ***simple path*** in a graph is path that does not repeat any nodes or edges.



(A path, not a simple path.)

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



A ***path*** 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.

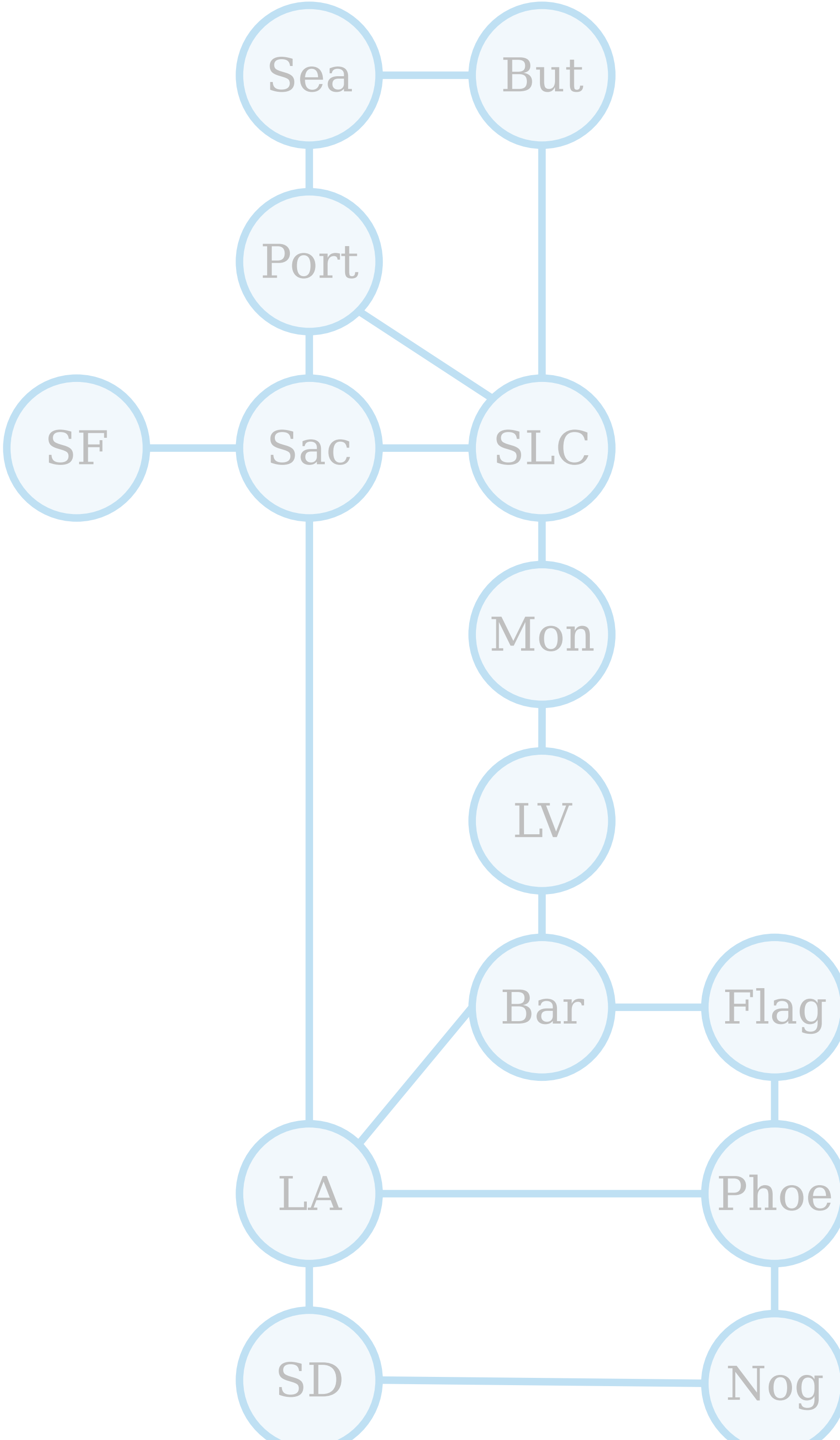
The ***length*** of the path $v_1, \ldots, v_n$ is $n - 1$.

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

A ***simple path*** in a graph is path that does not repeat any nodes or edges.

(This path has length six.)

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

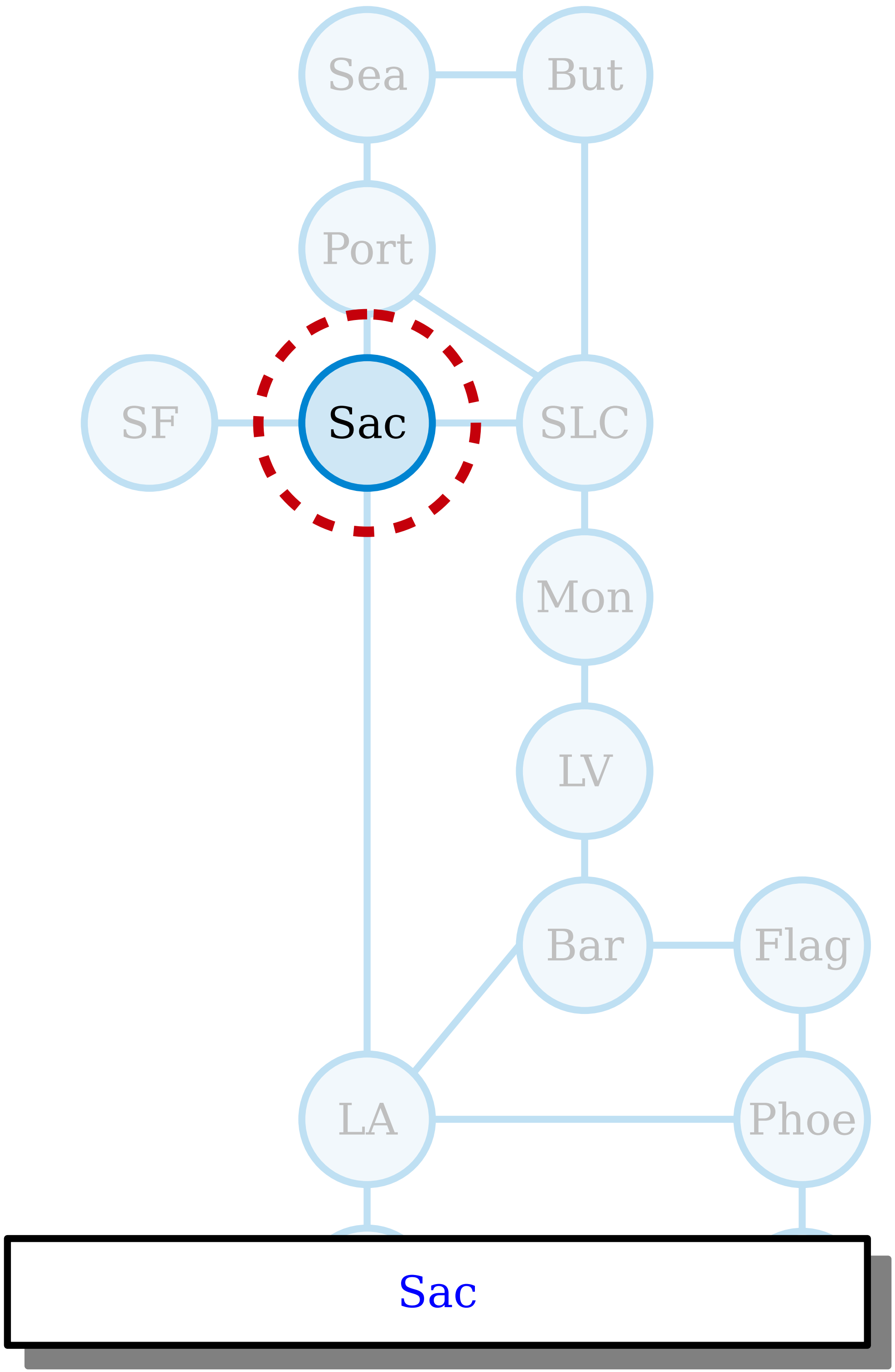


A ***path*** in a graph $G = (V, E)$ is a sequence of one or more nodes $v_1, v_2, v_3, \ldots, v_n$ such that any two consecutive nodes in the sequence are adjacent.

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

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

A ***simple path*** in a graph is path that does not repeat any nodes or edges.

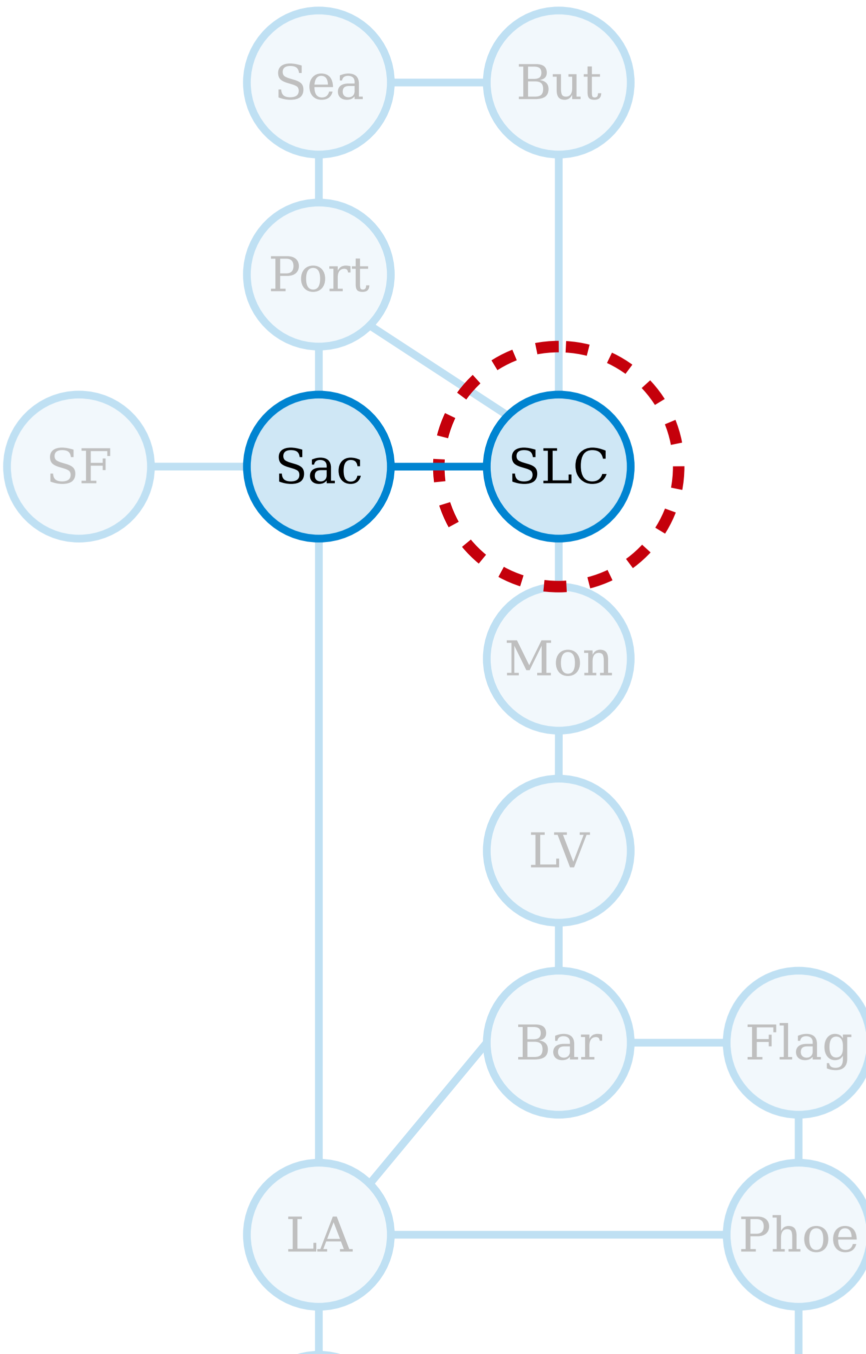


A ***path*** in a graph $G = (V, E)$ is a sequence of one or more nodes $v_1, v_2, v_3, \ldots, v_n$ such that any two consecutive nodes in the sequence are adjacent.

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

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

A ***simple path*** in a graph is path that does not repeat any nodes or edges.



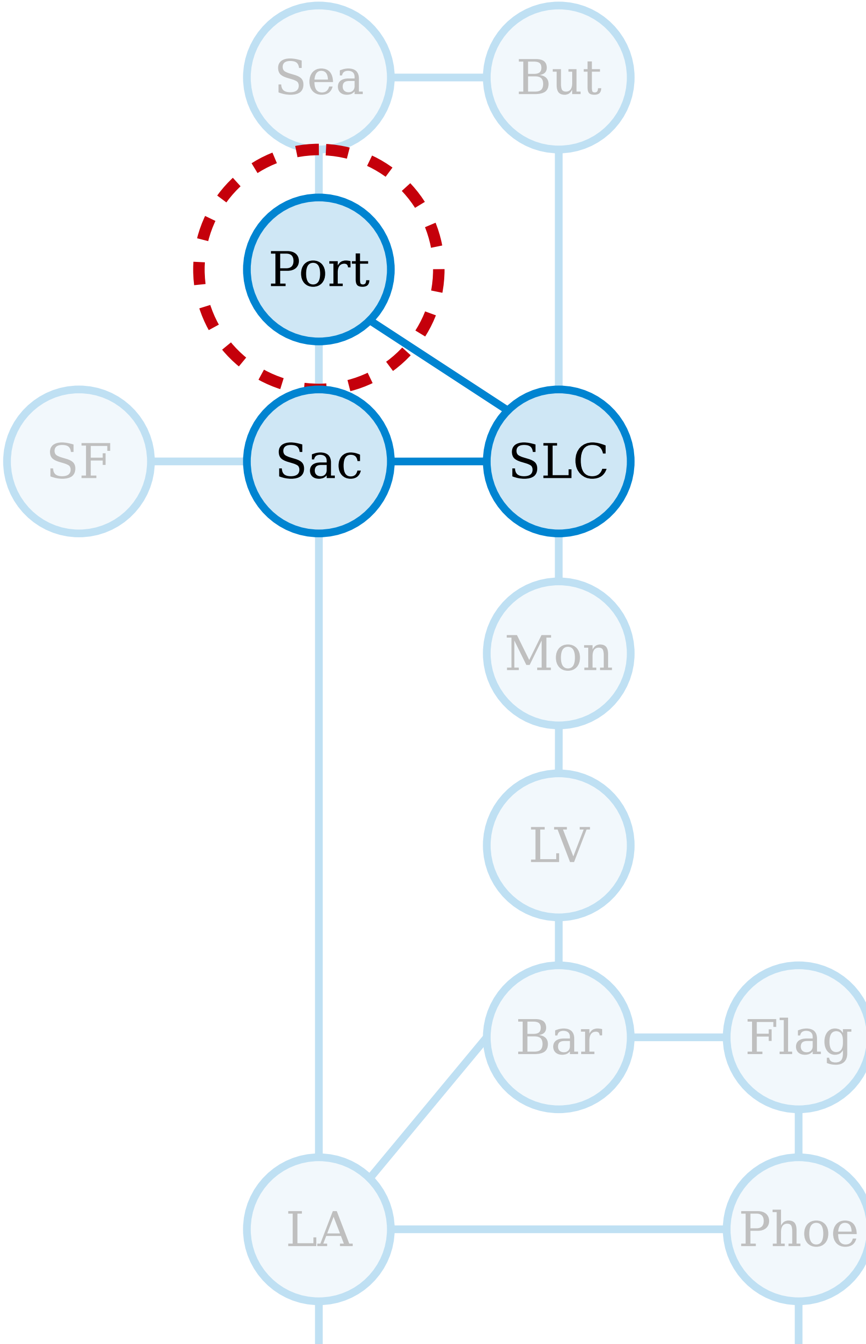
Sac, SLC

A ***path*** 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 path $v_1, \ldots, v_n$ is $n - 1$.

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

A ***simple path*** in a graph is path that does not repeat any nodes or edges.



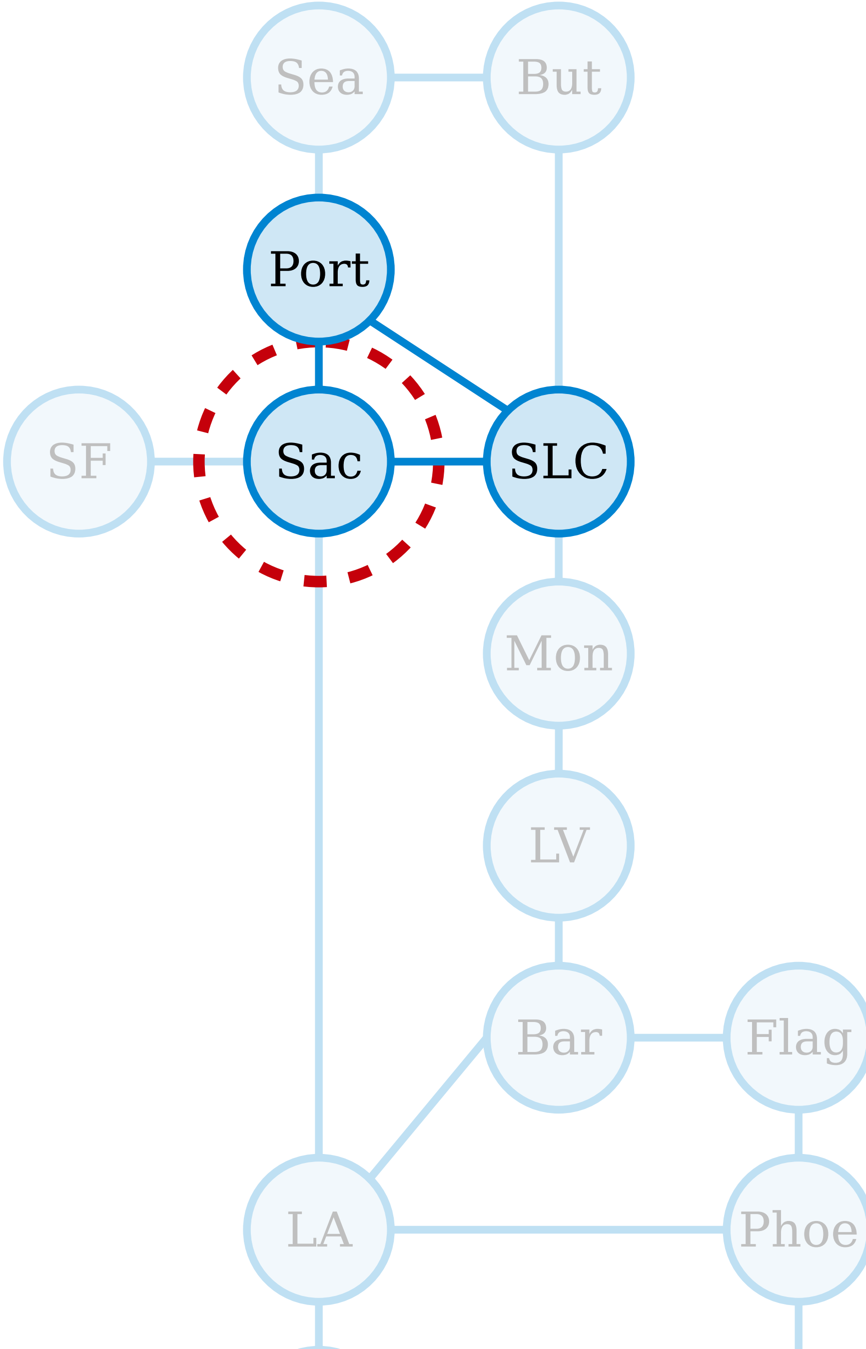
Sac, SLC, Port

A ***path*** 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 path $v_1, \ldots, v_n$ is $n - 1$.

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

A ***simple path*** in a graph is path that does not repeat any nodes or edges.



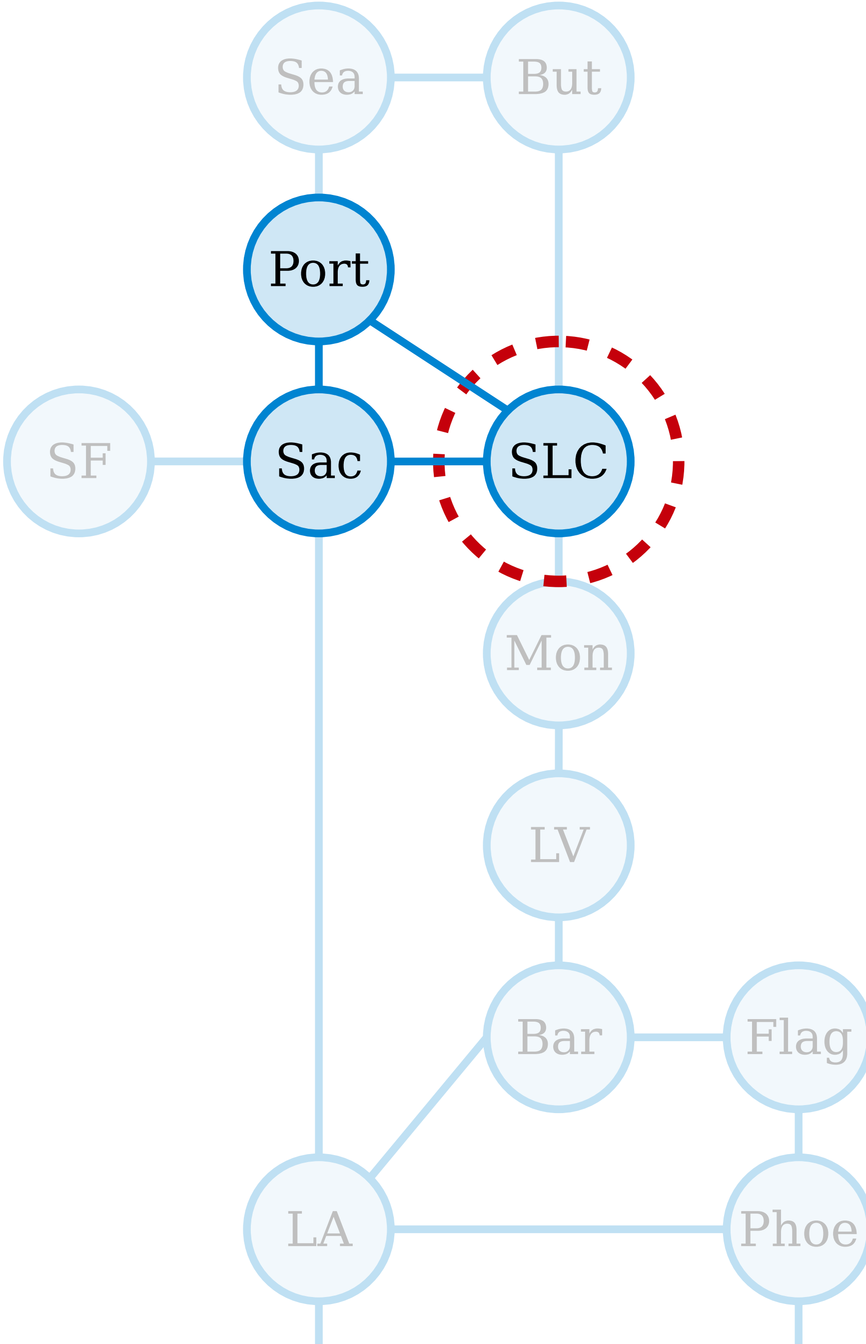
Sac, SLC, Port, Sac

A ***path*** 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 path $v_1, \ldots, v_n$ is $n - 1$.

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

A ***simple path*** in a graph is path that does not repeat any nodes or edges.



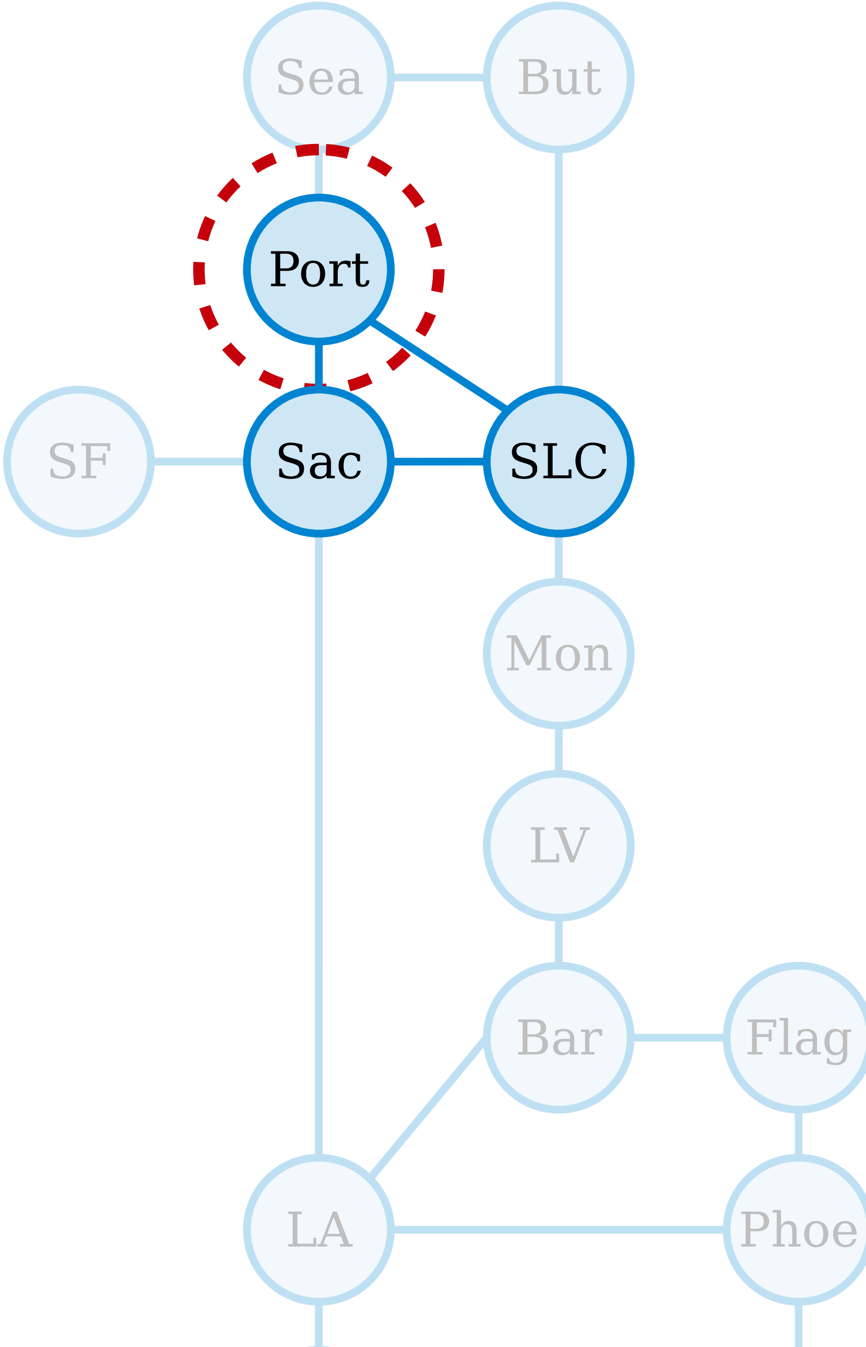
Sac, SLC, Port, Sac, SLC

A ***path*** 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 path $v_1, \ldots, v_n$ is $n - 1$.

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

A ***simple path*** in a graph is path that does not repeat any nodes or edges.



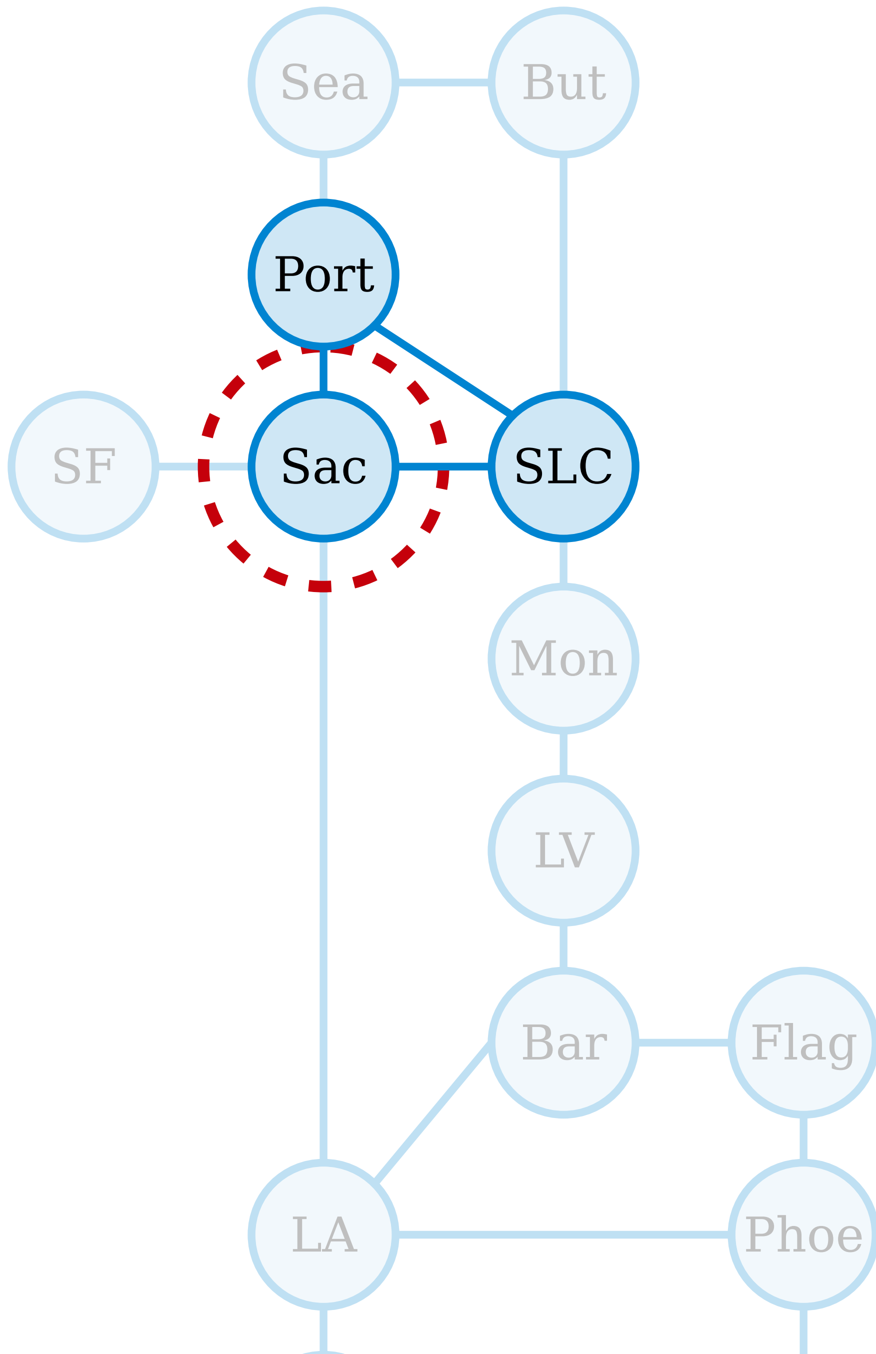
Sac, SLC, Port, Sac, SLC, Port

A ***path*** 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 path $v_1, \ldots, v_n$ is $n - 1$.

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

A ***simple path*** in a graph is path that does not repeat any nodes or edges.



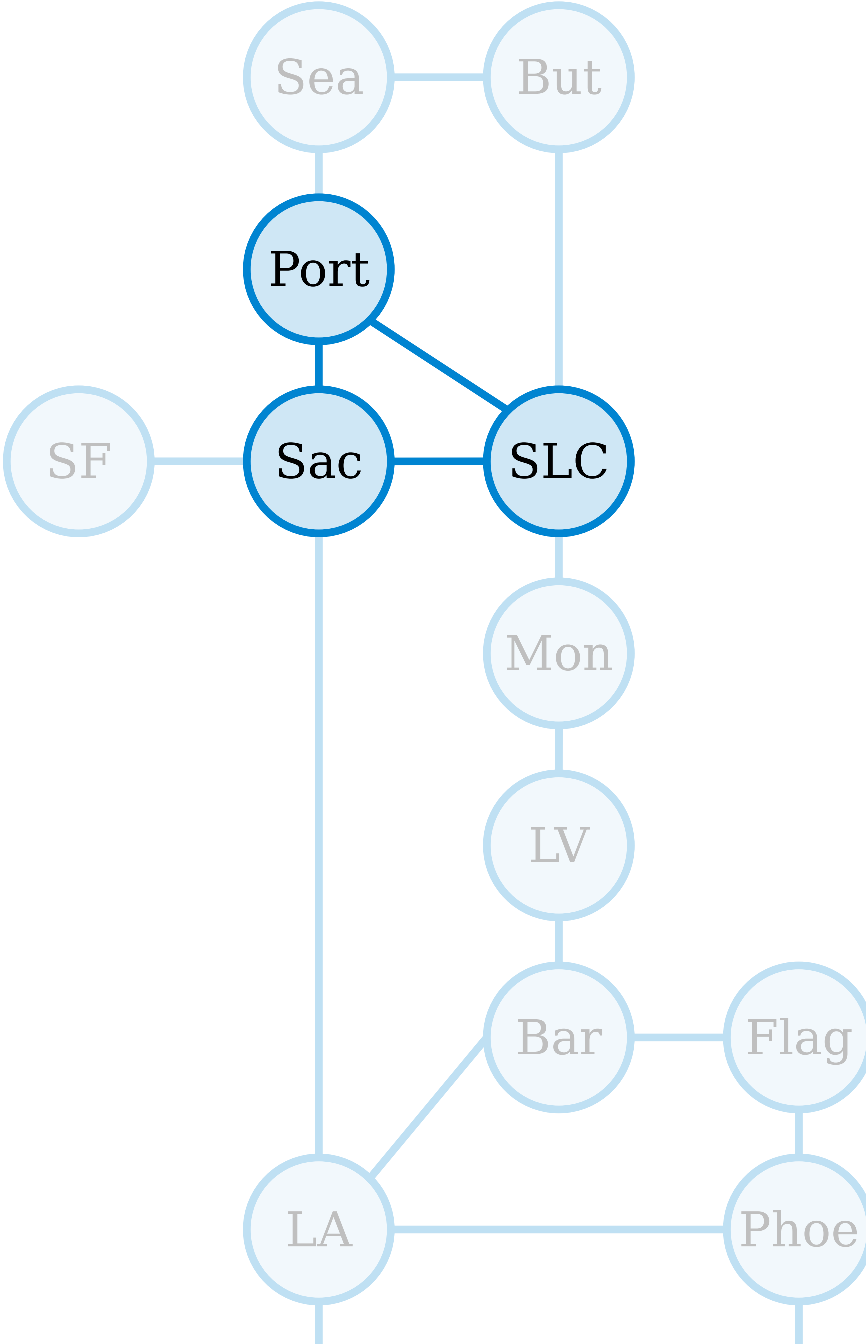
Sac, SLC, Port, Sac, SLC, Port, Sac

A ***path*** 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 path $v_1, \ldots, v_n$ is $n - 1$.

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

A ***simple path*** in a graph is path that does not repeat any nodes or edges.



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

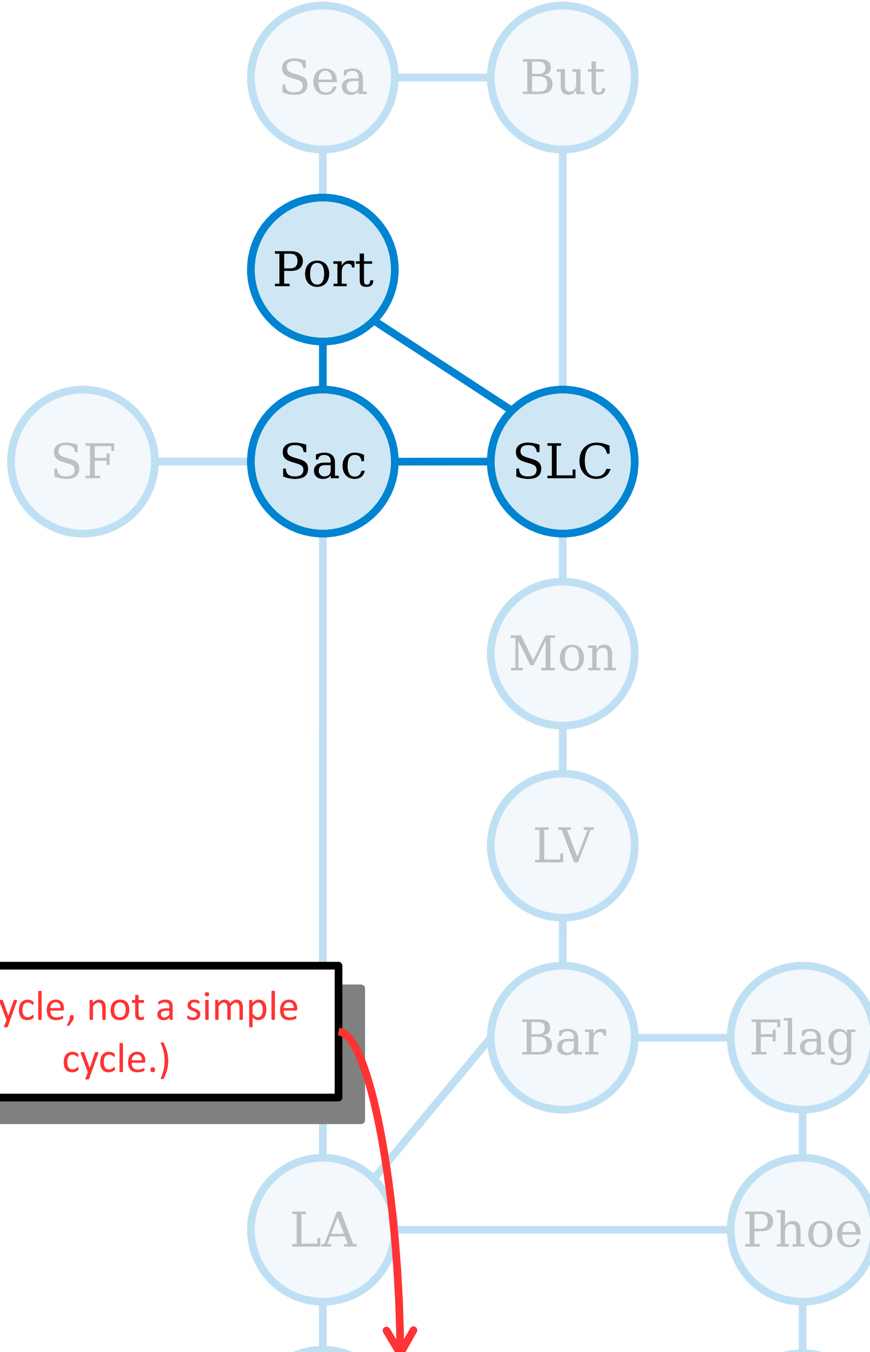
A ***path*** in a graph $G = (V, E)$ is a sequence of one or more nodes $v_1, v_2, v_3, \ldots, v_n$ such that any two consecutive nodes in the sequence are adjacent.

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

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

A ***simple path*** in a graph is path that does not repeat any nodes or edges.

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



(A cycle, not a simple cycle.)

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

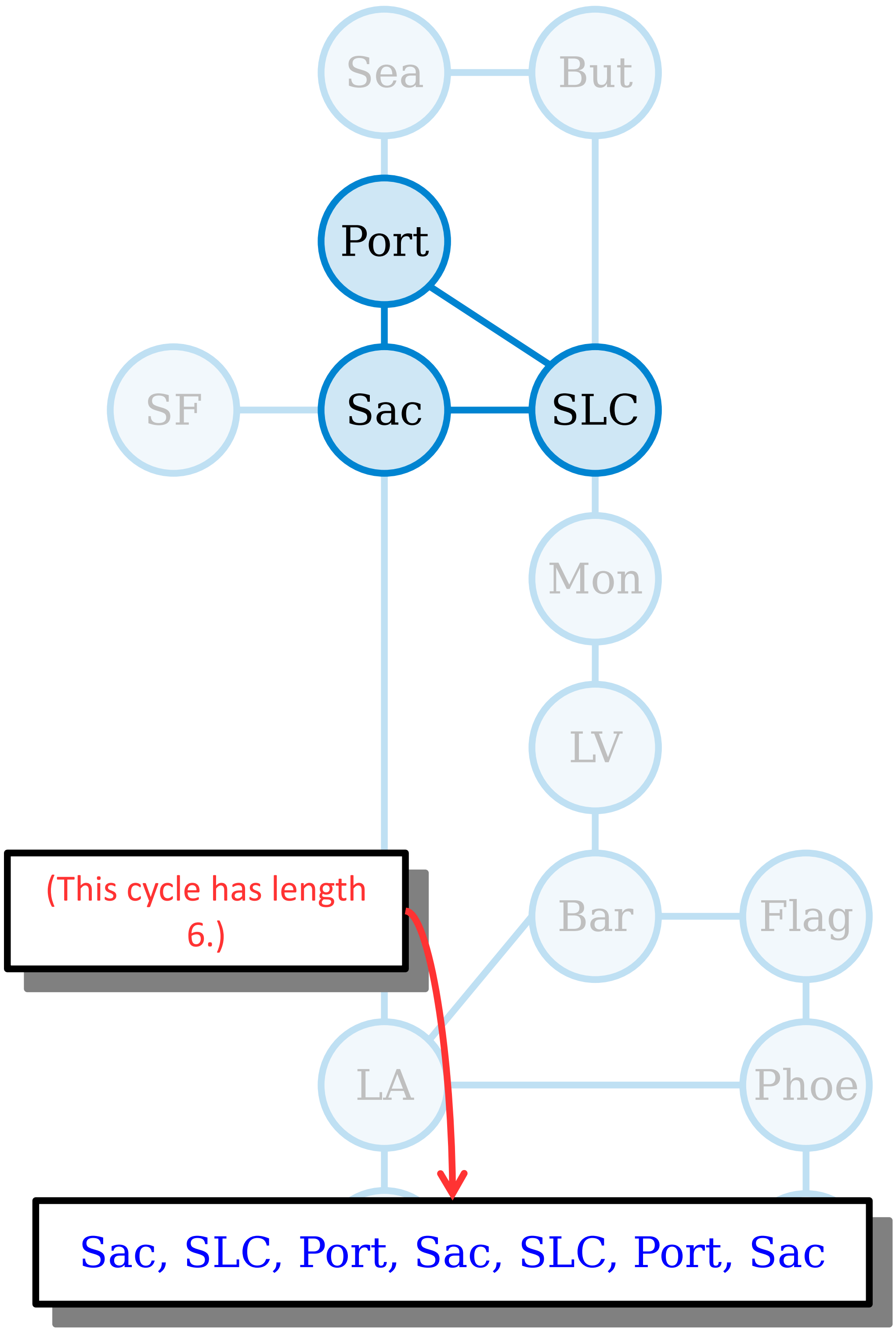
A ***path*** in a graph $G = (V, E)$ is a sequence of one or more nodes $v_1, v_2, v_3, \ldots, v_n$ such that any two consecutive nodes in the sequence are adjacent.

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

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

A ***simple path*** in a graph is path that does not repeat any nodes or edges.

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



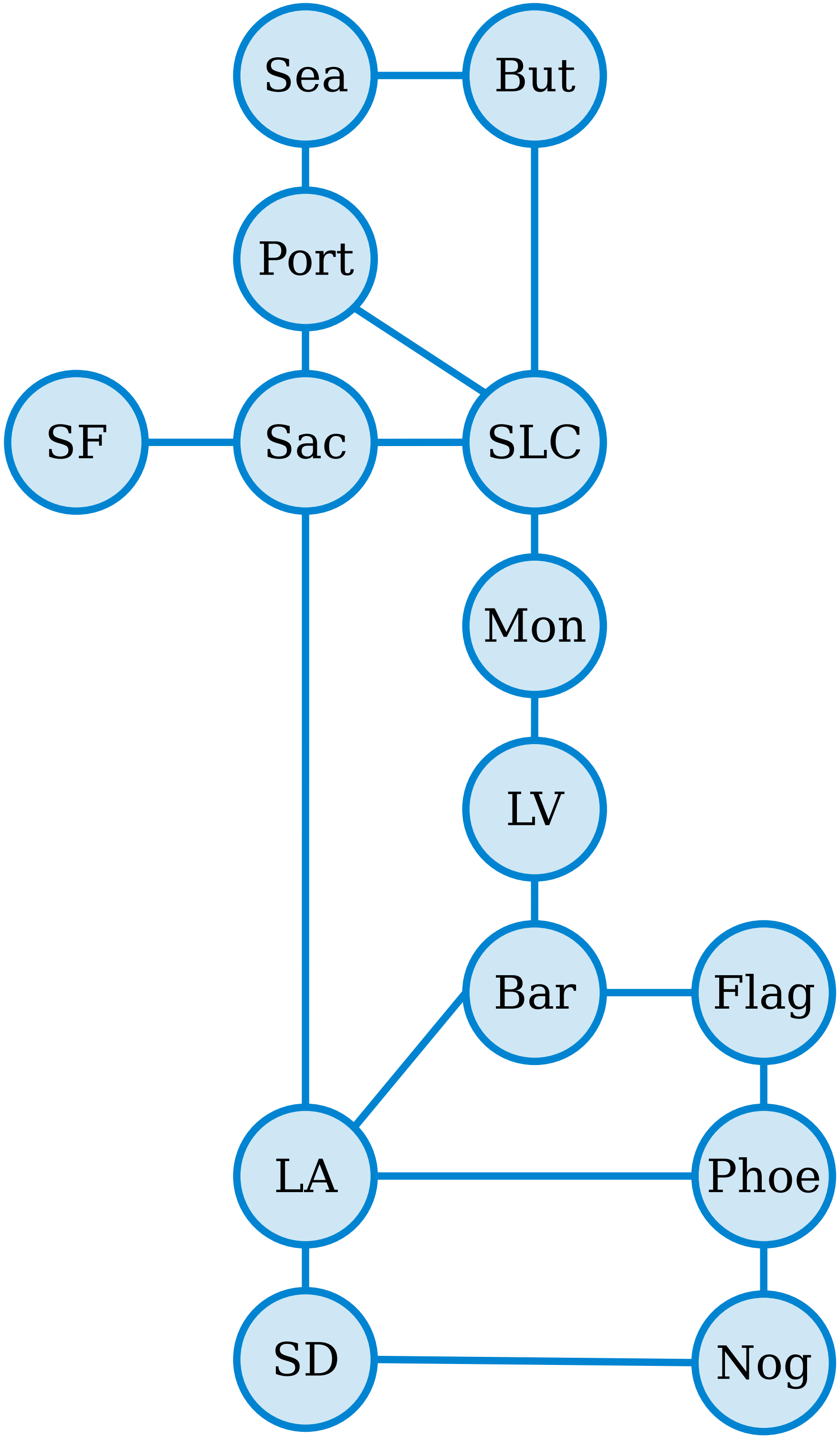
A ***path*** in a graph $G = (V, E)$ is a sequence of one or more nodes $v_1, v_2, v_3, \ldots, v_n$ such that any two consecutive nodes in the sequence are adjacent.

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

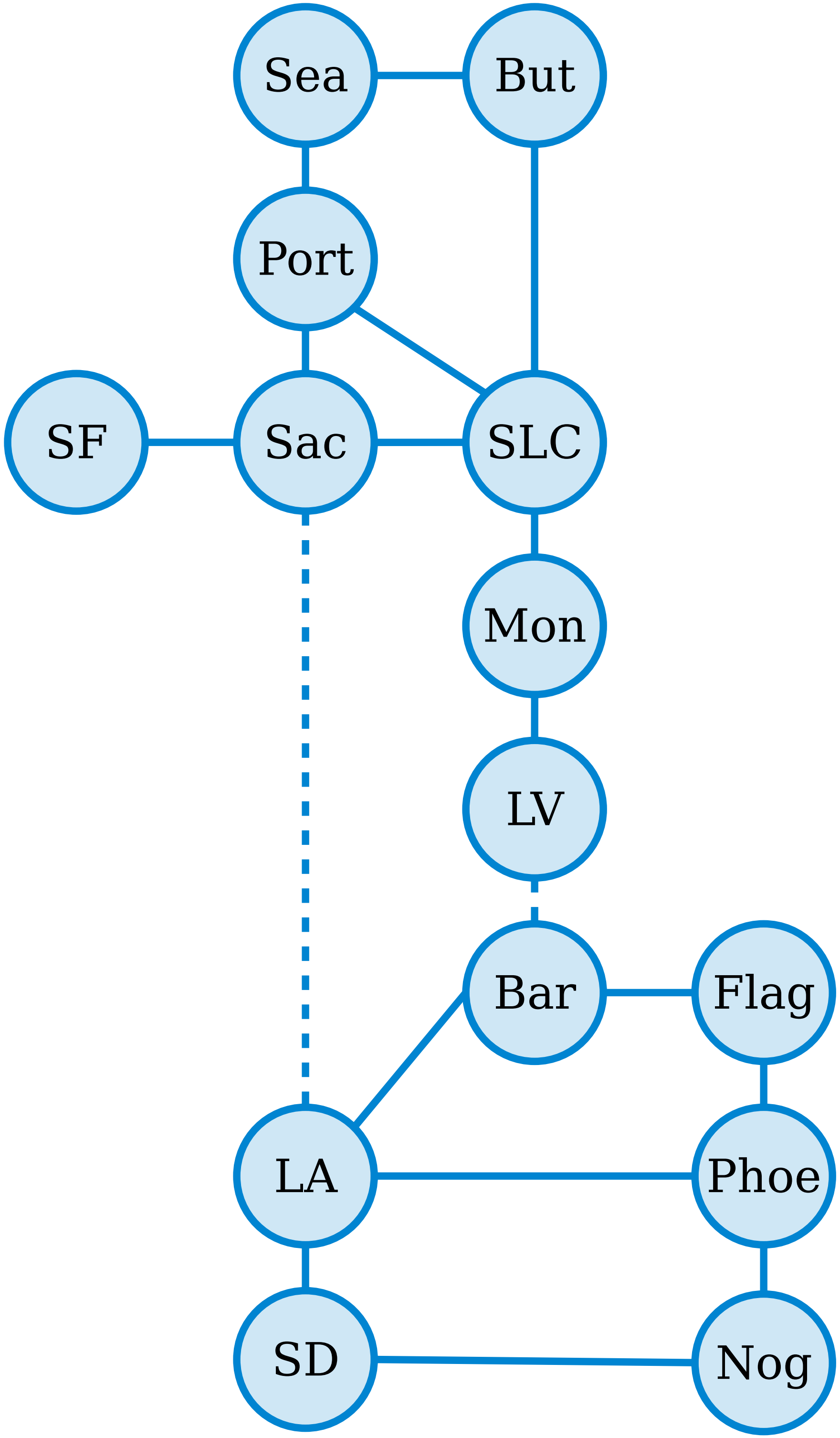
A ***cycle*** in a graph is a path from a node back to itself. (By convention, a cycle cannot have length zero.)

A ***simple path*** in a graph is path that does not repeat any nodes or edges.

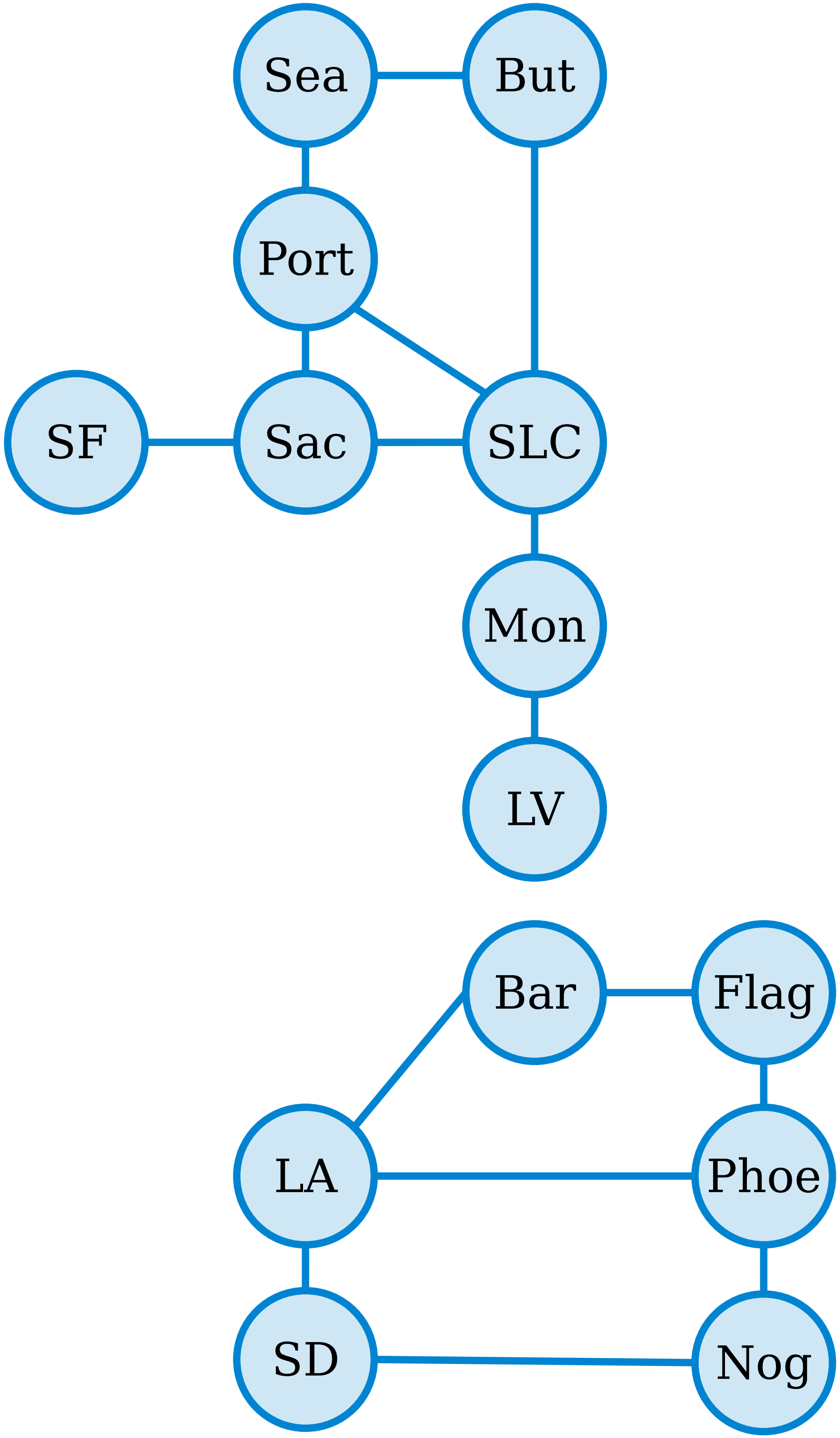
A ***simple cycle*** in a graph is cycle that does not repeat any nodes or edges except the first/last node.



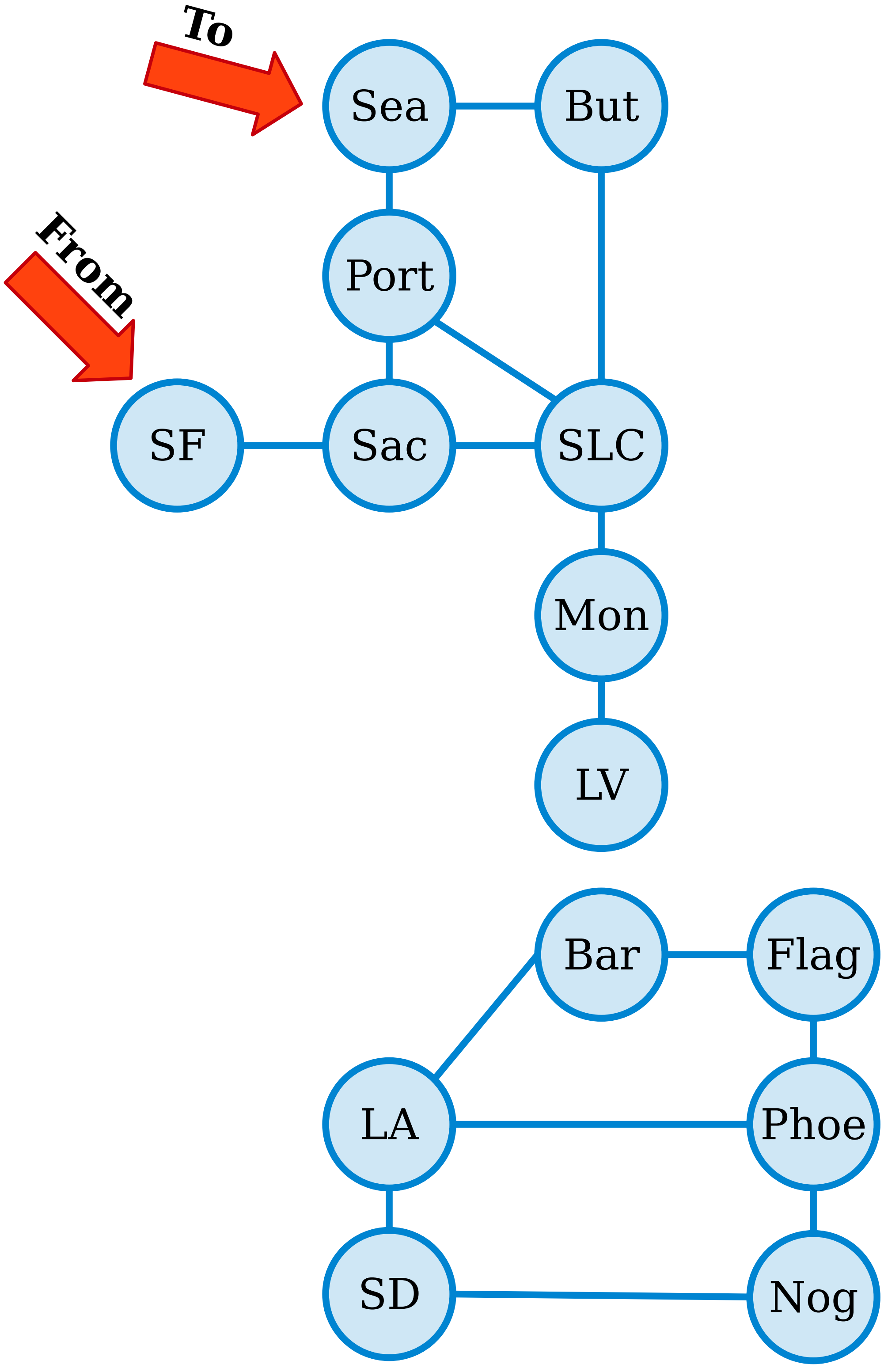
A ***path*** in a graph $G = (V, E)$ is a sequence of one or more nodes $v_1, v_2, v_3, \ldots, v_n$ such that any two consecutive nodes in the sequence are adjacent.



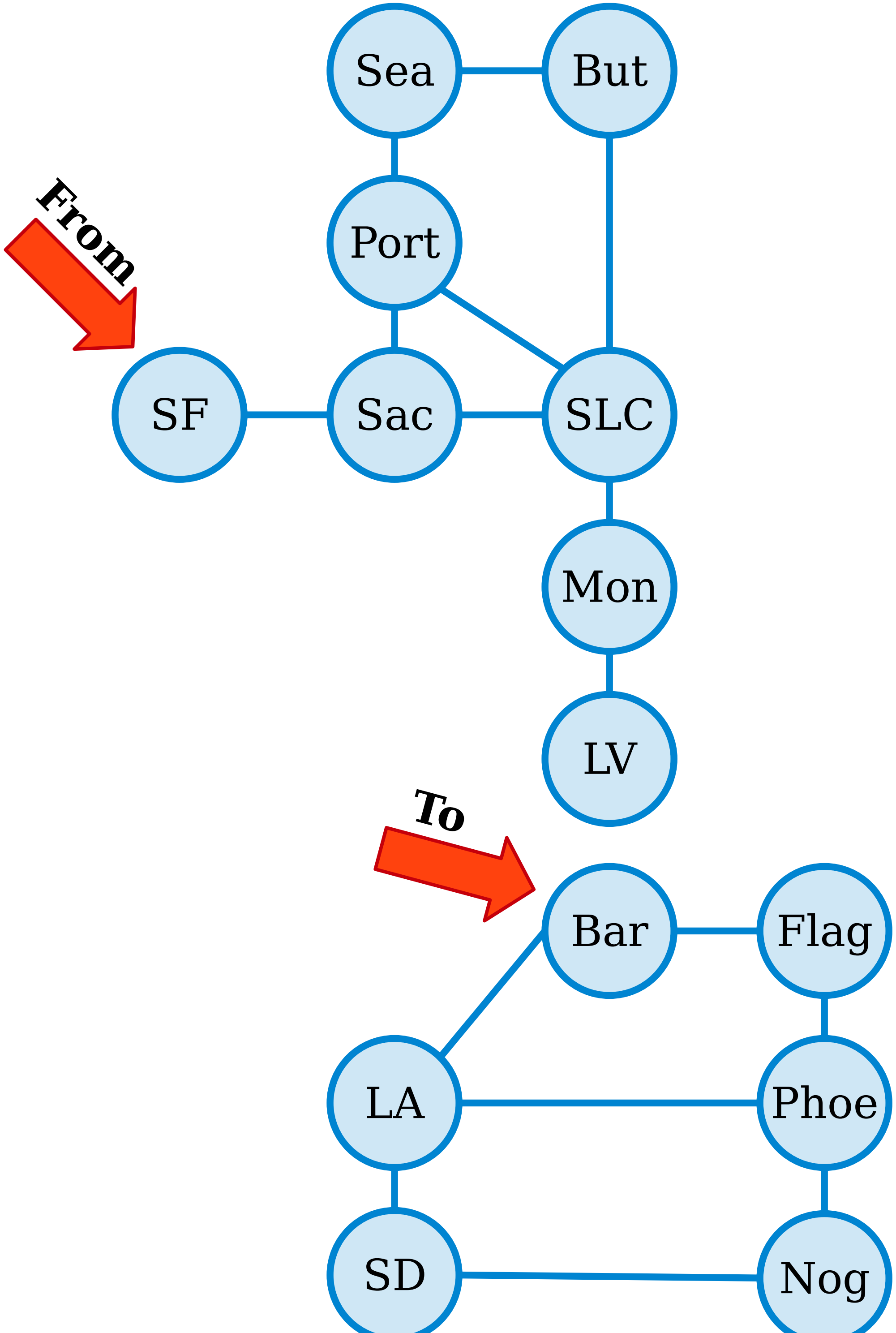
A ***path*** in a graph $G = (V, E)$ is a sequence of one or more nodes $v_1, v_2, v_3, \ldots, v_n$ such that any two consecutive nodes in the sequence are adjacent.



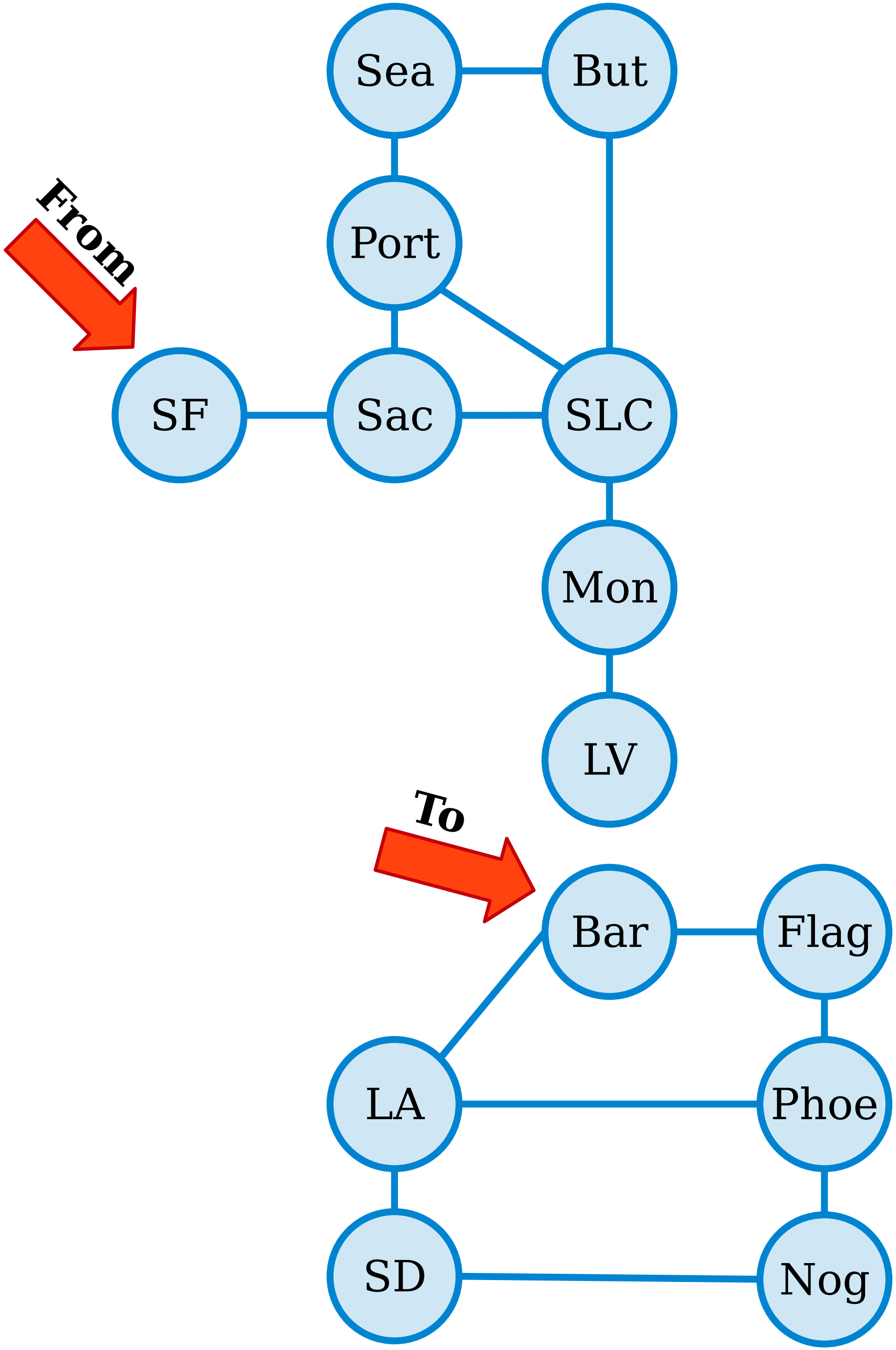
A ***path*** in a graph $G = (V, E)$ is a sequence of one or more nodes $v_1, v_2, v_3, \ldots, v_n$ such that any two consecutive nodes in the sequence are adjacent.



A ***path*** in a graph $G = (V, E)$ 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 $G = (V, E)$ 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.

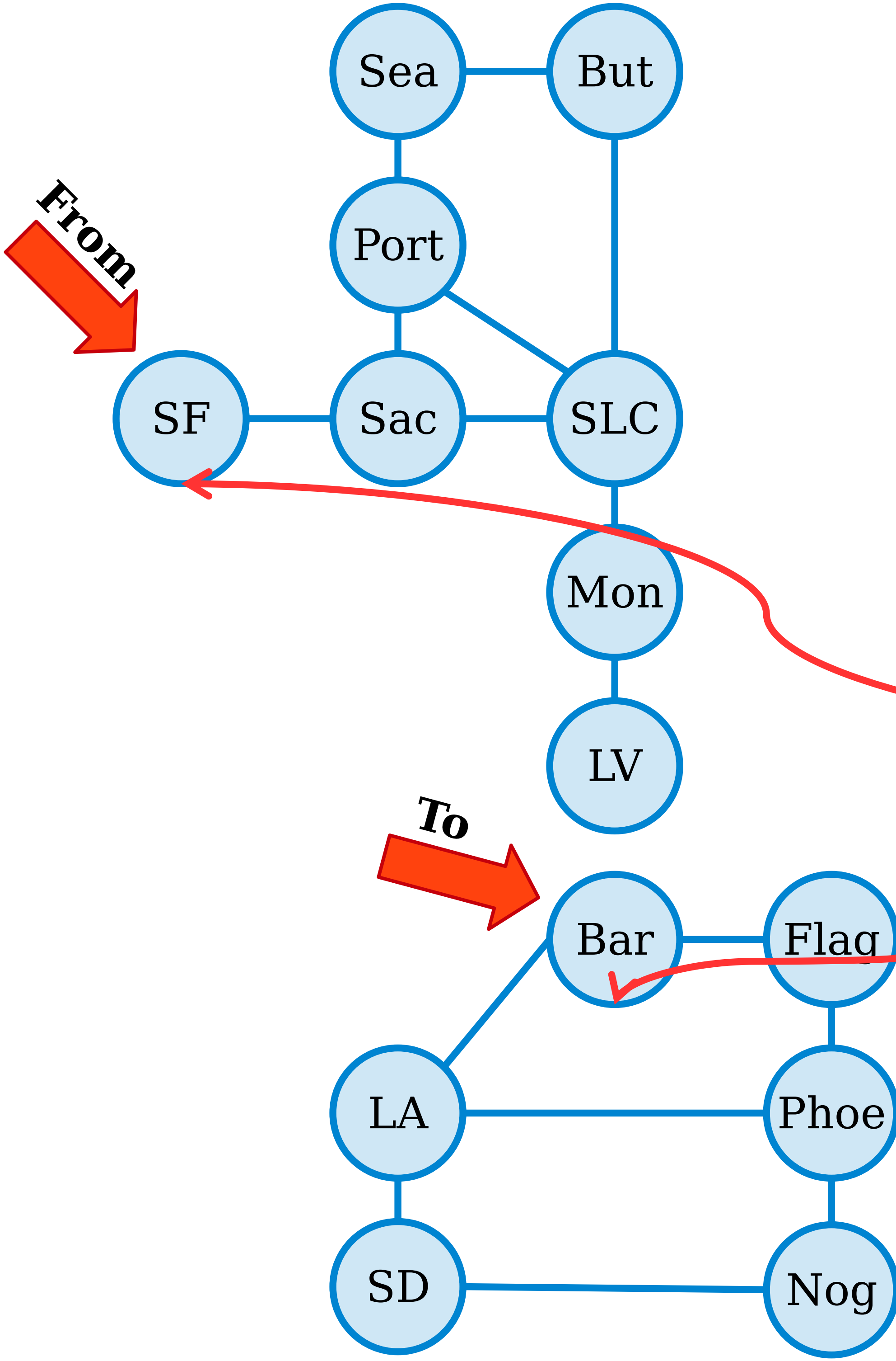


A ***path*** in a graph $G = (V, E)$ is a sequence of one or more nodes $v_1, v_2, v_3, \ldots, v_n$ such that any two consecutive nodes in the sequence are adjacent.

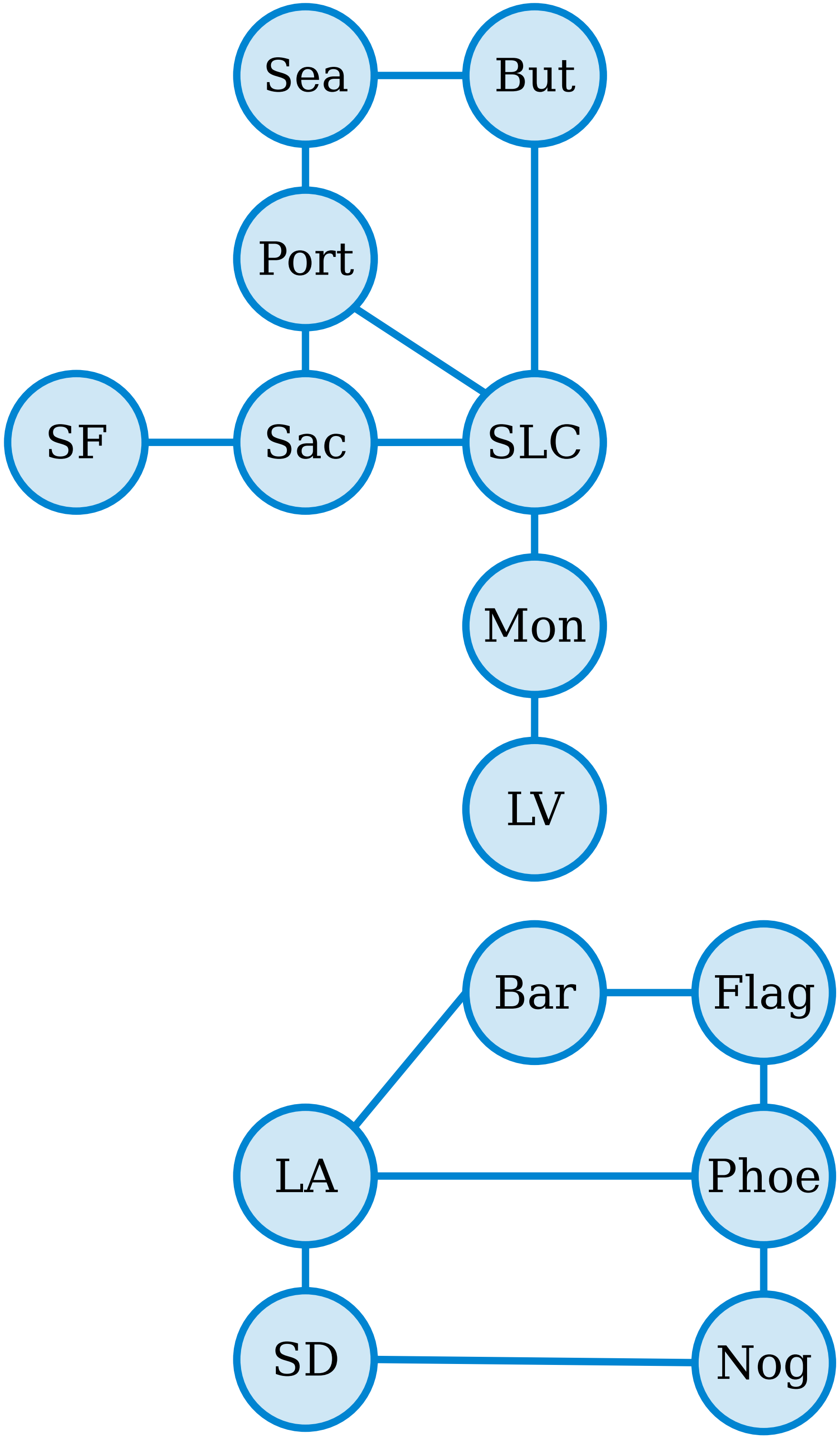
Two nodes in a graph are called ***connected*** if there is a path between them.

A ***path*** 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.

Two nodes in a graph are called ***connected*** if there is a path between them.



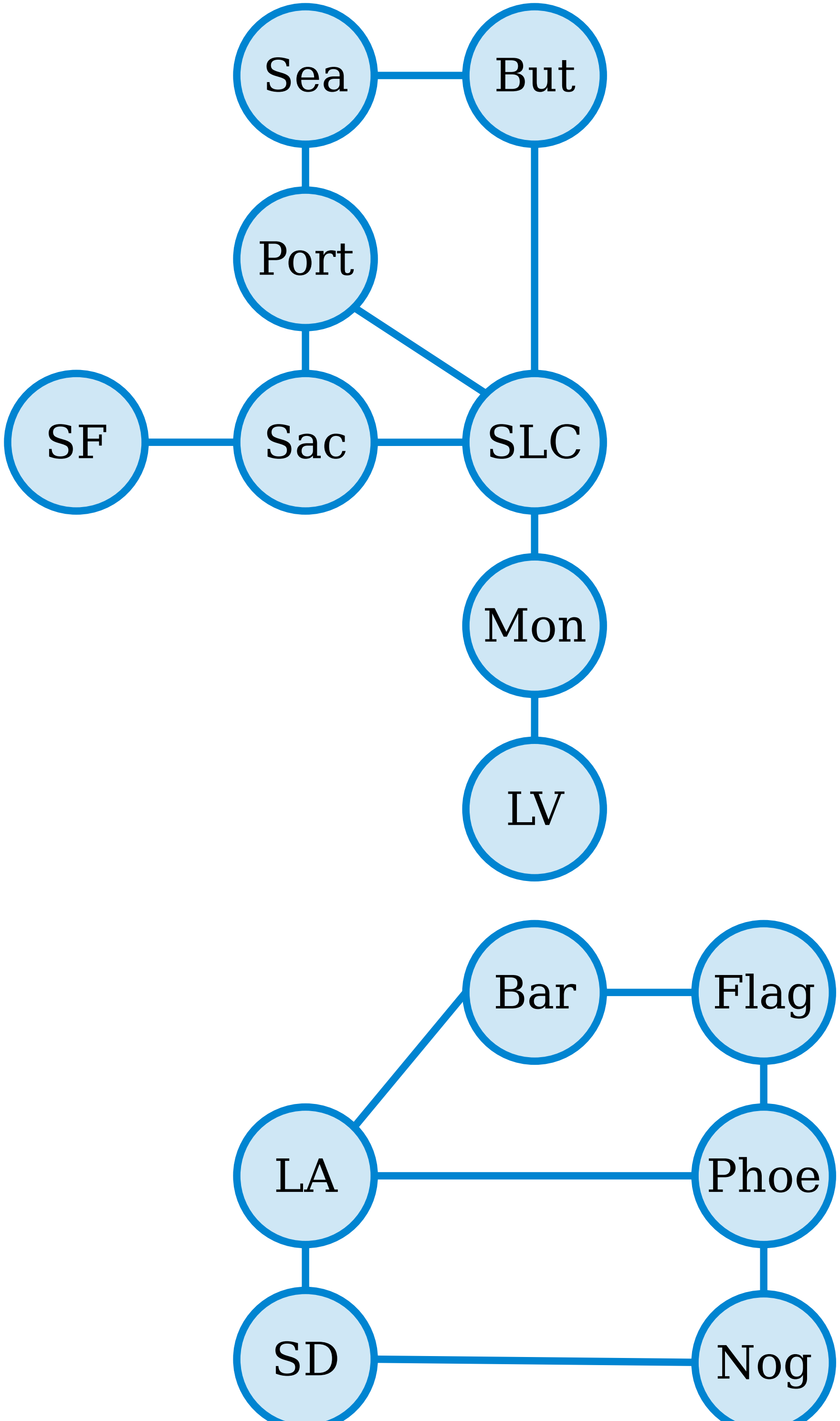
(These nodes are not connected. No Grand Canyon for you.)



A ***path*** 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.

Two nodes in a graph are called ***connected*** if there is a path between them.

A graph G as a whole is called ***connected*** if all pairs of nodes in G are connected.



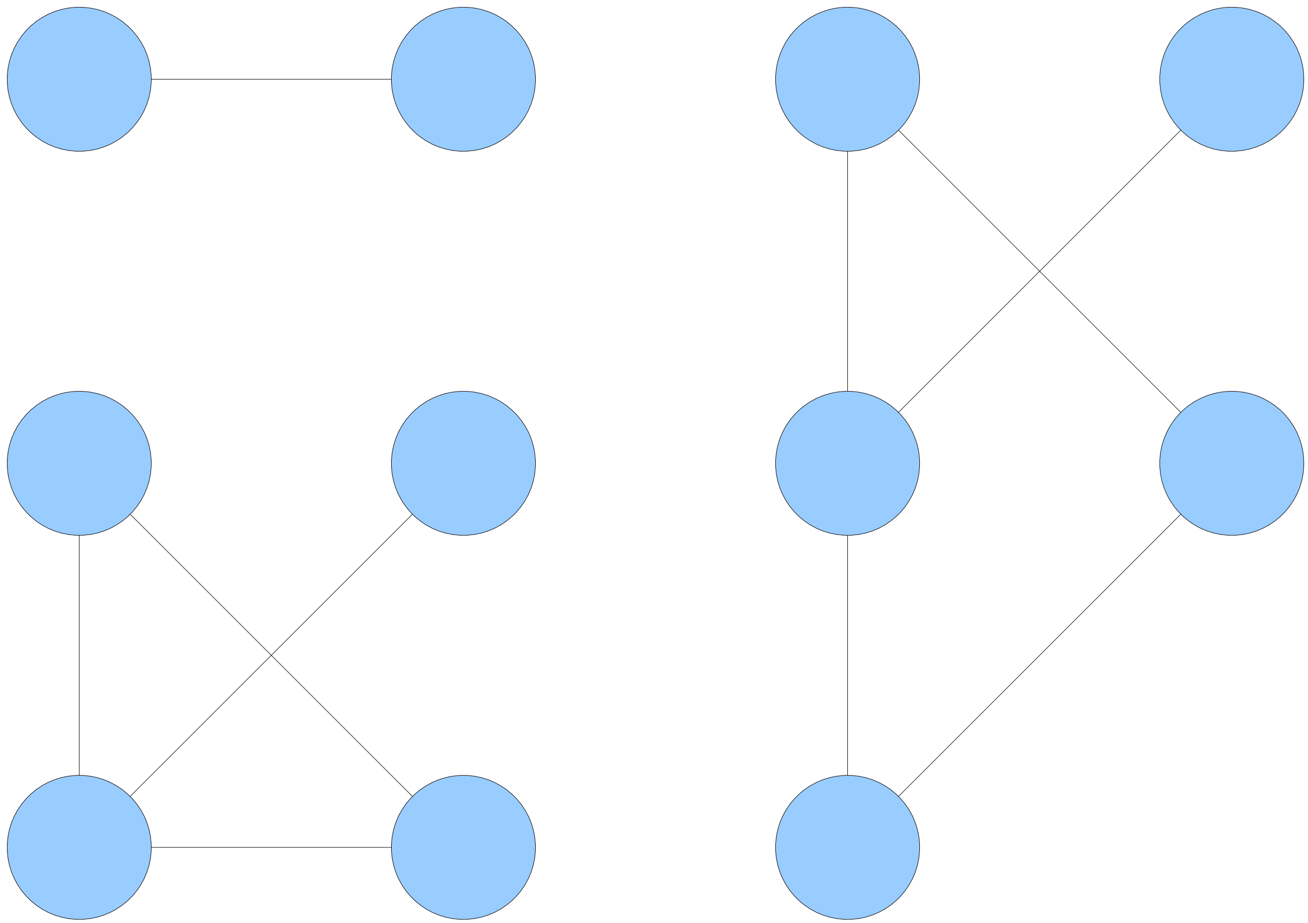
A ***path*** in a graph $G = (V, E)$ is a sequence of one or more nodes $v_1, v_2, v_3, \ldots, v_n$ such that any two consecutive nodes in the sequence are adjacent.

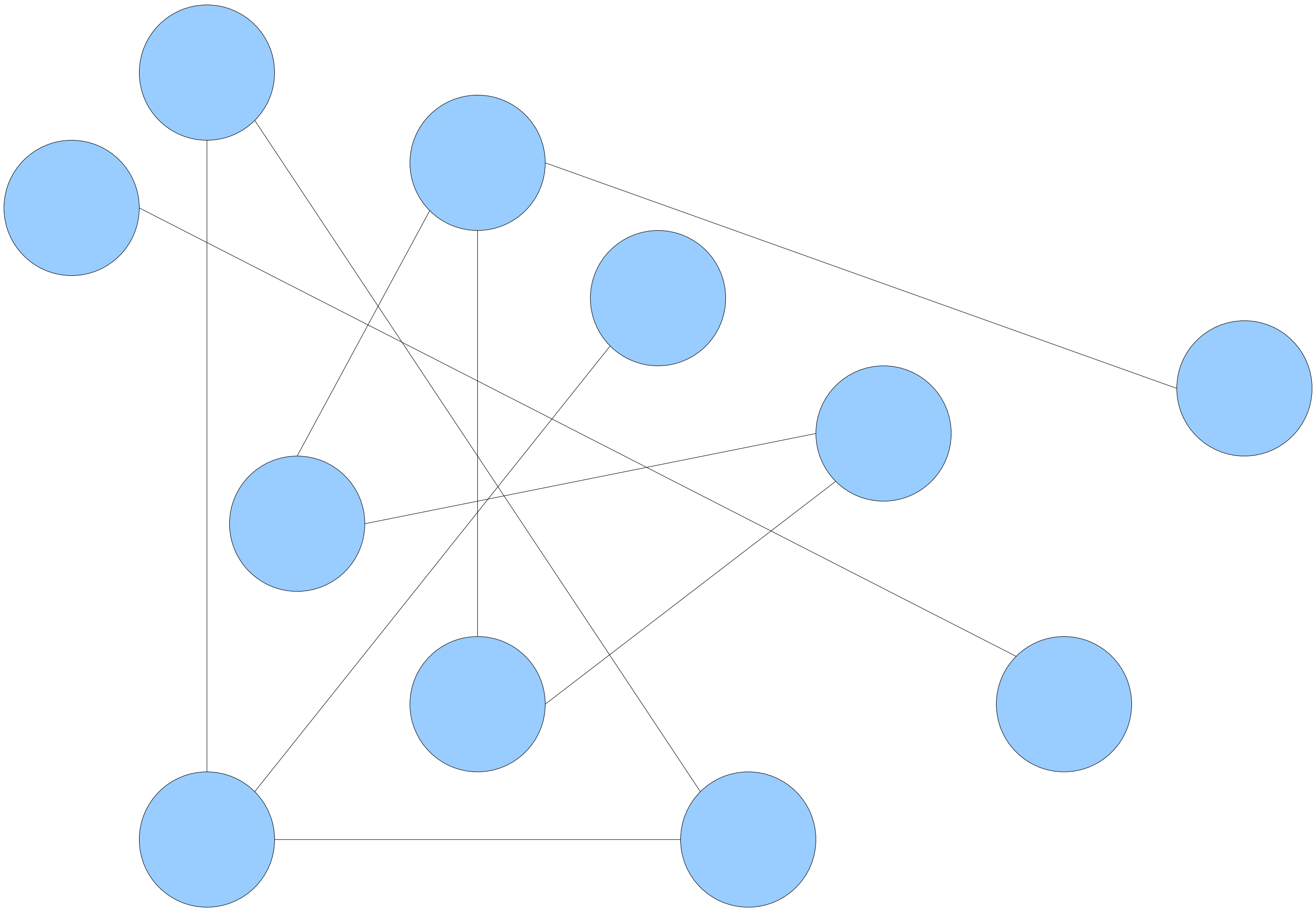
Two nodes in a graph are called ***connected*** if there is a path between them.

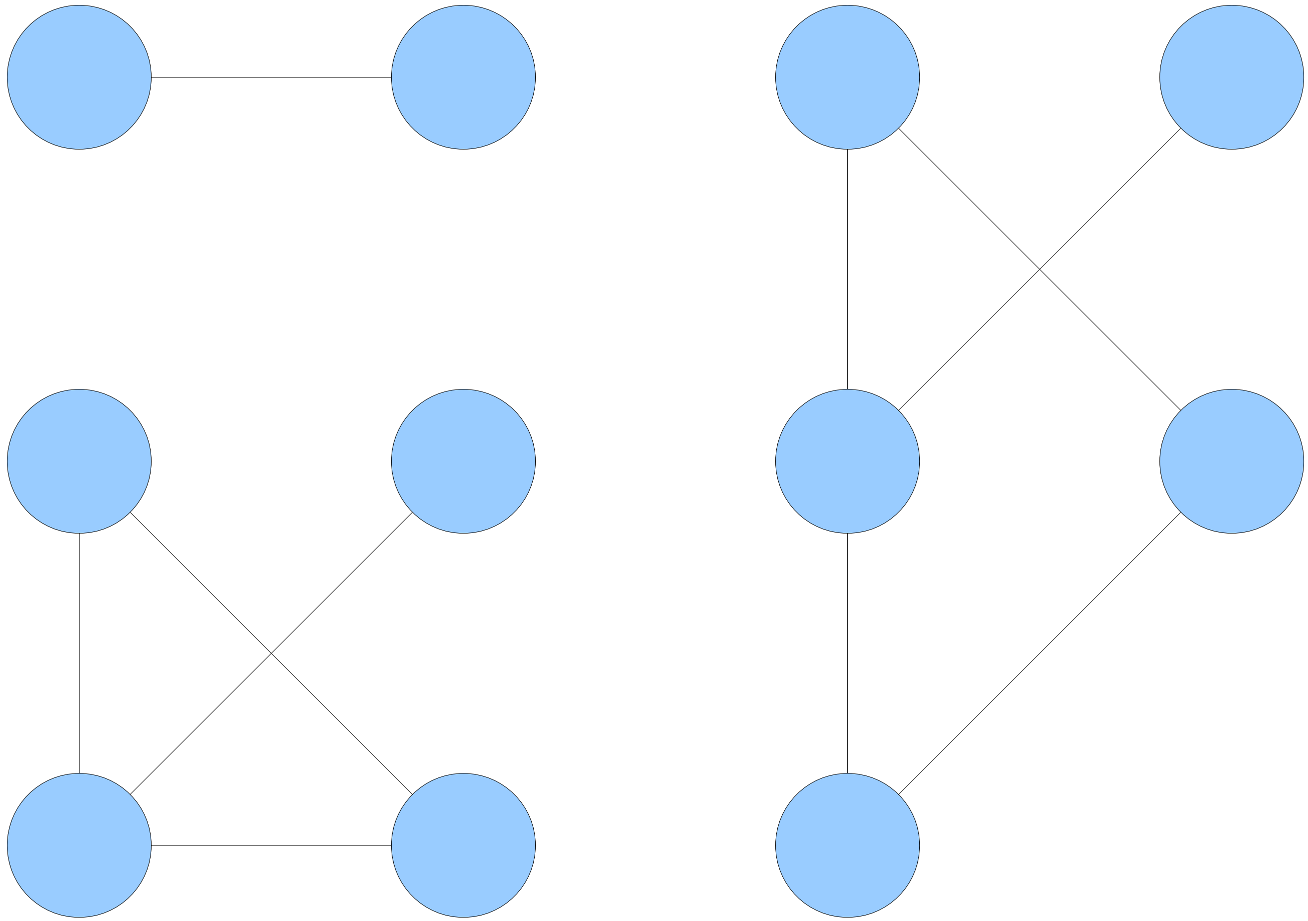
A graph G as a whole is called ***connected*** if all pairs of nodes in G are connected.

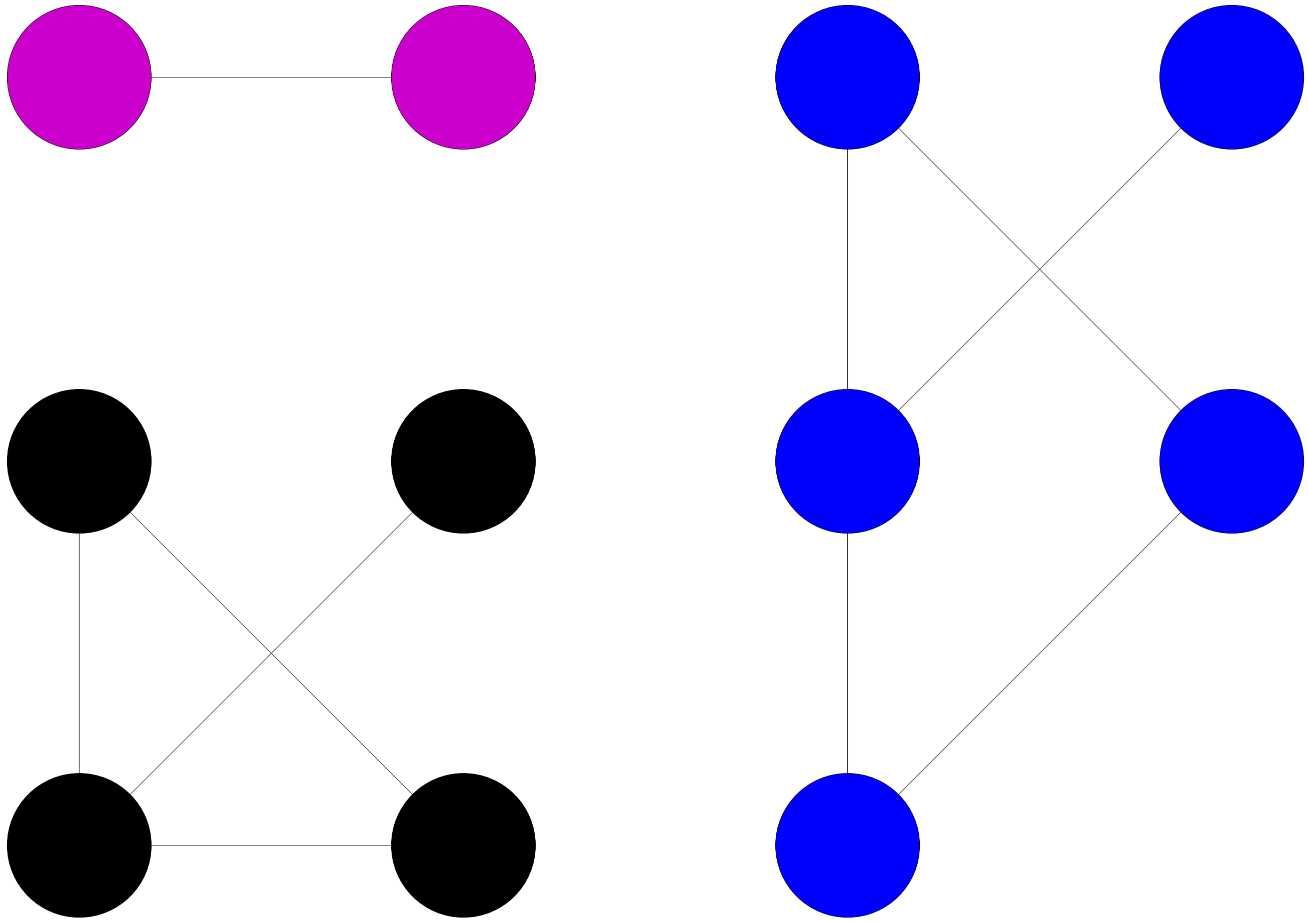
(This graph is not connected.)

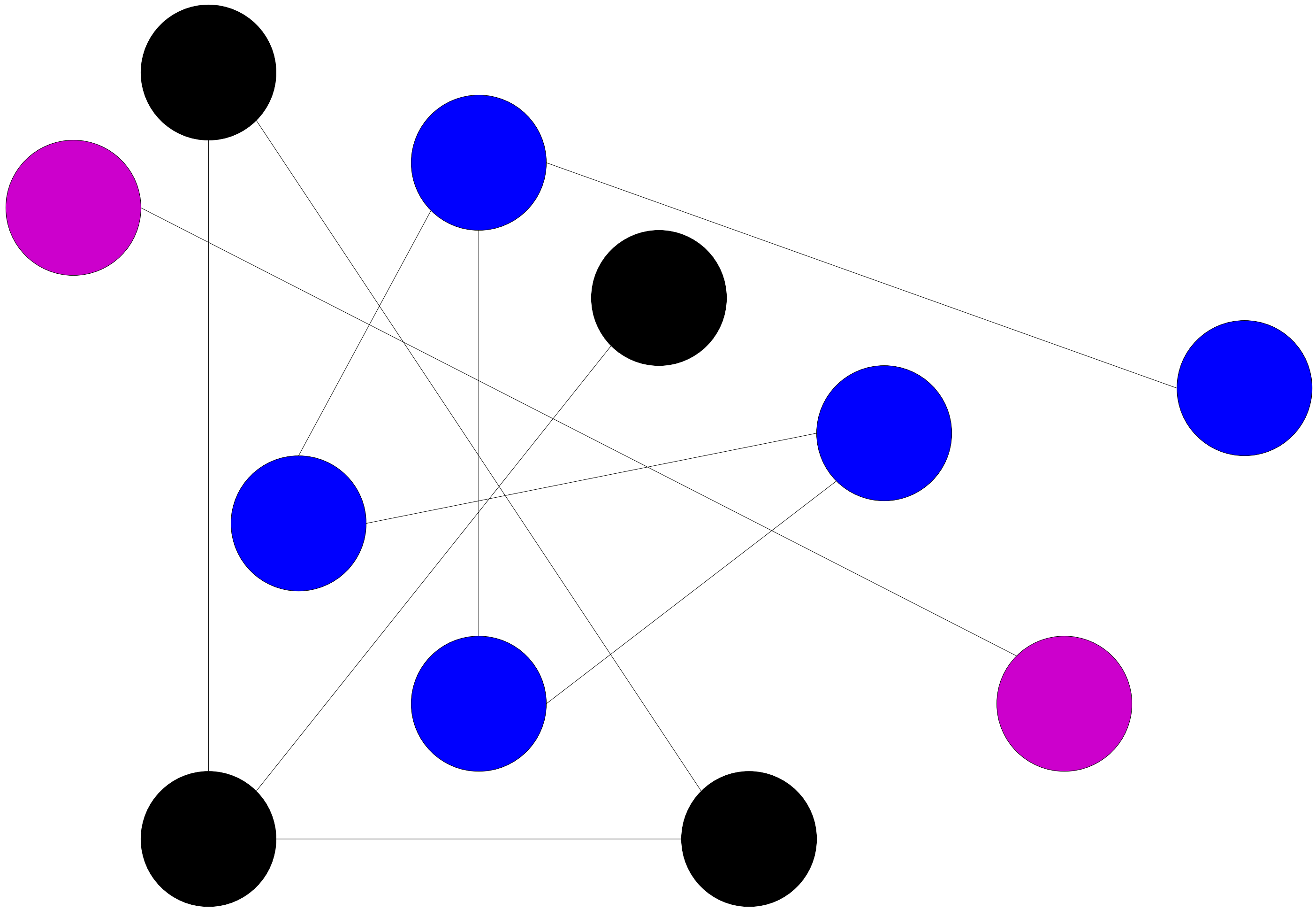
Connected Components

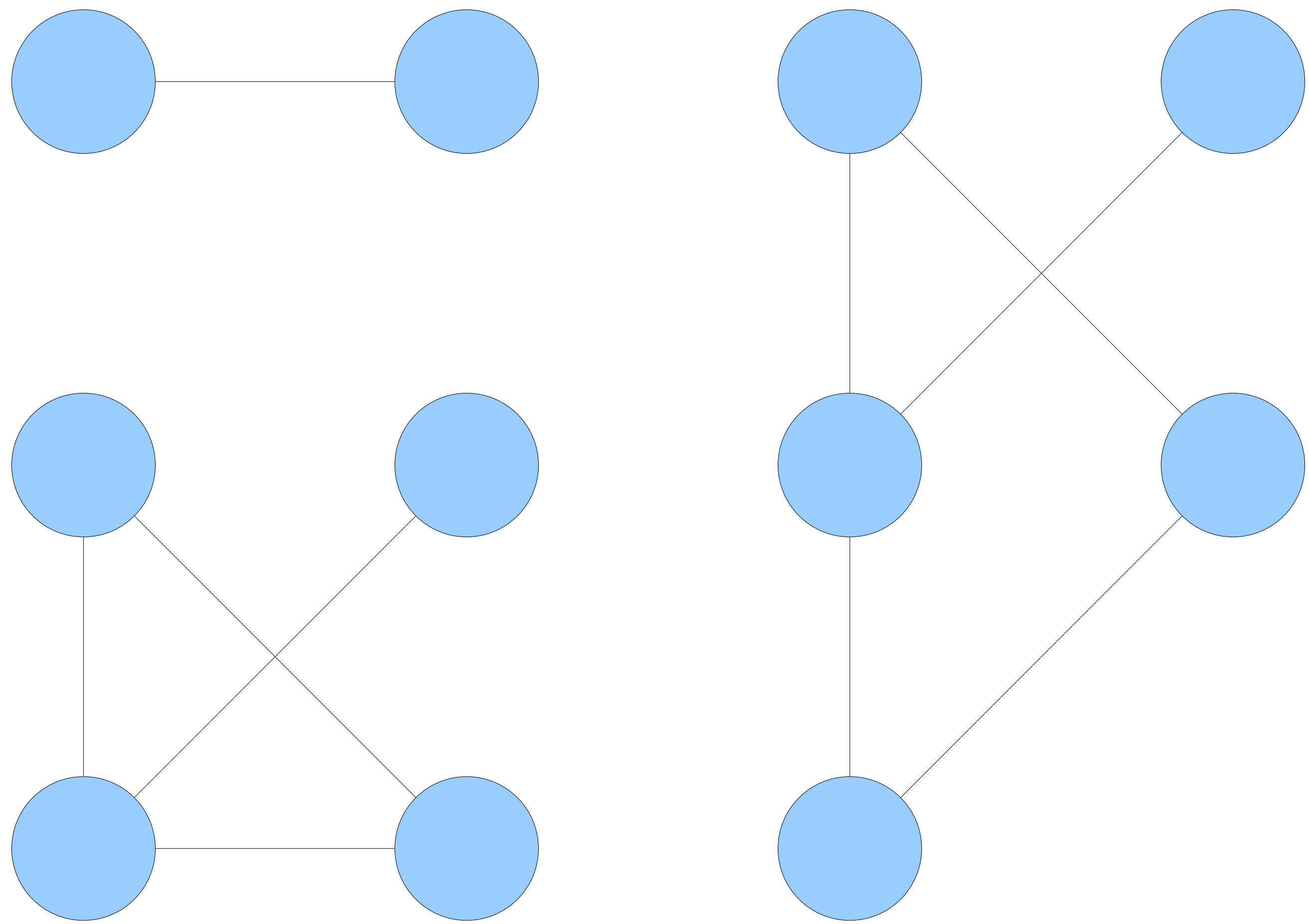


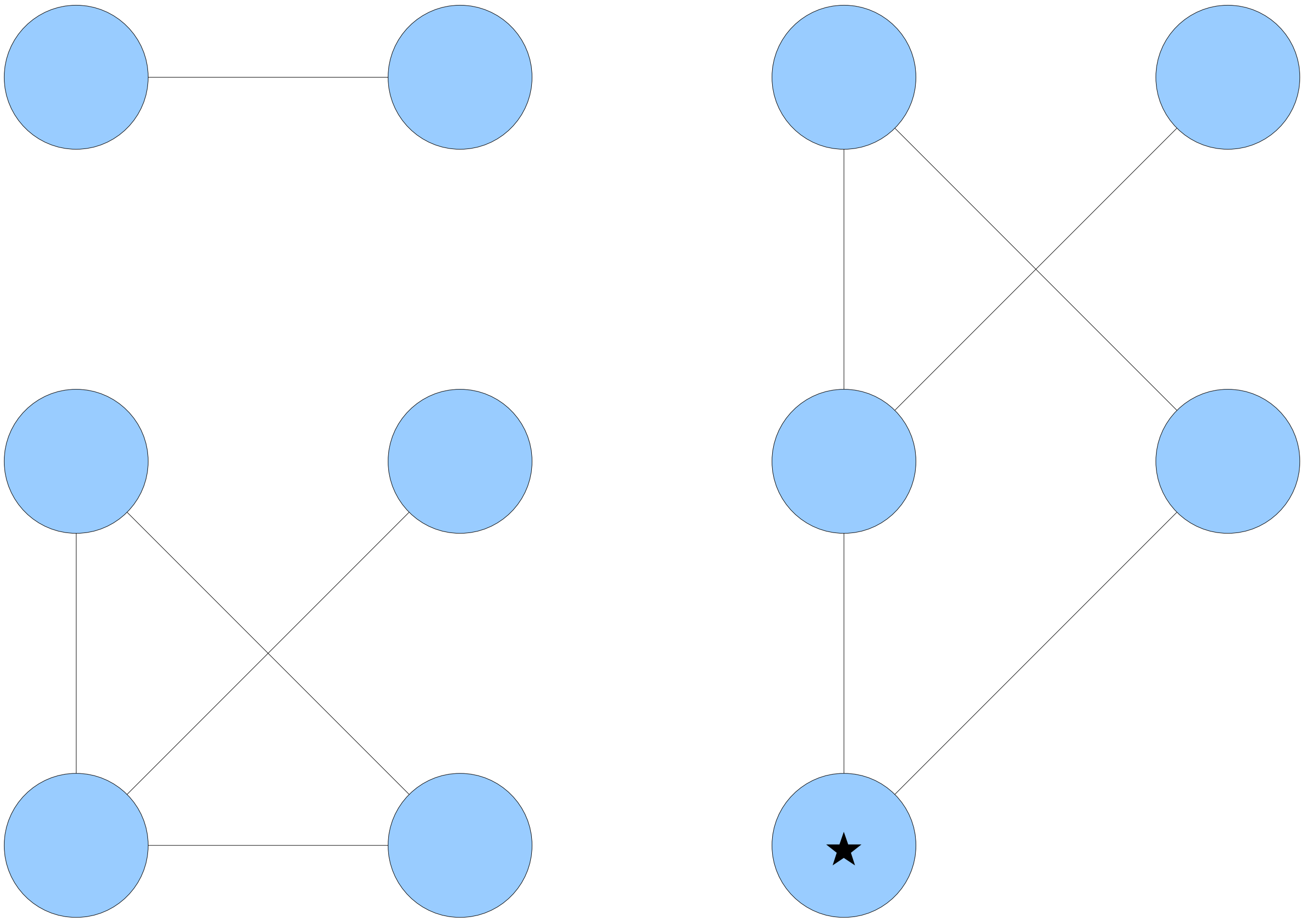


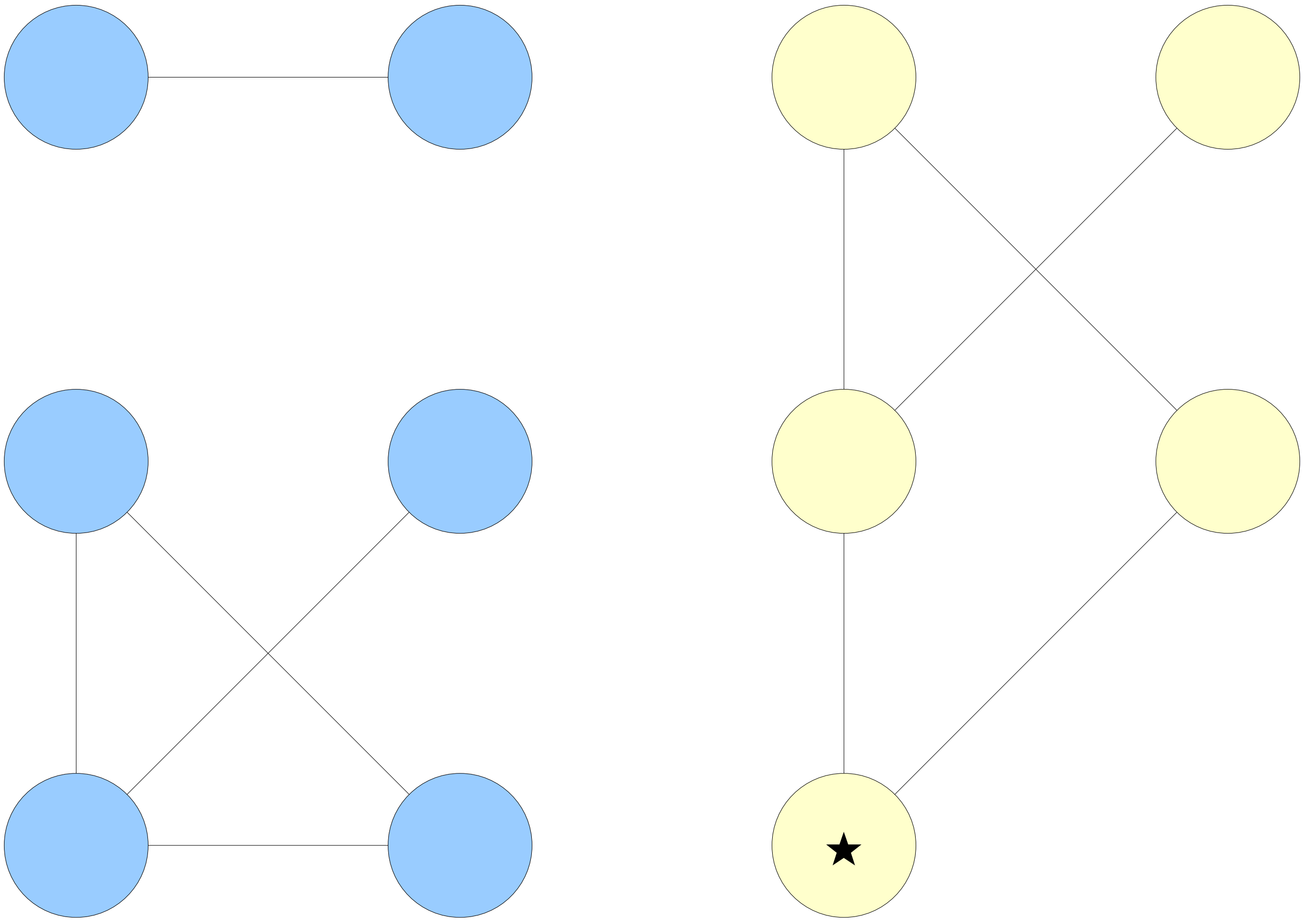












Connected Components

Let $G = (V, E)$ be a graph. For each $v \in V$, the ***connected component*** containing v is the set

$$[v] = \{ x \in V \mid v \text{ is connected to } x \}$$

Intuitively, a connected component is a "piece" of a graph in the sense we just talked about.

Question: How do we know that this particular definition of a "piece" of a graph is a good one?

Goal: Prove that any graph can be broken apart into different connected components.

We’re trying to reason about some way of partitioning the nodes in a graph into different groups.

What structure have we studied that captures the idea of a partition?

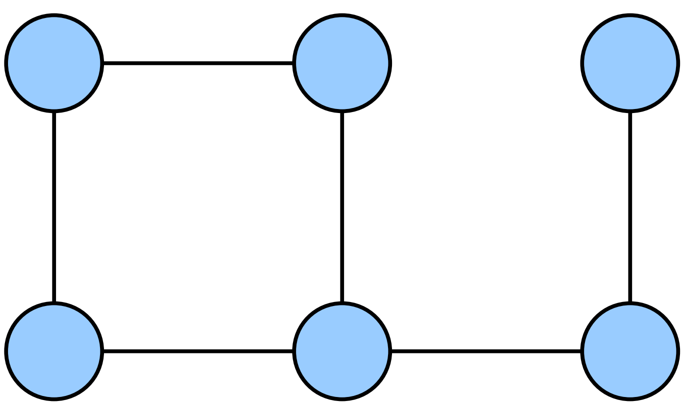
Connectivity

Claim: For any graph G, the “is connected to” relation is an equivalence relation.

Is it reflexive?

Is it symmetric?

Is it transitive?



Connectivity

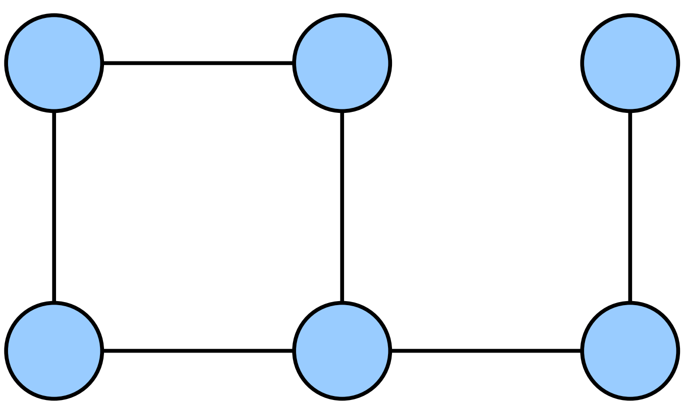
Claim: For any graph G, the "is connected to" relation is an equivalence relation.

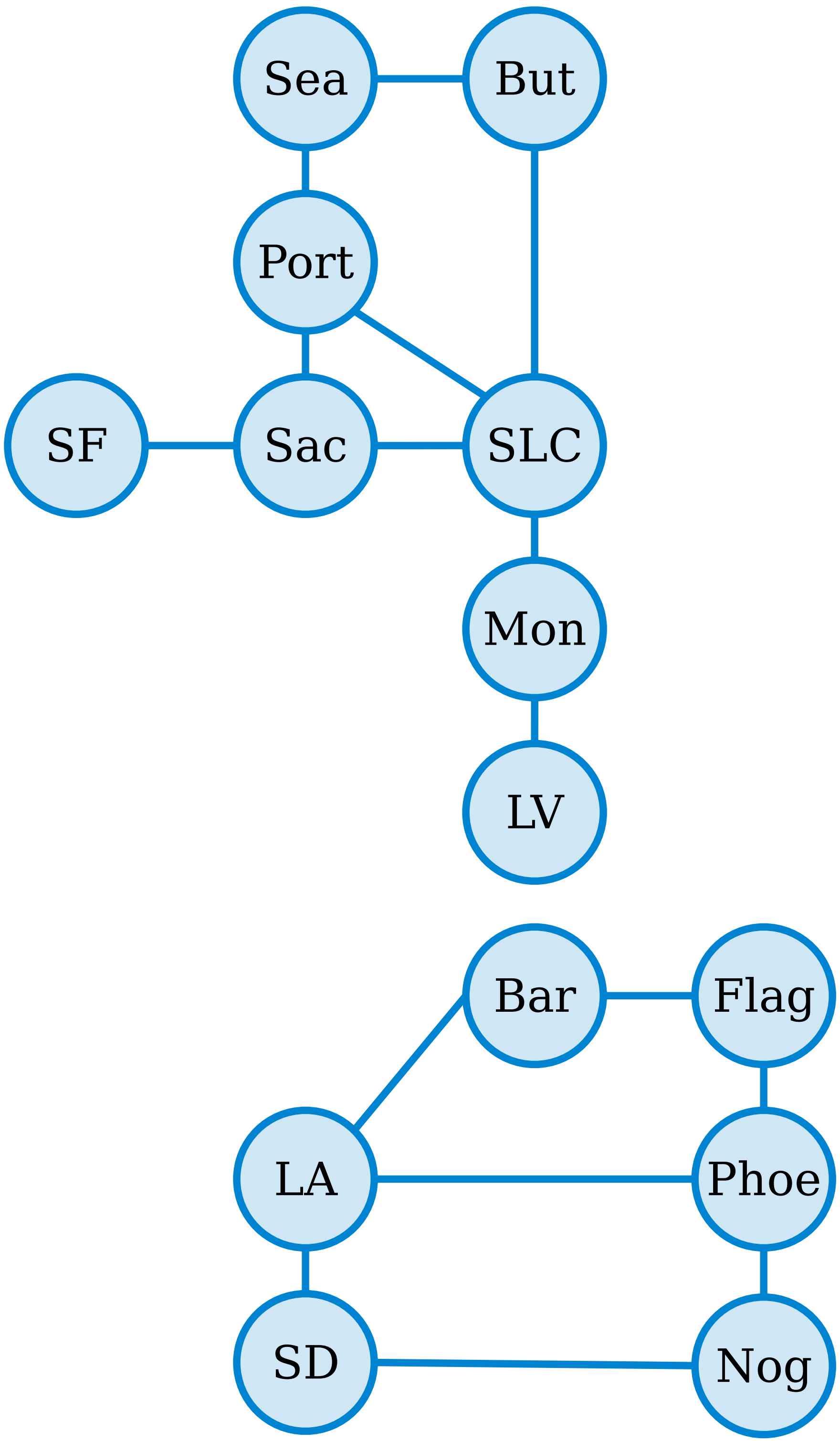
Is it reflexive?

Is it symmetric?

Is it transitive?

$$\forall v \in V.\ Conn(v, v)$$





A ***path*** 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.

Two nodes in a graph are called ***connected*** if there is a path between them

Connectivity

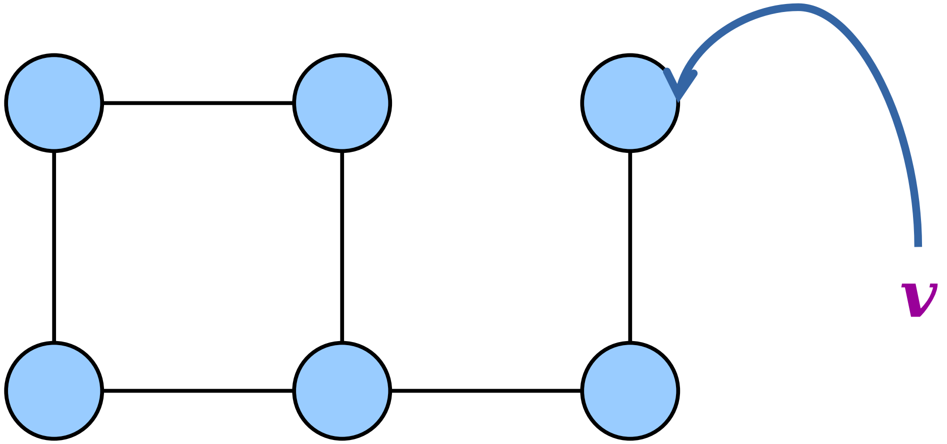
Claim: For any graph *G*, the "is connected to" relation is an equivalence relation.

Is it reflexive?

Is it symmetric?

Is it transitive?

$\forall v \in V.\ Conn(v, v)$



A ***path*** 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.

Connectivity

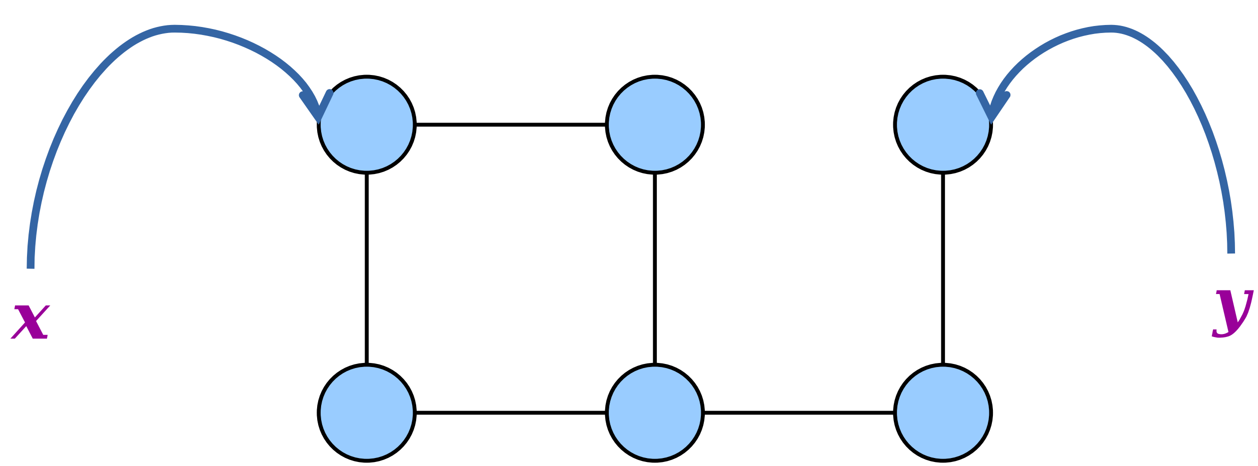
Claim: For any graph G, the “is connected to” relation is an equivalence relation.

$$\forall x \in V.\ \forall y \in V.\ (Conn(x, y) \rightarrow Conn(y, x))$$

Is it reflexive?

Is it symmetric?

Is it transitive?



Connectivity

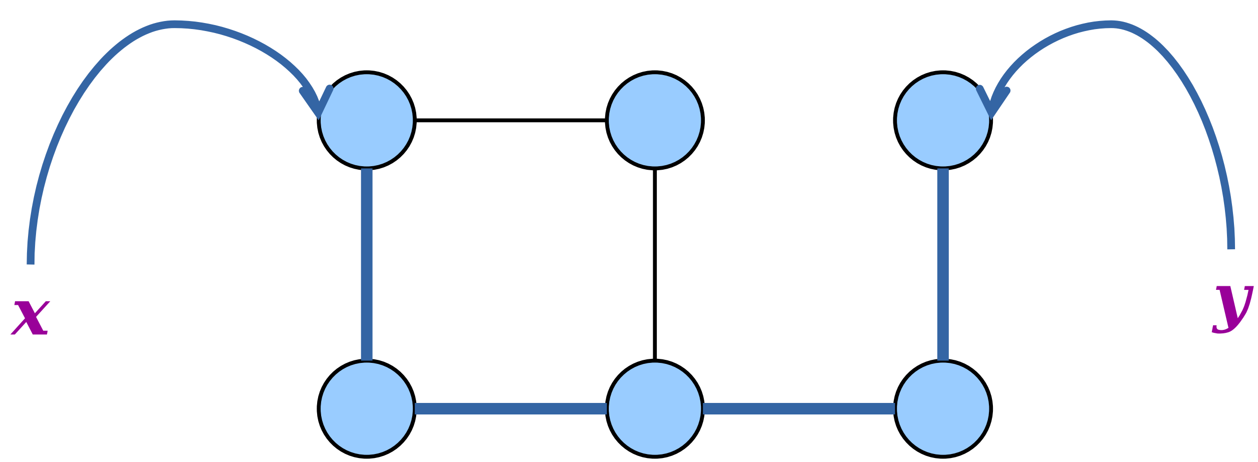
Claim: For any graph G, the "is connected to" relation is an equivalence relation.

$$\forall x \in V.\ \forall y \in V.\ (Conn(x, y) \to Conn(y, x))$$

Is it reflexive?

Is it symmetric?

Is it transitive?



Connectivity

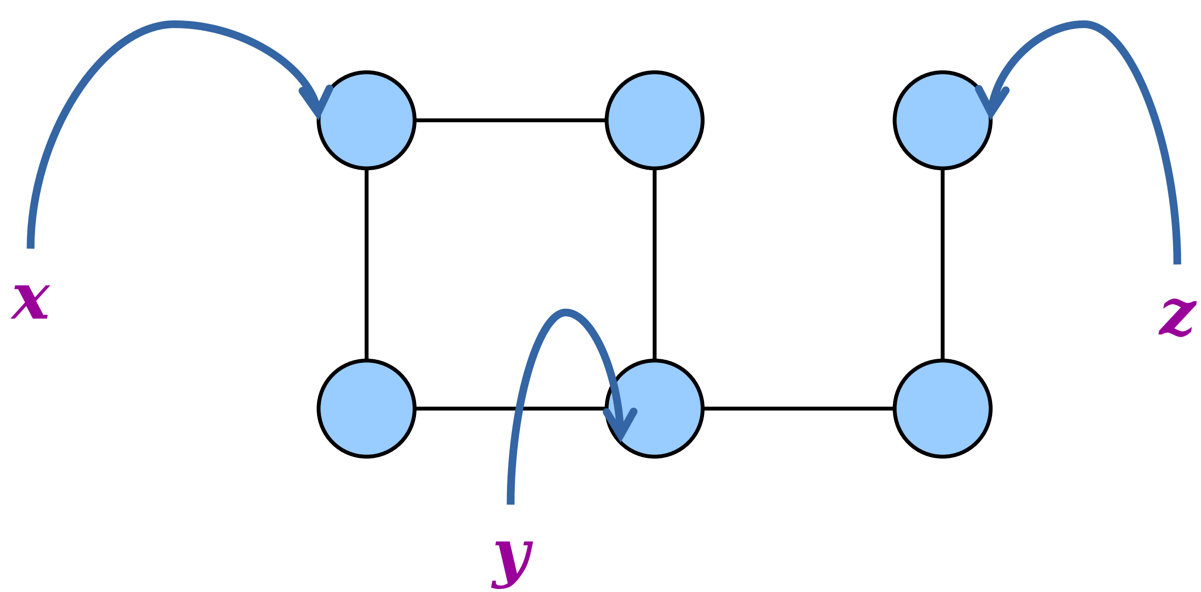
Claim: For any graph *G*, the "is connected

$$\forall x \in V.\ \forall y \in V.\ \forall z \in V.\ (Conn(x, y) \land Conn(y, z) \rightarrow Conn(x, z))$$

Is it reflexive?

Is it symmetric?

Is it transitive?



Connectivity

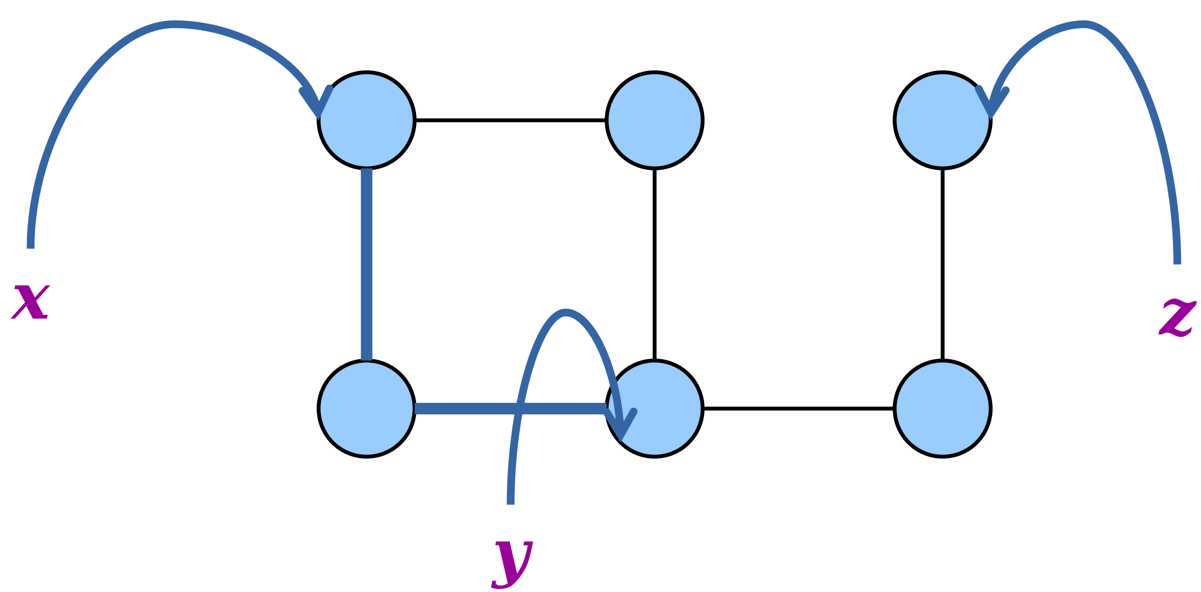
Claim: For any graph G, the “is connected

$$\forall x \in V.\ \forall y \in V.\ \forall z \in V.\ (Conn(x, y) \land Conn(y, z) \to Conn(x, z))$$

Is it reflexive?

Is it symmetric?

Is it transitive?



Theorem: Let $G = (V, E)$ be a graph. Then the connectivity relation over V is an equivalence relation.

Proof: Consider an arbitrary graph $G = (V, E)$. We will prove that the connectivity relation over V is reflexive, symmetric, and transitive.

To show that connectivity is reflexive, consider any $v \in V$. Then the singleton path v is a path from v to itself. Therefore, v is connected to itself, as required.

To show that connectivity is symmetric, consider any $x, y \in V$ where x is connected to y. We need to show that y is connected to x. Since x is connected to y, there is some path $x, v_1, \ldots, v_n, y$ from x to y. Then $y, v_n, \ldots, v_1, x$ is a path from y back to x, so y is connected to x.

Finally, to show that connectivity is transitive, let $x, y, z \in V$ be arbitrary nodes where x is connected to y and y is connected to z. We will prove that x is connected to z. Since x is connected to y, there is a path $x, u_1, \ldots, u_n, y$ from x to y. Since y is connected to z, there is a path $y, v_1, \ldots, v_k, z$ from y to z. Then the path $x, u_1, \ldots, u_n, y, v_1, \ldots, v_k, z$ goes from x to z. Thus x is connected to z, as required. ■

Putting Things Together

Earlier, we defined the connected component of a node v to be

$$[v] = \{\ x \in V \mid v \text{ is connected to } x\ \}$$

Connectivity is an equivalence relation! So what's the equivalence class of a node v with respect to connectivity?

$$[v]_{conn} = \{\ x \in V \mid v \text{ is connected to } x\ \}$$

Connected components are equivalence classes of the connectivity relation!

Theorem: If $G = (V, E)$ is a graph, then every node in G belongs to exactly one connected component of G.

Proof: Let $G = (V, E)$ be an arbitrary graph and let $v \in V$ be any node in G. The connected components of G are just the equivalence classes of the connectivity relation in G. The Fundamental Theorem of Equivalence Relations guarantees that v belongs to exactly one equivalence class of the connectivity relation. Therefore, v belongs to exactly one connected component in G. ■

Time out for announcements!

Problem Set 3

- Due tomorrow (Thursday) at 11:59pm PDT.
- Use a late period to extend this to Saturday at 11:59pm PDT.
- Any last questions? Come to office hours or ask on Campuswire.

Midterm

- Thursday July 23rd

- Will cover material up to Monday's lecture (psets 1, 2, 3).

- 24-hour window to start the exam. Begins at 9:30AM PDT on Thursday July 23rd.

- Once you click start on Gradscope, Gradescope will give you access to the exam. You'll have 3 hours to complete the exam, plus 15 minutes to upload your exam to Gradescope.

- Please make sure any OAE letters get sent to the staff mailing list as soon as possible.