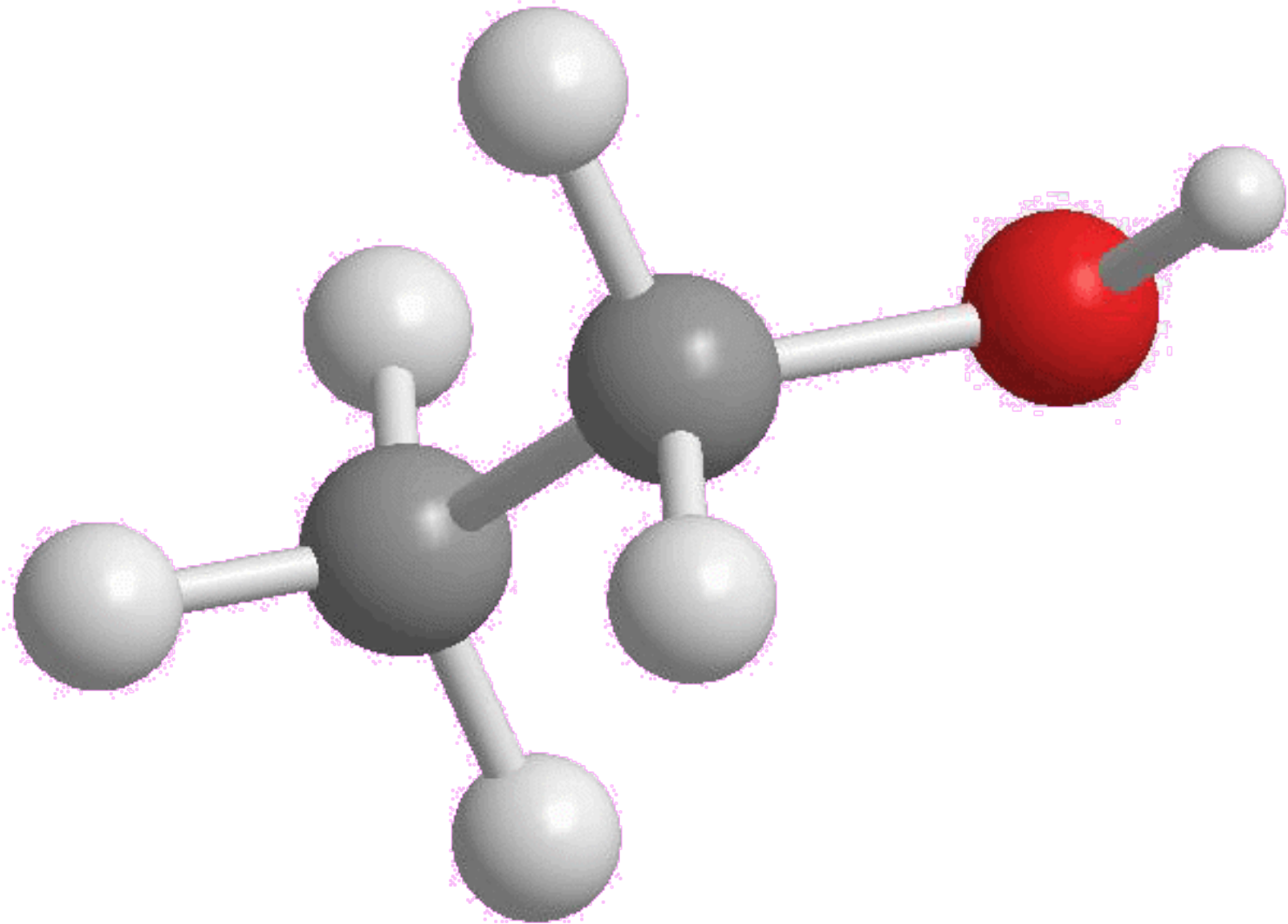
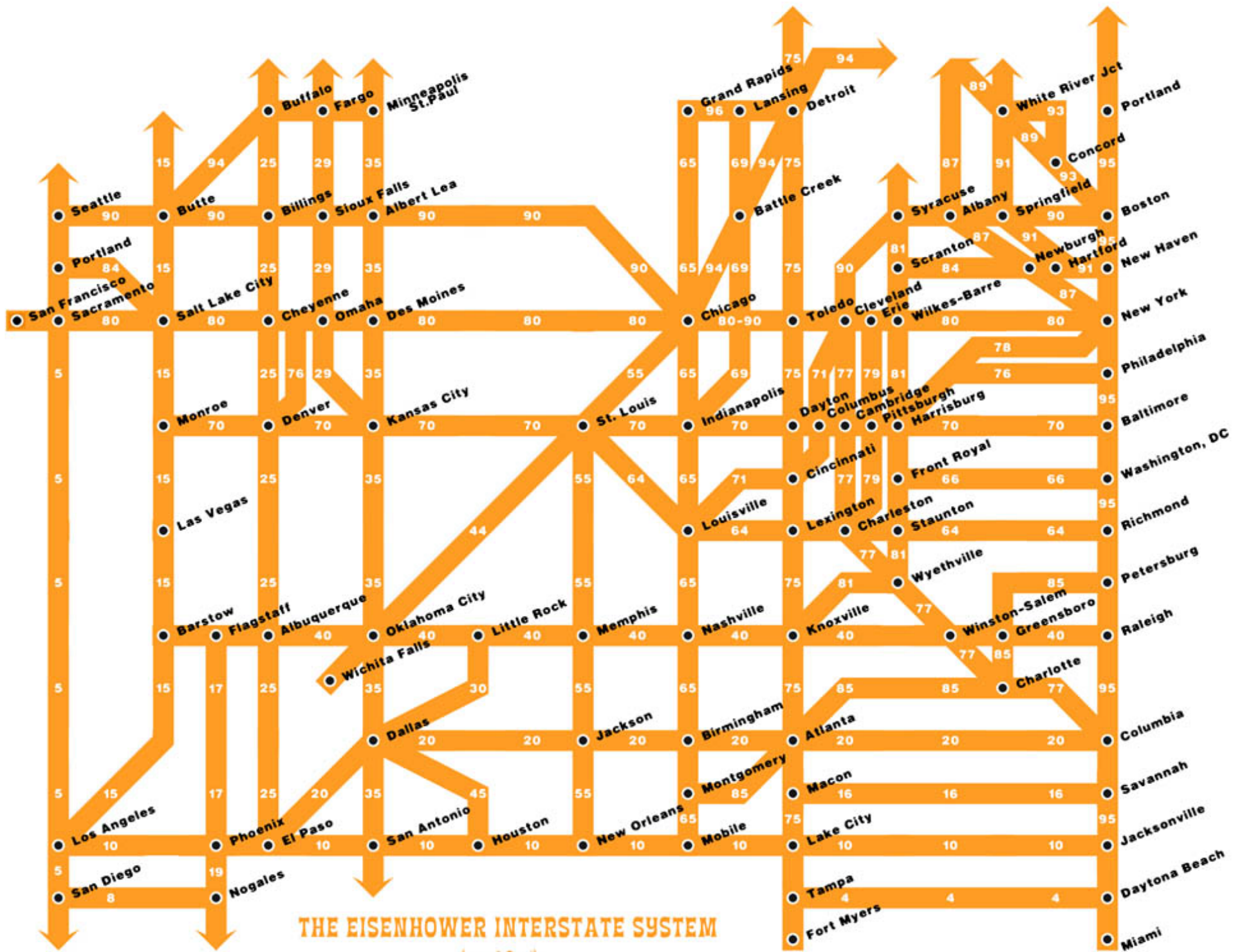


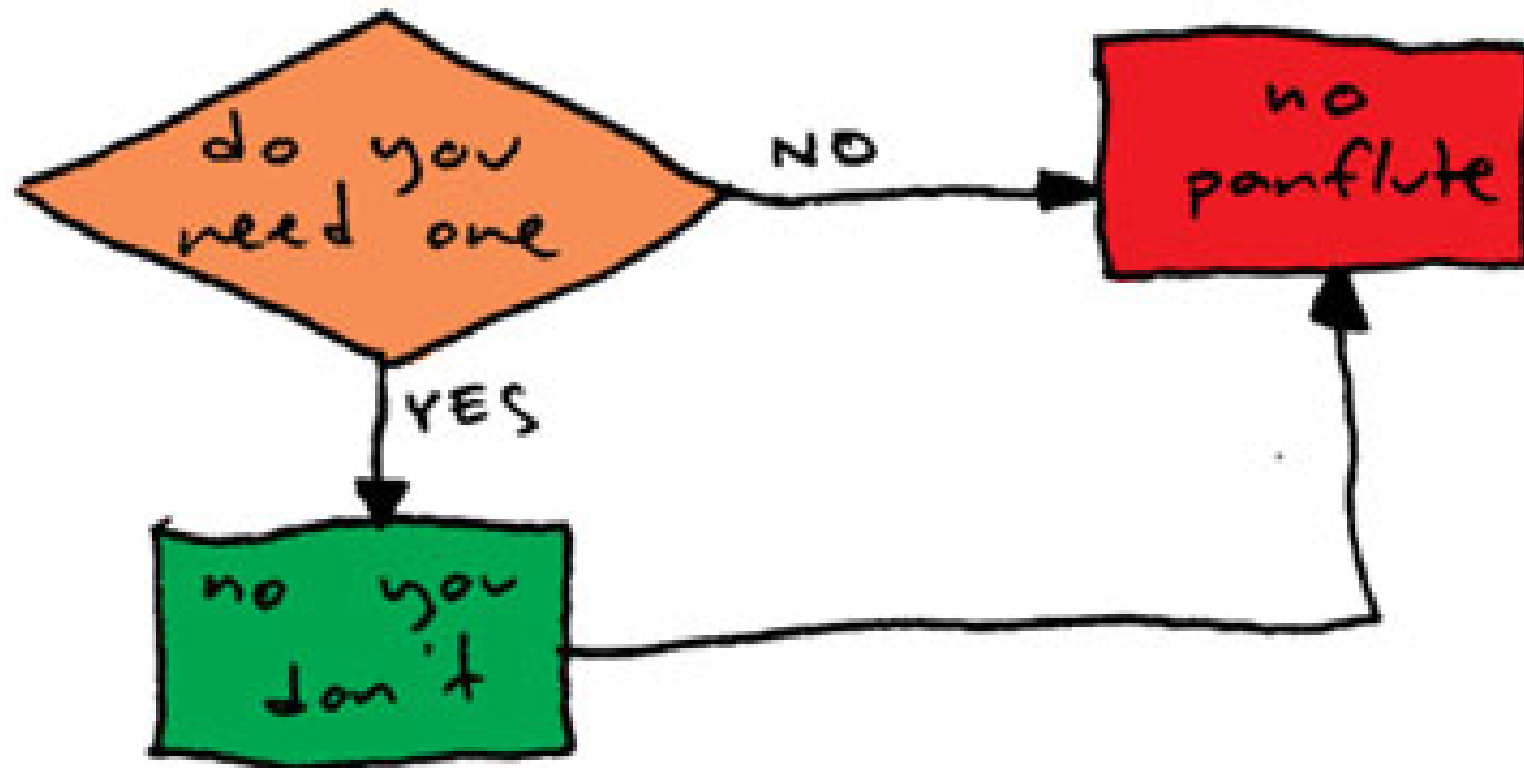
# Graphs

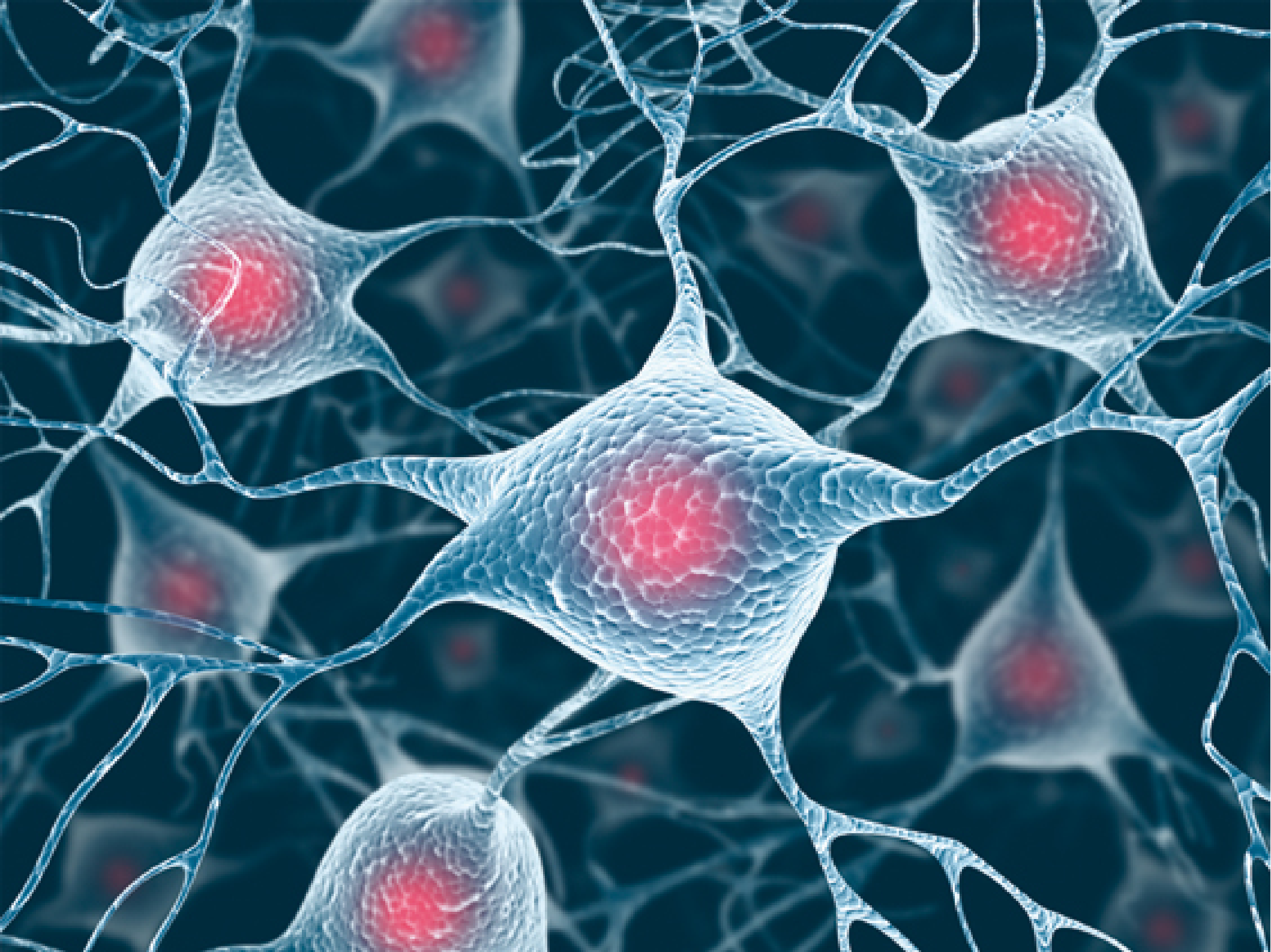
# Chemical Bonds





# PANFLUTE FLOWCHART





**facebook®**

facebook®

Me too!

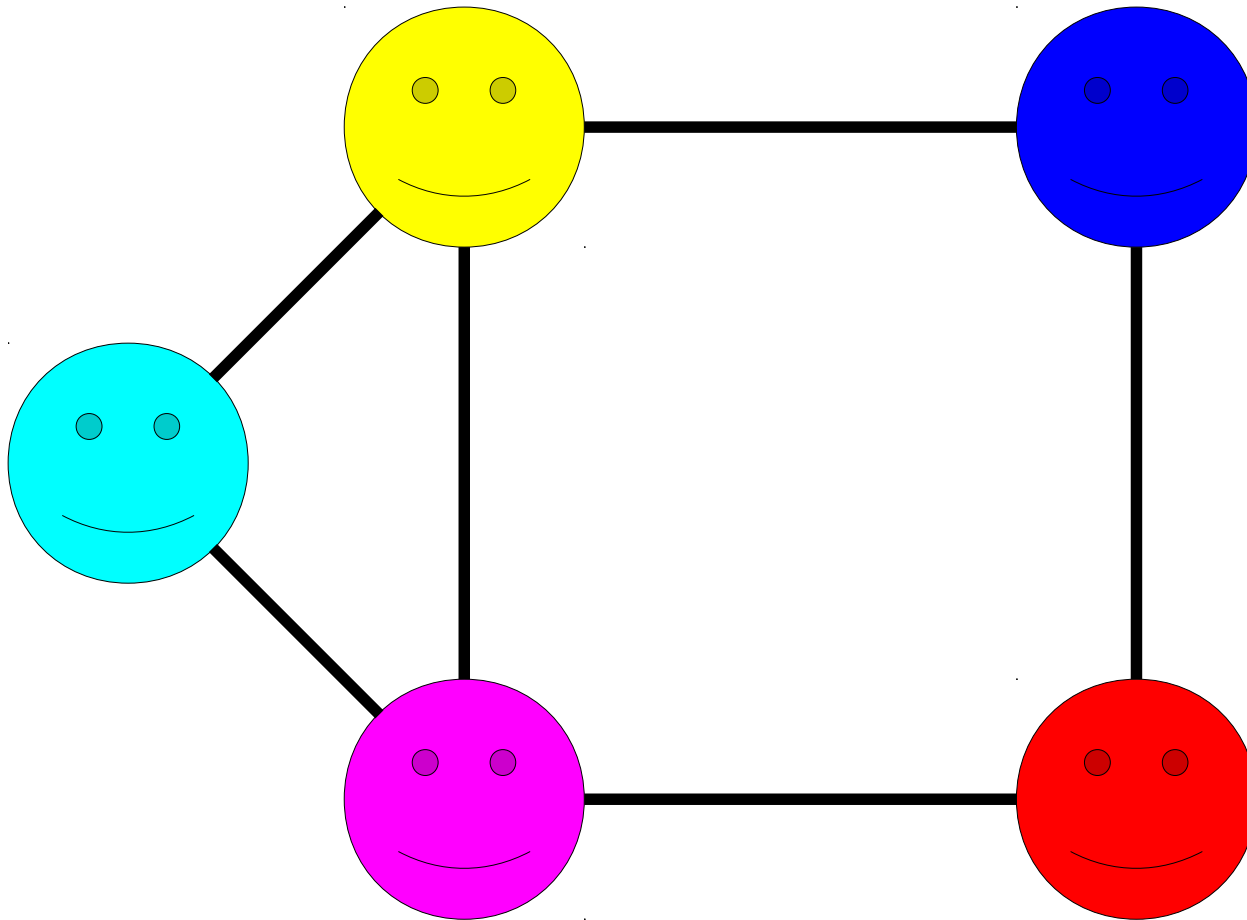




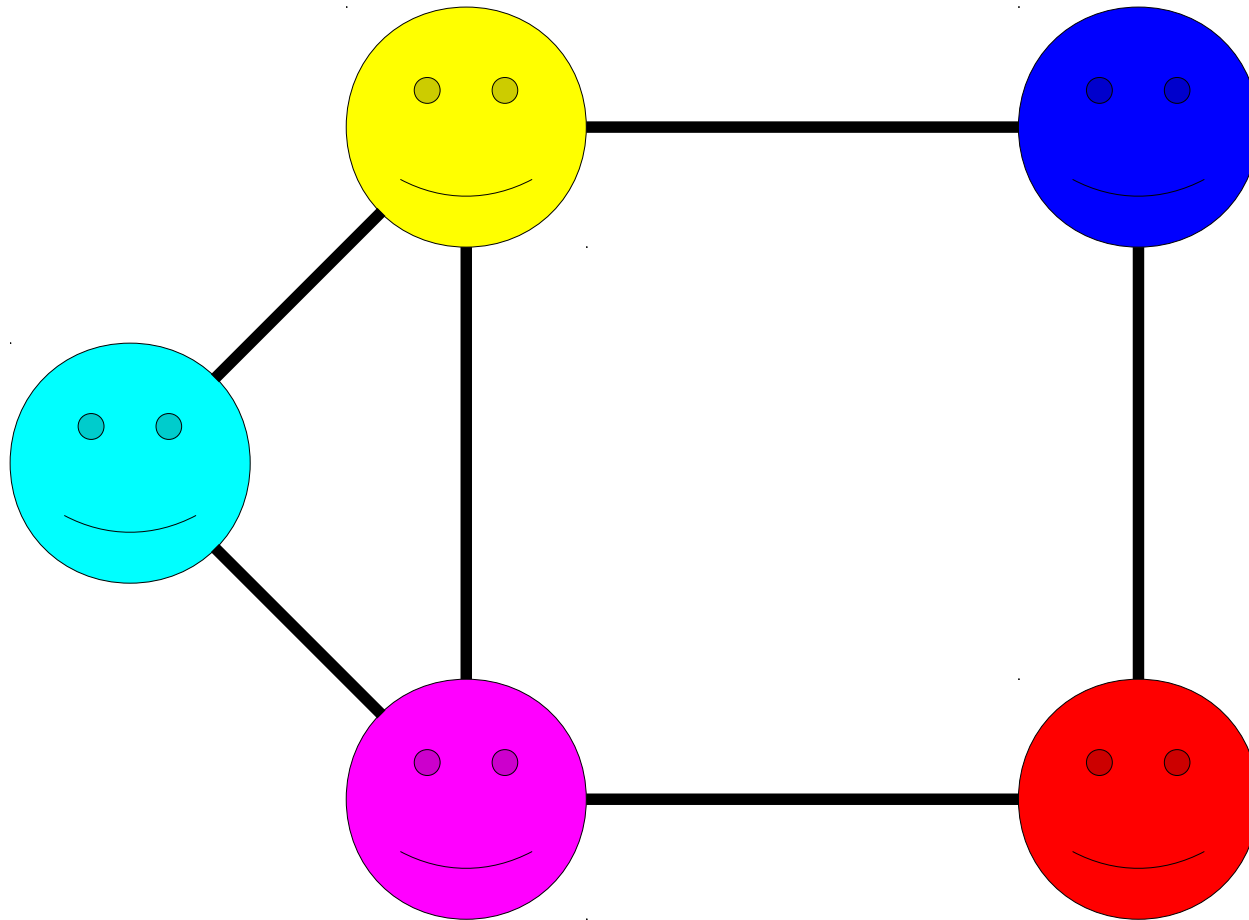
# What's in Common

- Each of these structures consists of
  - Individual 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.

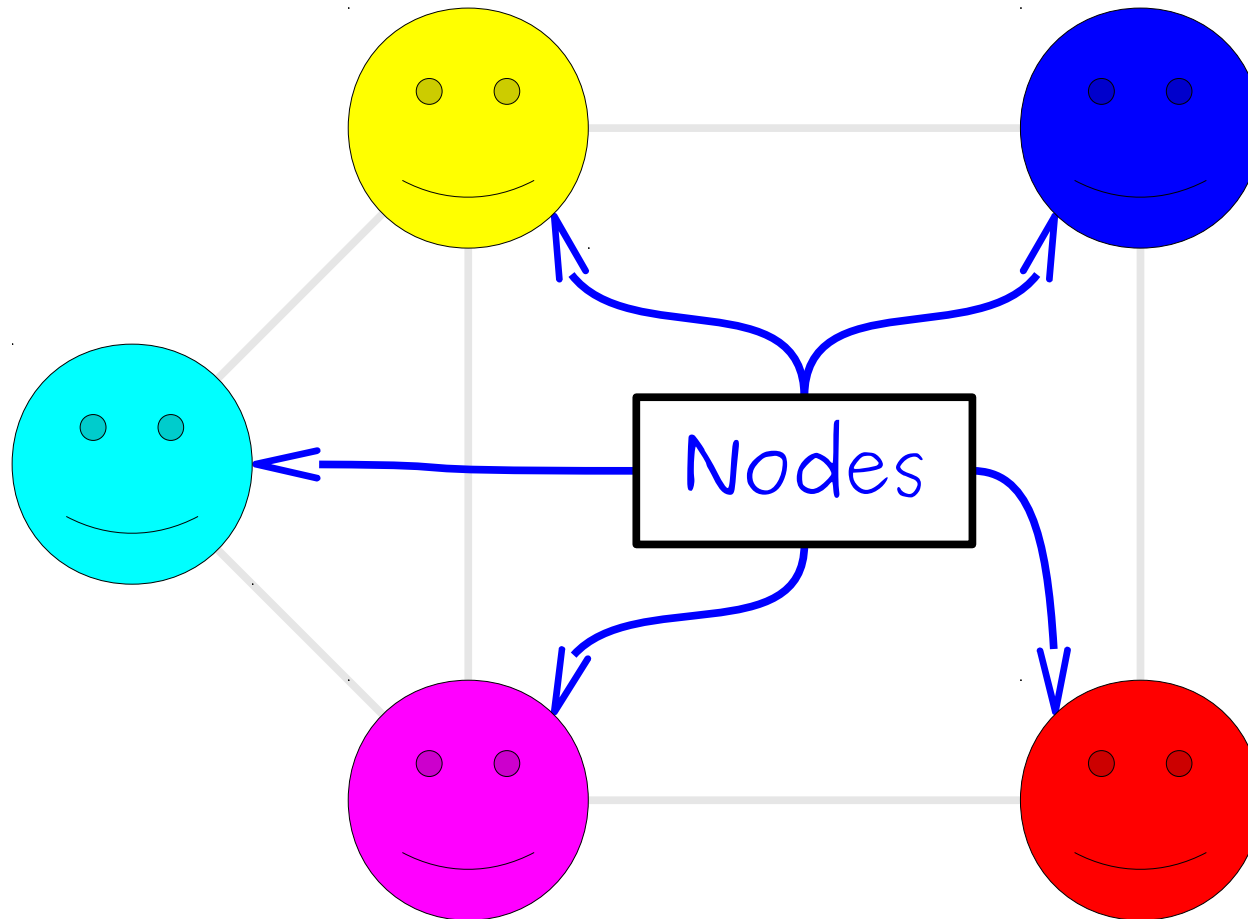


A **graph** is a mathematical structure for representing relationships.



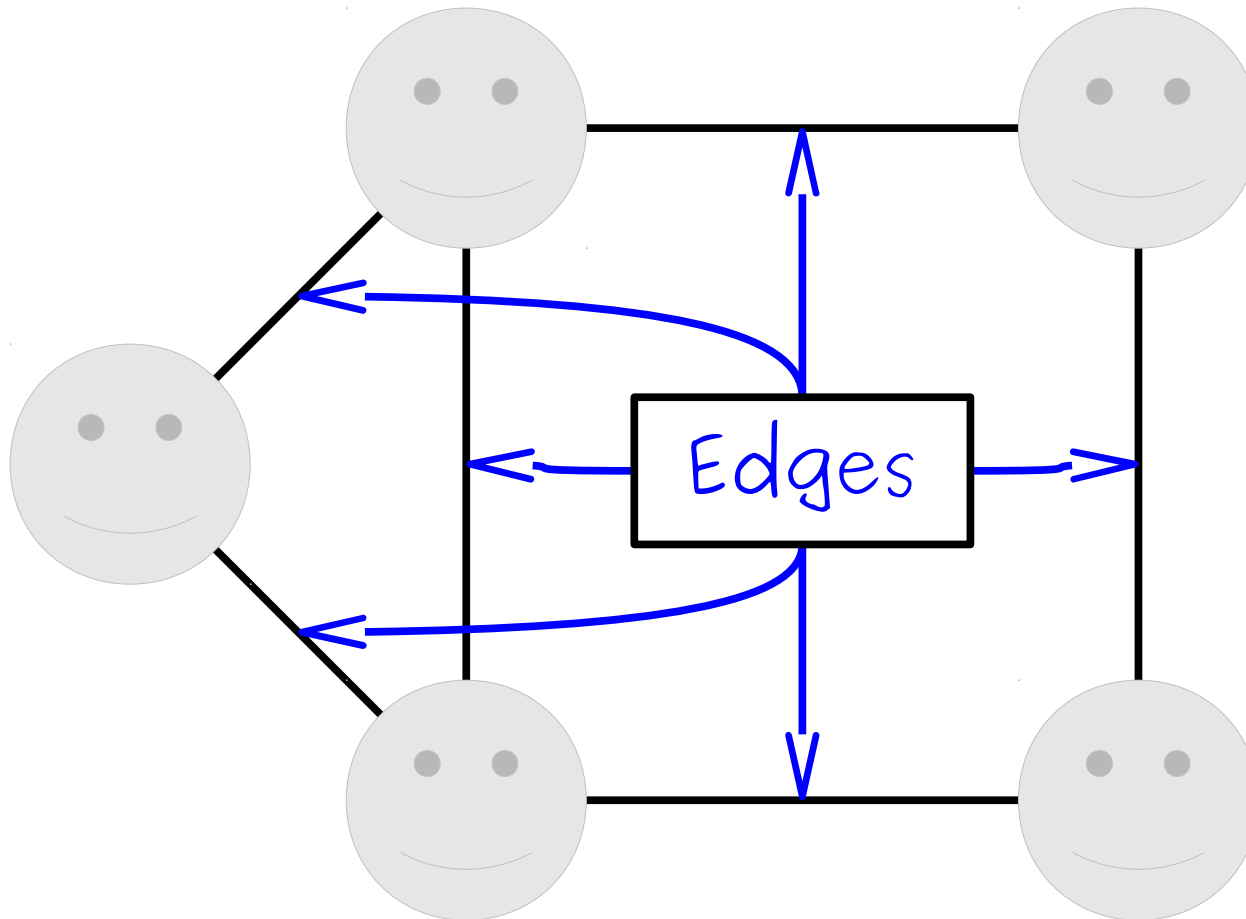
A graph consists of a set of **nodes** (or **vertices**) connected by **edges** (or **arcs**)

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



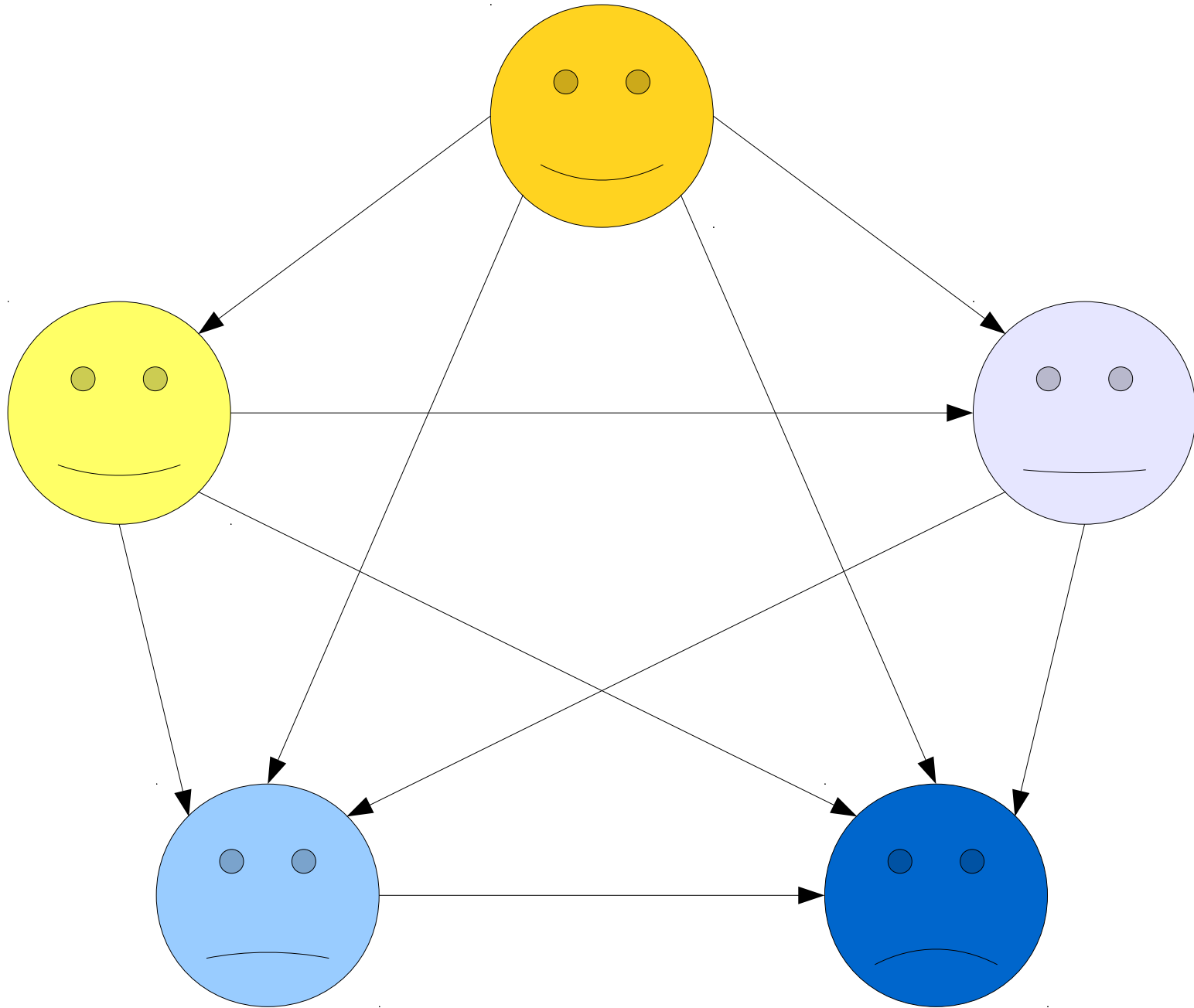
A graph consists of a set of **nodes** (or **vertices**) connected by **edges** (or **arcs**)

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

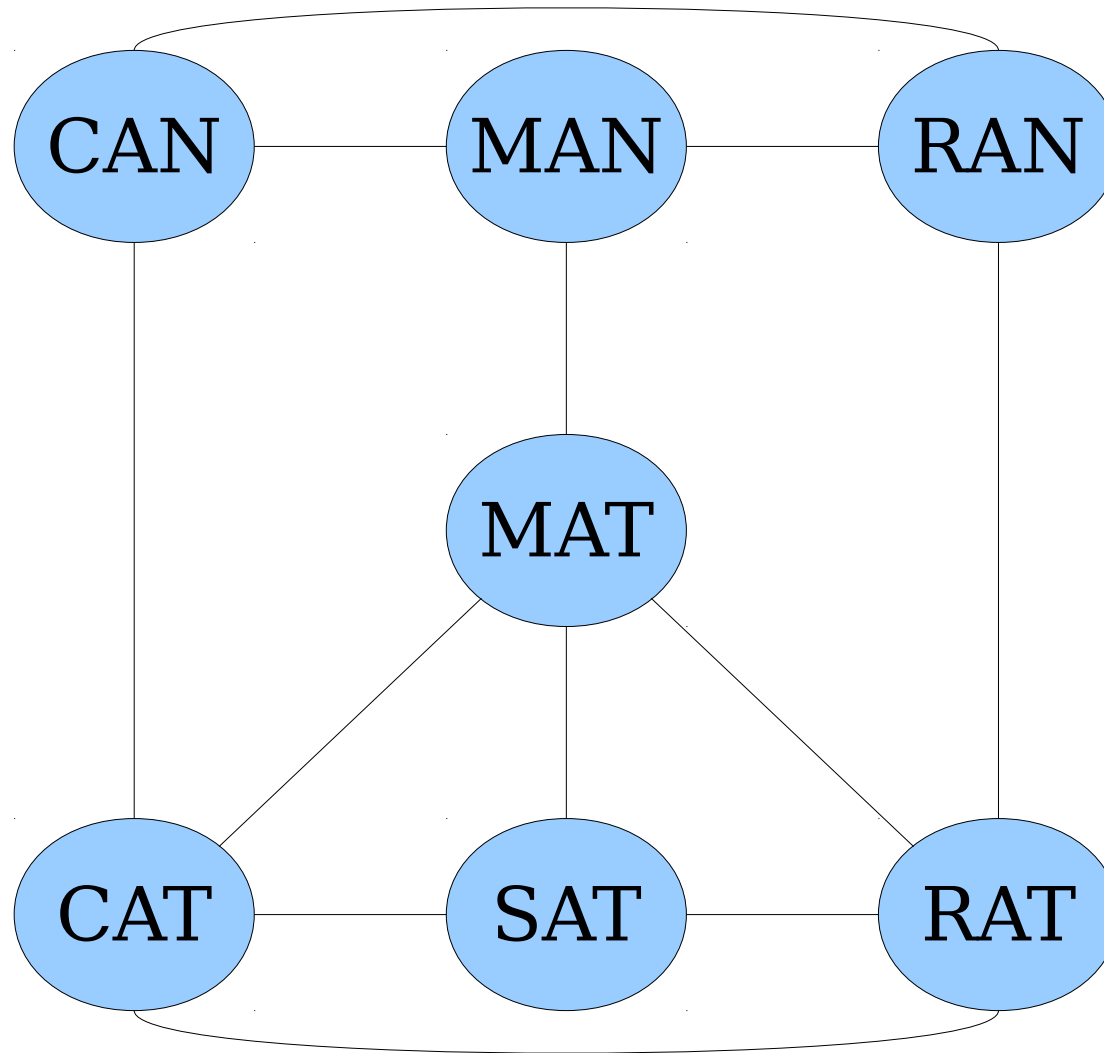


A graph consists of a set of **nodes** (or **vertices**) connected by **edges** (or **arcs**)

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 unless specified otherwise.

# Formalizing Graphs

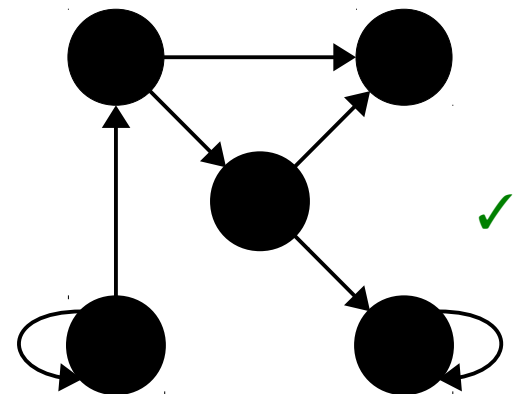
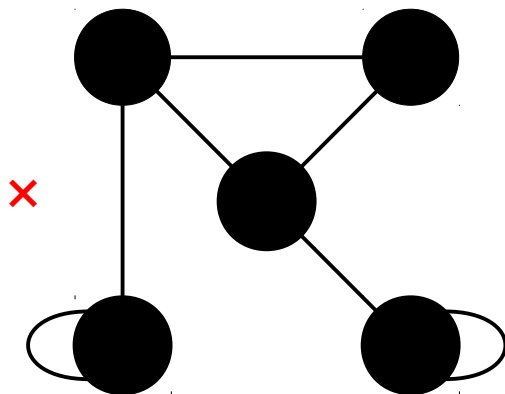
- How might we define a graph mathematically?
- Need to specify
  - What the nodes in the graph are, and
  - What the edges are in the graph.
- The nodes can be pretty much anything.
- What about the edges?

# Formalizing Graphs

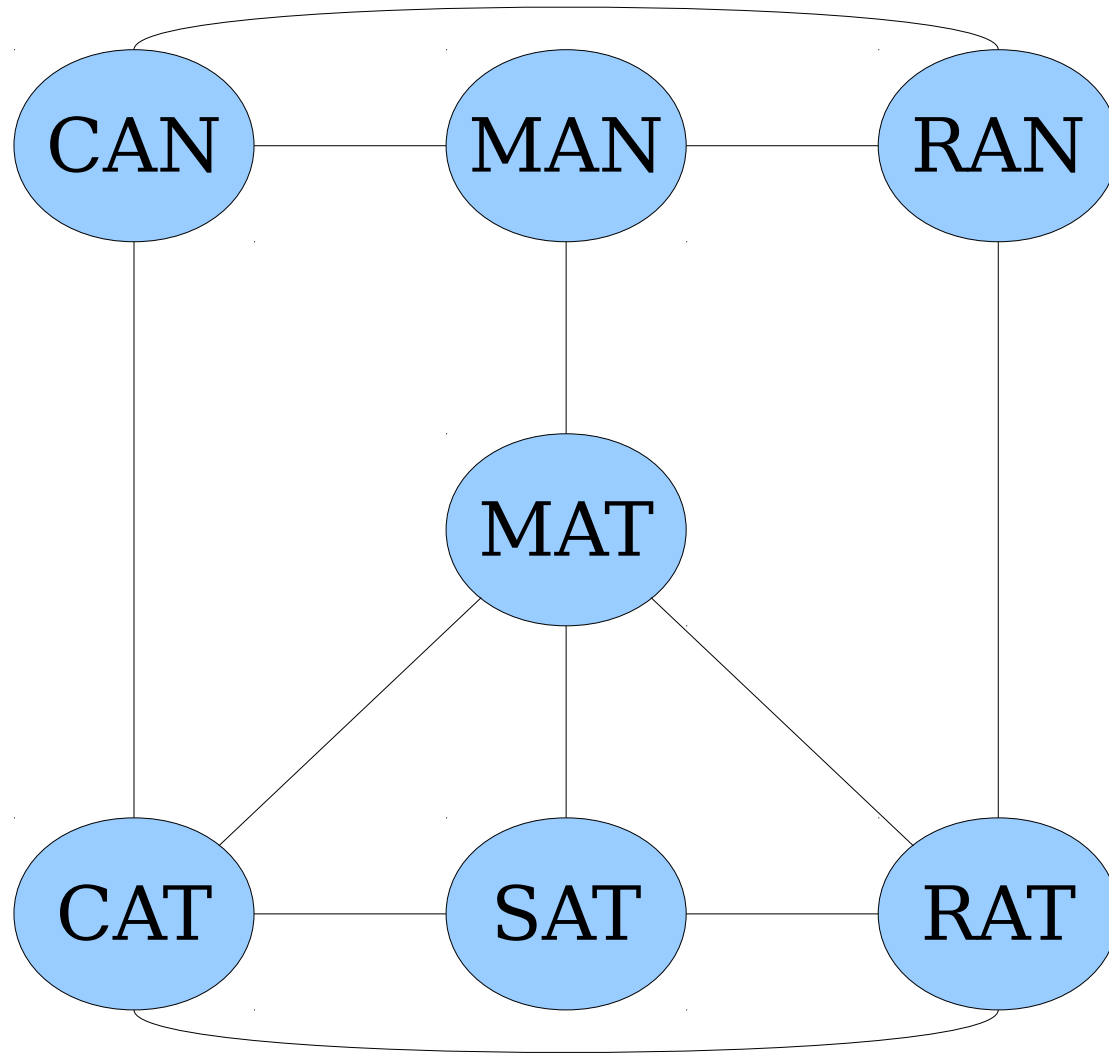
- An ***unordered pair*** is a set  $\{a, b\}$  of two elements (remember that sets are unordered).
  - $\{0, 1\} = \{1, 0\}$
- Formally, an undirected graph is an ordered pair  $G = (V, E)$ , where
  - $V$  is a set of nodes, which can be anything, and
  - $E$  is a set of edges, which are unordered pairs of nodes.
- Formally, 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.

# Self-Loops

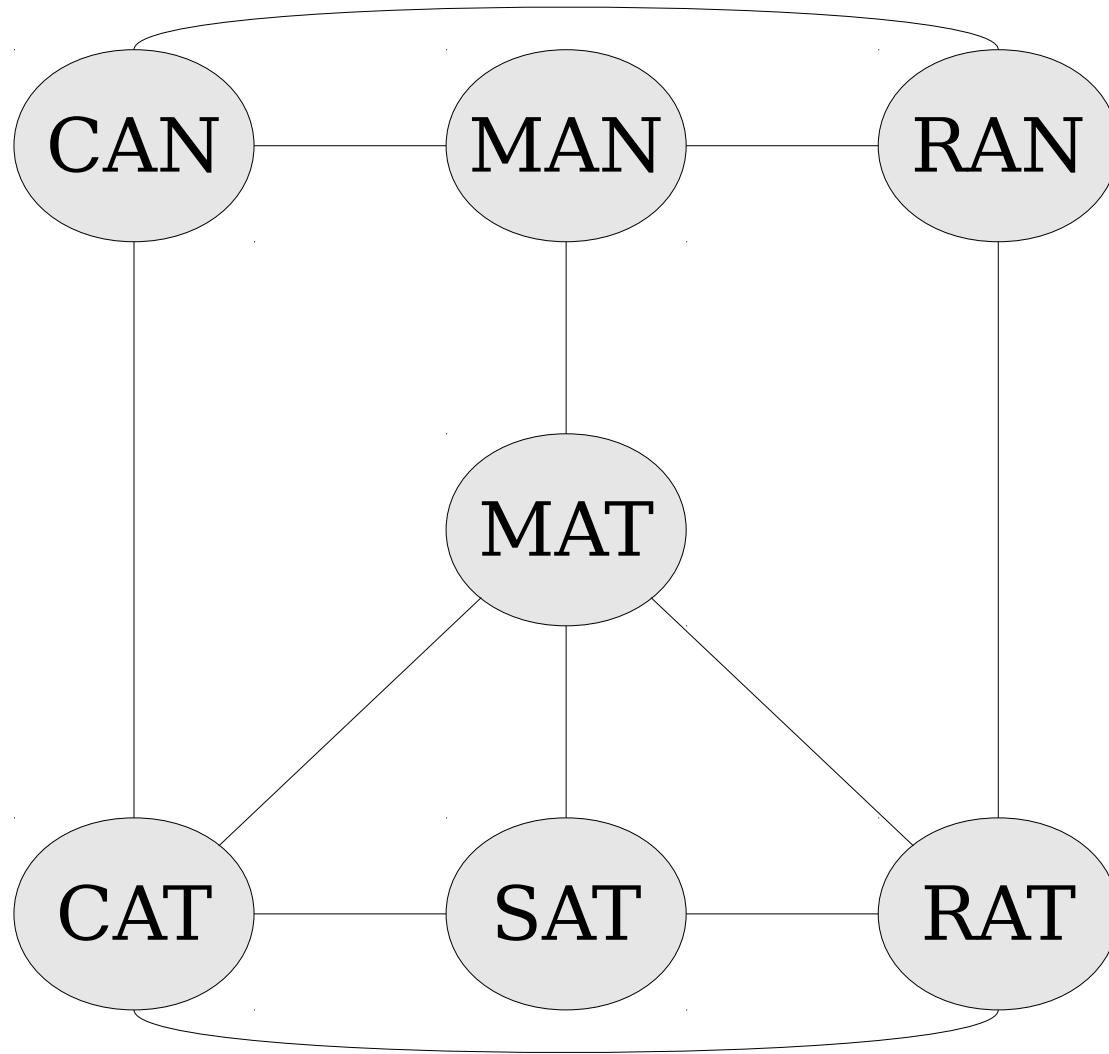
- An edge from a node to itself is called a ***self-loop***.
- In undirected graphs, self-loops are generally not allowed unless specified otherwise.
  - This is mostly to keep the math easier. If you allow self-loops, a lot of results get messier and harder to state.
- In directed graphs, self-loops are generally allowed unless specified otherwise.



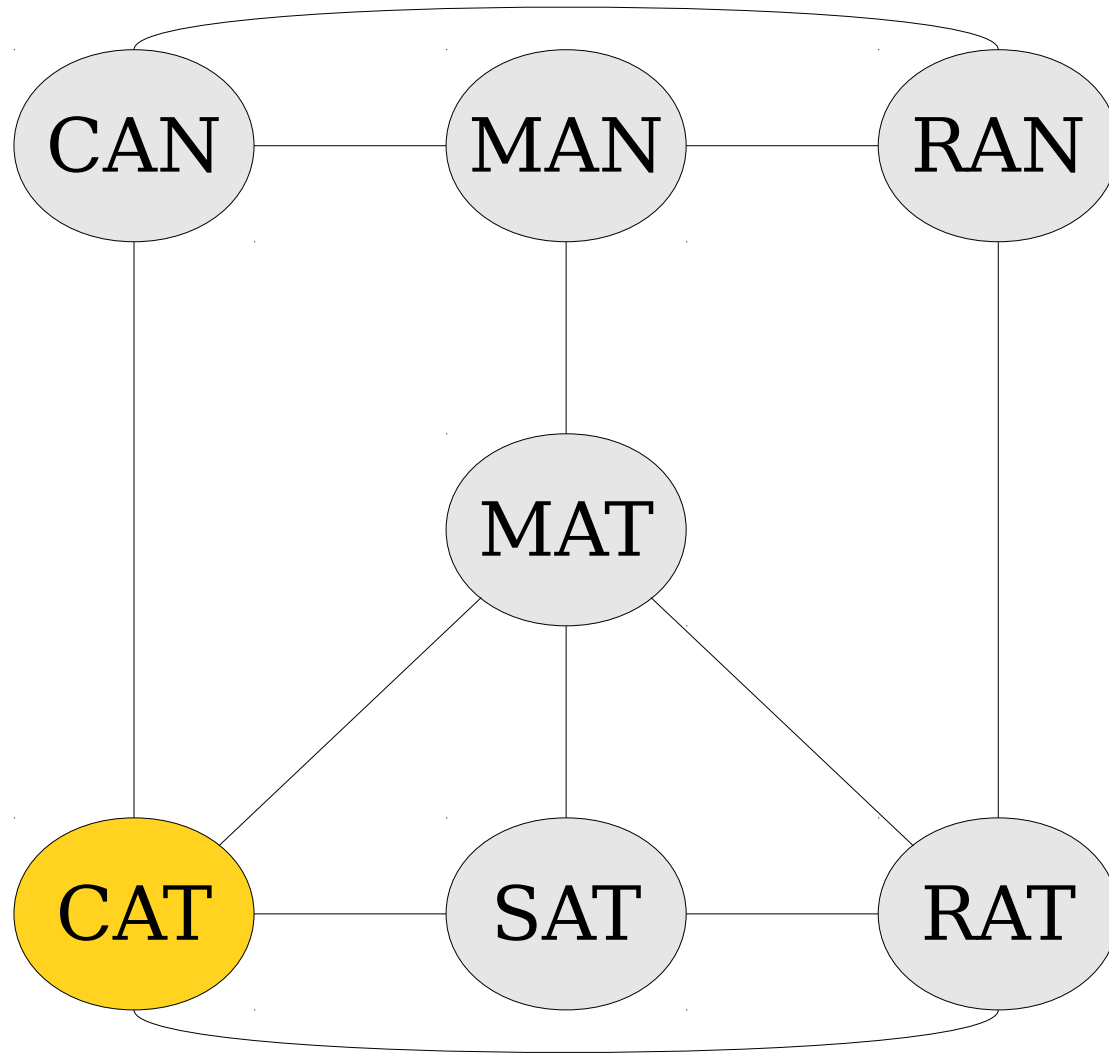
# Standard Graph Terminology



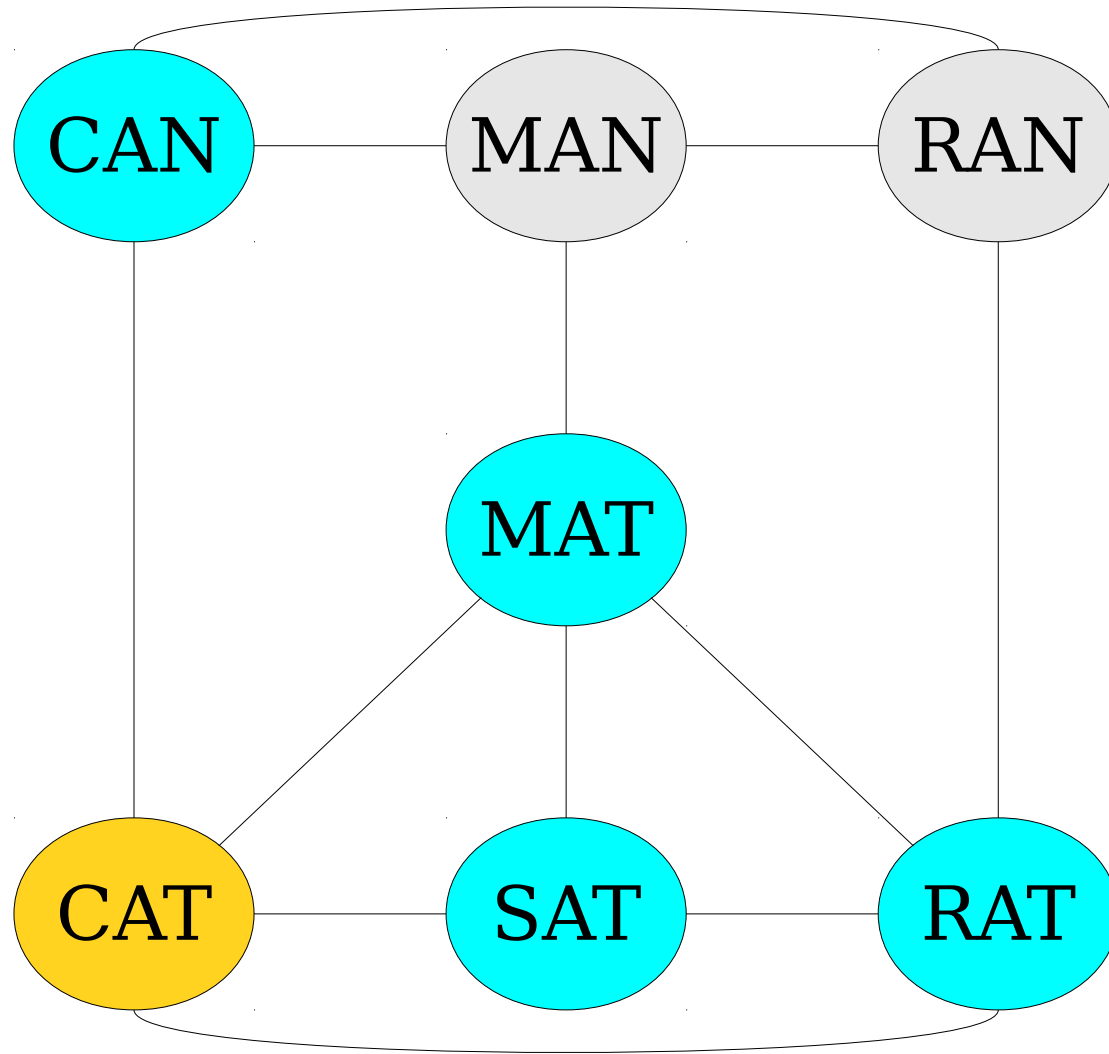
Two nodes are called *adjacent* if there is an edge between them.



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



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

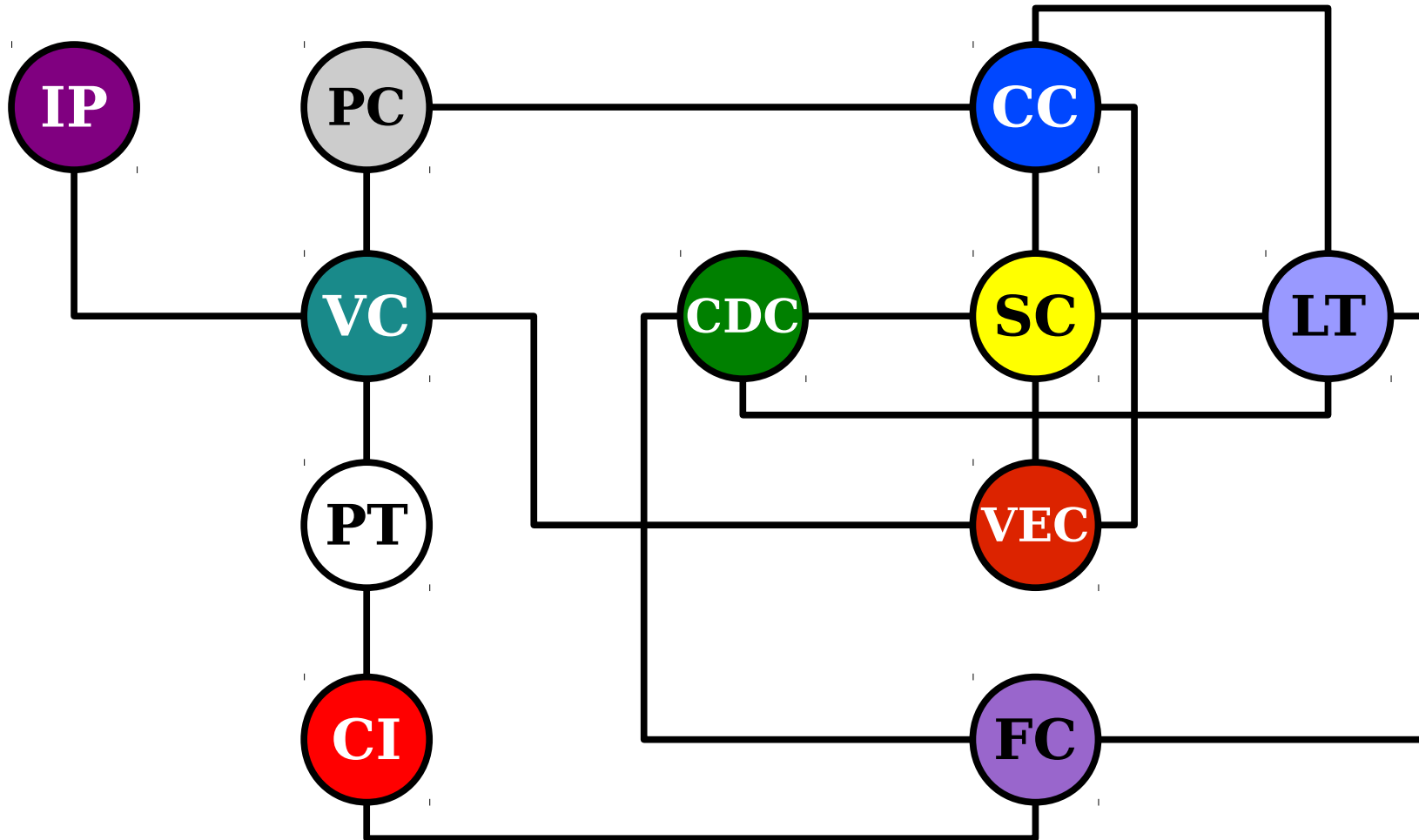


Two nodes are called *adjacent* if there is an edge between them.

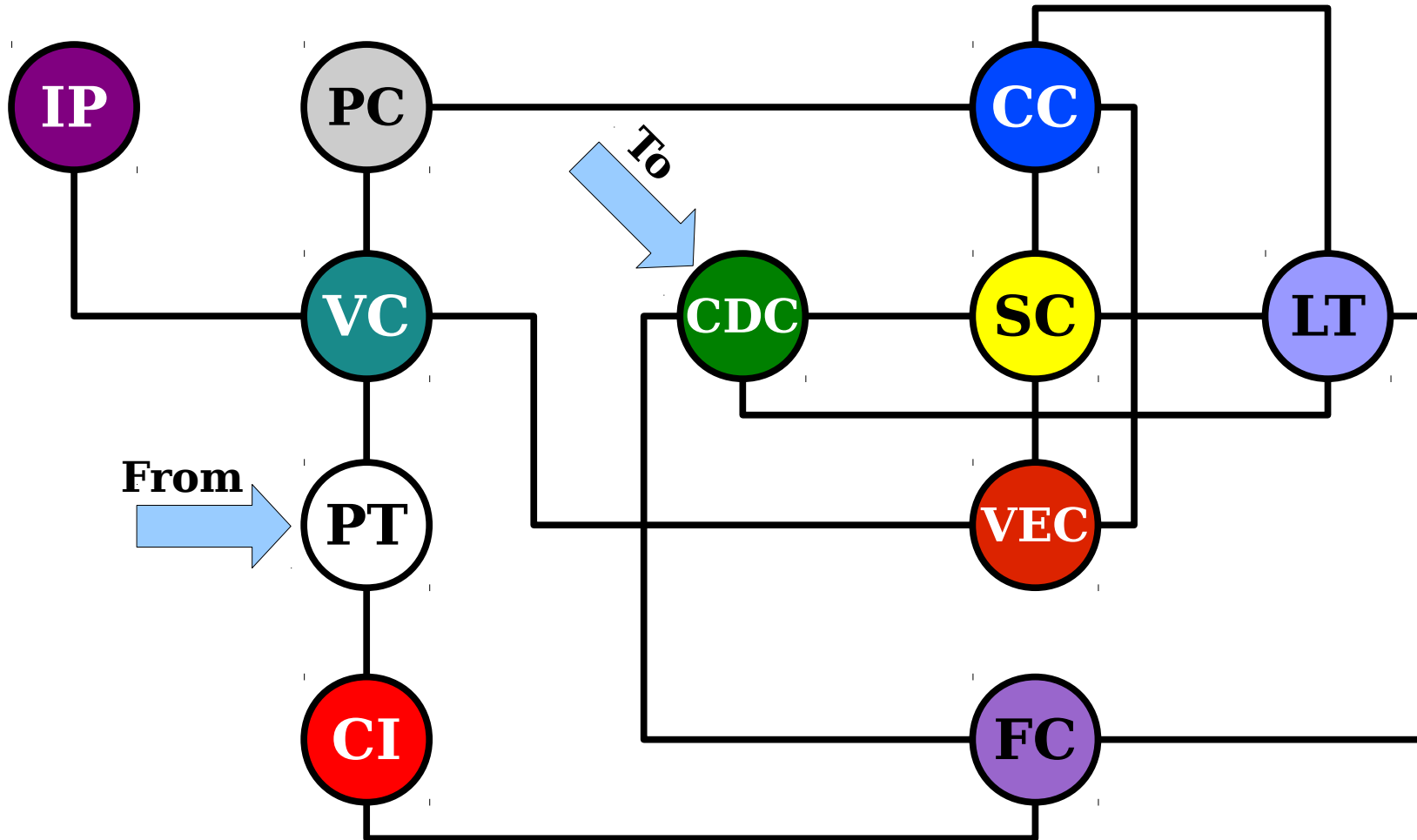
# Using our Formalisms

- Let  $G = (V, E)$  be 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  $\{u, v\} \in E$ .

# Navigating a Graph

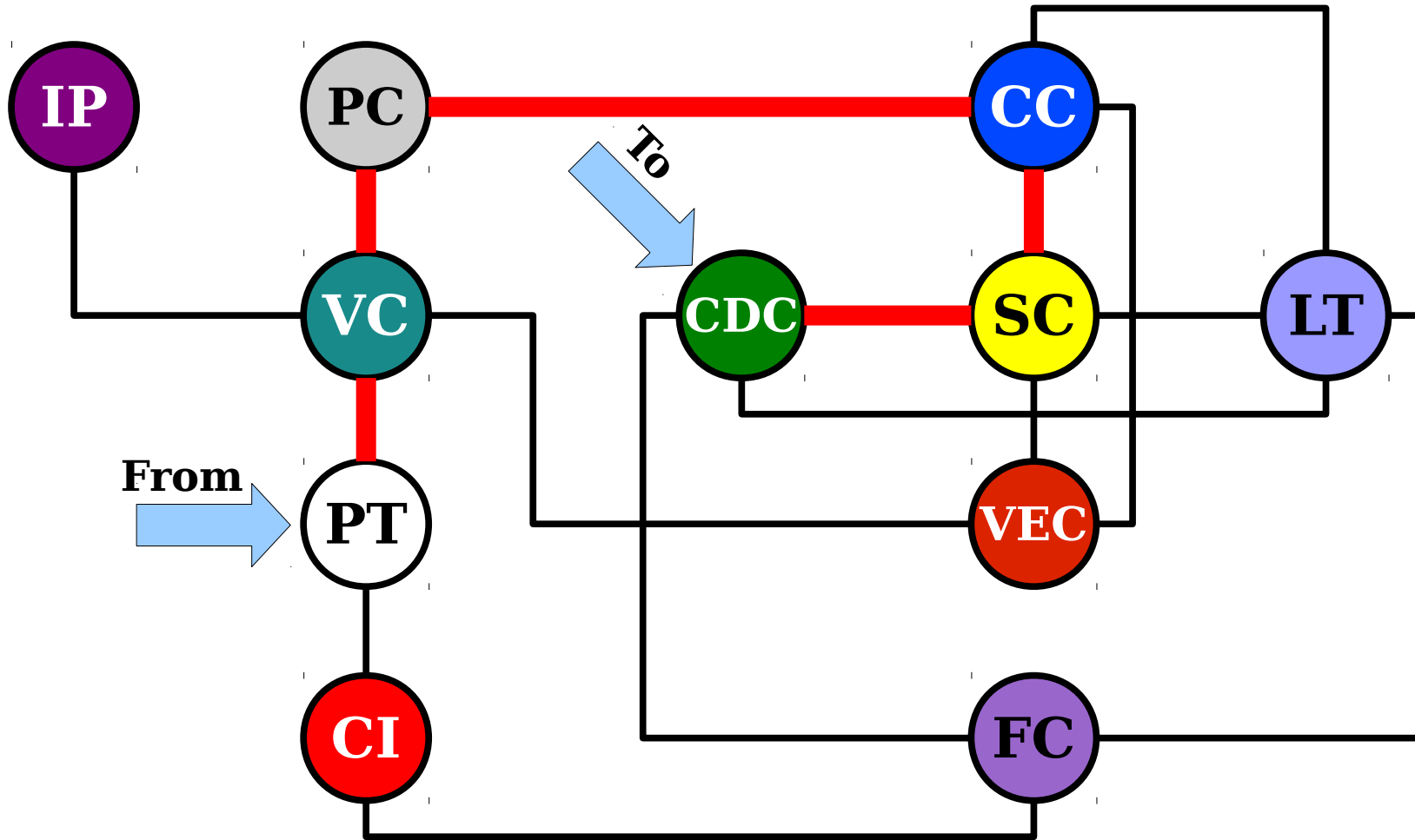


# Navigating a Graph



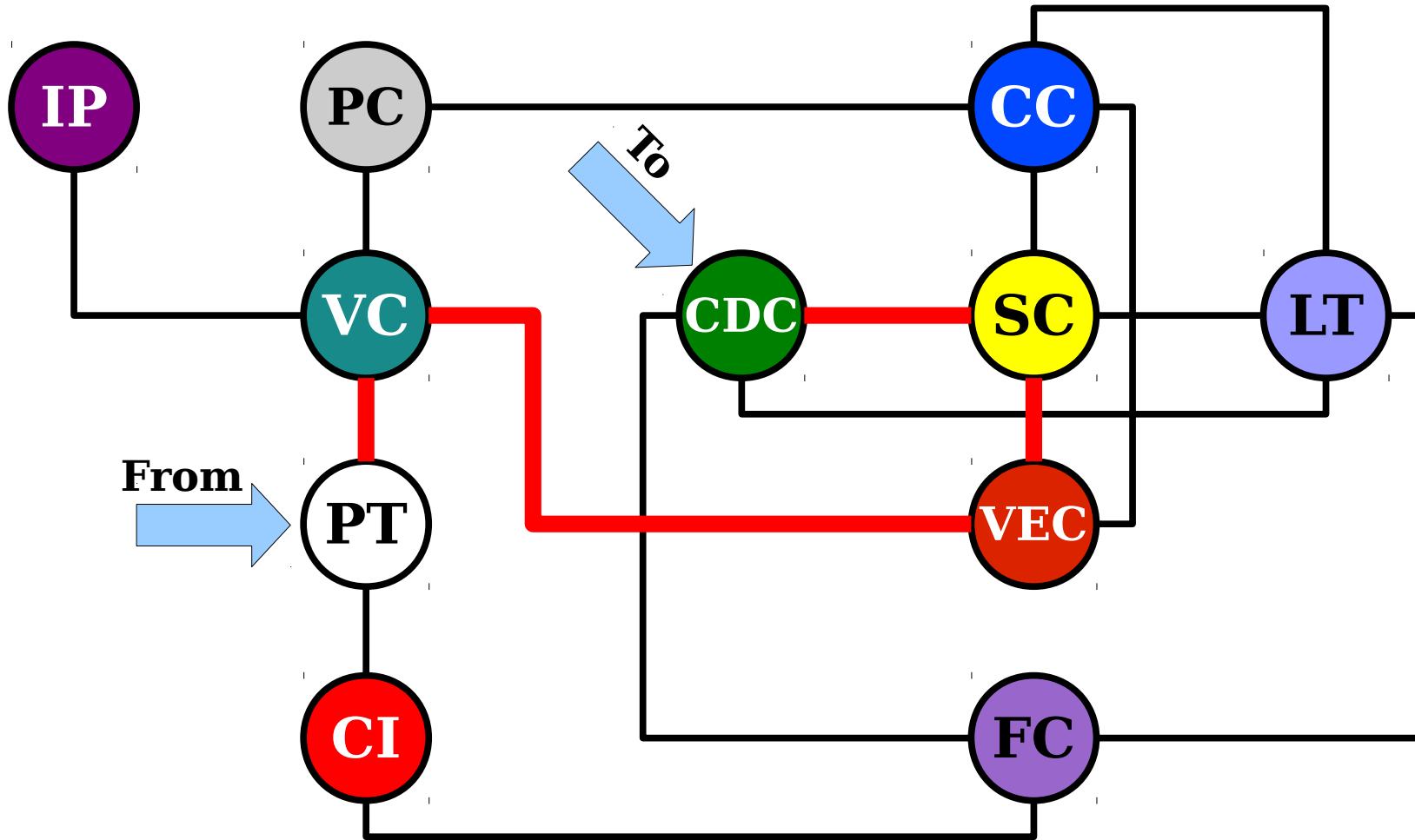


# Navigating a Graph

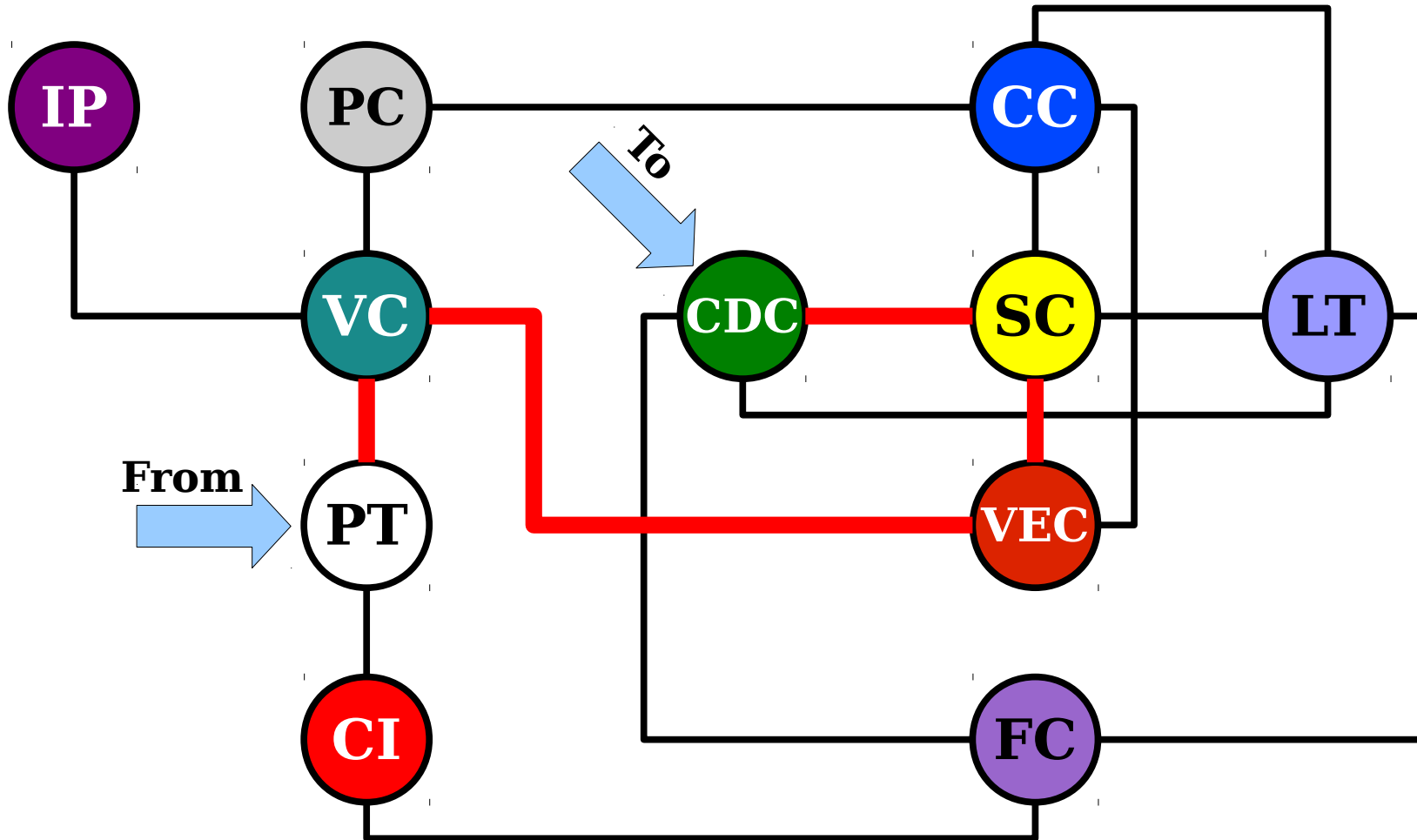


PT → VC → PC → CC → SC → CDC

# Navigating a Graph

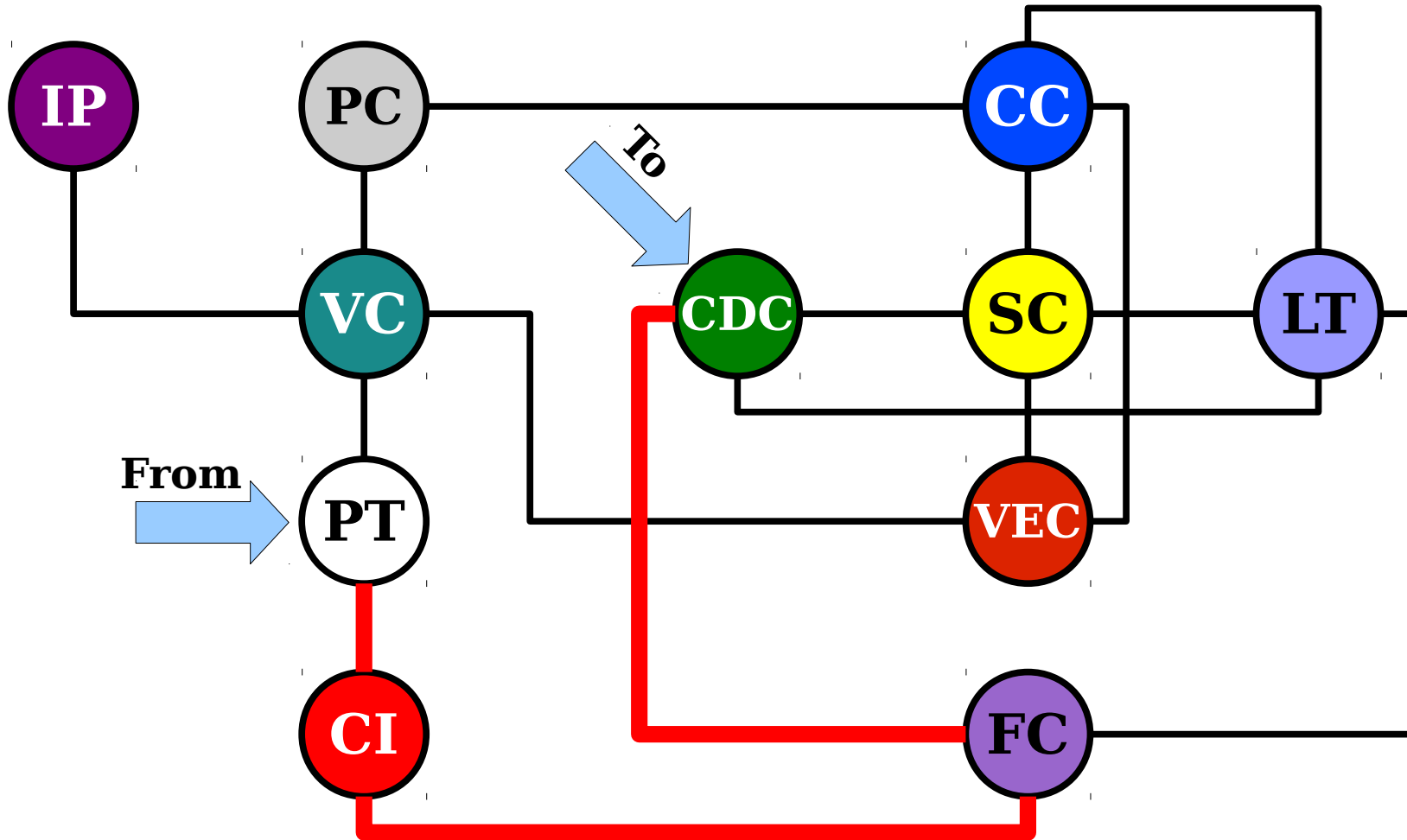


# Navigating a Graph

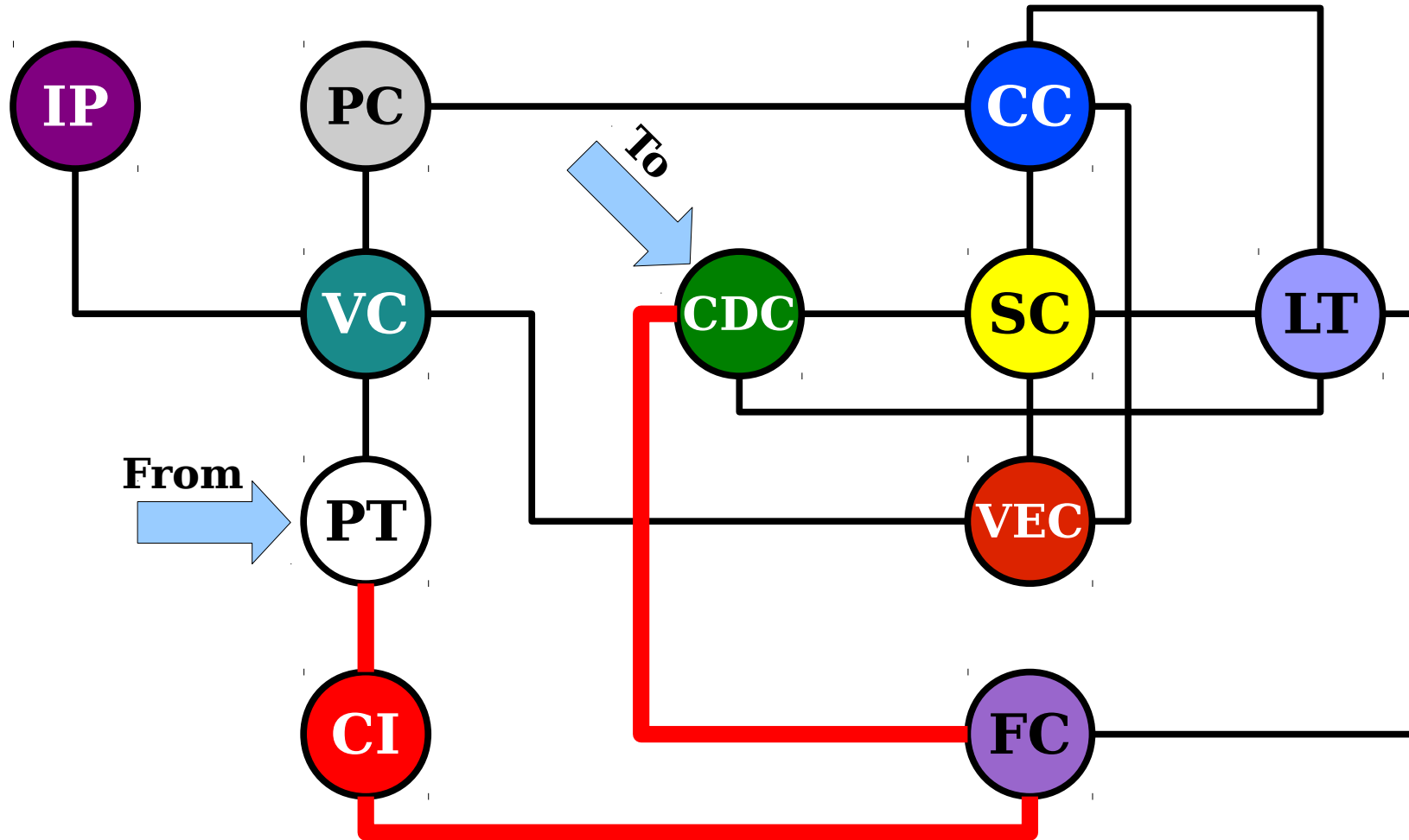


PT → VC → VEC → SC → CDC

# Navigating a Graph



# Navigating a Graph



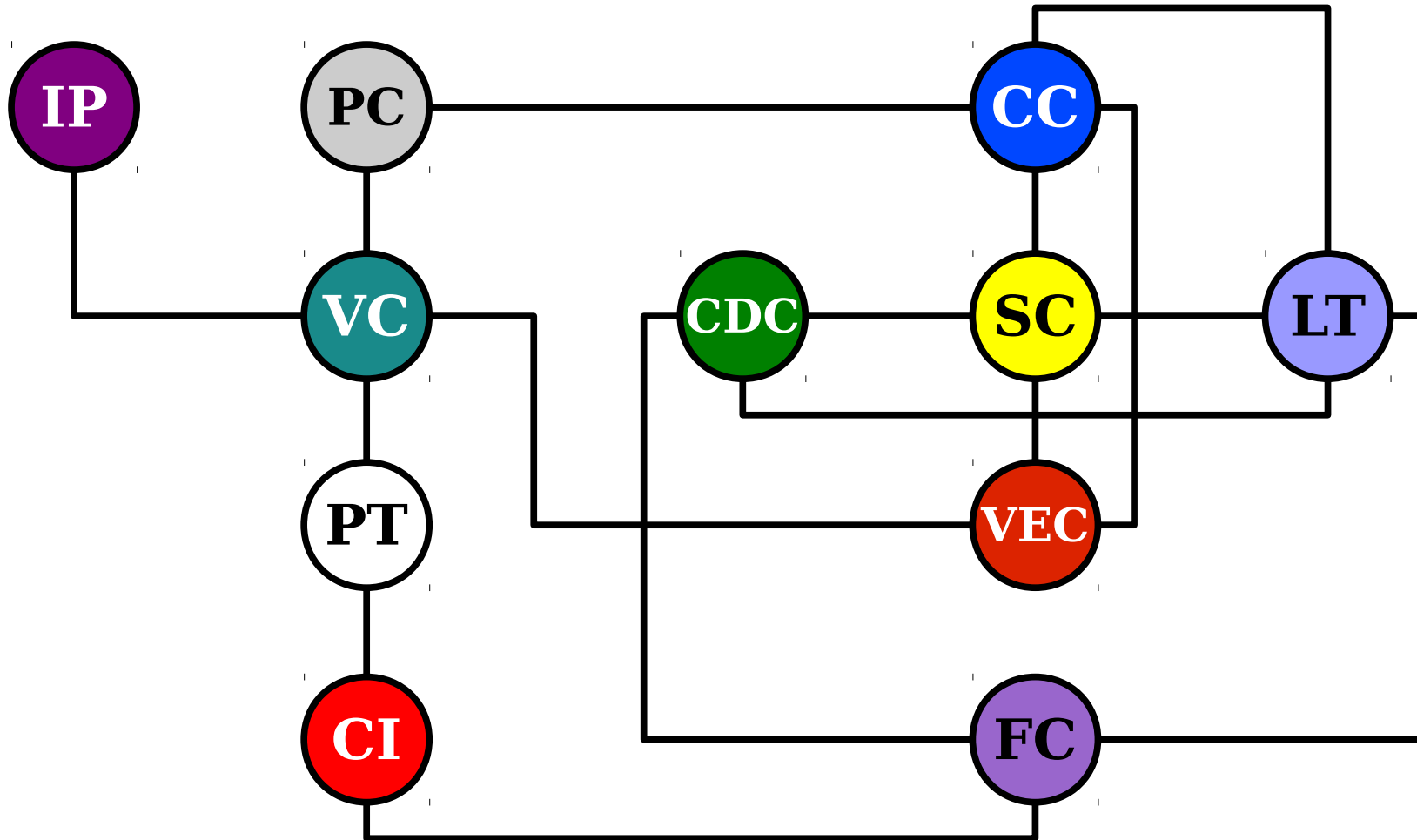
PT → CI → FC → CDC

A **path** from  $v_1$  to  $v_n$  is a sequence of nodes  $v_1, v_2, \dots, v_n$  where  $\{v_k, v_{k+1}\} \in E$  for all natural numbers in the range  $1 \leq k \leq n - 1$ .

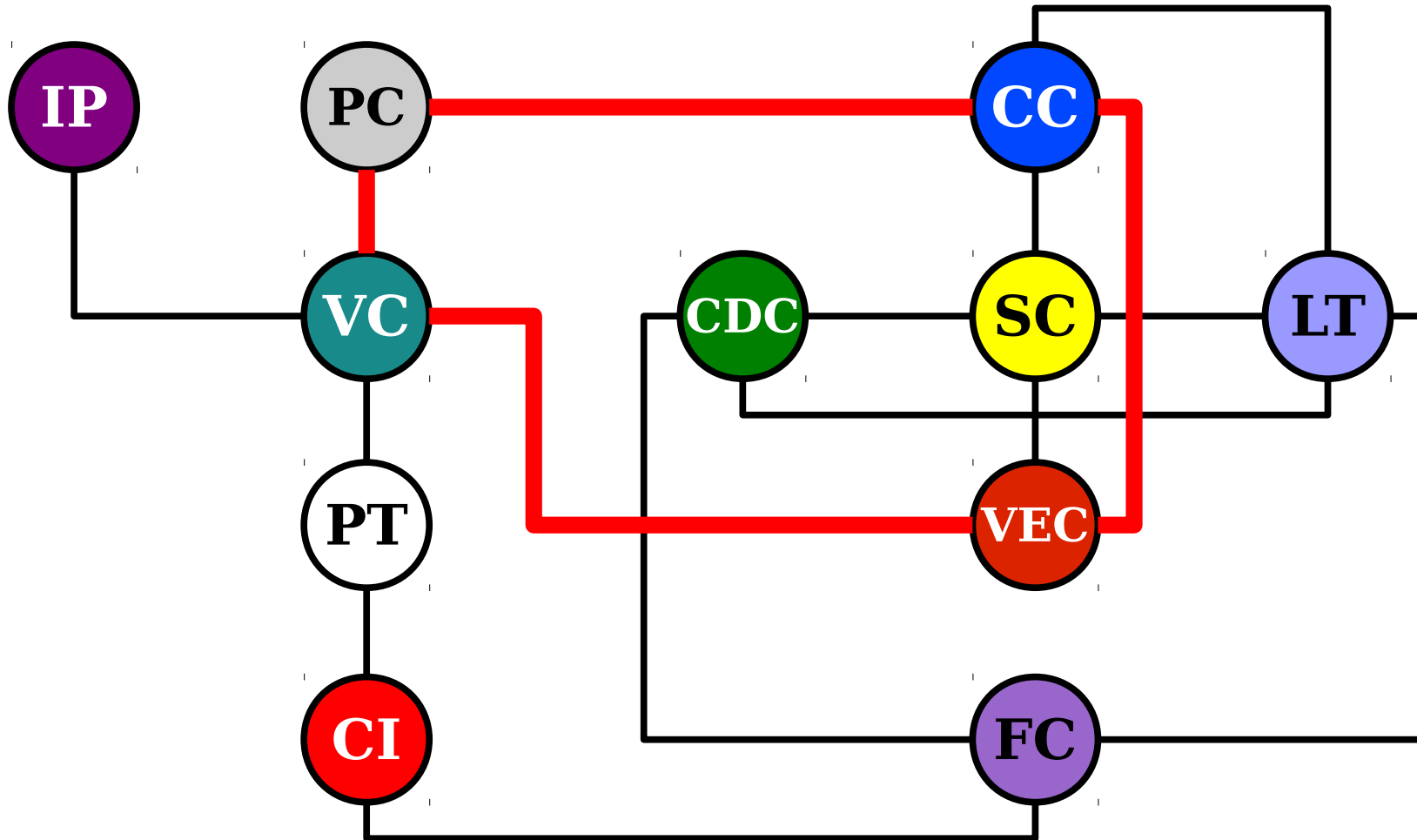
The **length** of a path is the number of edges it contains, which is one less than the number of nodes in the path.

As an edge case, paths of length zero are allowed.

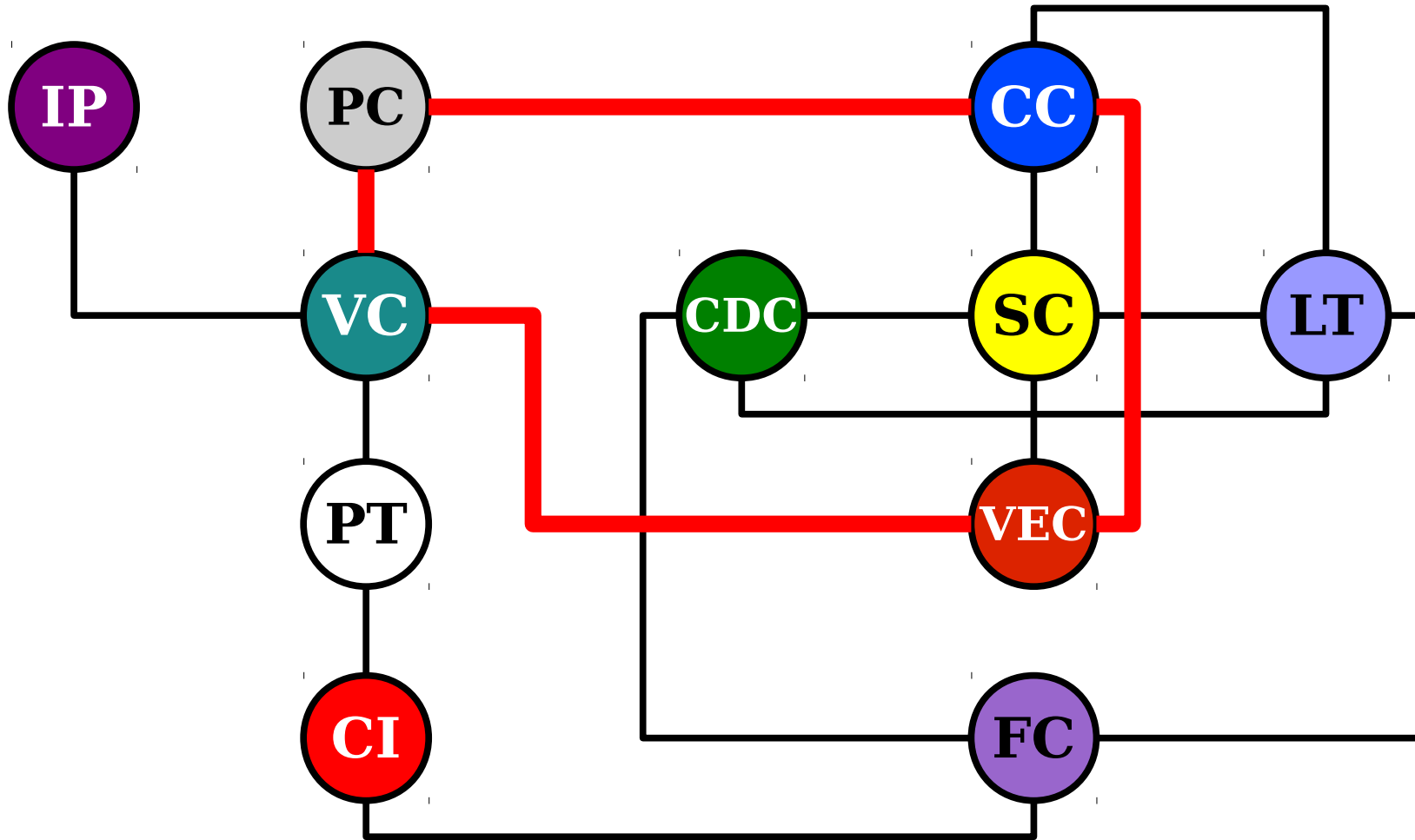
# Navigating a Graph



# Navigating a Graph

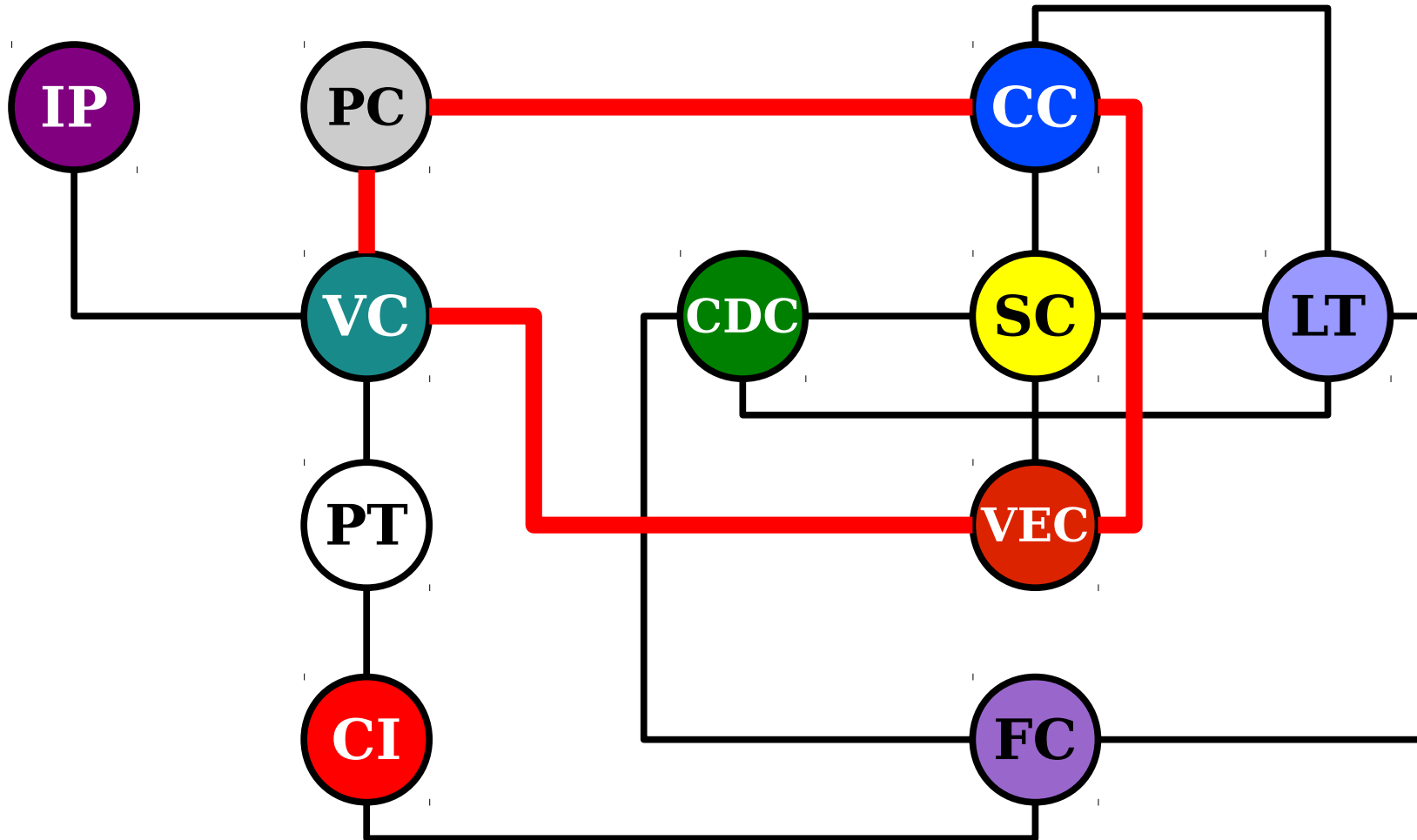


# Navigating a Graph



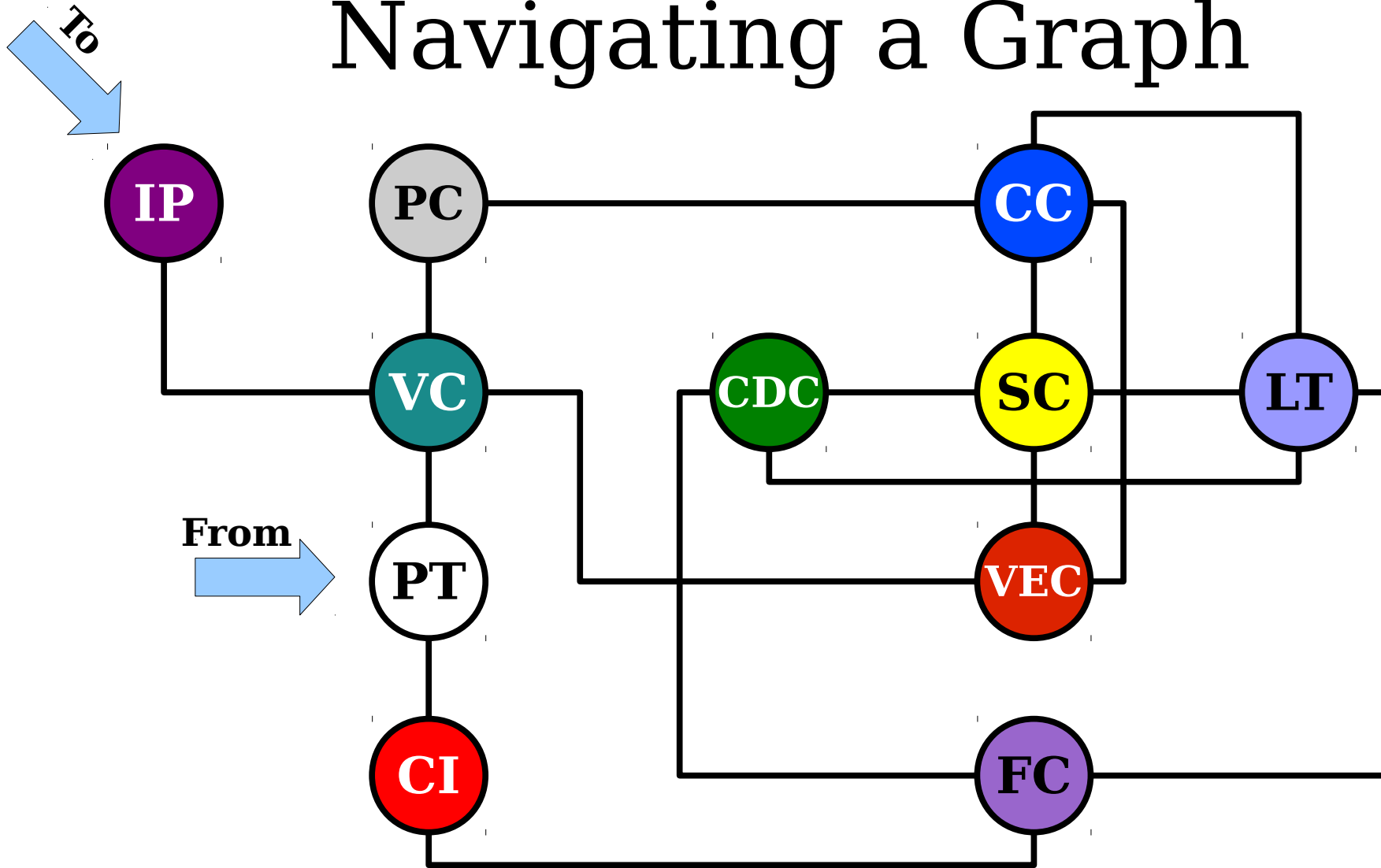
PC → CC → VEC → VC → PC

# Navigating a Graph

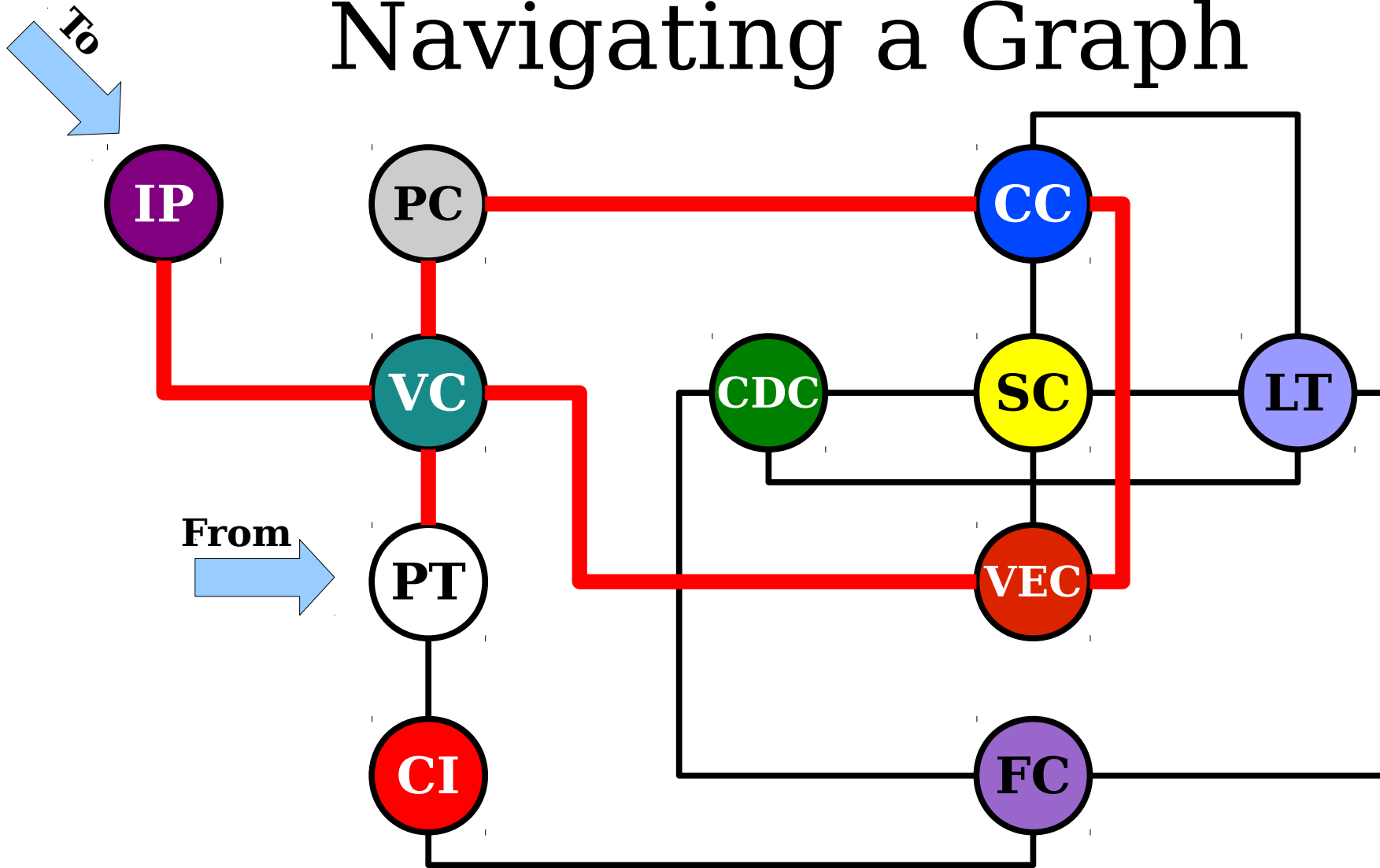


PC → CC → VEC → VC → PC → CC → VEC → VC → PC

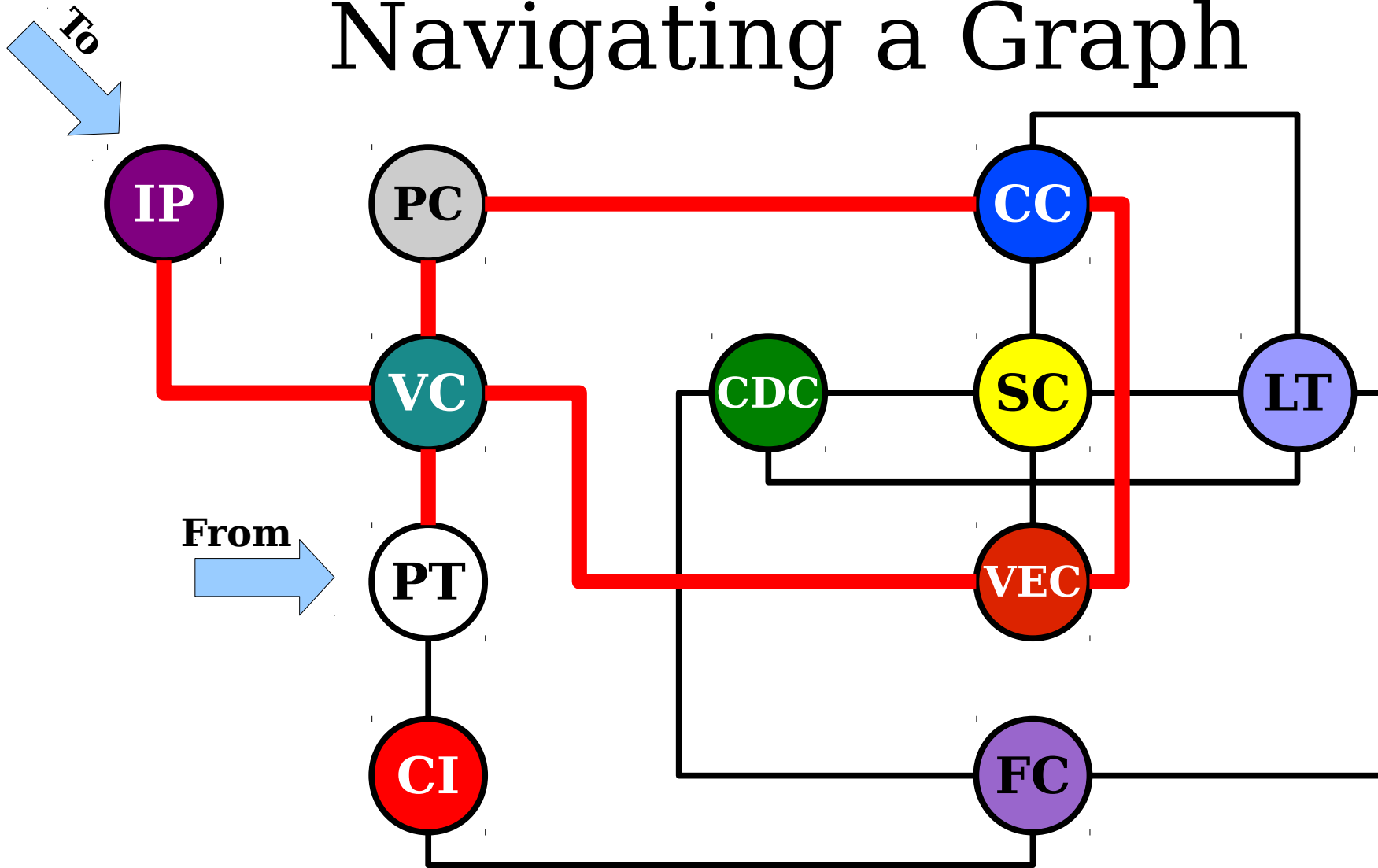
# Navigating a Graph



# Navigating a Graph



# Navigating a Graph



PT → VC → PC → CC → VEC → VC → IP

A ***cycle*** in a graph is a path from a node to itself.

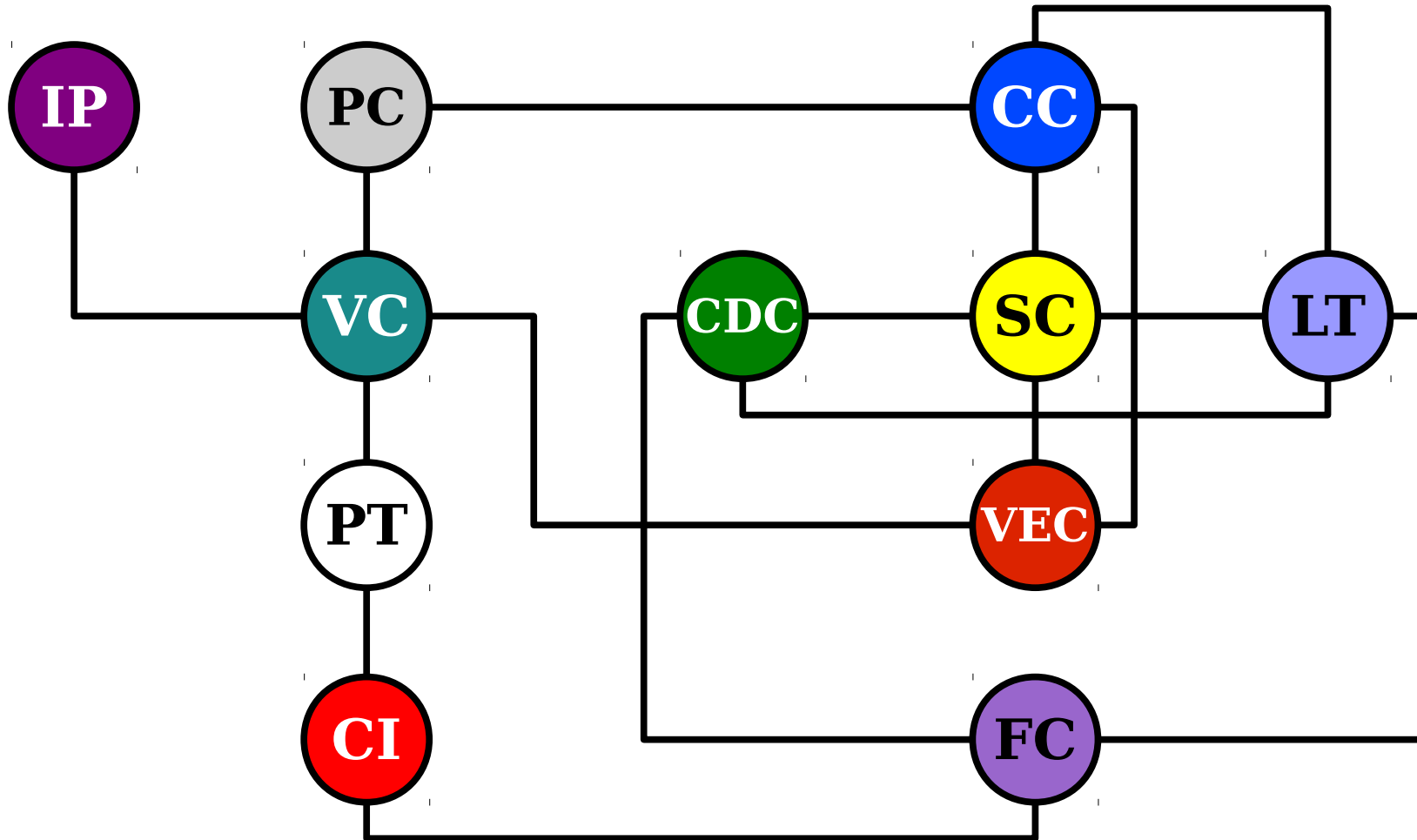
The ***length*** of a cycle is the number of edges in that cycle.

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

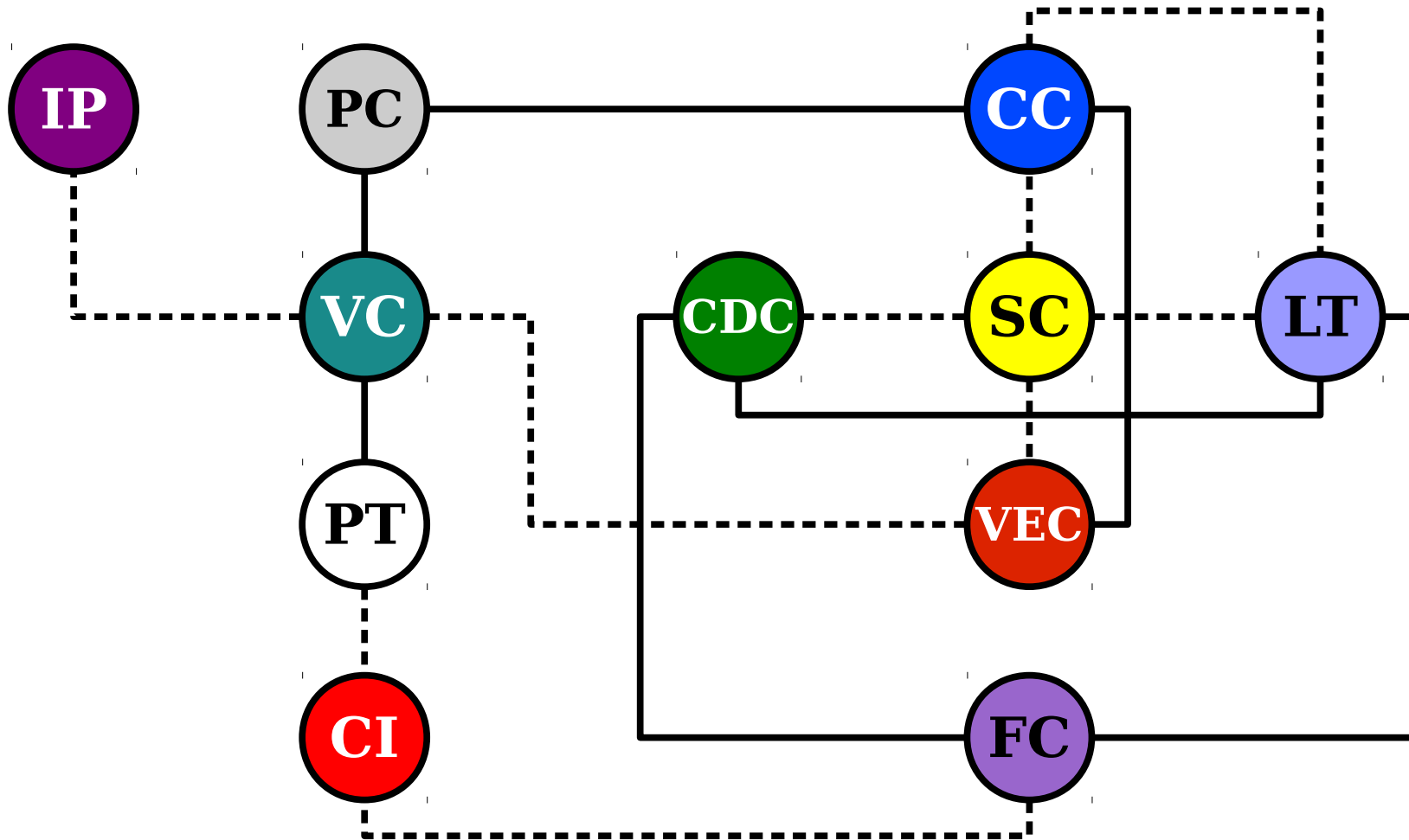
A ***simple cycle*** in a graph is a cycle that does not revisit any nodes or edges (except the start/end node).

Usually, the ***empty path*** starting and ending at a node and containing no edges is considered a simple path.

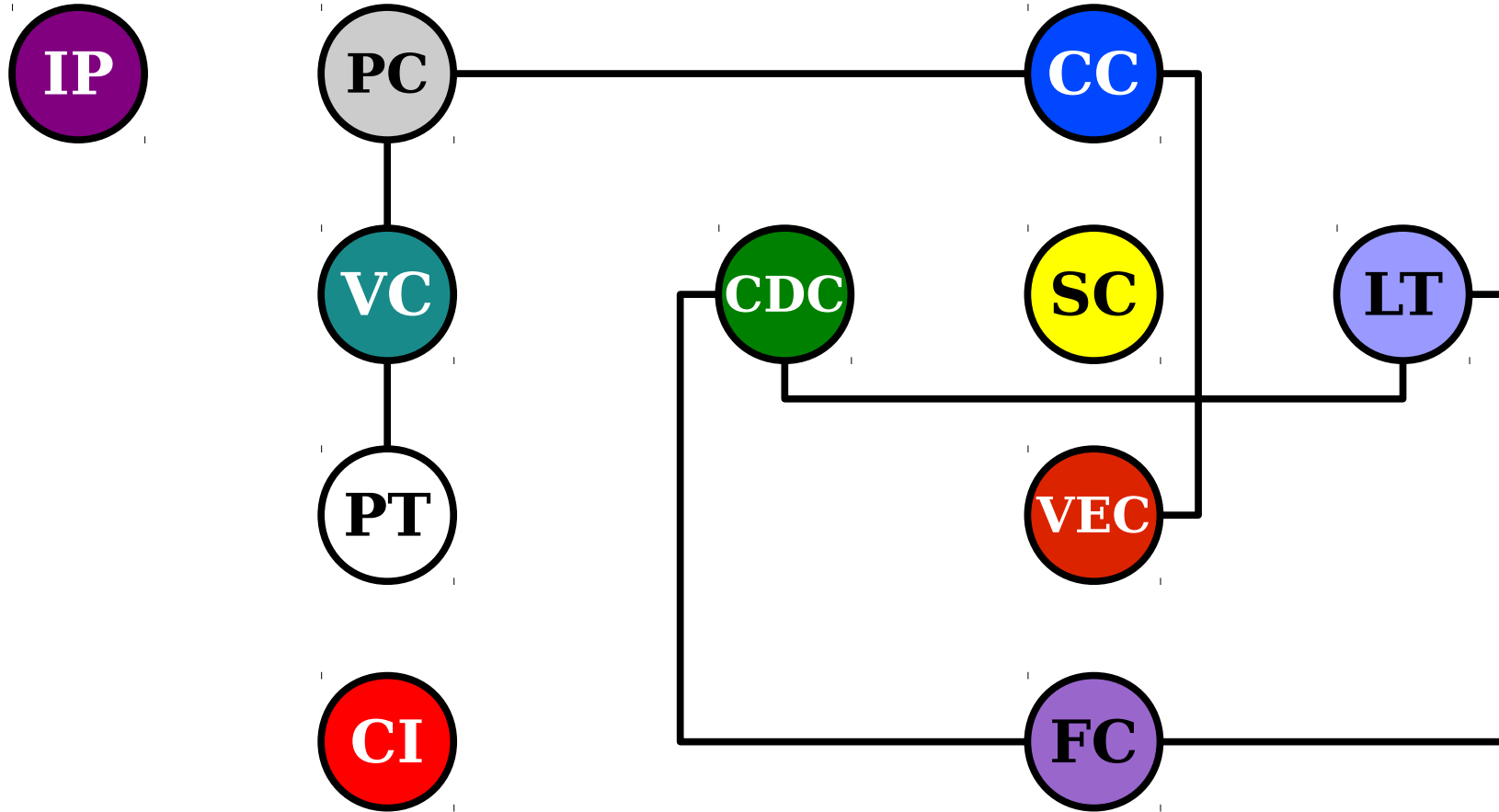
# Navigating a Graph



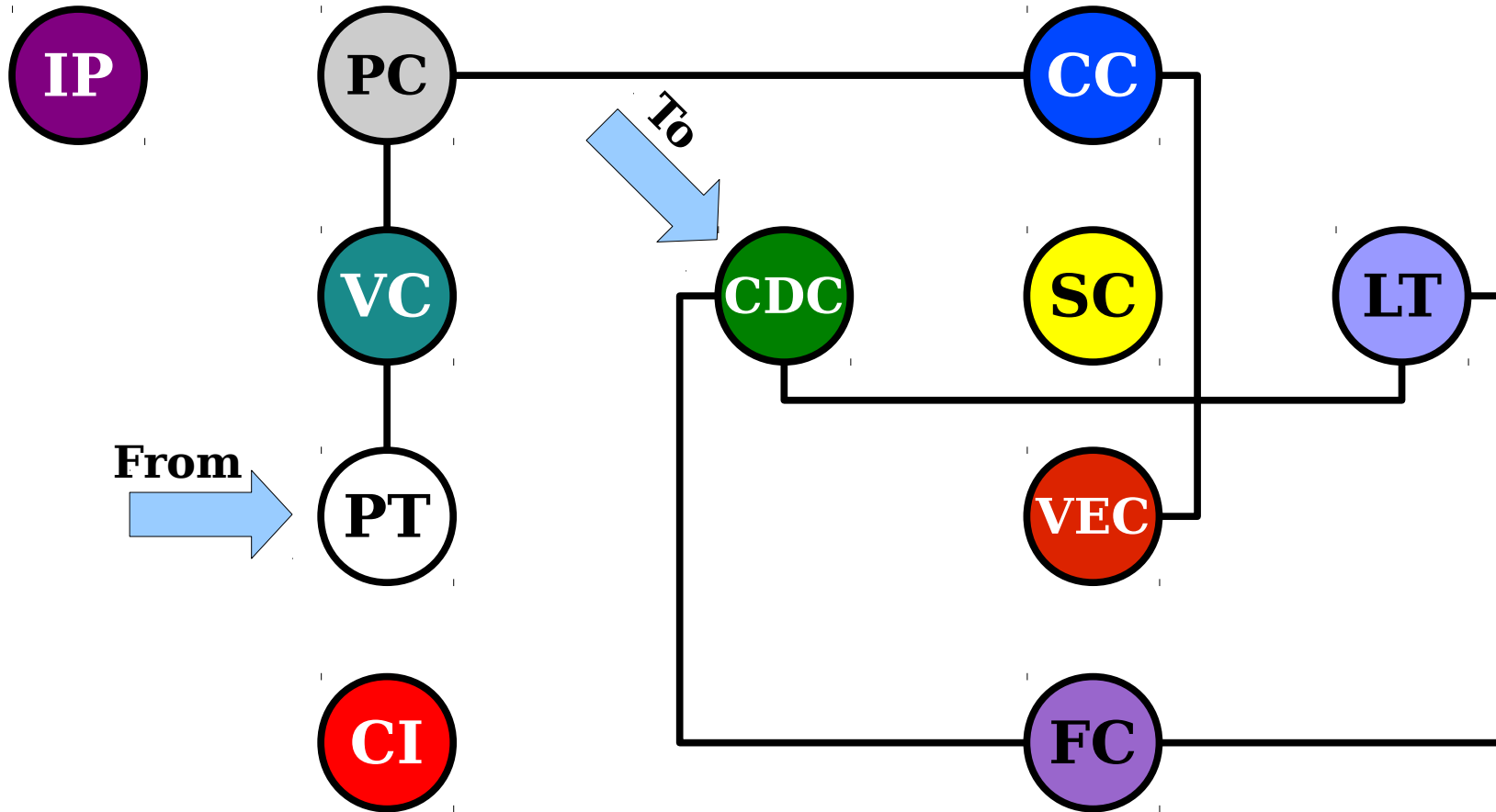
# Navigating a Graph



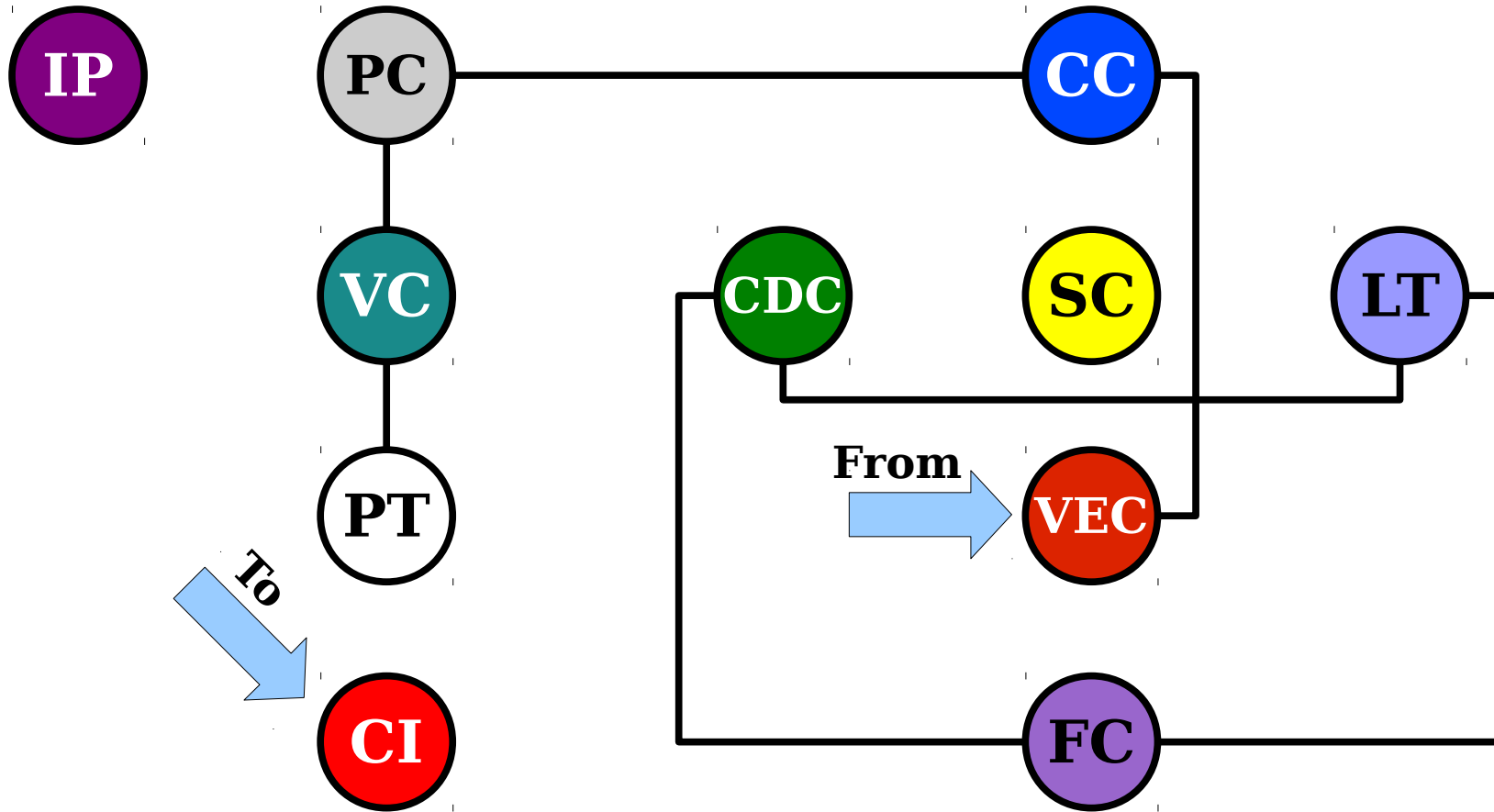
# Navigating a Graph



# Navigating a Graph



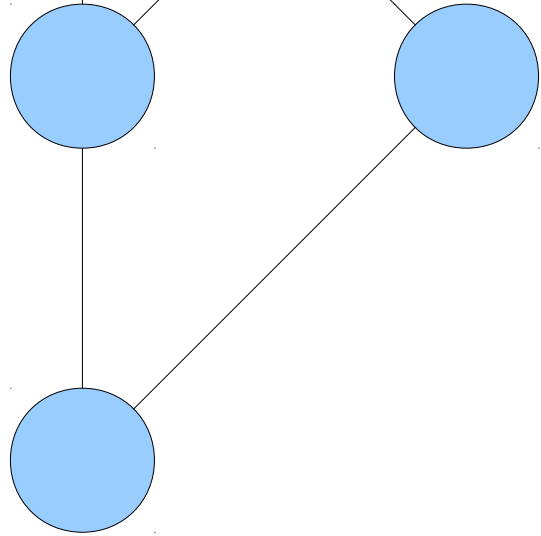
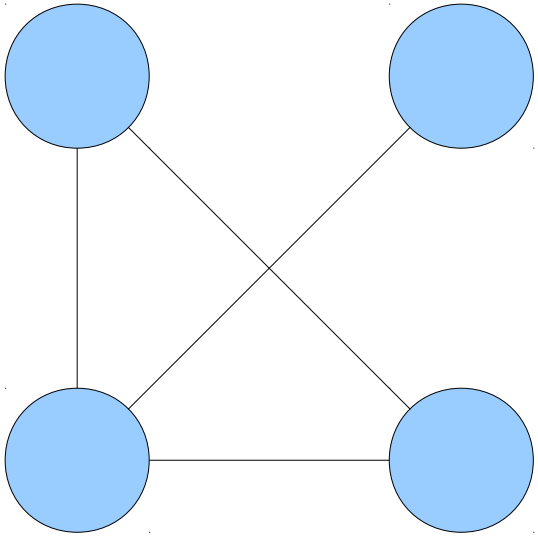
# Navigating a Graph

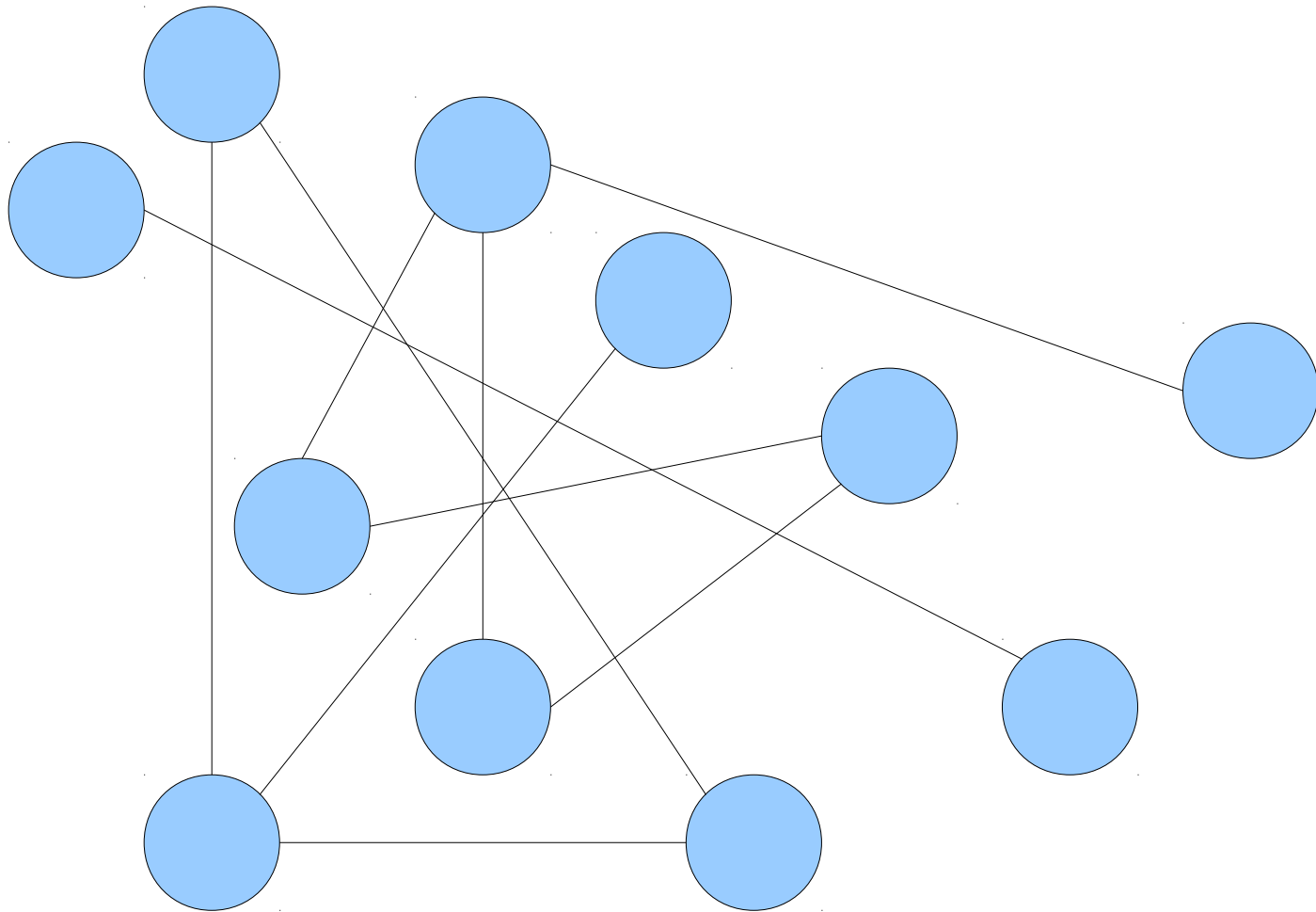


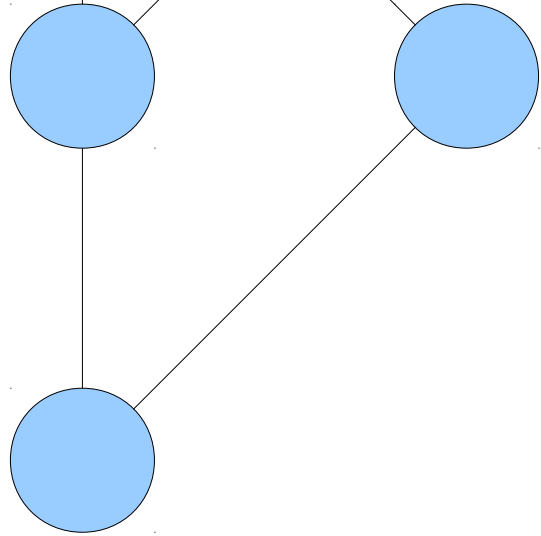
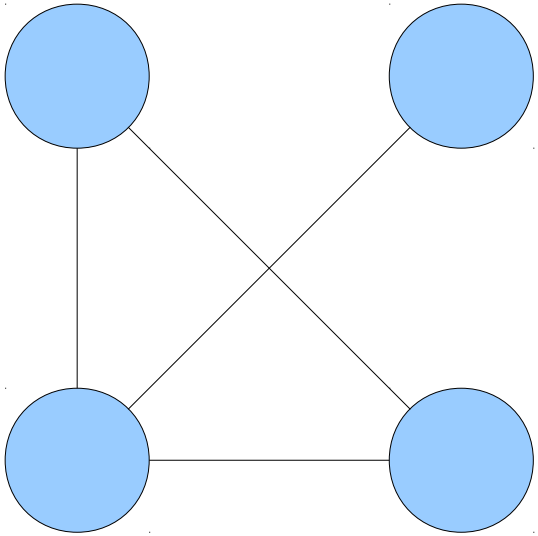
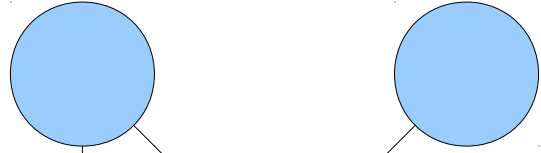
Two nodes  $u$  and  $v$  are called ***connected*** if there is a path from  $u$  to  $v$ .

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

# Connected Components







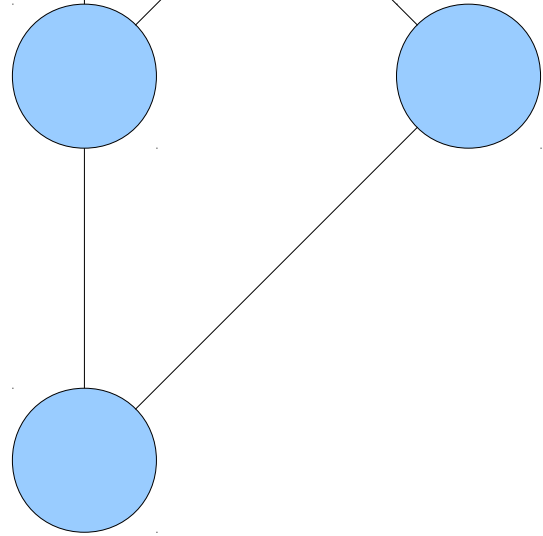
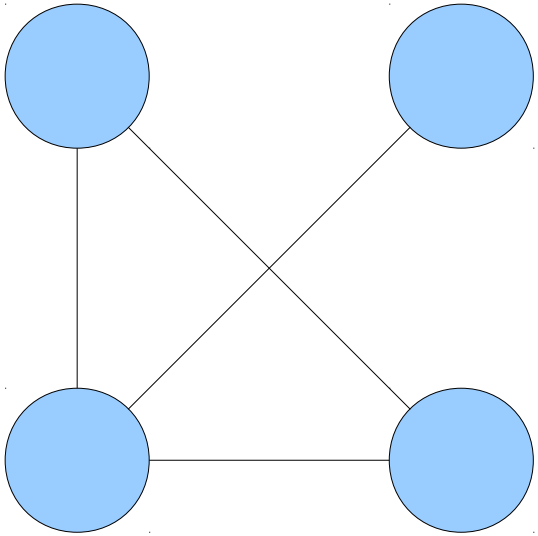
# An Initial Definition

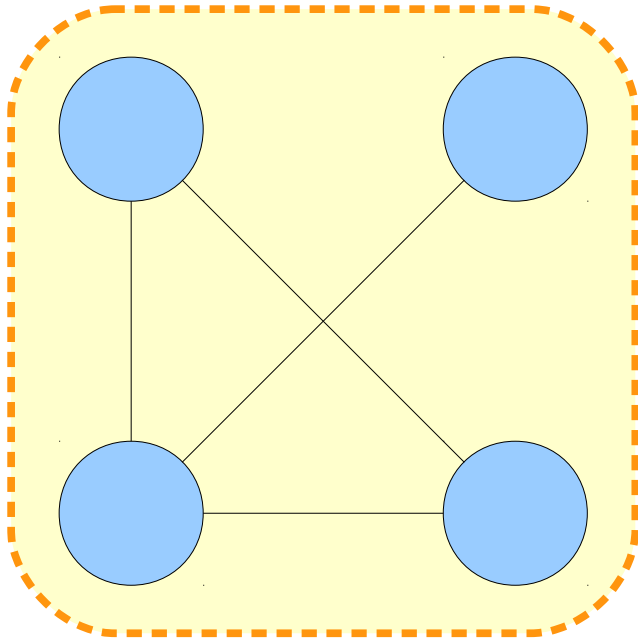
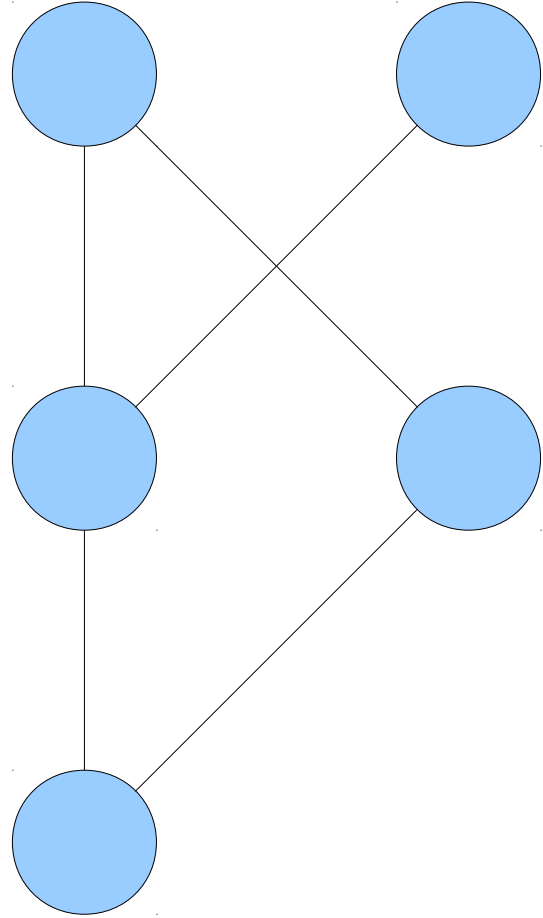
- **Attempted Definition #1:** A *piece* of an undirected graph  $G = (V, E)$  is a set  $C \subseteq V$  where

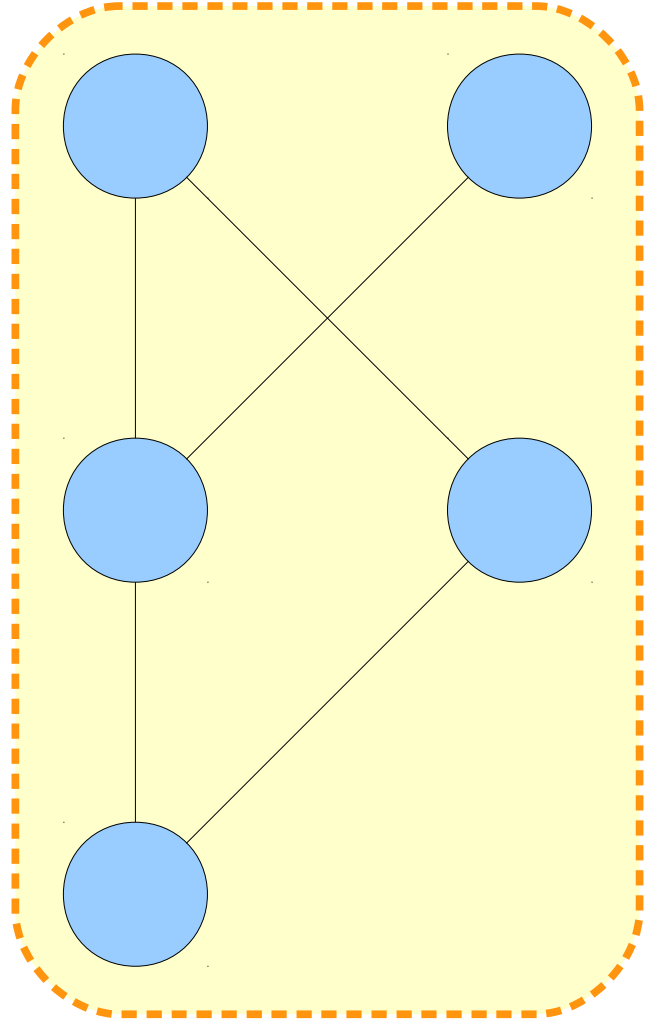
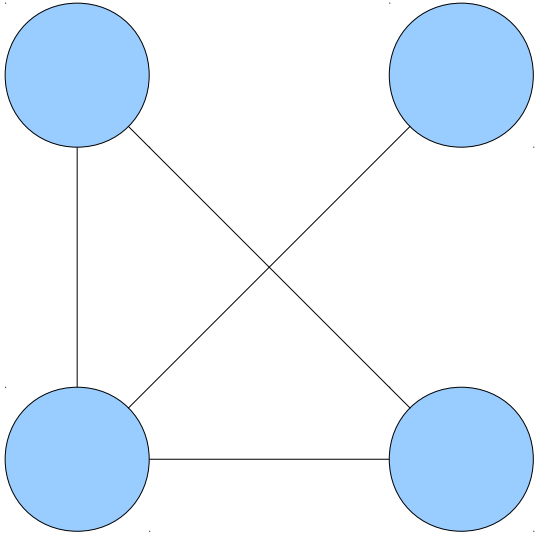
$$\forall u \in C. \forall v \in C. \text{Connected}(u, v)$$

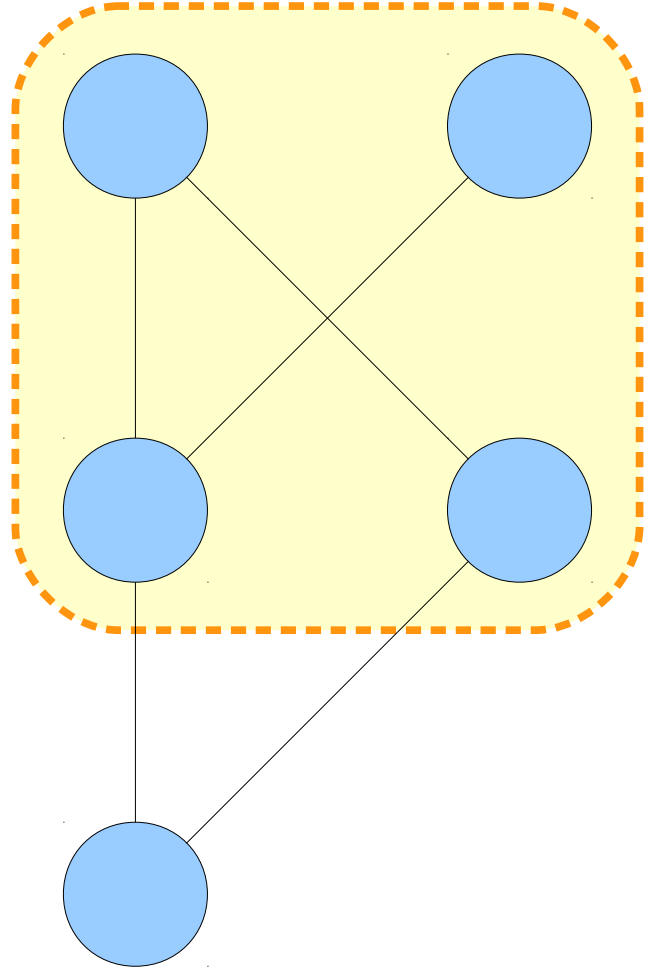
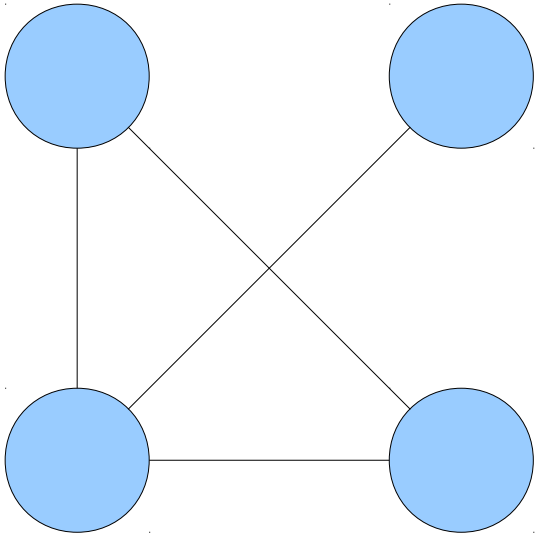
- Intuition: a piece of a graph is a set of nodes that are all connected to one another.

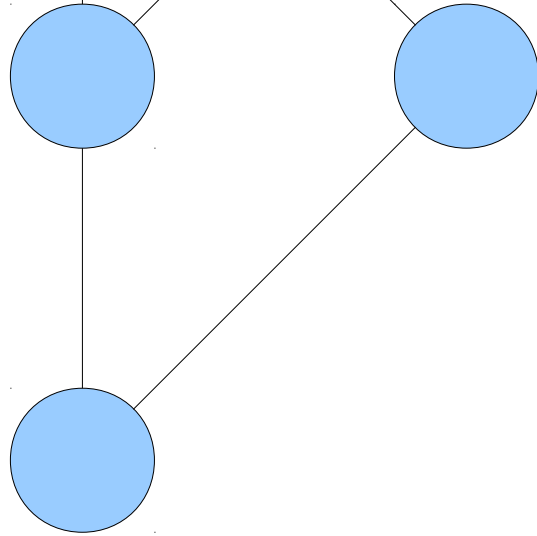
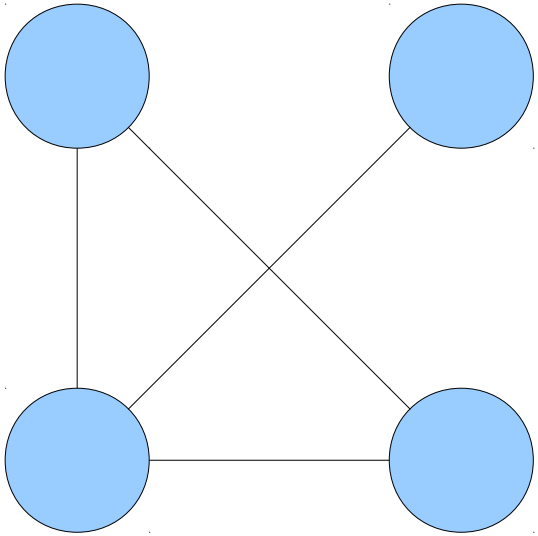
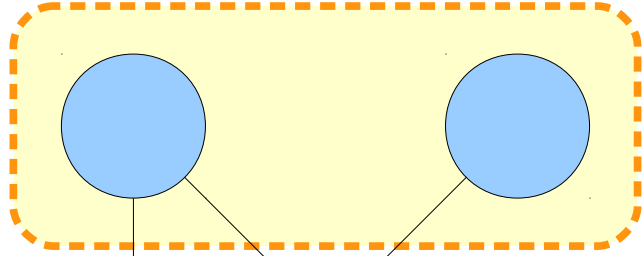
⚠ *This definition has some problems; ⚠  
please don't use it as a reference.*

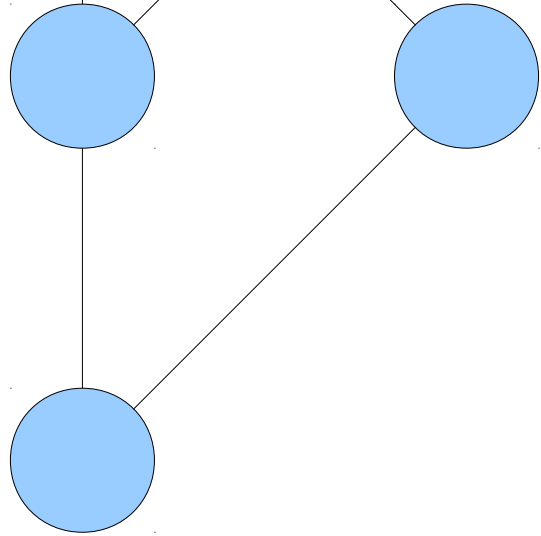
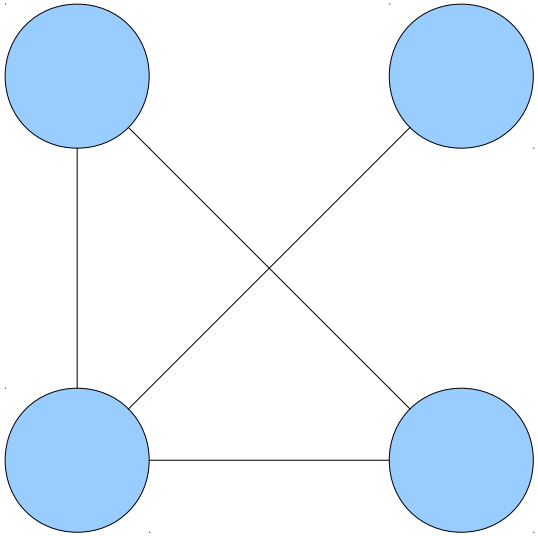
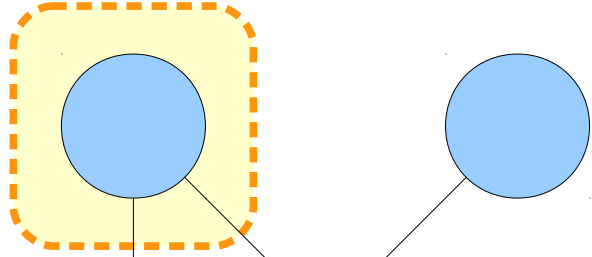






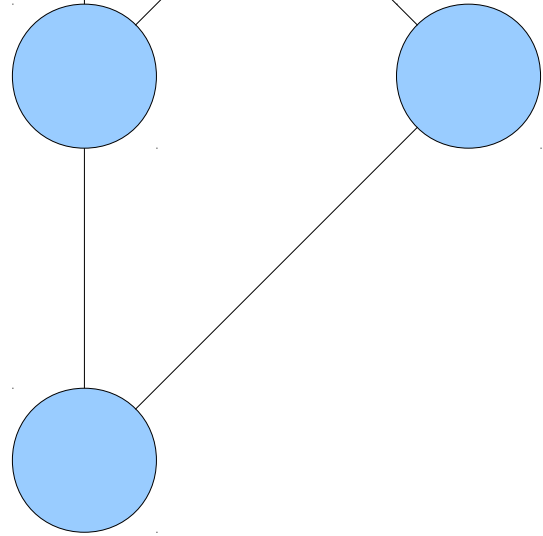
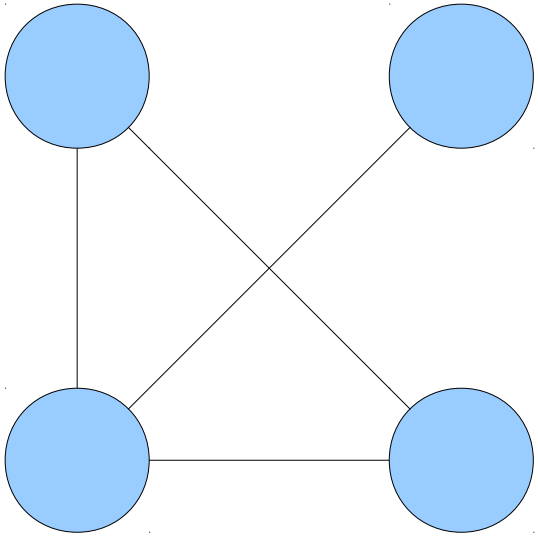


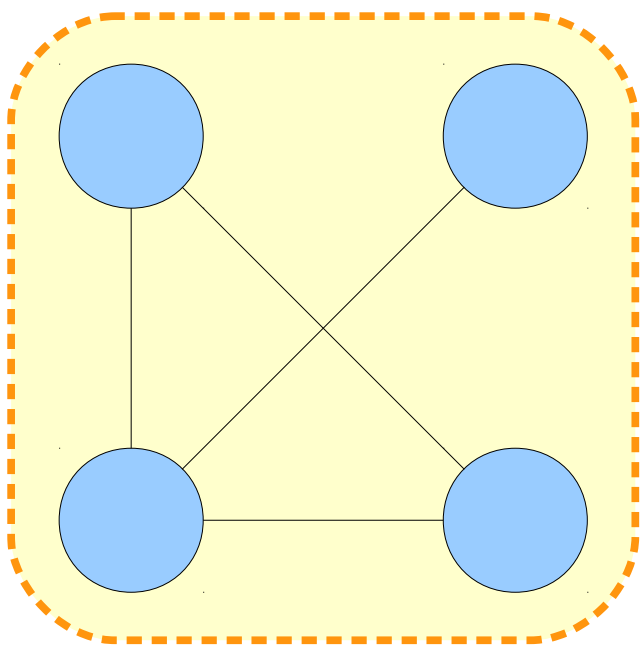
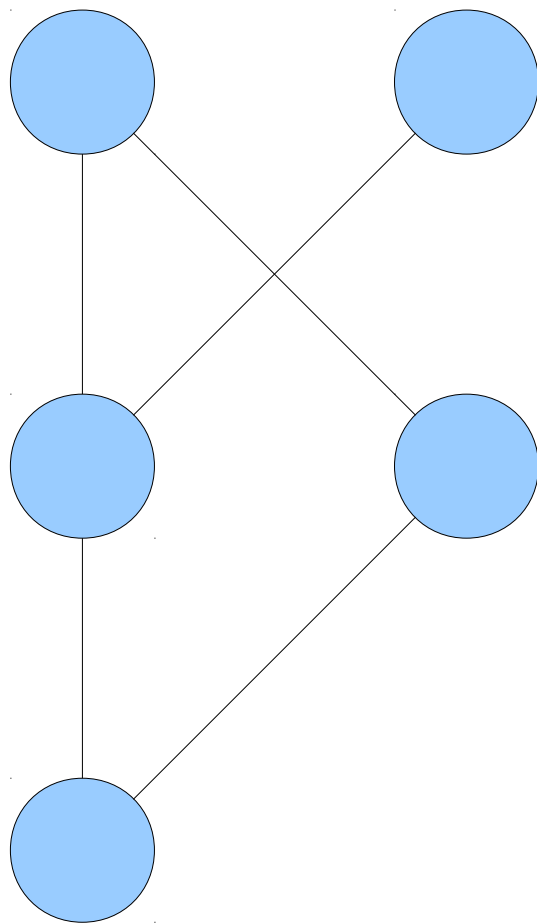


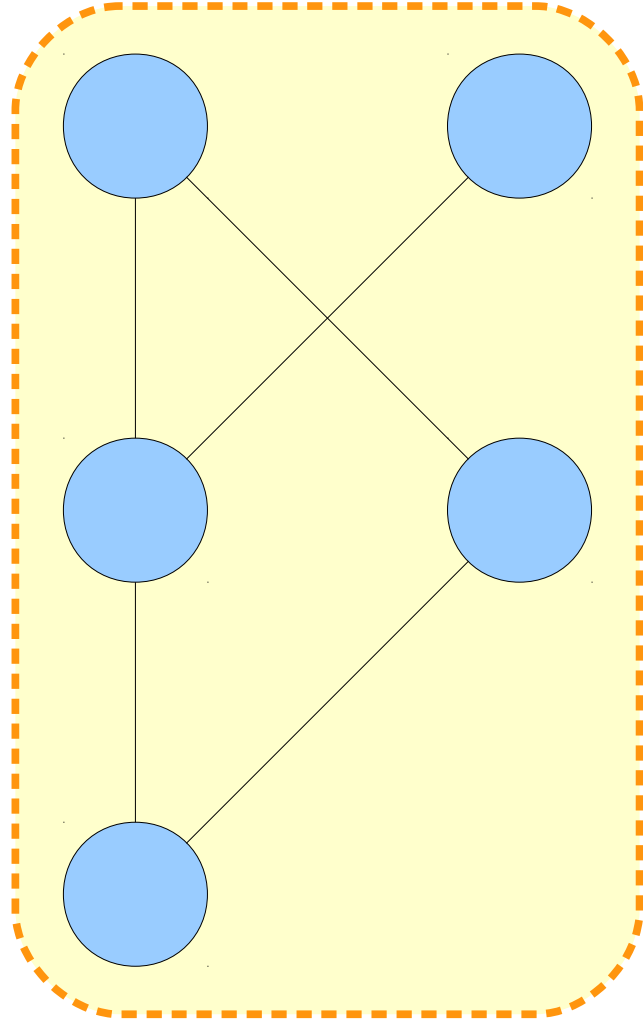
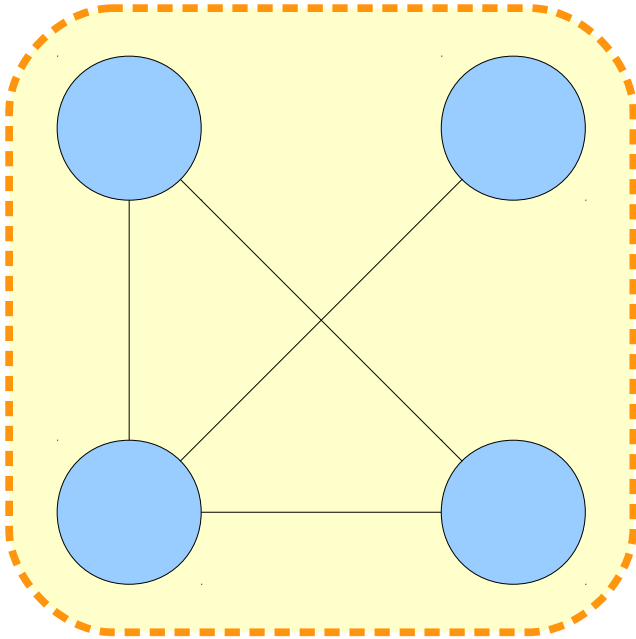


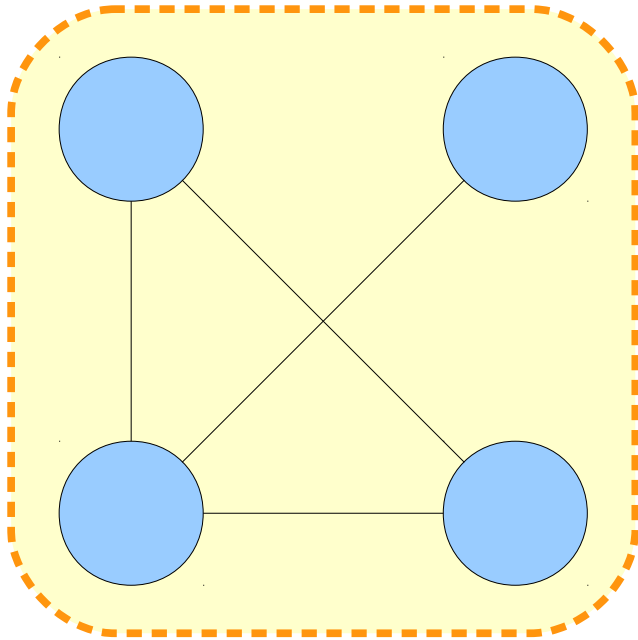
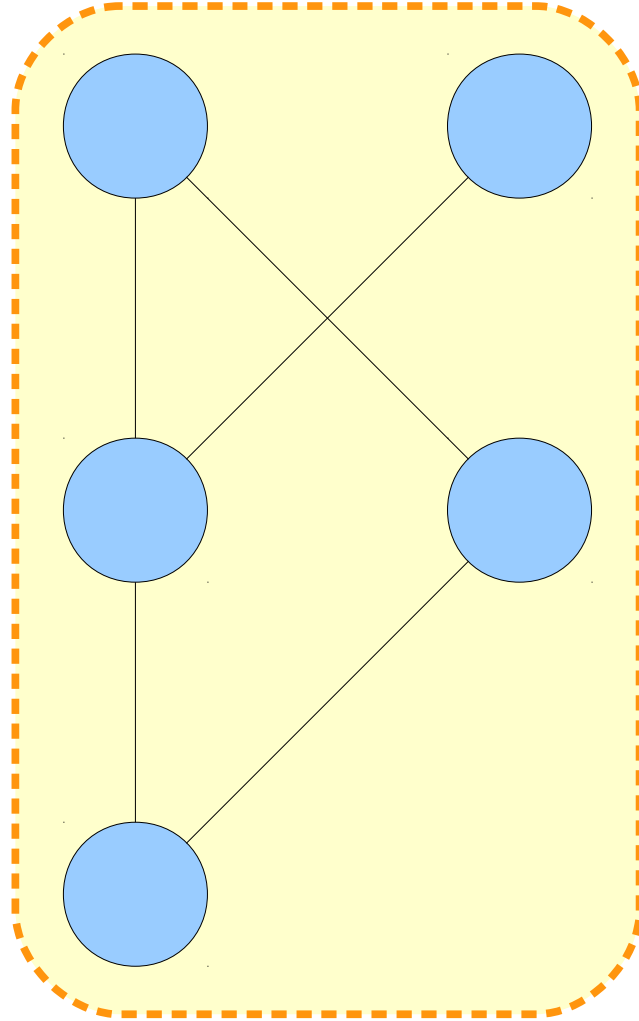
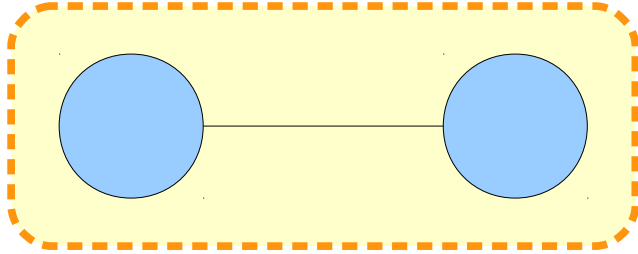
# An Updated Definition

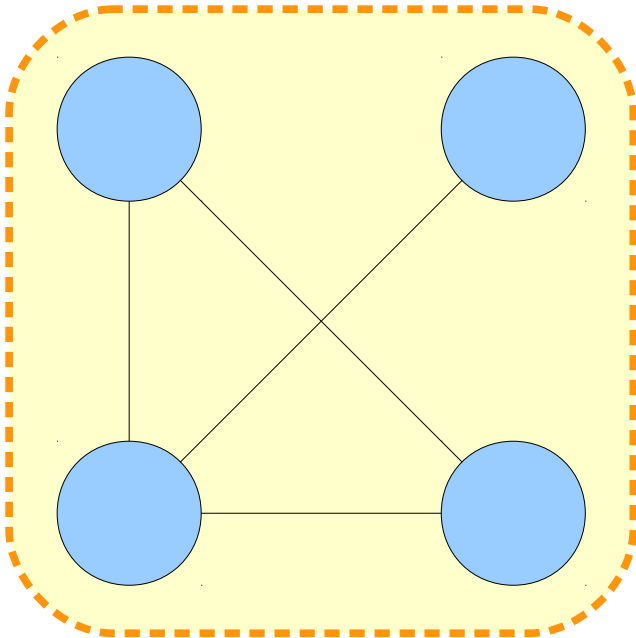
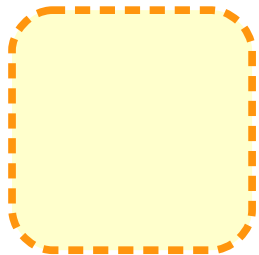
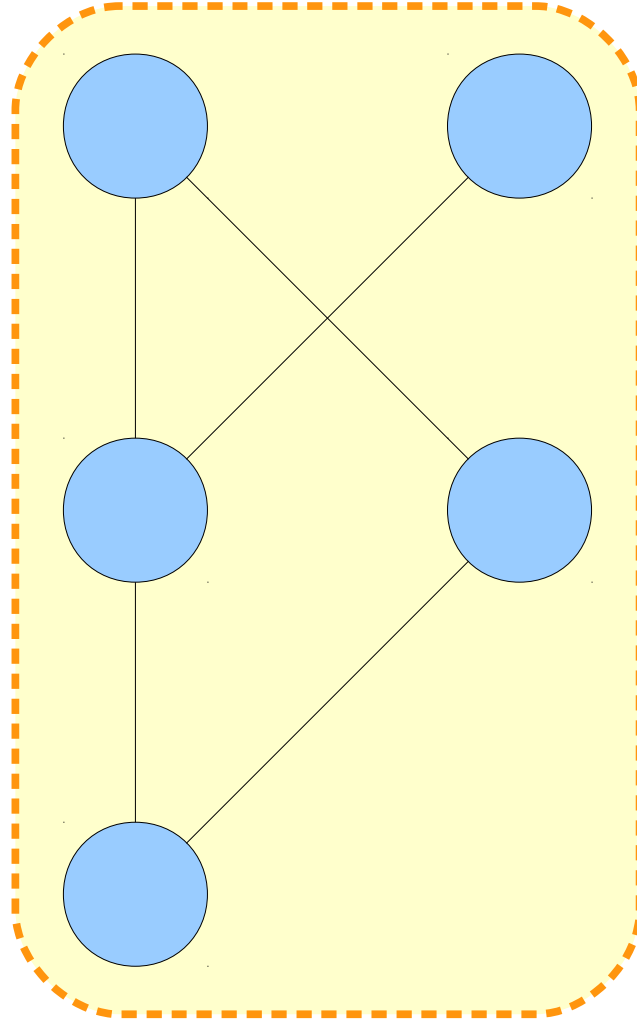
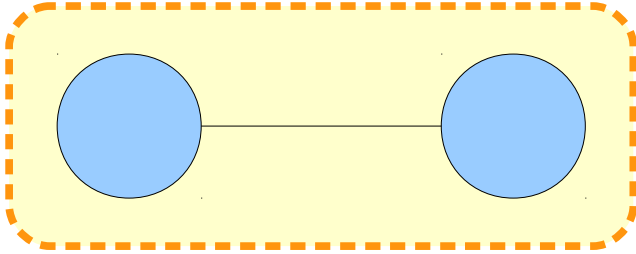
- **Attempted Definition #2:** A *piece* of an undirected graph  $G = (V, E)$  is a set  $C \subseteq V$  where
  - $\forall u \in C. \forall v \in C. \text{Connected}(u, v)$
  - $\forall u \in C. \forall v \in V - C. \neg \text{Connected}(u, v)$
- Intuition: a piece of a graph is a set of nodes that are all connected to one another that doesn't “miss” any nodes.
  - △ *This definition has some problems; △ please don't use it as a reference.*

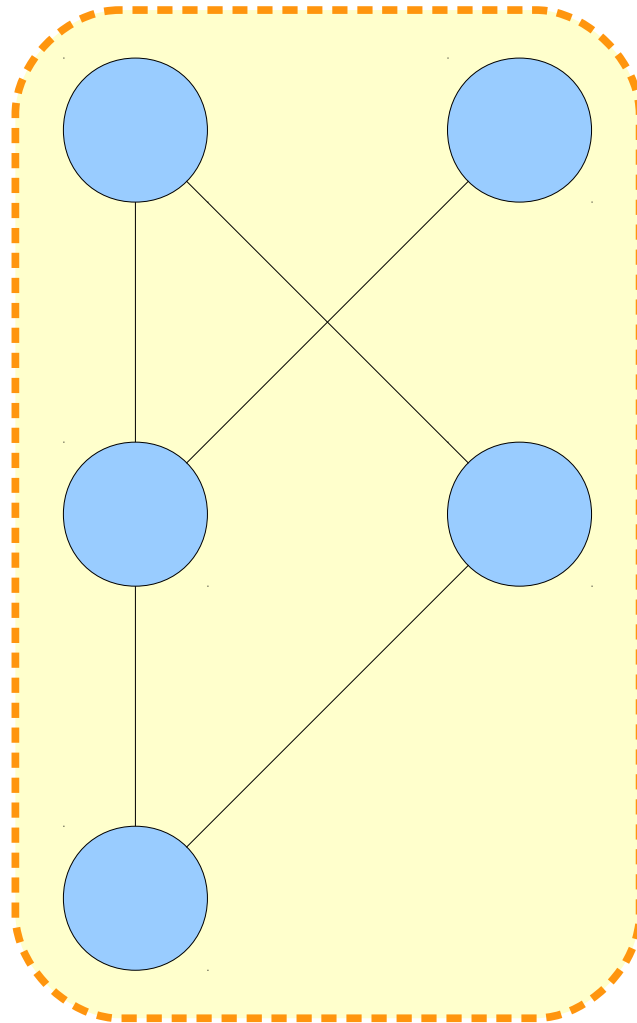
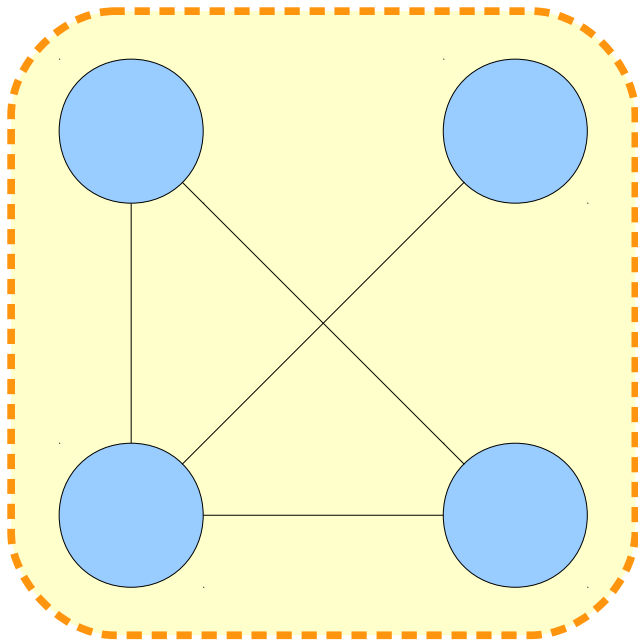
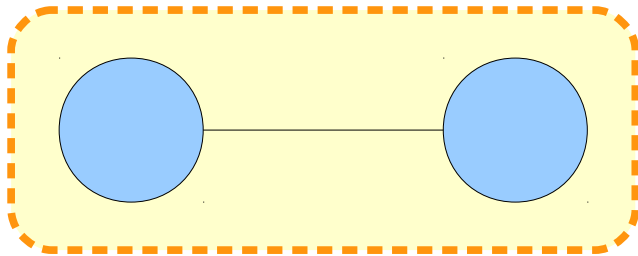
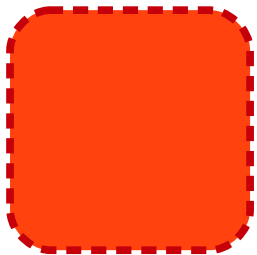








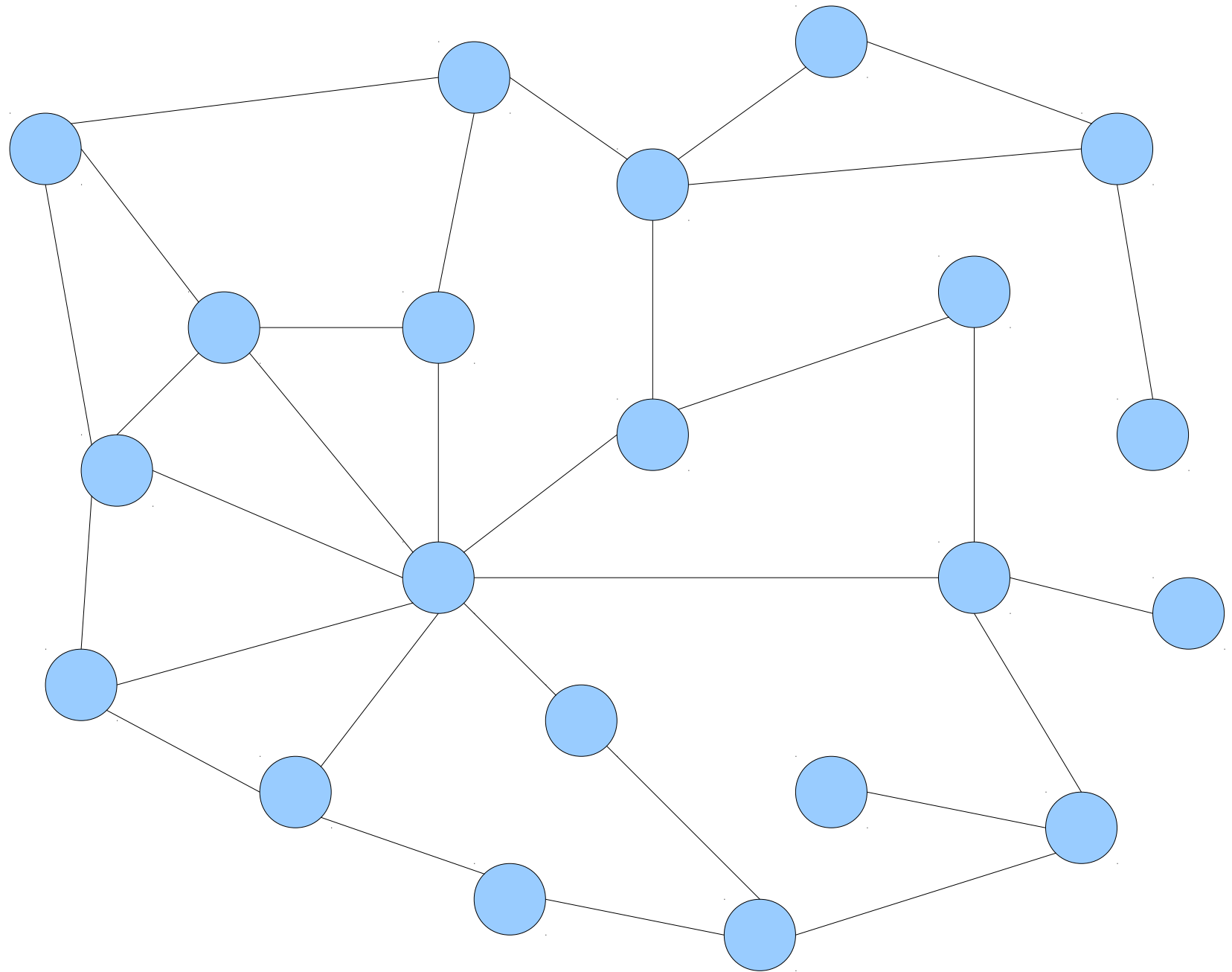


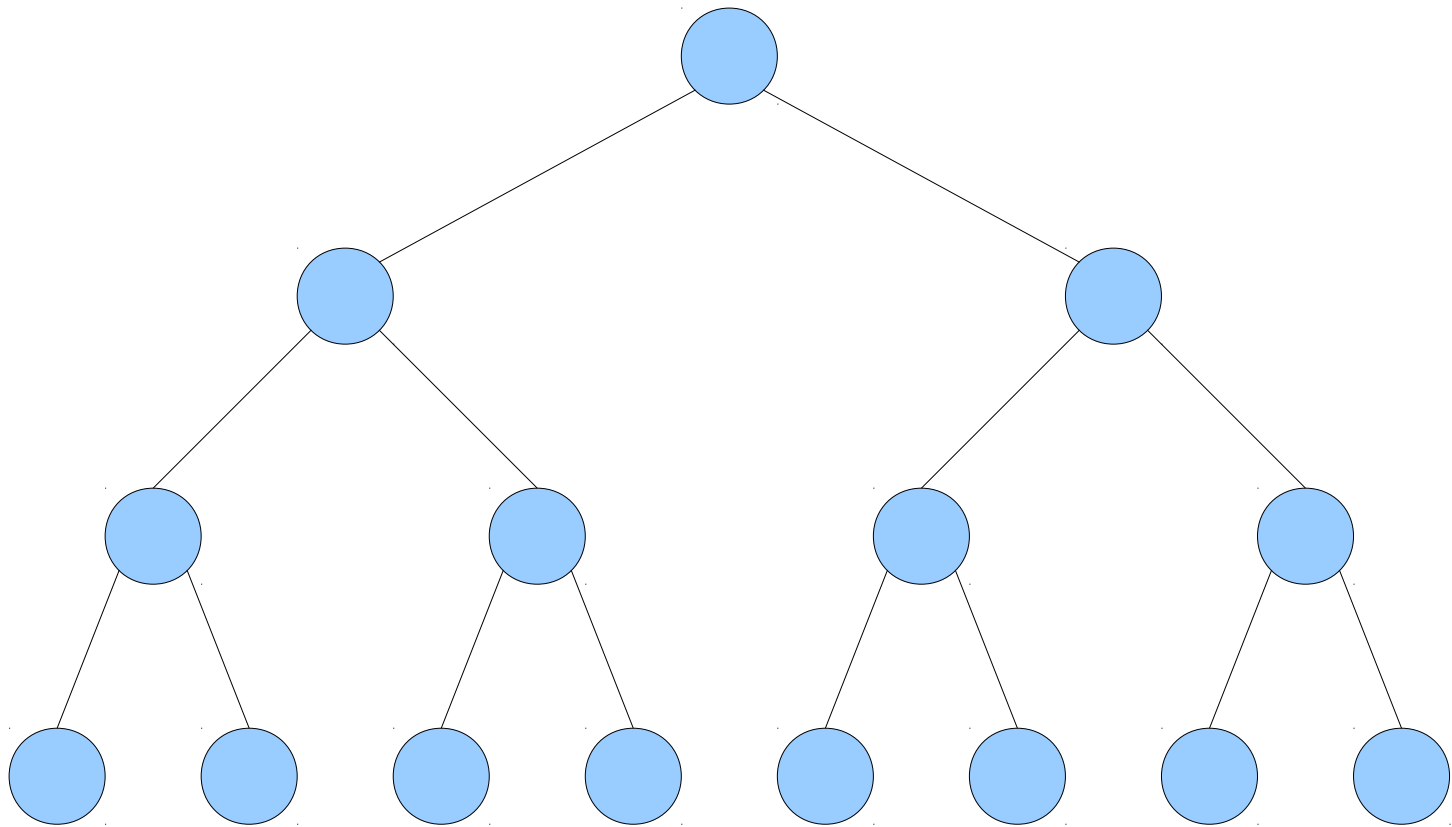


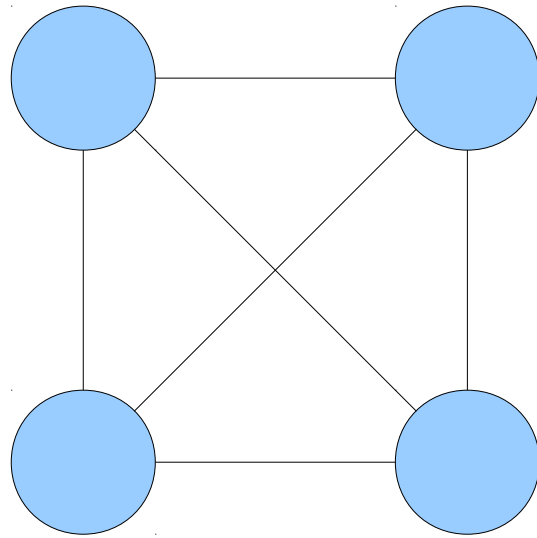
# A Final Definition

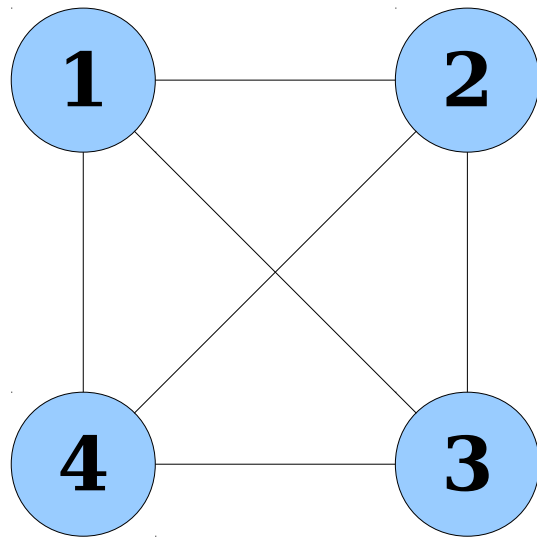
- **Definition:** A ***connected component*** of an undirected graph  $G = (V, E)$  is a *nonempty* set  $C \subseteq V$  where
  - $\forall u \in C. \forall v \in C. \text{Connected}(u, v)$
  - $\forall u \in C. \forall v \in V - C. \neg \text{Connected}(u, v)$
- Intuition: a connected component is a nonempty set of nodes that are all connected to one another that includes as many nodes as possible.

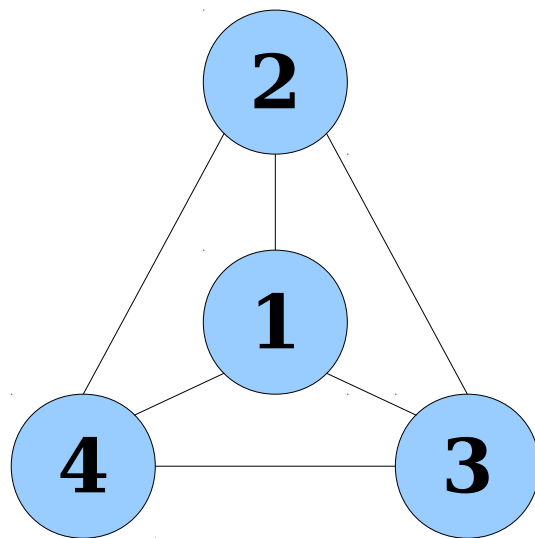
# Planar Graphs

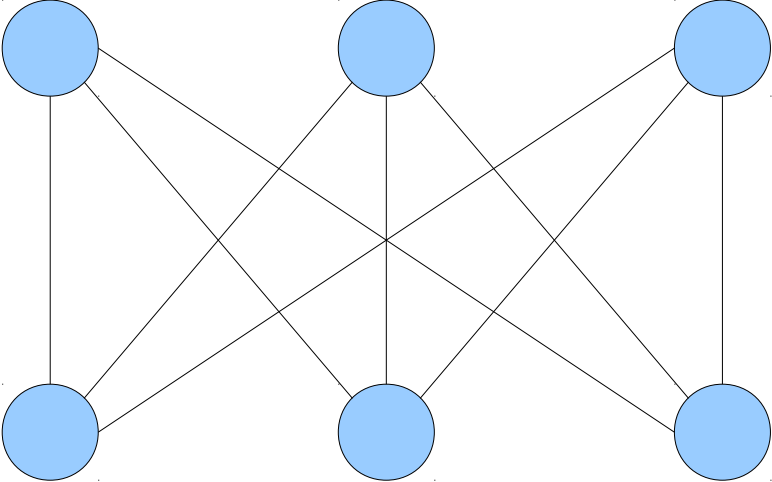


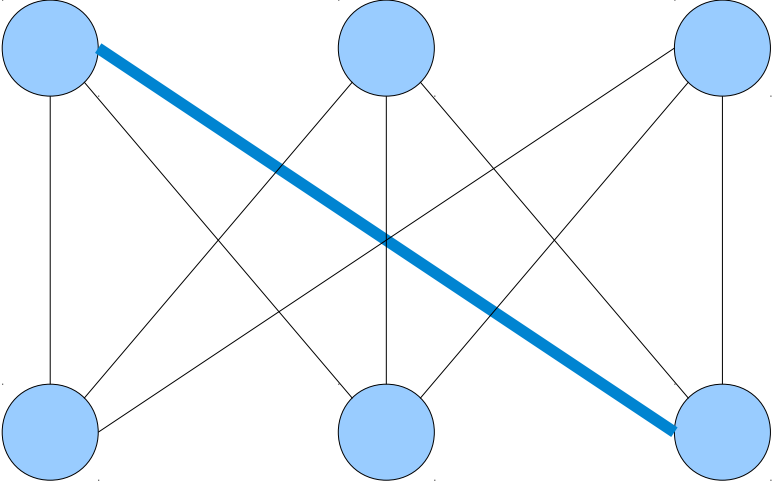


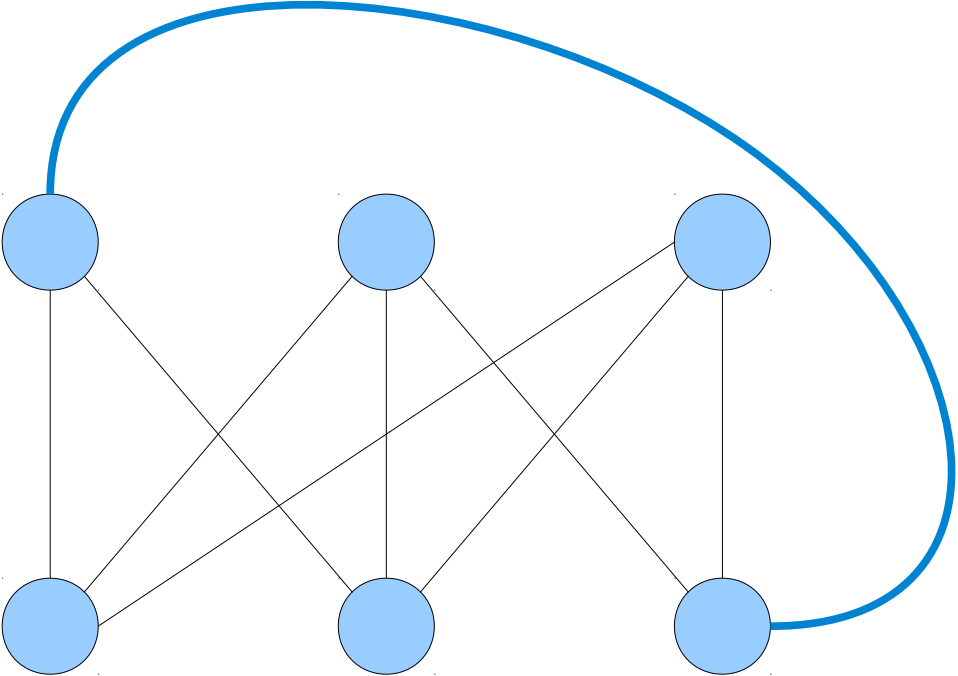


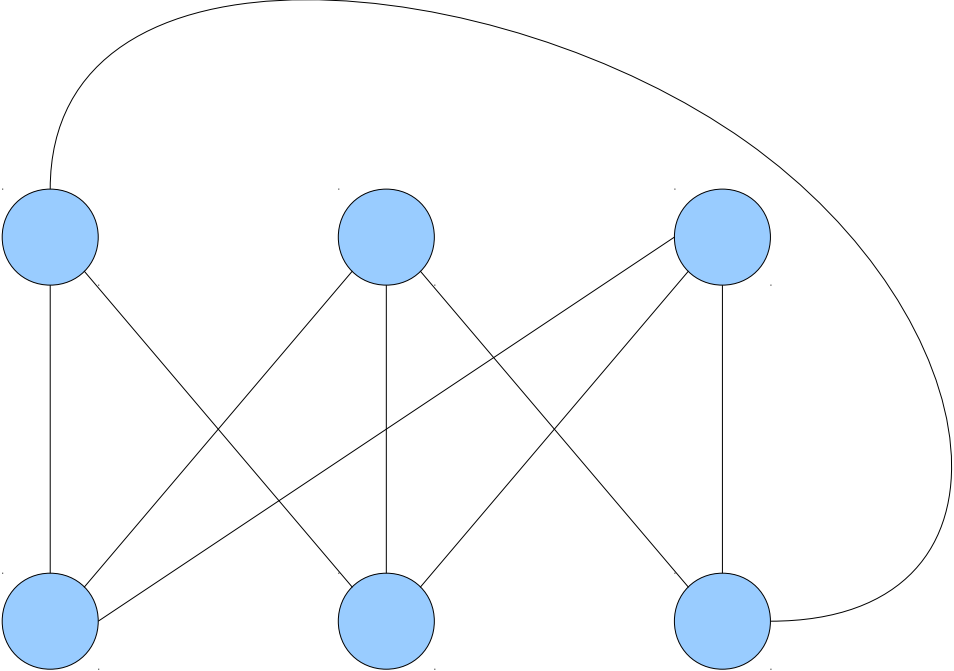


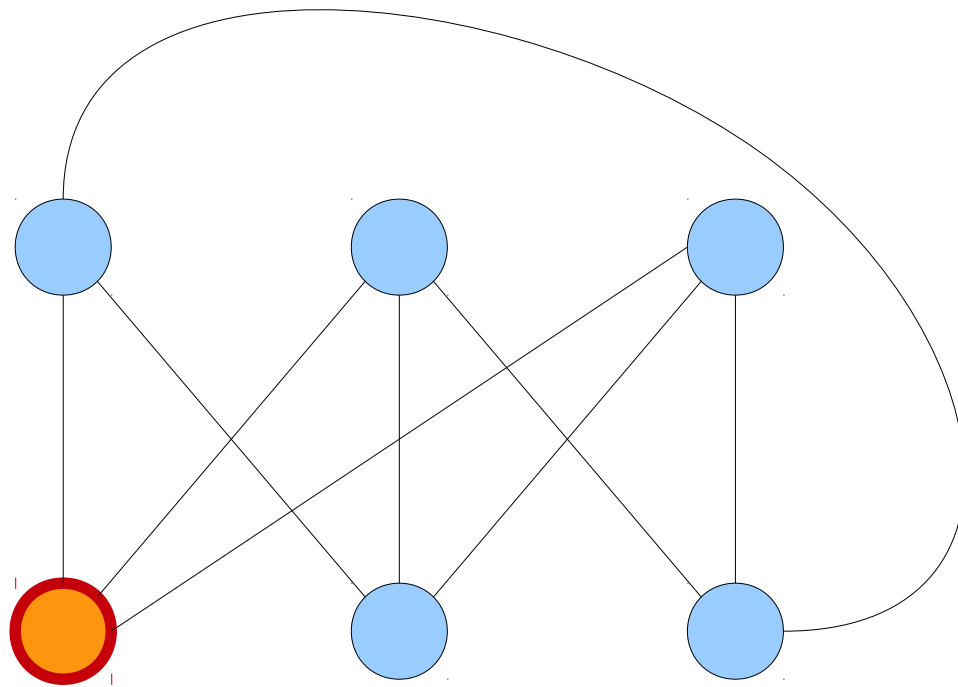


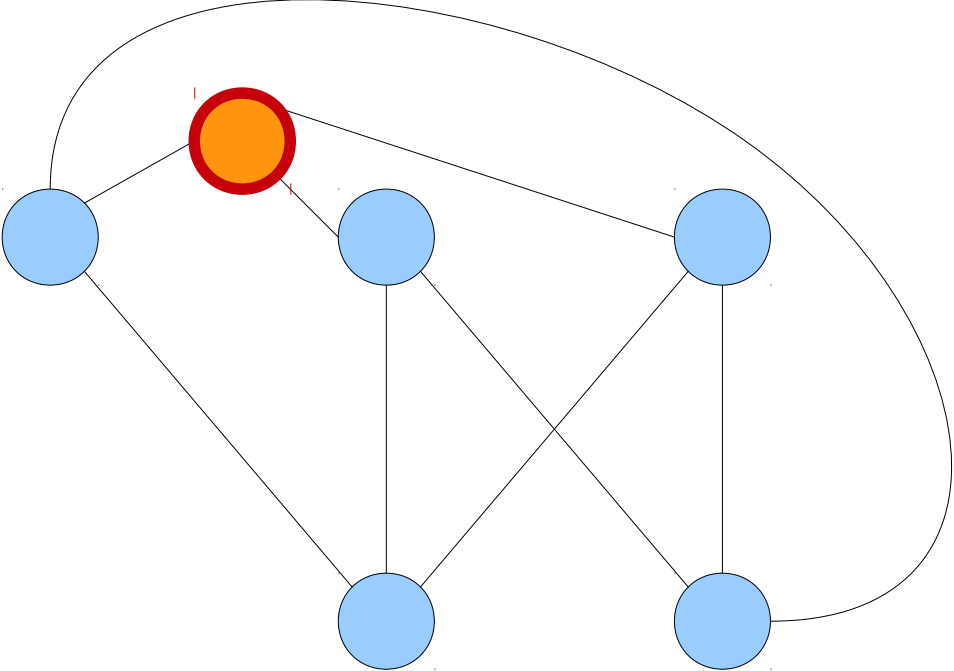


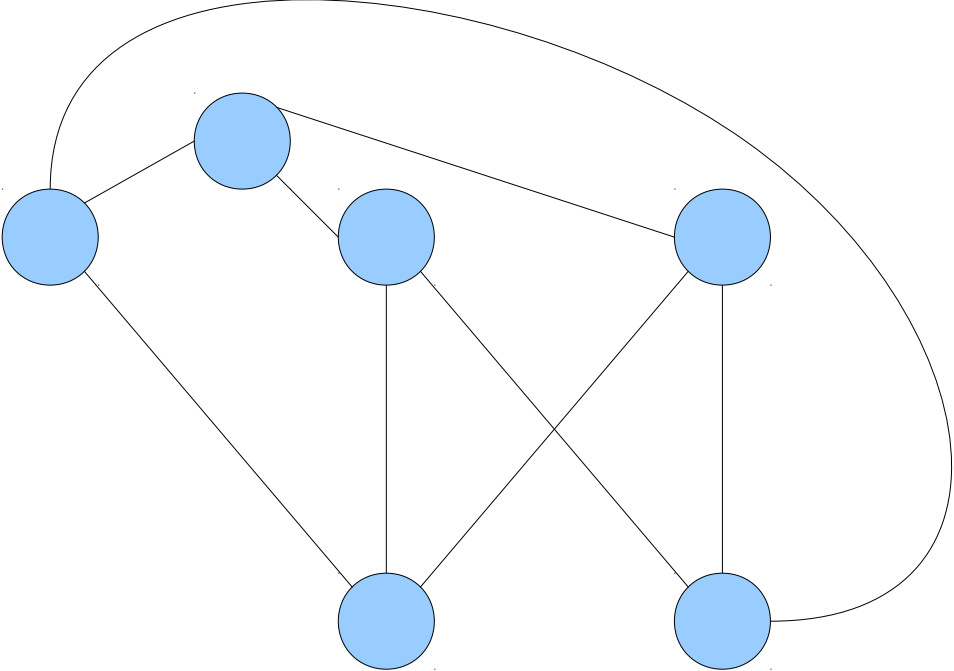


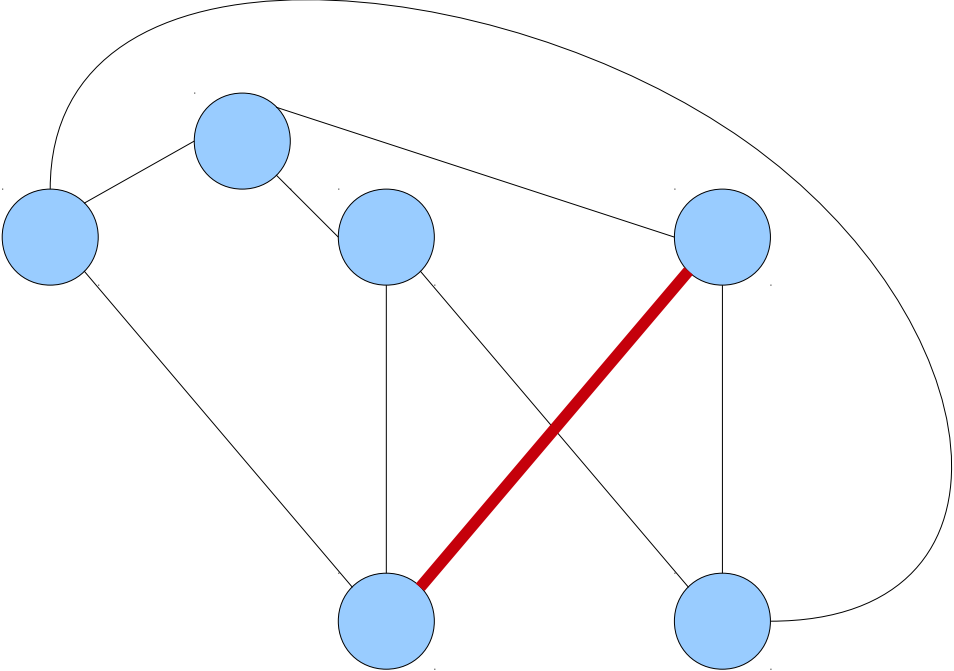


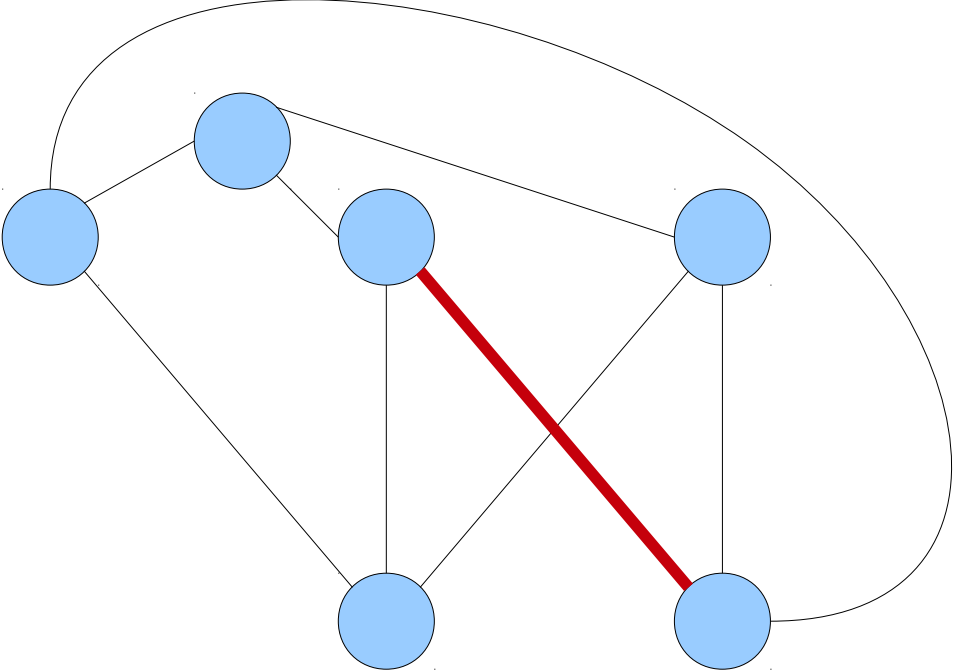


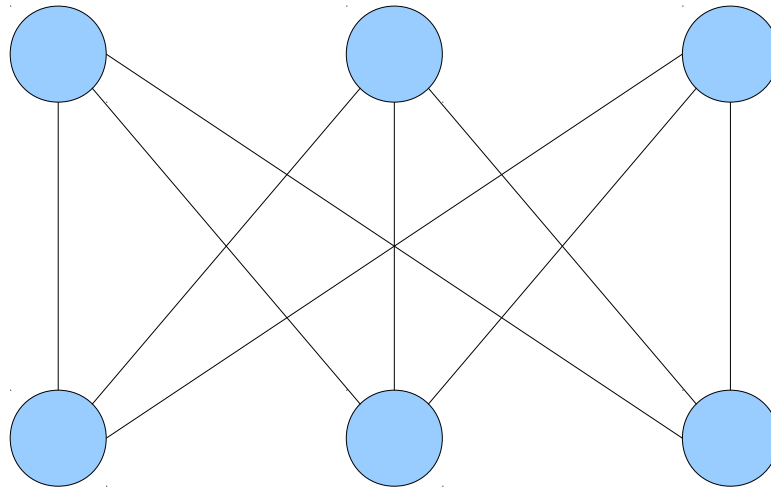








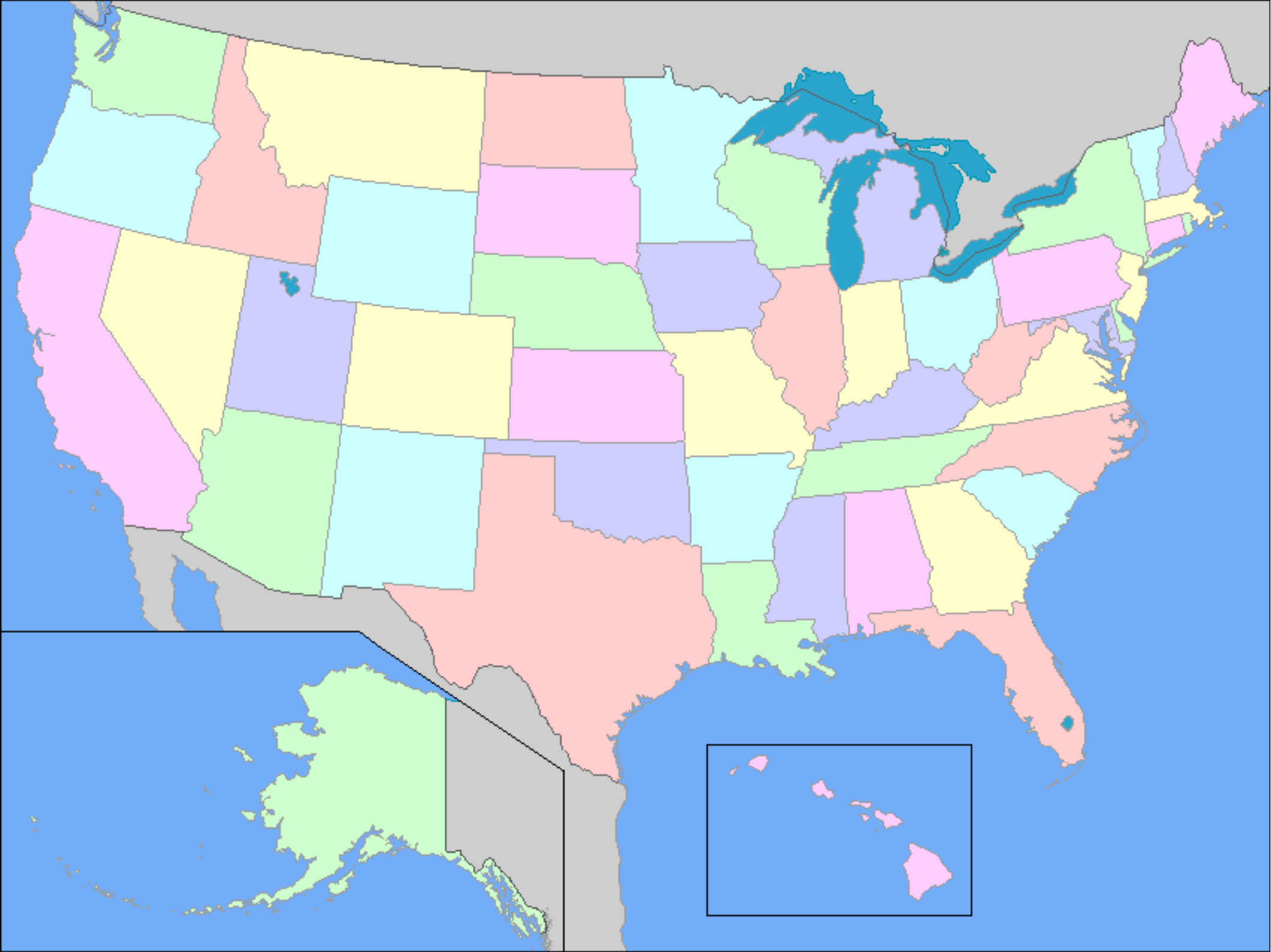


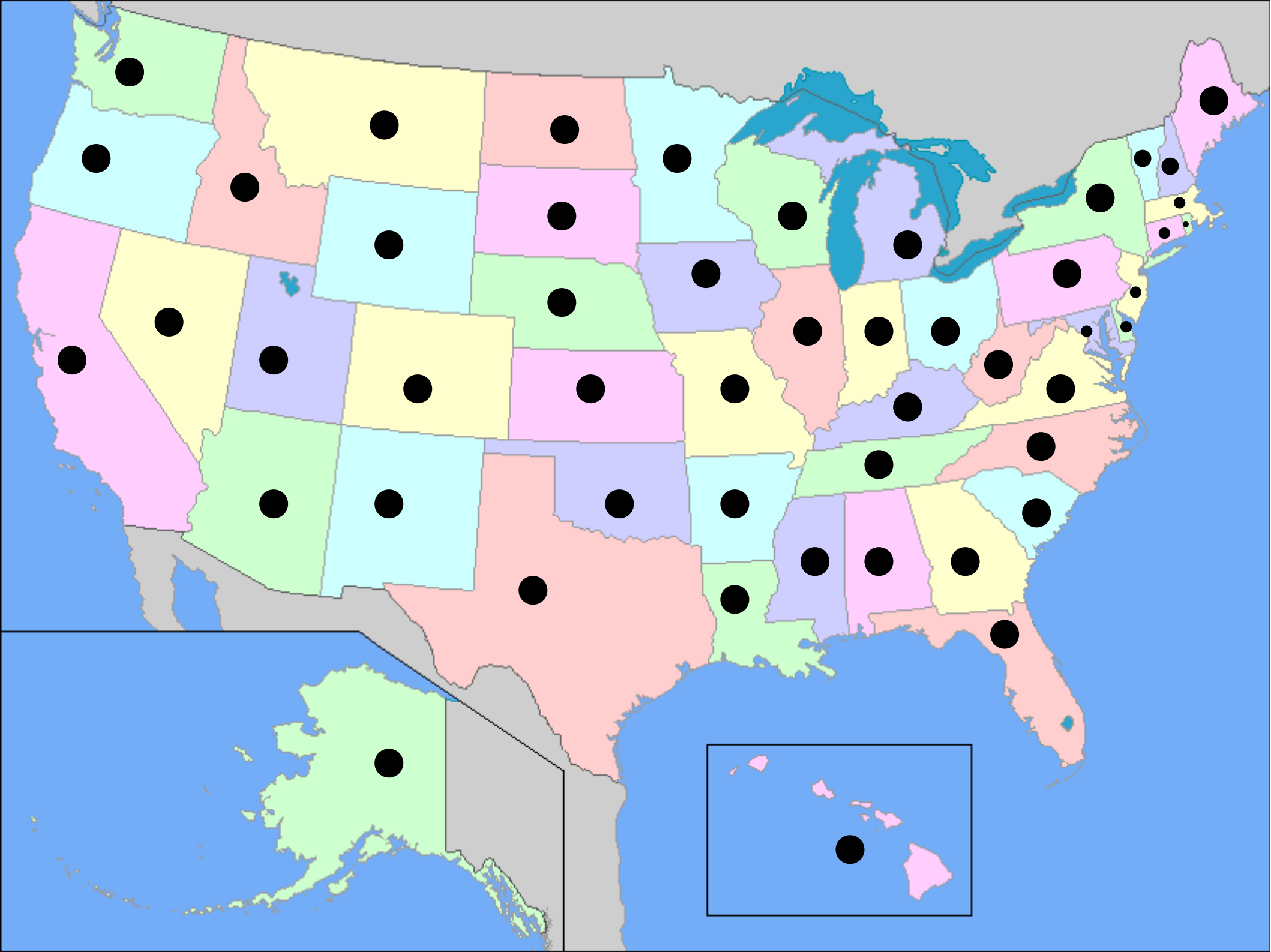


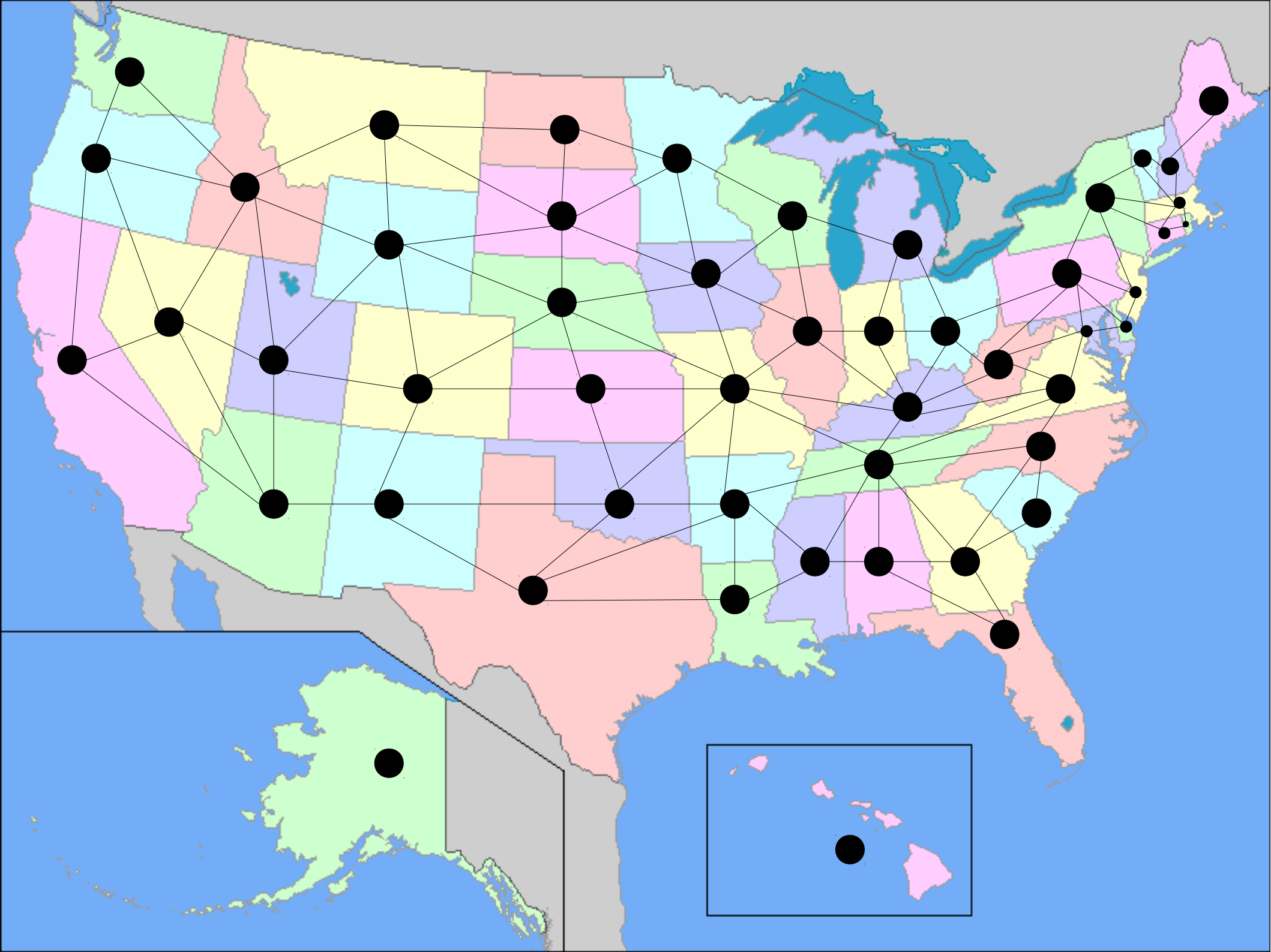
This graph is called the ***utility graph***. There is no way to draw it in the plane without edges crossing.

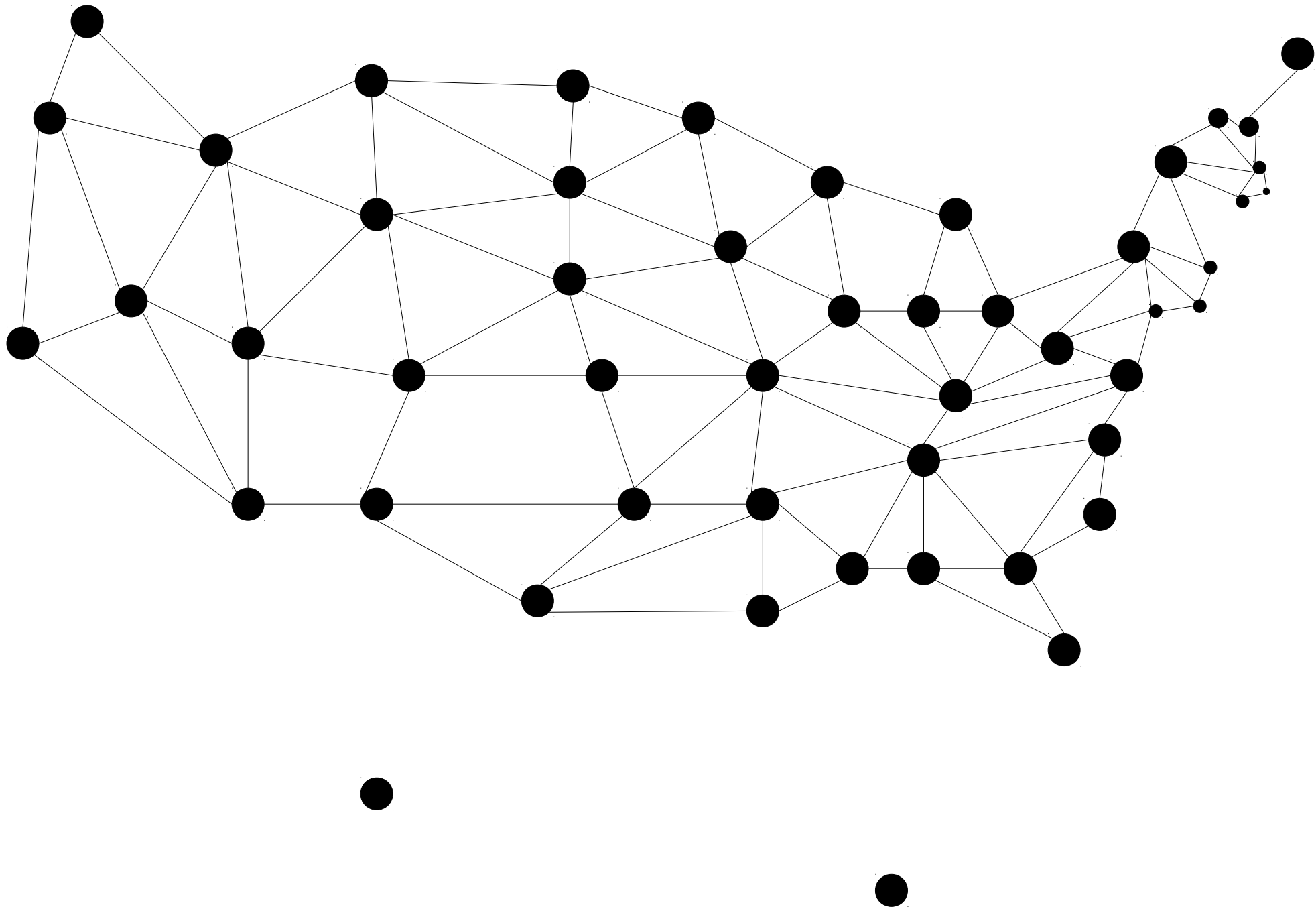
A graph is called a ***planar graph*** if there is some way to draw it in a 2D plane without any of the edges crossing.

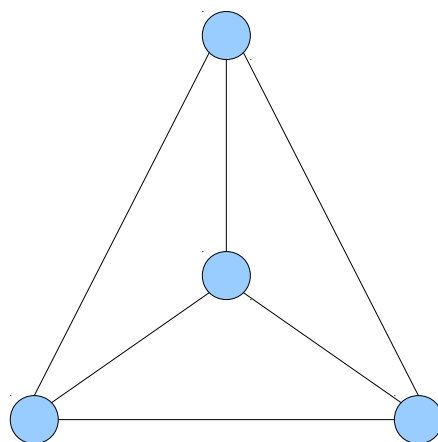
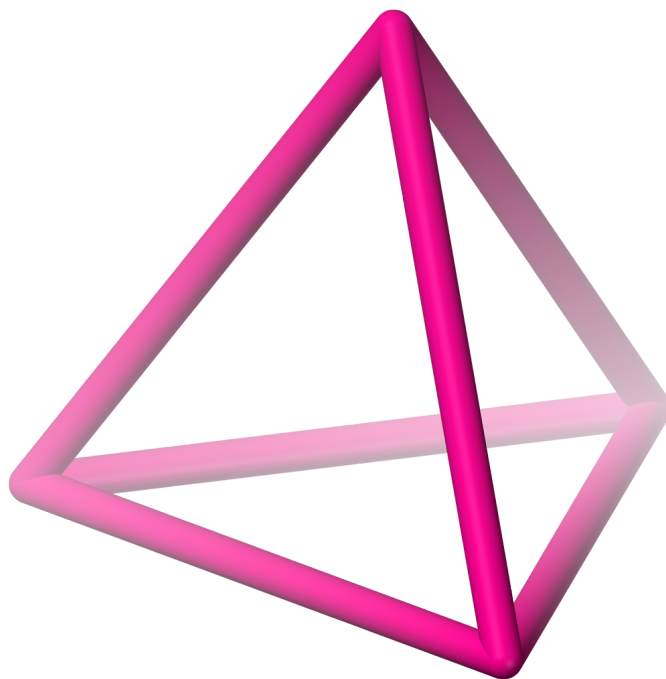
A Fun (And Strangely Addicting) Game:  
<http://planarity.net/>

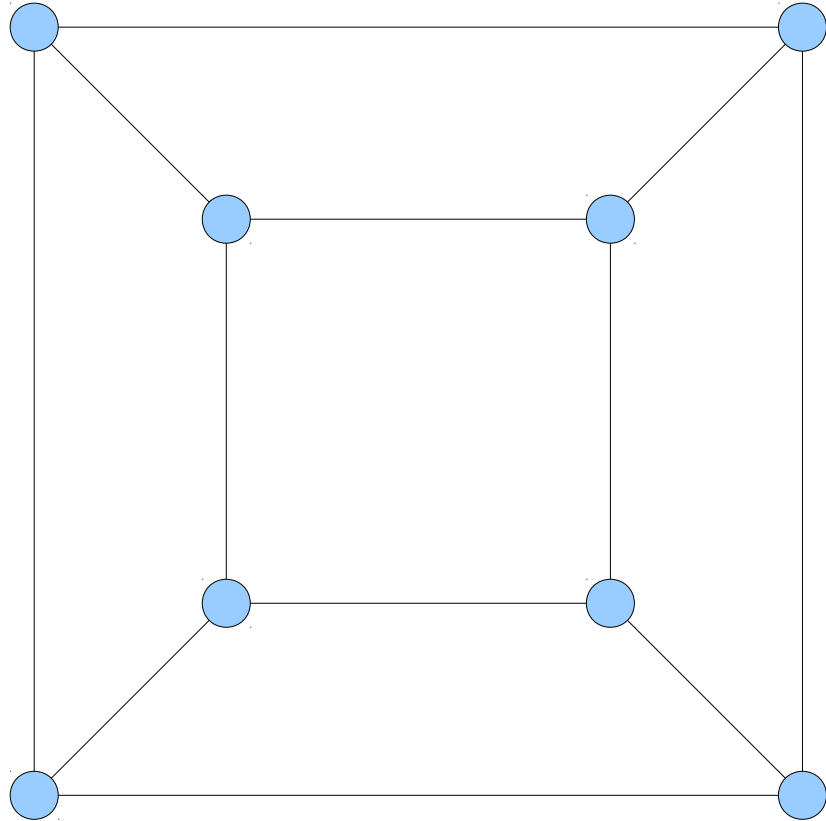
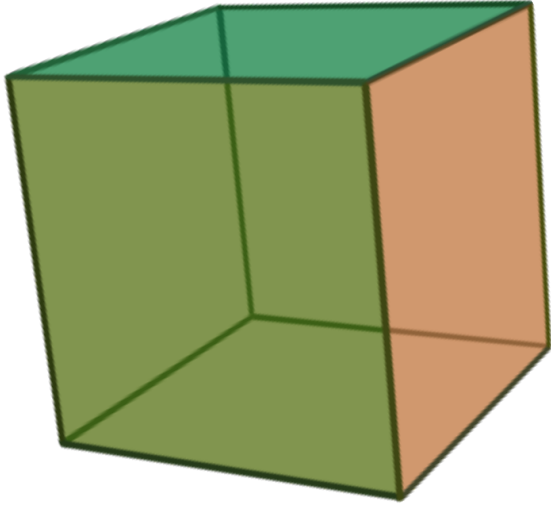


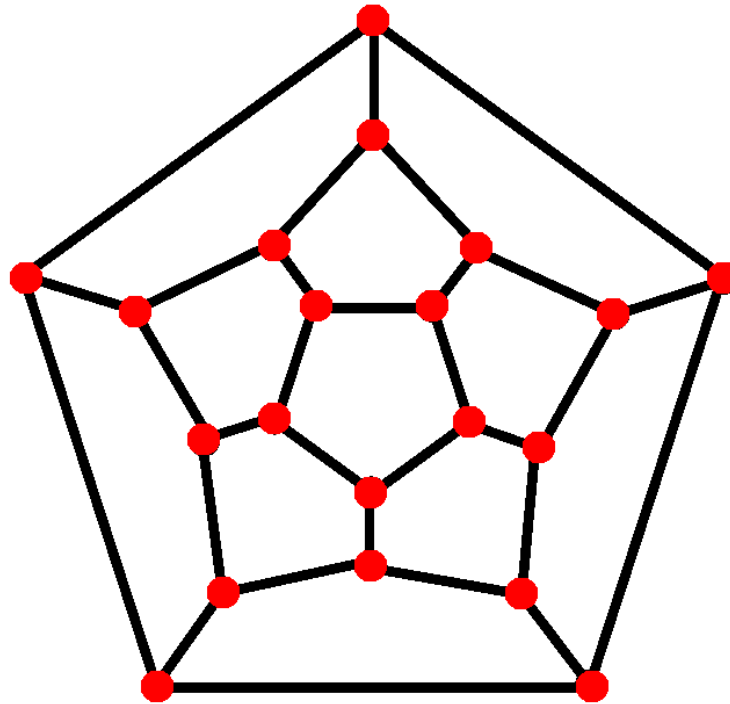
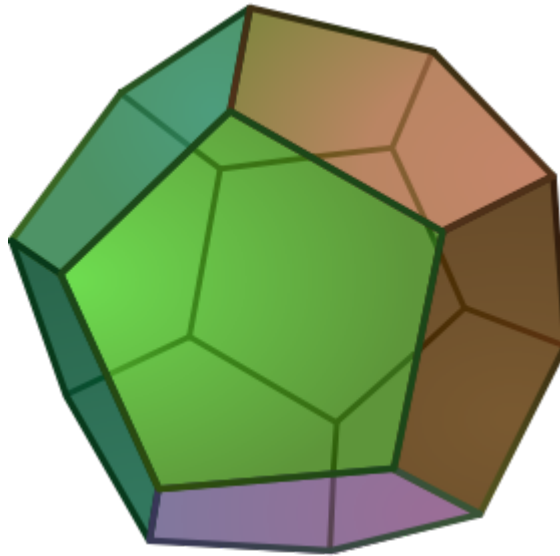




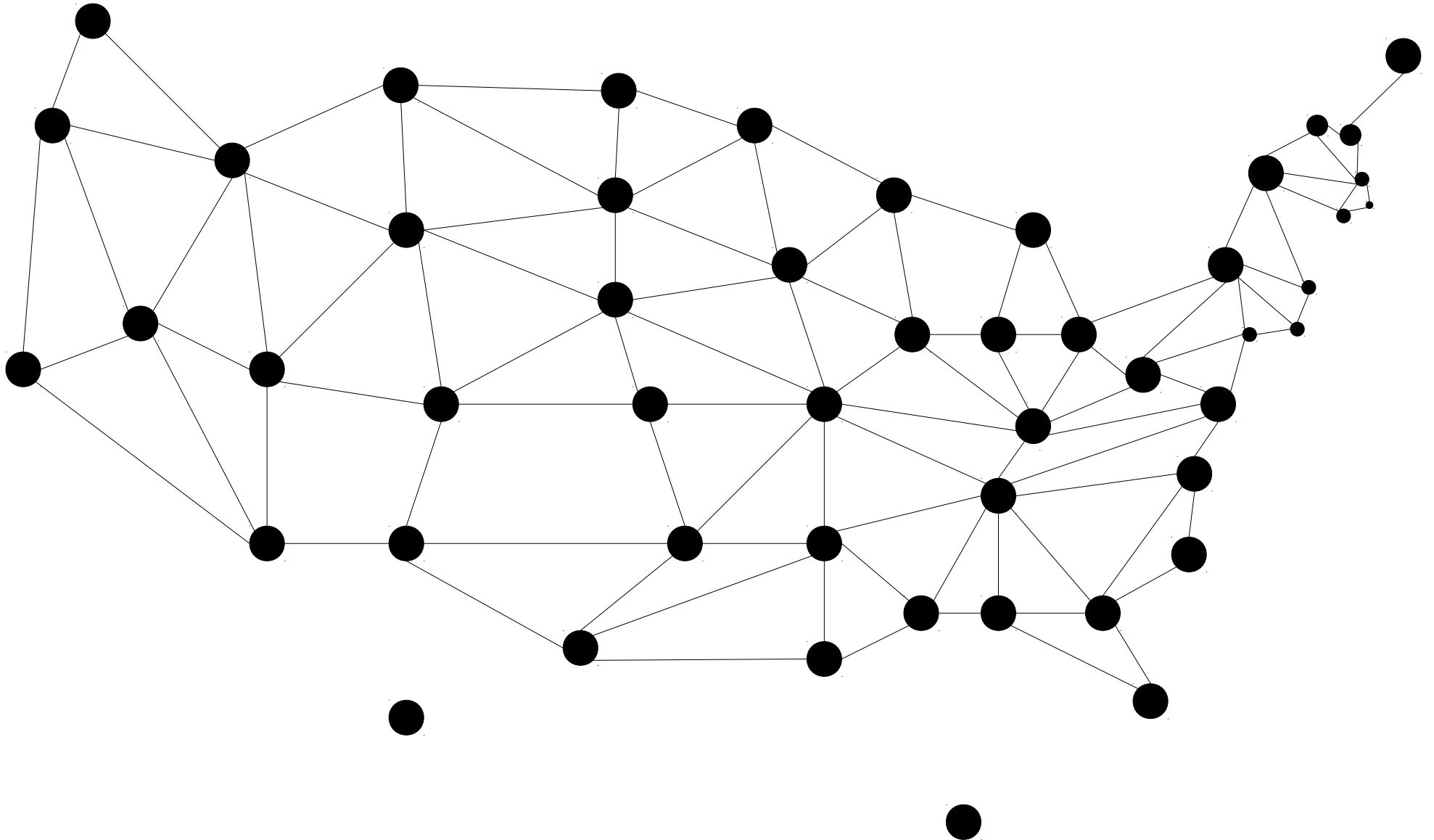




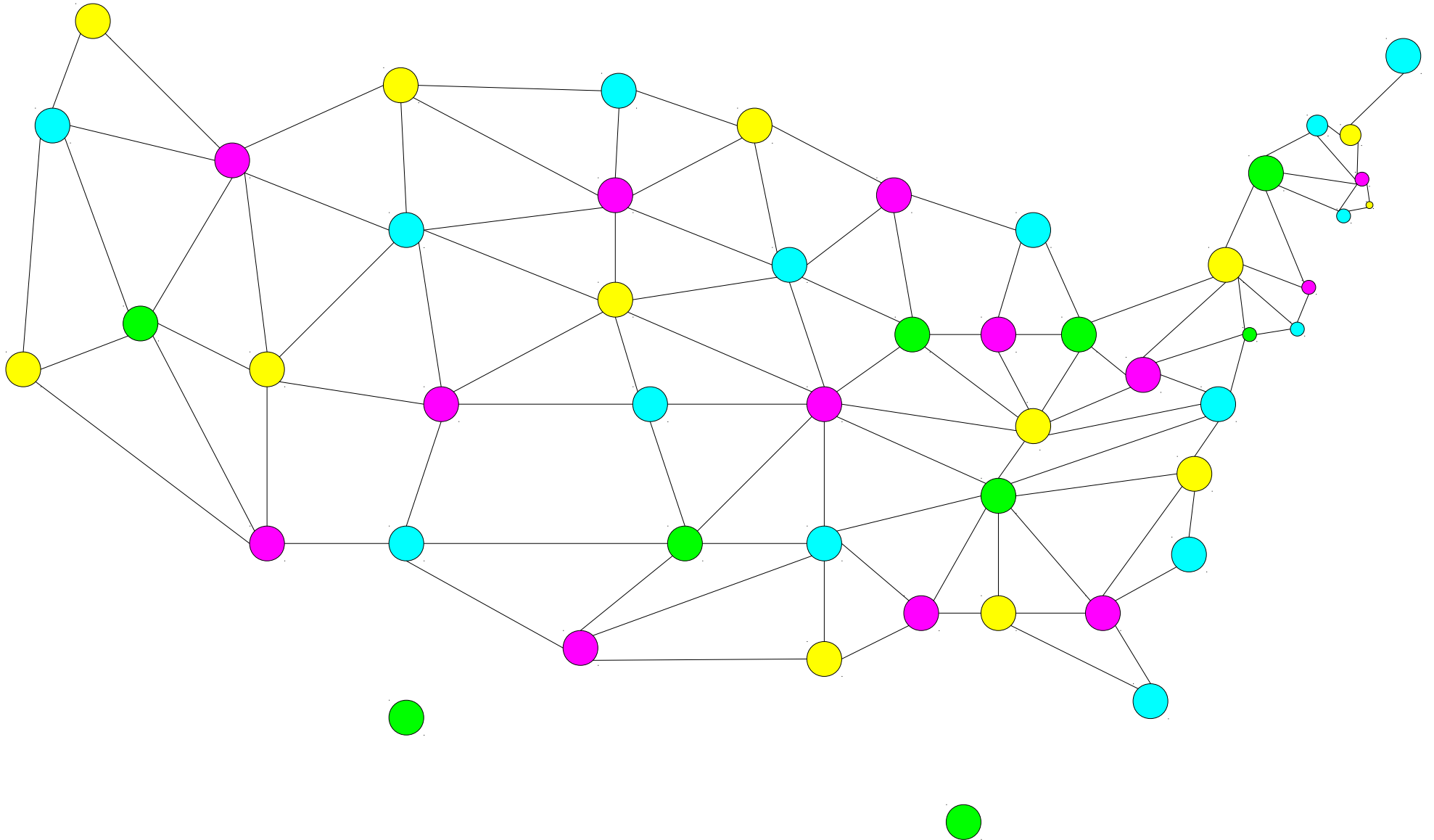




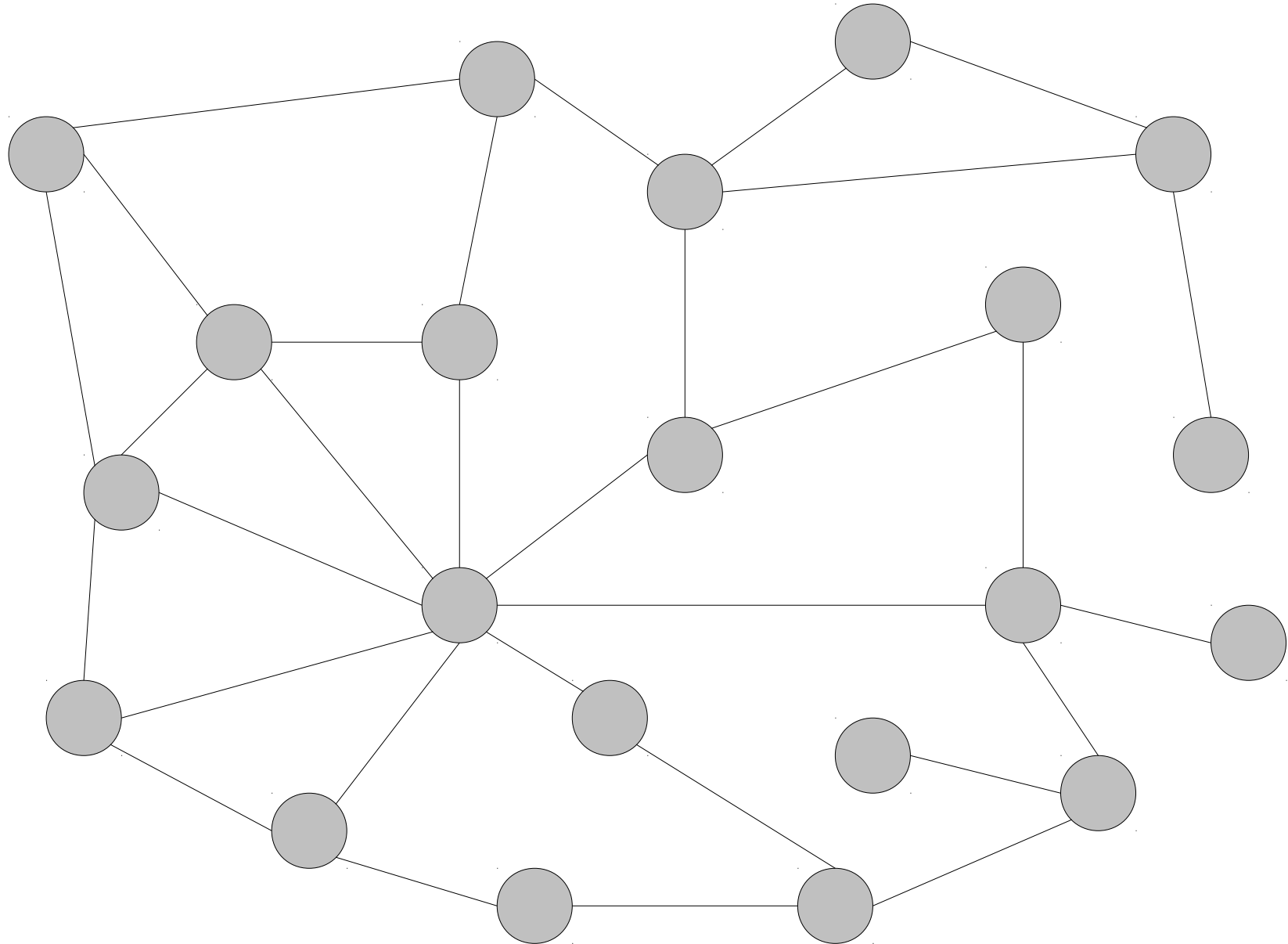
# Graph Coloring



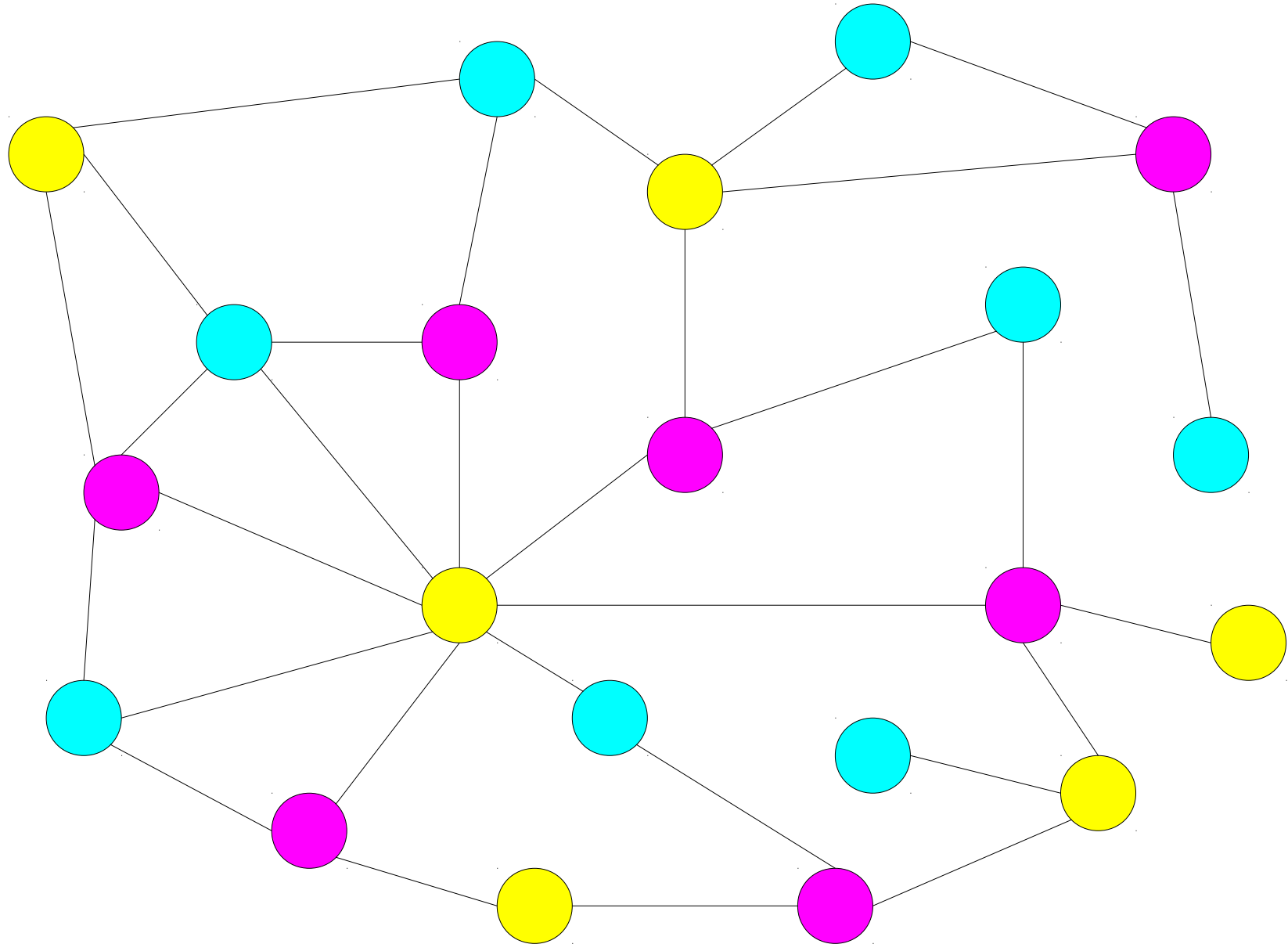
# Graph Coloring



# Graph Coloring



# Graph Coloring



# Graph Coloring

- An undirected graph  $G = (V, E)$  with no self-loops (edges from a node to itself) is called ***k-colorable*** if the nodes in  $V$  can be assigned one of  $k$  different colors such that no two nodes of the same color are joined by an edge.
- The minimum number of colors needed to color a graph is called that graph's ***chromatic number***.
  - The chromatic number of a graph  $G$  is usually denoted  **$\chi(G)$** , from the Greek  $\chi\rho\acute{\omega}\mu\alpha$  (“color”).

***Theorem (Four-Color Theorem):*** Every planar graph is 4-colorable.

- **1850s:** Four-Color Conjecture posed.
- **1879:** Kempe proves the Four-Color Theorem.
- **1890:** Heawood finds a flaw in Kempe's proof.
- **1976:** Appel and Haken design a computer program that proves the Four-Color Theorem. The program checked 1,936 specific cases that are “minimal counterexamples;” any counterexample to the theorem must contain one of the 1,936 specific cases.
- **1980s:** Doubts rise about the validity of the proof due to errors in the software.
- **1989:** Appel and Haken revise their proof and show it is indeed correct. They publish a book including a 400-page appendix of all the cases to check.
- **1996:** Roberts, Sanders, Seymour, and Thomas reduce the number of cases to check down to 633.
- **2005:** Werner and Gonthier repeat the proof using an established automatic theorem prover (Coq), improving confidence in the truth of the theorem.