CS103
Winter 2025

Lecture 09:

Graph Theory

Part 1 of 3

But first…

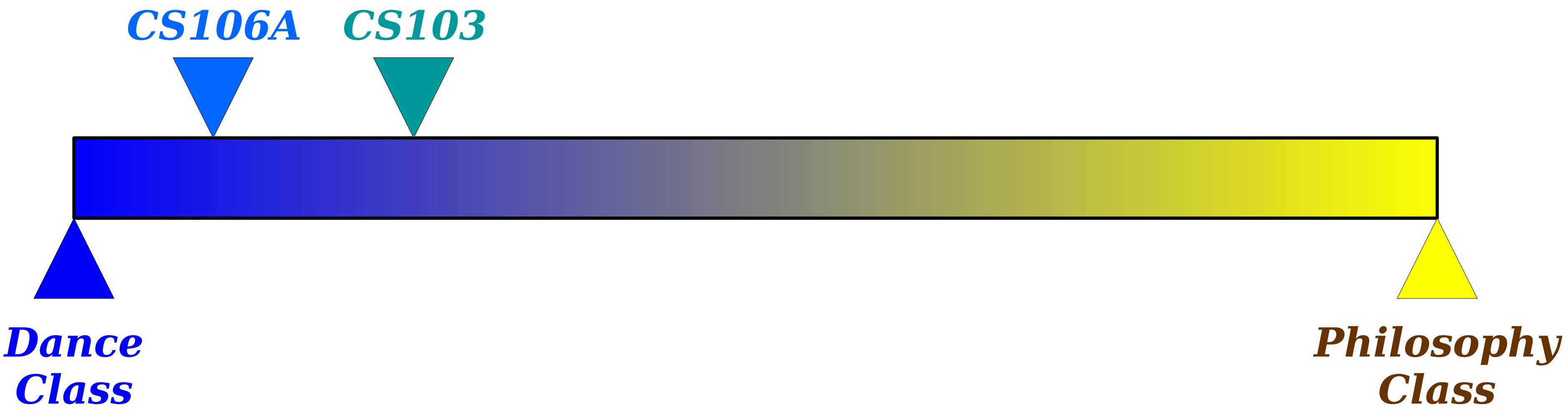
Midterm Exam Logistics

- Our first midterm exam is next ***Tuesday, February 4th***, from ***7:00 – 9:00 PM***. Locations vary (mostly CEMEX).
- You’re responsible for Lectures 00 – 05 and topics covered in PS1 – PS2. Later lectures (functions forward) and problem sets (PS3 onward) won’t be tested here. Exam problems may build on the written or coding components from the problem sets.
- The exam is closed-book, closed-computer, and limited-note. You can bring a double-sided, 8.5” × 11” sheet of notes with you to the exam, decorated however you’d like.
- Students with alternate exam arrangements: these will be confirmed via our seating assignment website.

Midterm Exam

- ***We want you to do well on this exam!***
  - We're not trying to “weed out” weak students.
  - We're not trying to enforce a curve where there isn't one.
  - We want you to show what you've learned up to this point so that you get a sense for where you stand and where you can improve.
- The purpose of this midterm is to give you a chance to show what you've learned in the past few weeks.
- It is not designed to assess your “mathematical potential” or “innate mathematical ability.”

Preparing for the Exam



Learn by doing.

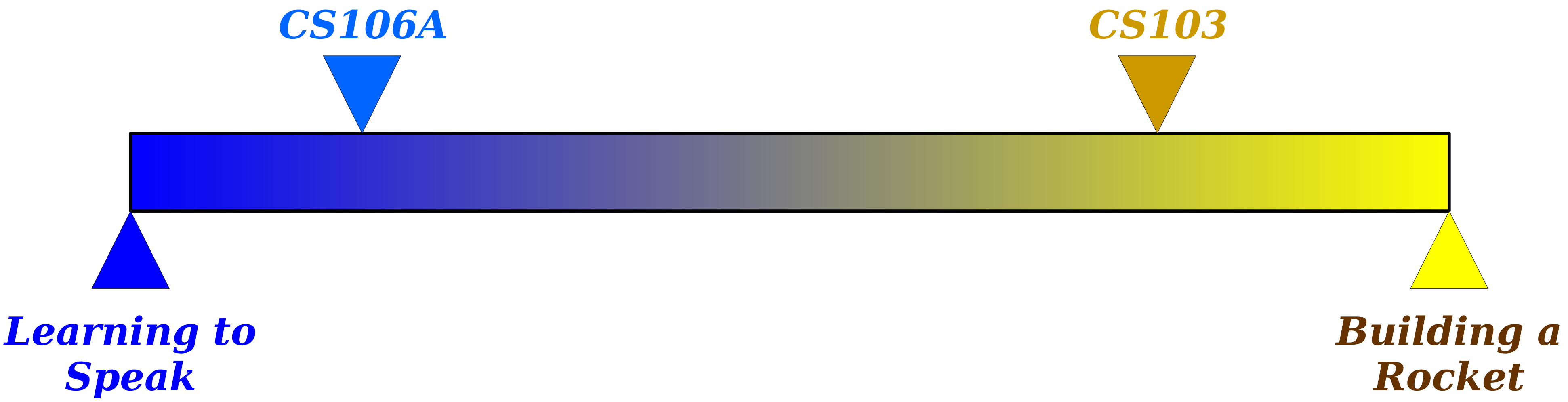
Learn by reading.

Extra Practice Problems

- Up on the course website, you'll find Extra Practice Problems 1, a collection of seven practice midterms and an assortment of other questions.
- ***Our Recommendation:***
  - Work through one or two practice exams under realistic conditions (block off two hours, have your notes sheet, use pencil and paper).
  - Review the solutions only when you're done. ***Don't peek!*** You can't do that on the actual exam.
  - Ping the course staff to ask questions, whether that's "please review this proof I wrote for one of the exam questions" or "why doesn't the solution do X, which seems easier than Y, which is what it did?"
  - ***Internalize the feedback***. What areas do you need more practice with? Study up on those topics. What transferrable skills did you learn in the course of solving the problems? If you aren't sure, ask!
  - Repeat!
- Realistically, we don't expect you to do seven practice exams. We've provided those just so you can get a sense of what's out there.

You can always run your code and just see what happens!

Checking a proof requires human expertise.



CS106A

CS103

Learning to Speak

Building a Rocket

Rapid iteration.
Constant, small feedback.

Slower iteration.
Infrequent, large feedback.

Doing Practice Problems

- As you work through practice problems, ***keep other humans in the loop!***
- Ask your problem set partner to review your answers and offer feedback – and volunteer to do the same!
- Post your answers as private questions on Ed and ask for TA feedback!
- ***Feedback loops are key to improving!***

Preparing for the Exam

- We’ve posted an ***Exam Logistics*** 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!

Exam Day Logistics

- We’ll have proctors in the room.
- We have assigned seating; see course website. These will be posted sometime tomorrow (Thursday).
- Check out your seat in advance, and screenshot it!
- No phones, calculators, or other digital devices during the exam.
- Must have Stanford student ID to turn in exam.

Back to course content!

Today’s Main Topic: Graphs!

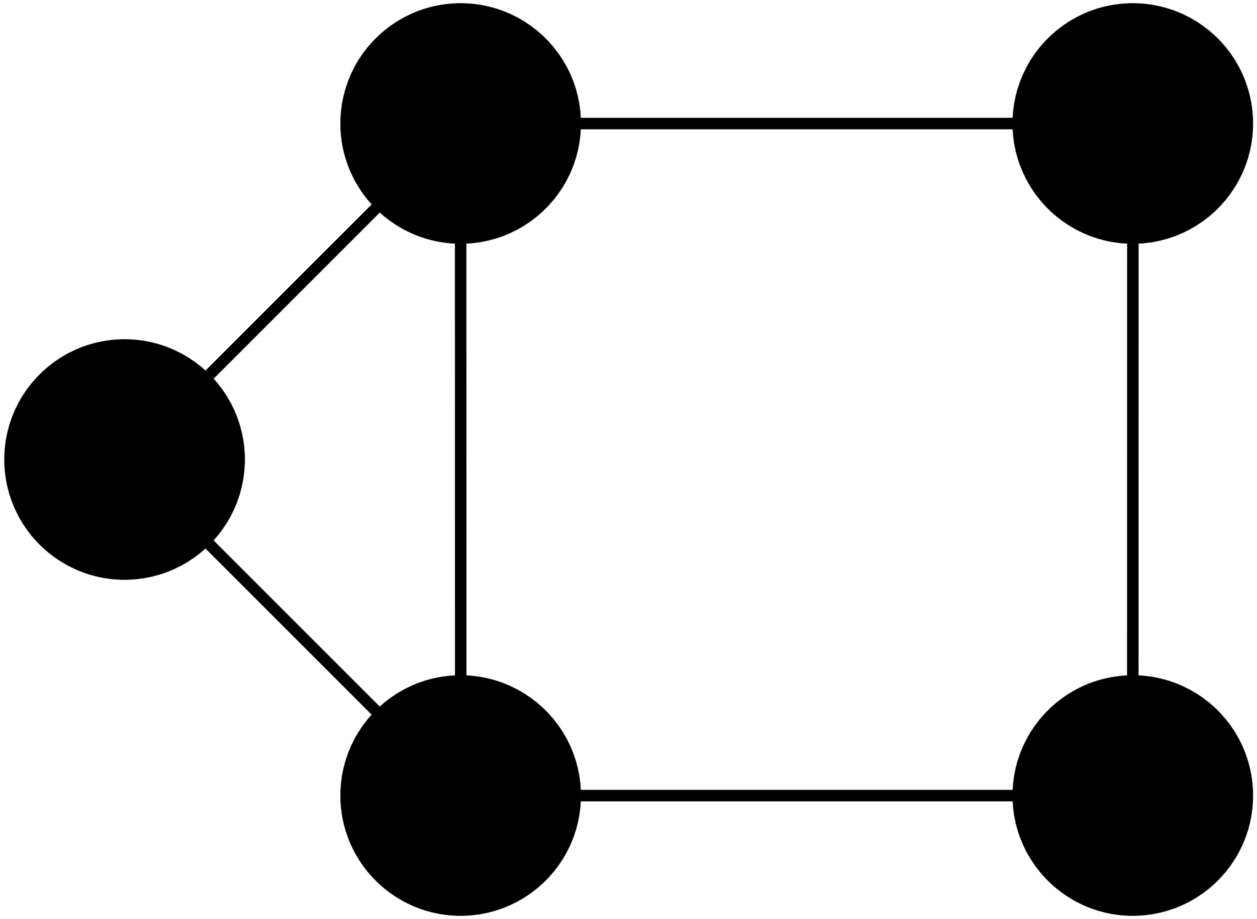
- A fun new toy to play with!
- “Magnum opus” of data structures with respect to representational versatility.
- A vehicle for exploring proof techniques.

Outline for Today

- ***Graphs and Digraphs***
  - Two fundamental mathematical structures.
- ***Independent Sets and Vertex Covers***
  - Two structures in graphs.
- ***Proofs on Graphs***
  - Reprising themes from last week.

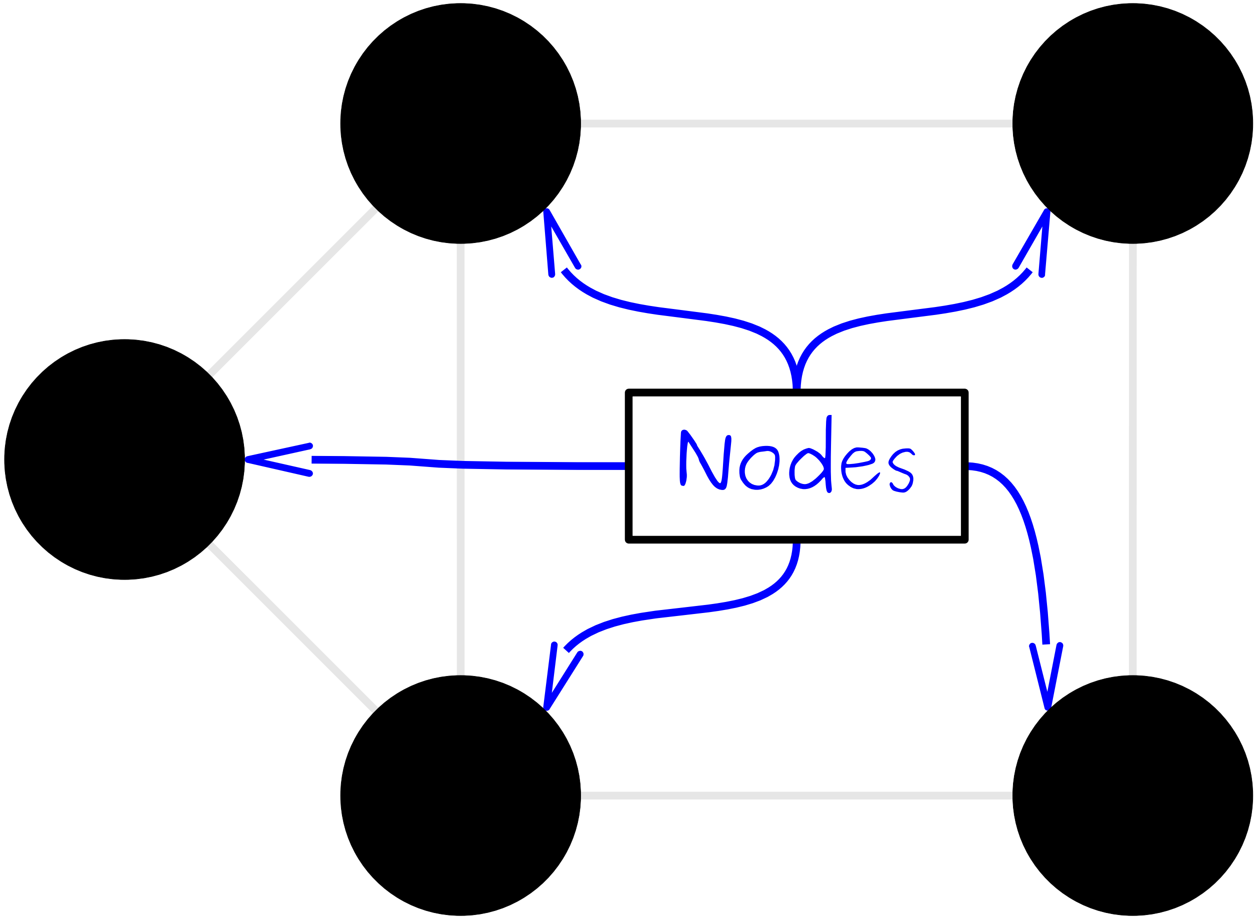
Graphs: An Overview

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



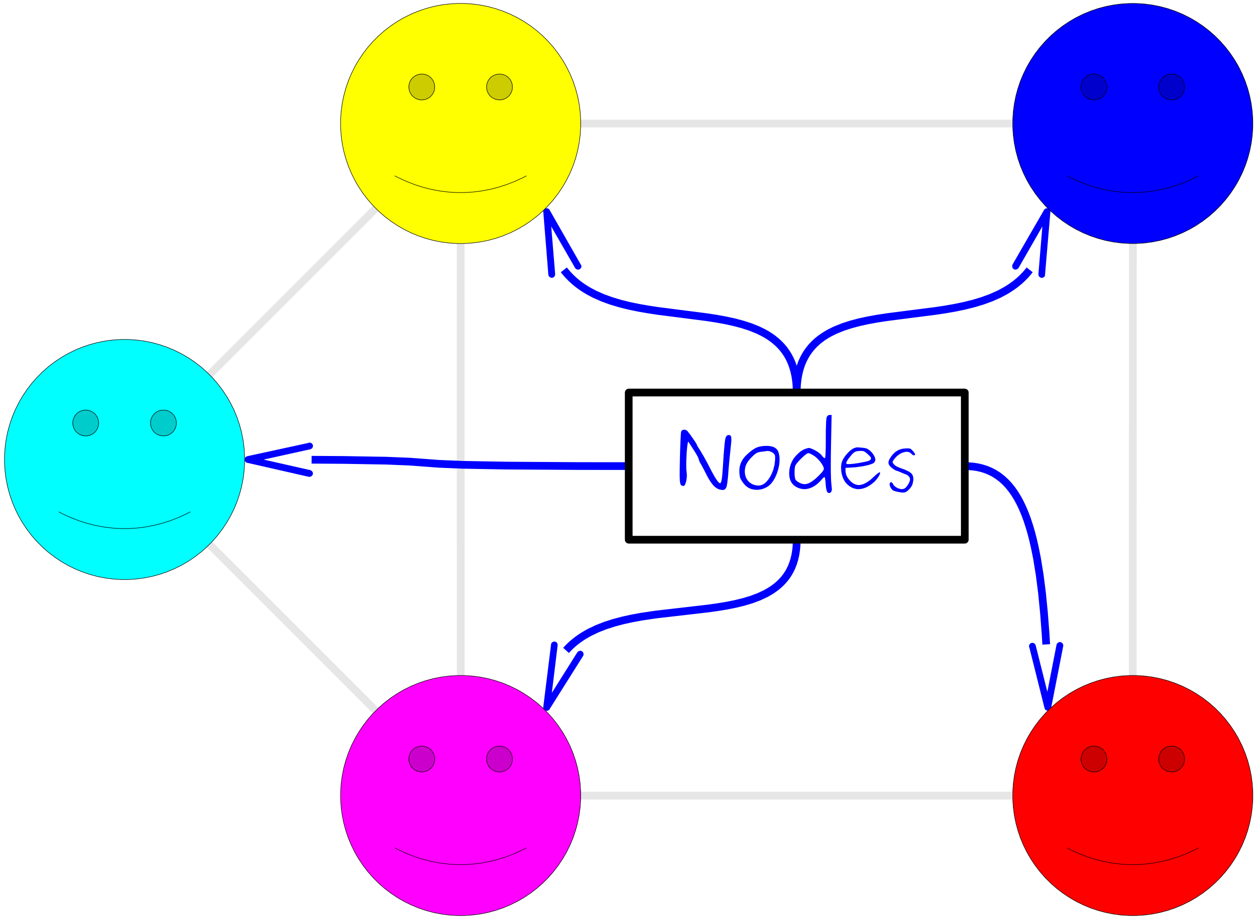
A graph consists of a set of ***nodes*** (or ***vertices***) connected by ***edges*** (or ***arcs***)

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



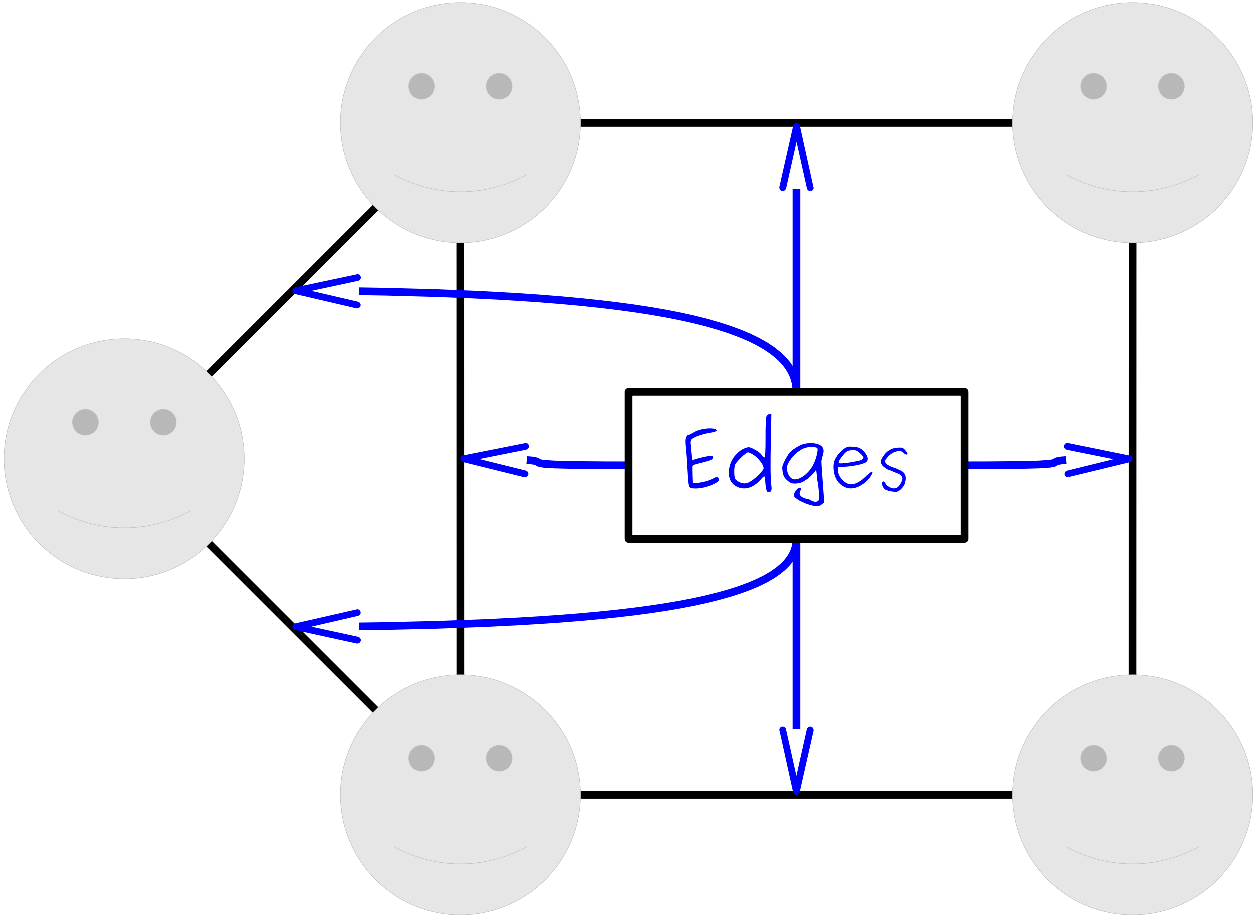
A graph consists of a set of ***nodes*** (or ***vertices***) connected by ***edges*** (or ***arcs***)

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



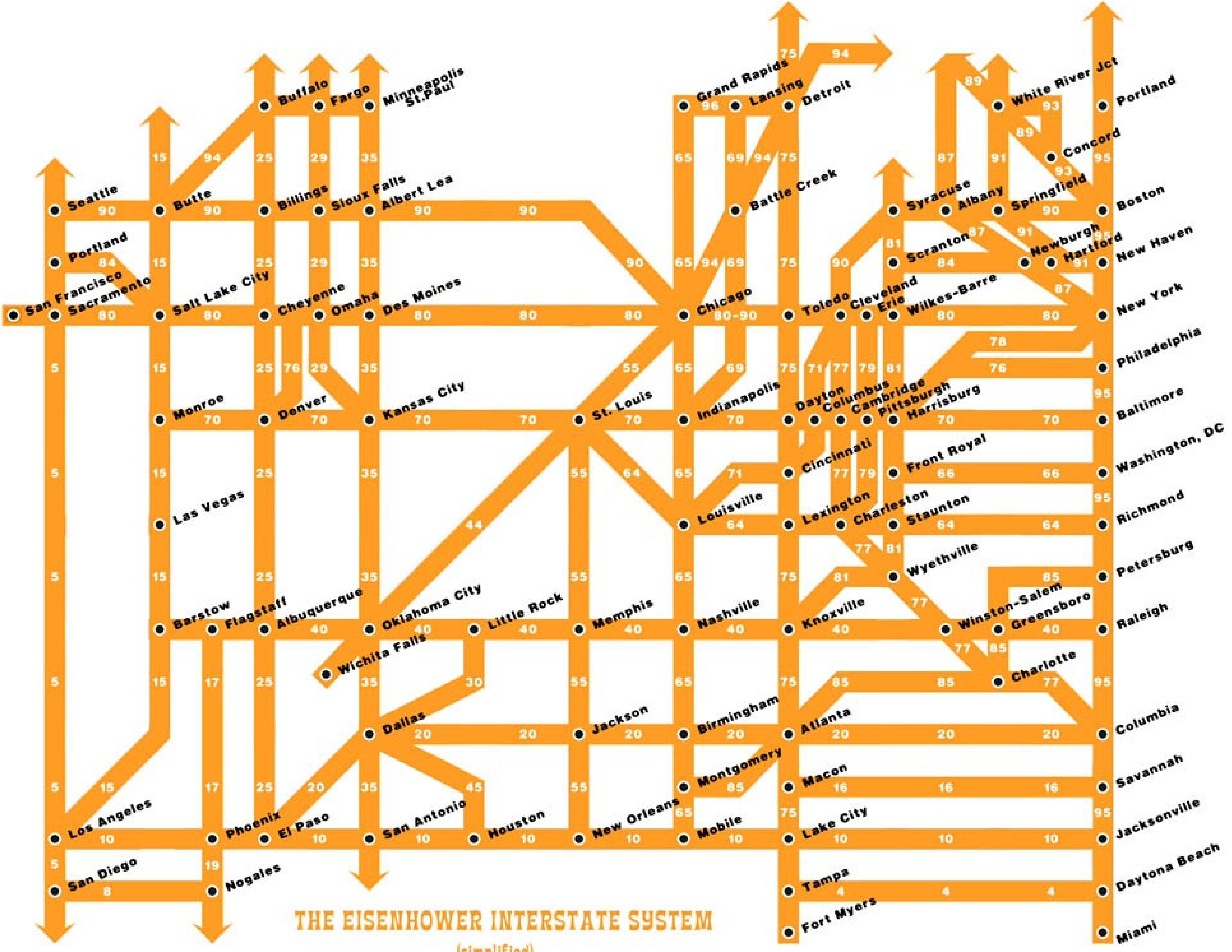
A graph consists of a set of ***nodes*** (or ***vertices***) connected by ***edges*** (or ***arcs***)

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



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

Examples



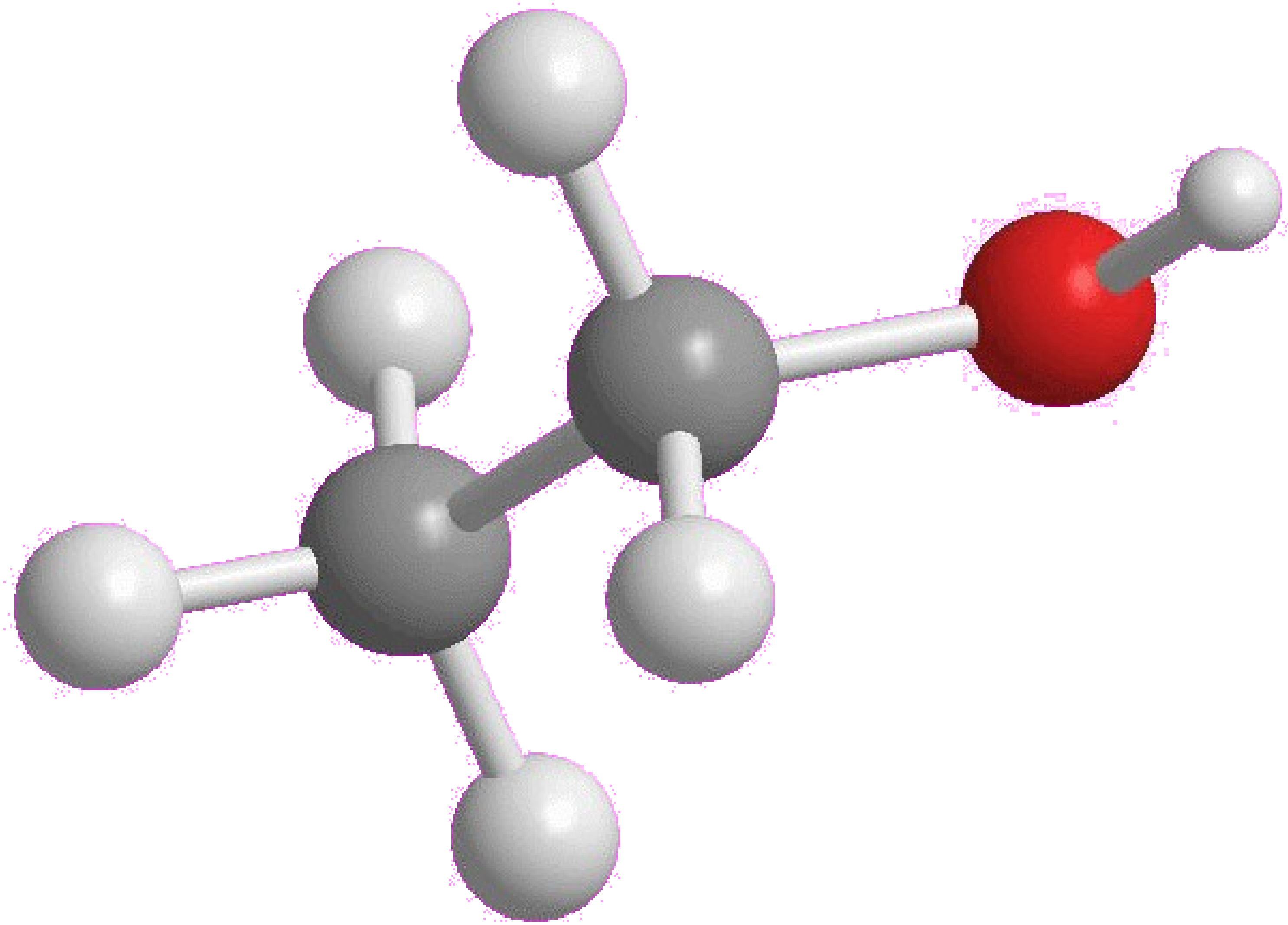
THE EISENHOWER INTERSTATE SYSTEM

(simplified)

CHRIS YATES 2007

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

Chemical Bonds

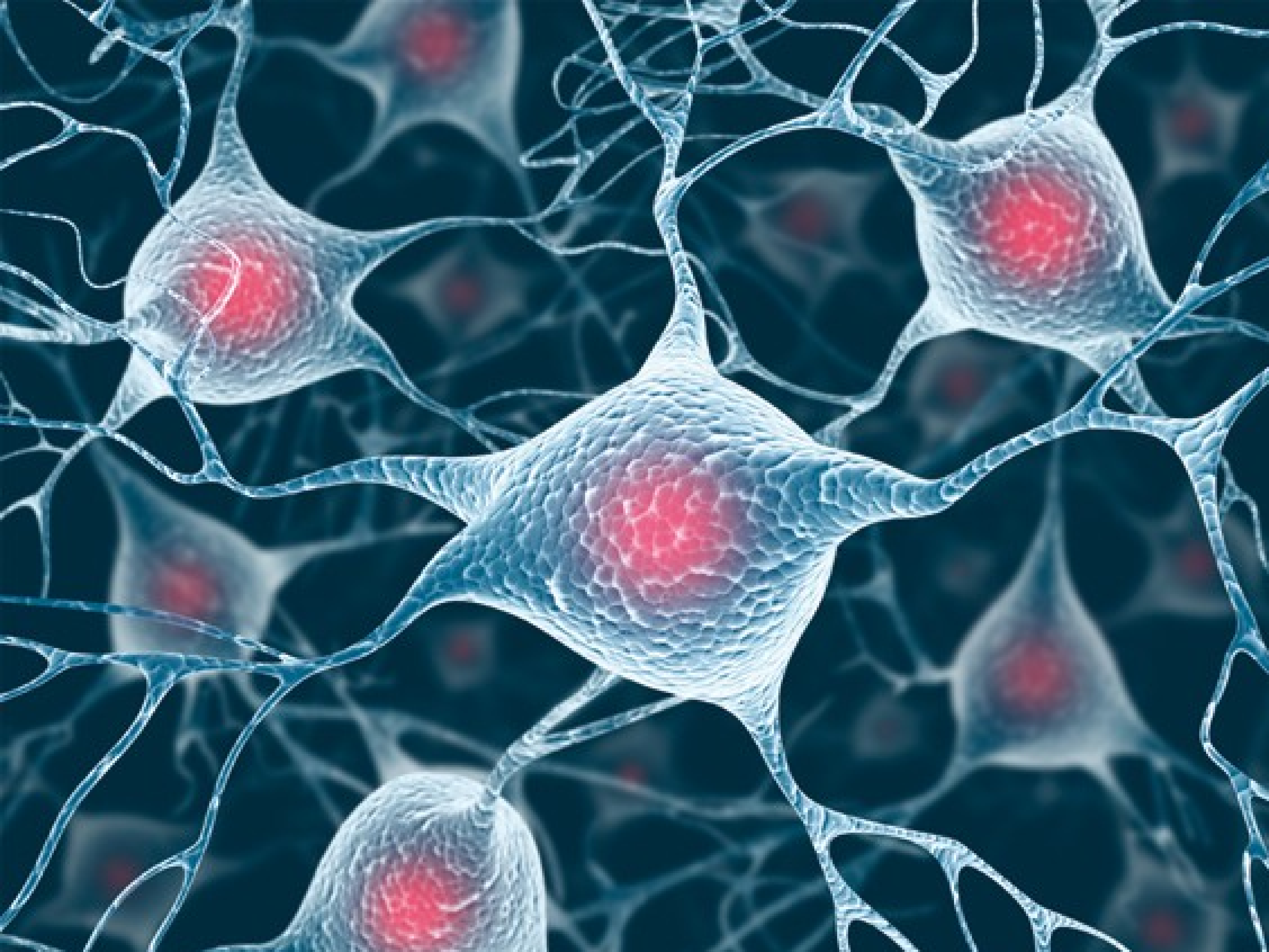


http://4.bp.blogspot.com/-xCtBJ8lKHqA/Tjm0BONWBRI/AAAAAAAAAK4/-mHrbAUOHHg/s1600/

facebook®

LinkedIn

WeChat

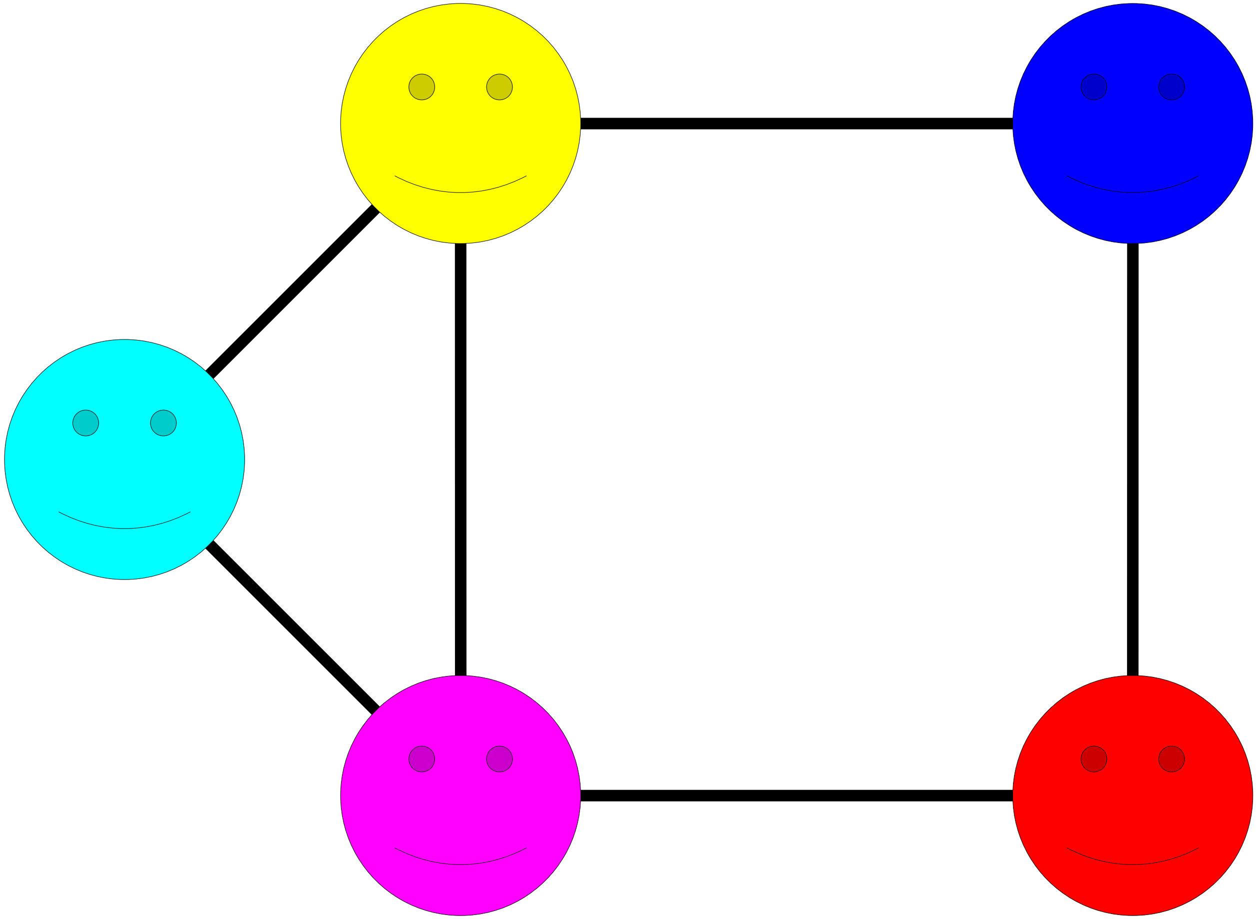


What's in Common

- Each of these structures consists of
  - a collection of objects and
  - links between those objects.
- ***Goal:*** Develop a general framework for describing structures like these that generalizes the idea across a wide domain.

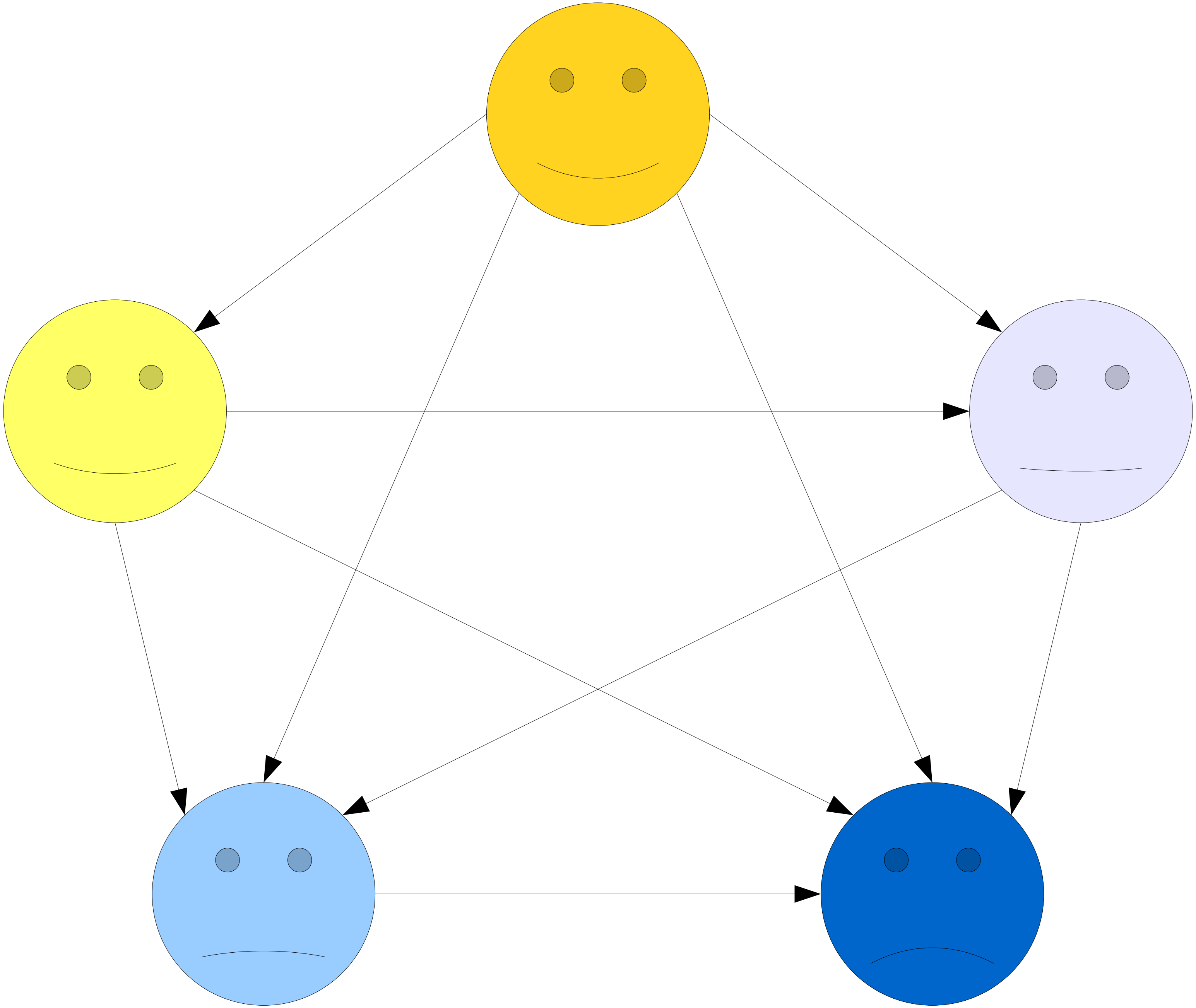
Graphs vs. Digraphs

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

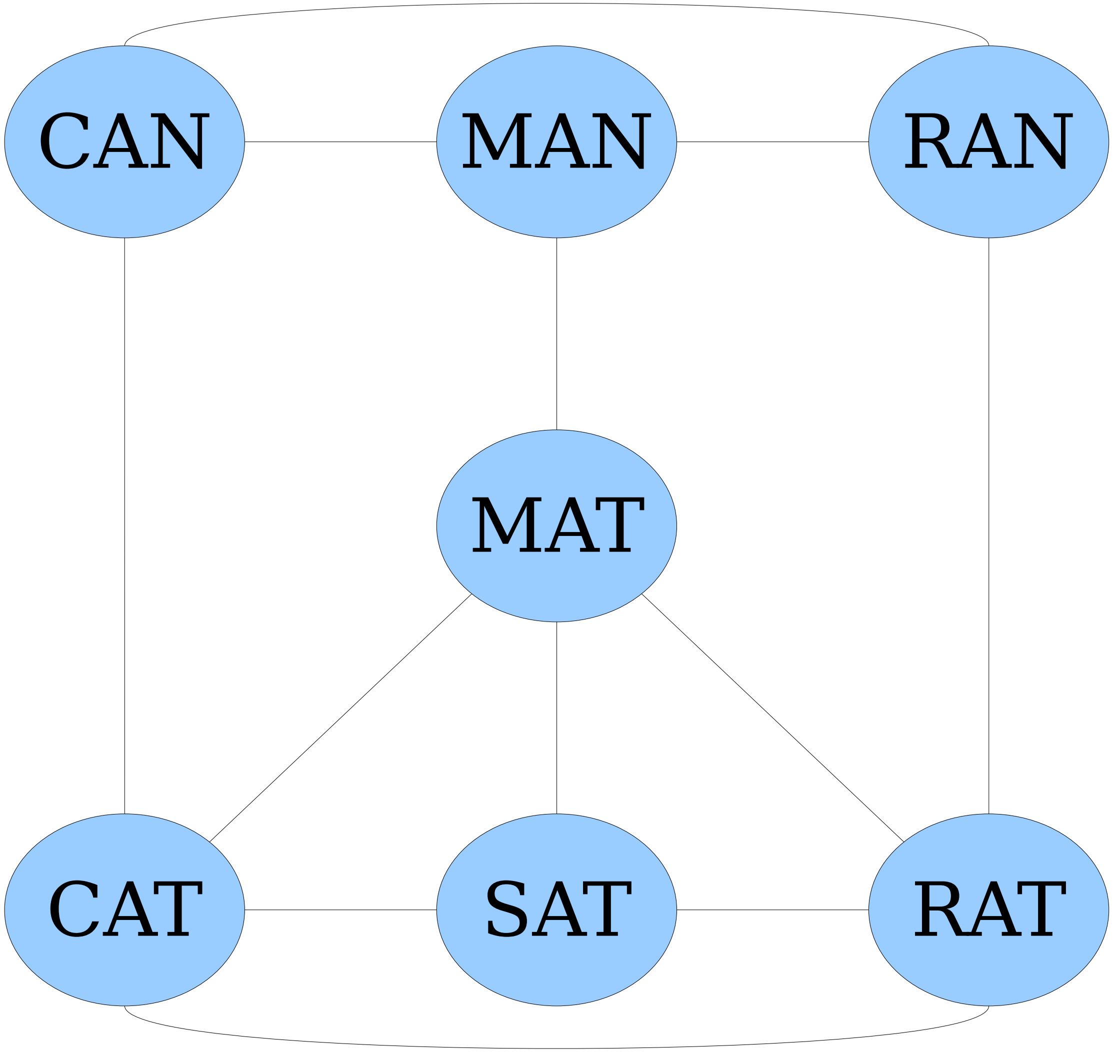


A graph consists of a set of ***nodes*** (or ***vertices***) connected by ***edges*** (or ***arcs***)

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 types of edges, but they’re fairly uncommon and we won’t talk about them.)
- Unless specified otherwise:

☞ ***“Graph” means “undirected graph”*** ☜

Formalizing Graphs

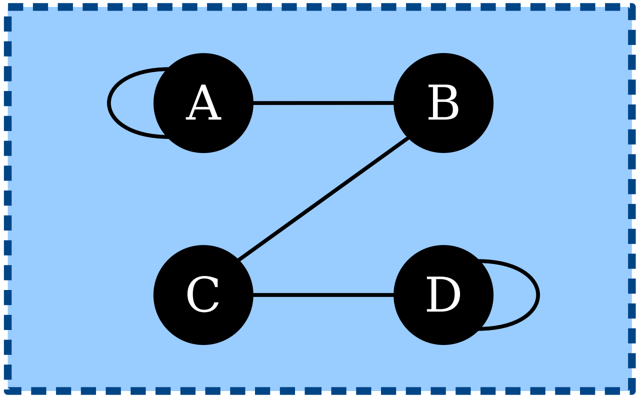
- How might we define a graph mathematically?
- We need to specify
  - what the nodes in the graph are, and
  - which edges are in the graph.
- The nodes can be pretty much anything.
- This is pretty broad, but that’s a good thing!
- What about edges?

Formalizing Graphs

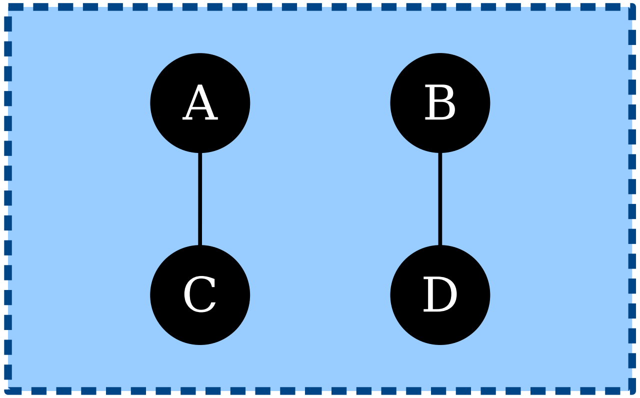
- An ***unordered pair*** is a set $\{a, b\}$ of two elements $a \neq b$. (Remember that sets are unordered.)
  - For example, $\{0, 1\} = \{1, 0\}$
- An ***undirected graph*** is an ordered pair $G = (V, E)$, where
  - V is a set of nodes, which can be anything, and
  - E is a set of edges, which are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

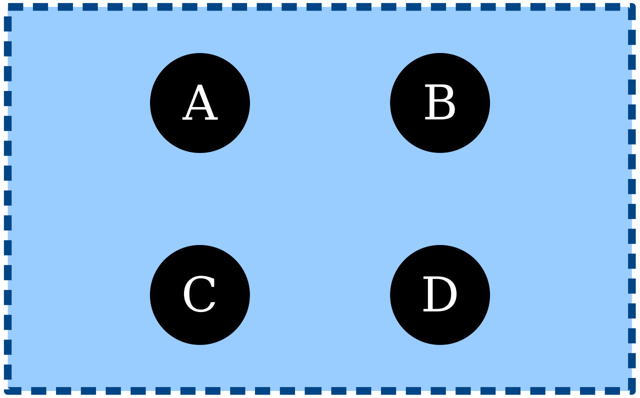
1



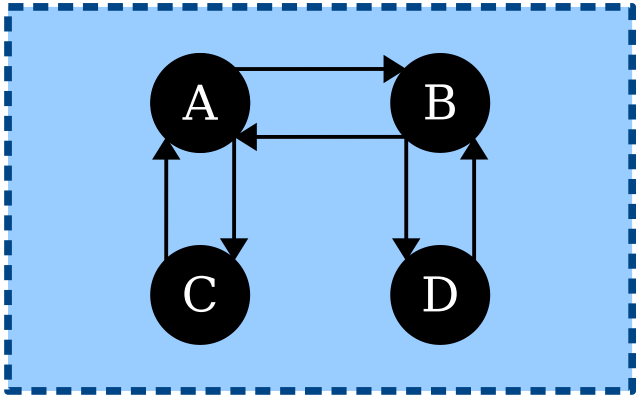
2



3



4



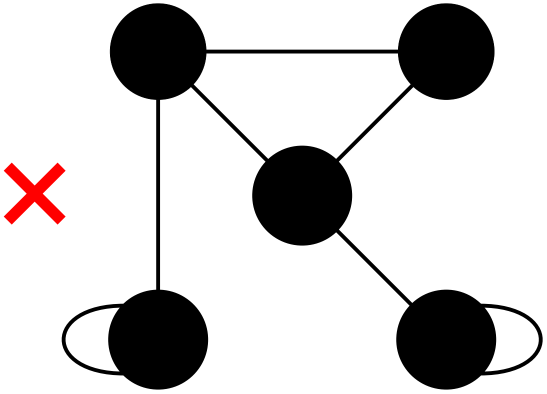
Which of these are drawings of undirected graphs?

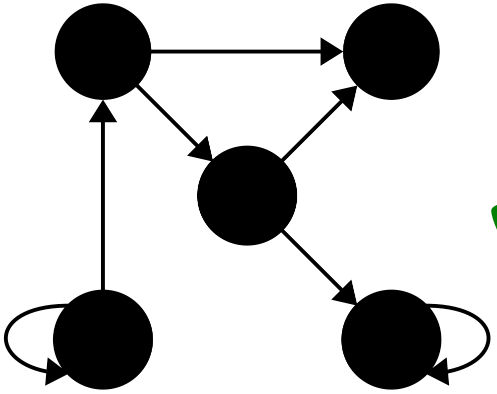
Answer at ***https://cs103.stanford.edu/pollev***

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.

✗

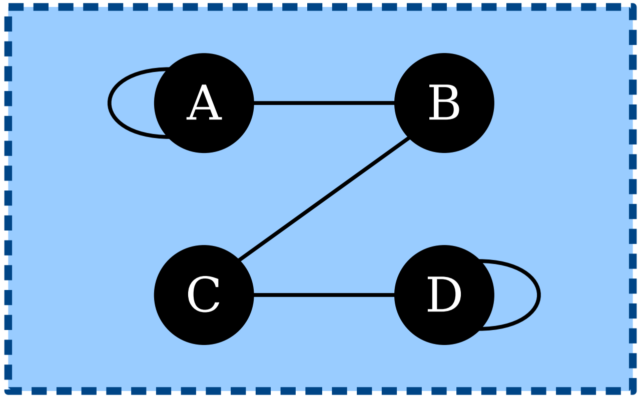




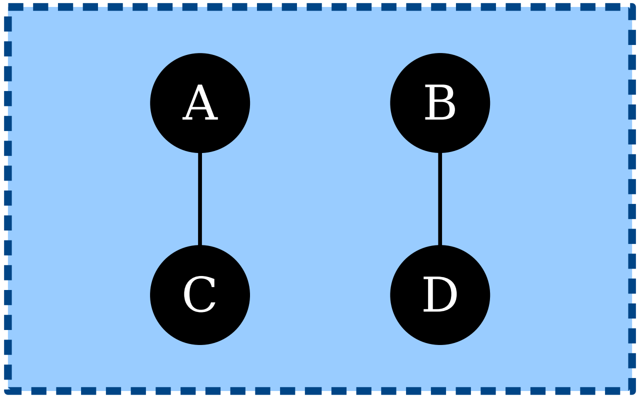
✓

- 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.

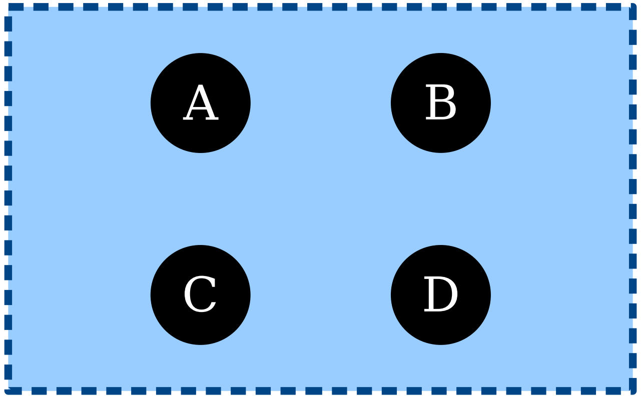
1



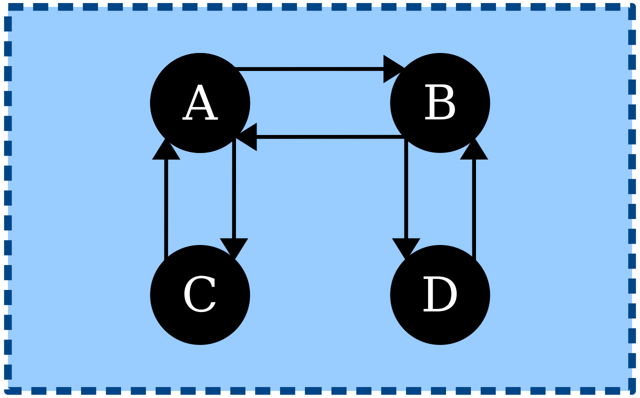
2



3



4



Which of these are drawings of undirected graphs?

Time-Out for Announcements!

^

more

PS2 Solutions Released

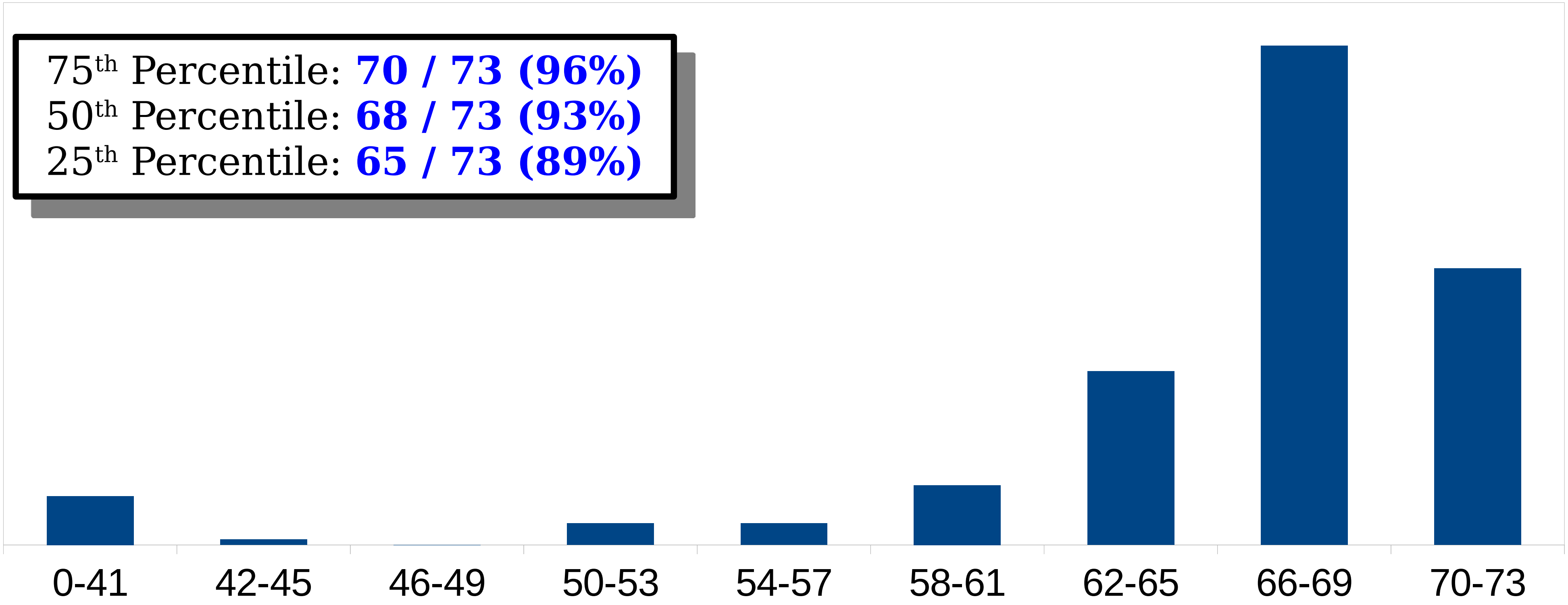
- Solutions to Problem Set 2 posted on course website.
  - Generally no solutions to autograded problems.
  - Questions? Ping us privately on Ed or visit office hours!
- PS3 is due this Friday at 1:00PM.
  - Ask questions if you have them! That’s what we’re here for. You can ask on Ed or in office hours.
  - Please tag problems. :)

In-Person Practice Midterm

- Saturday, 3-5 PM
- See Elena’s post on Ed for details

Problem Set Two Graded

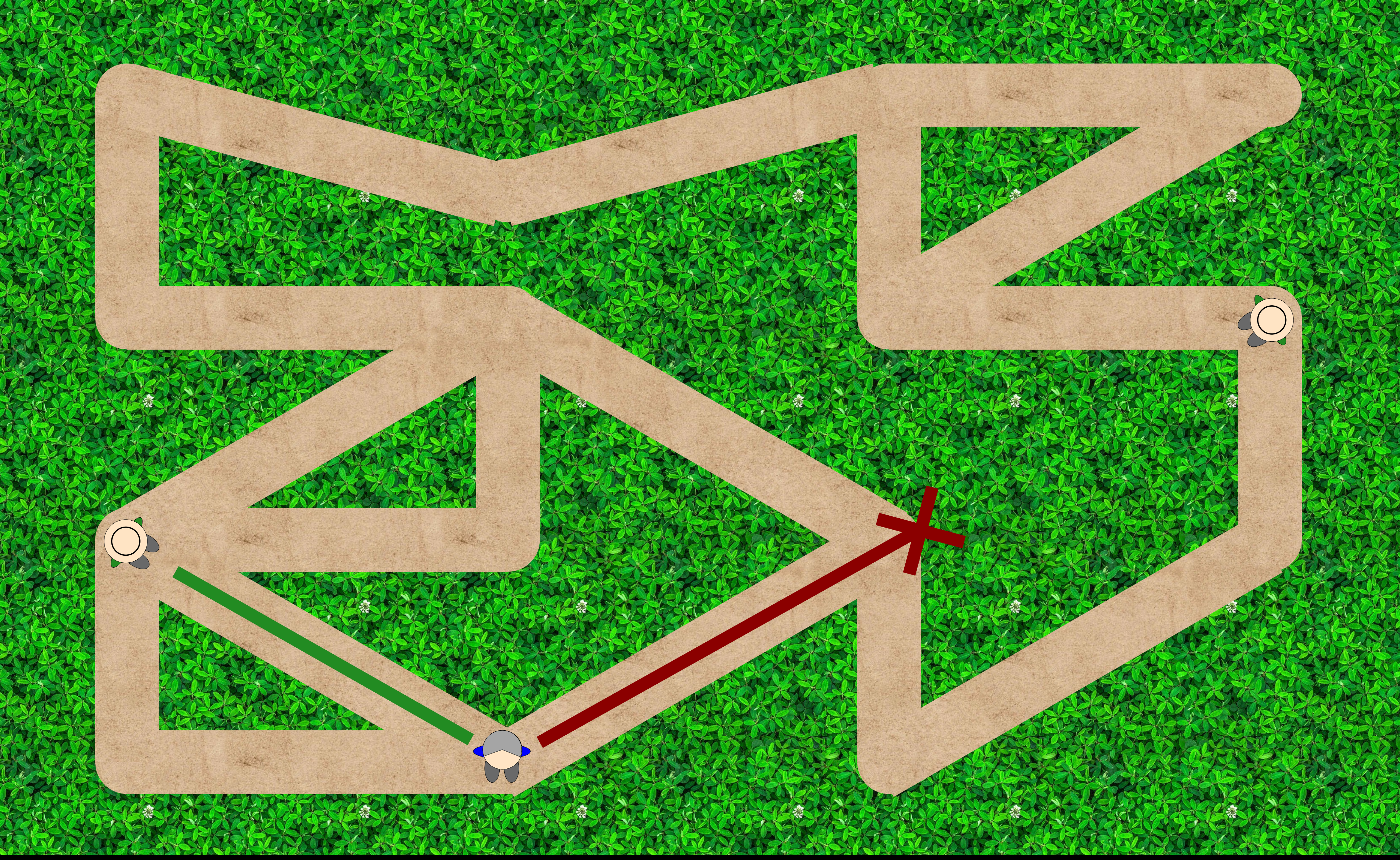
75th Percentile: **70 / 73 (96%)**
50th Percentile: **68 / 73 (93%)**
25th Percentile: **65 / 73 (89%)**



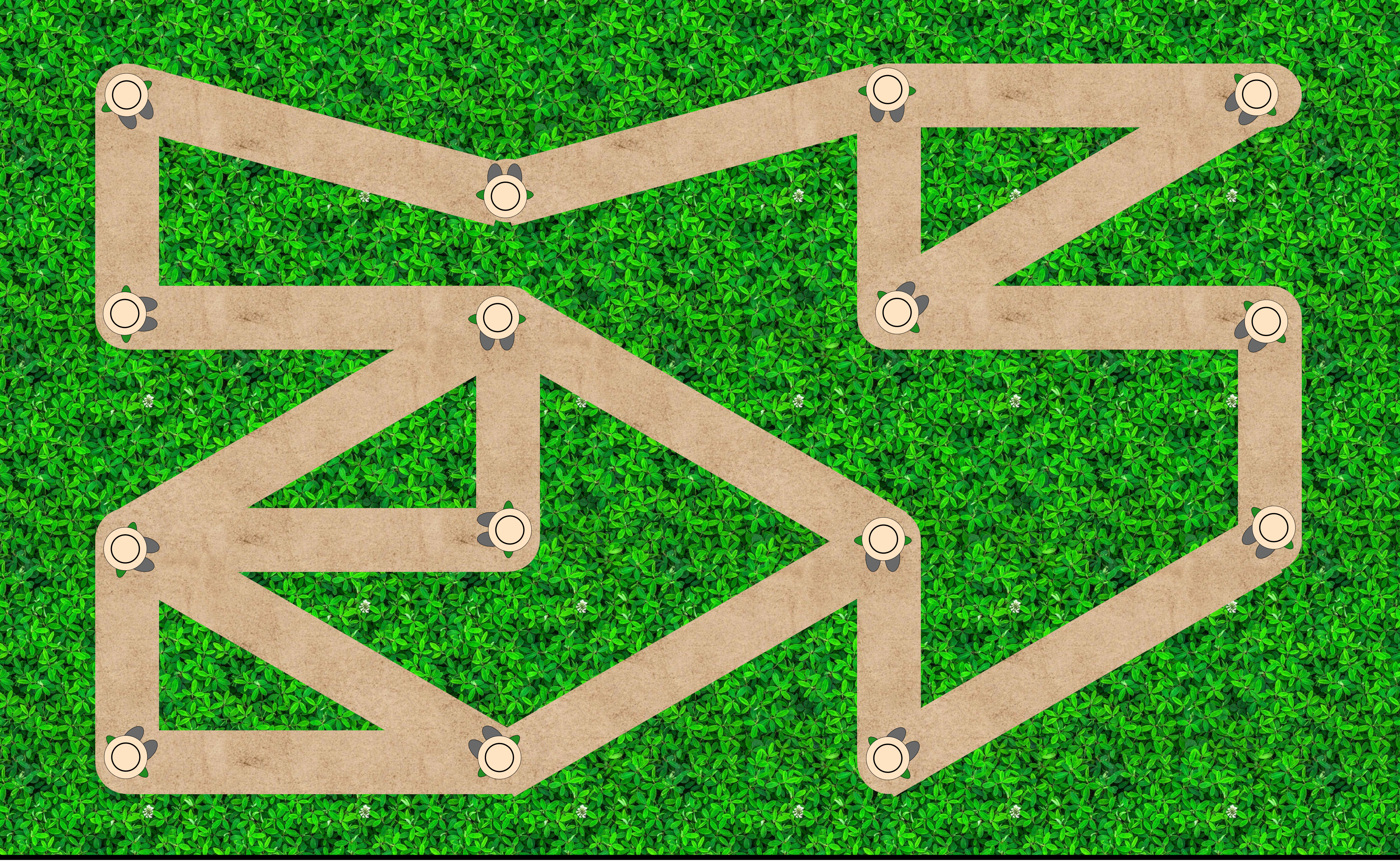
Back to CS103!

Independent Sets and Vertex Covers

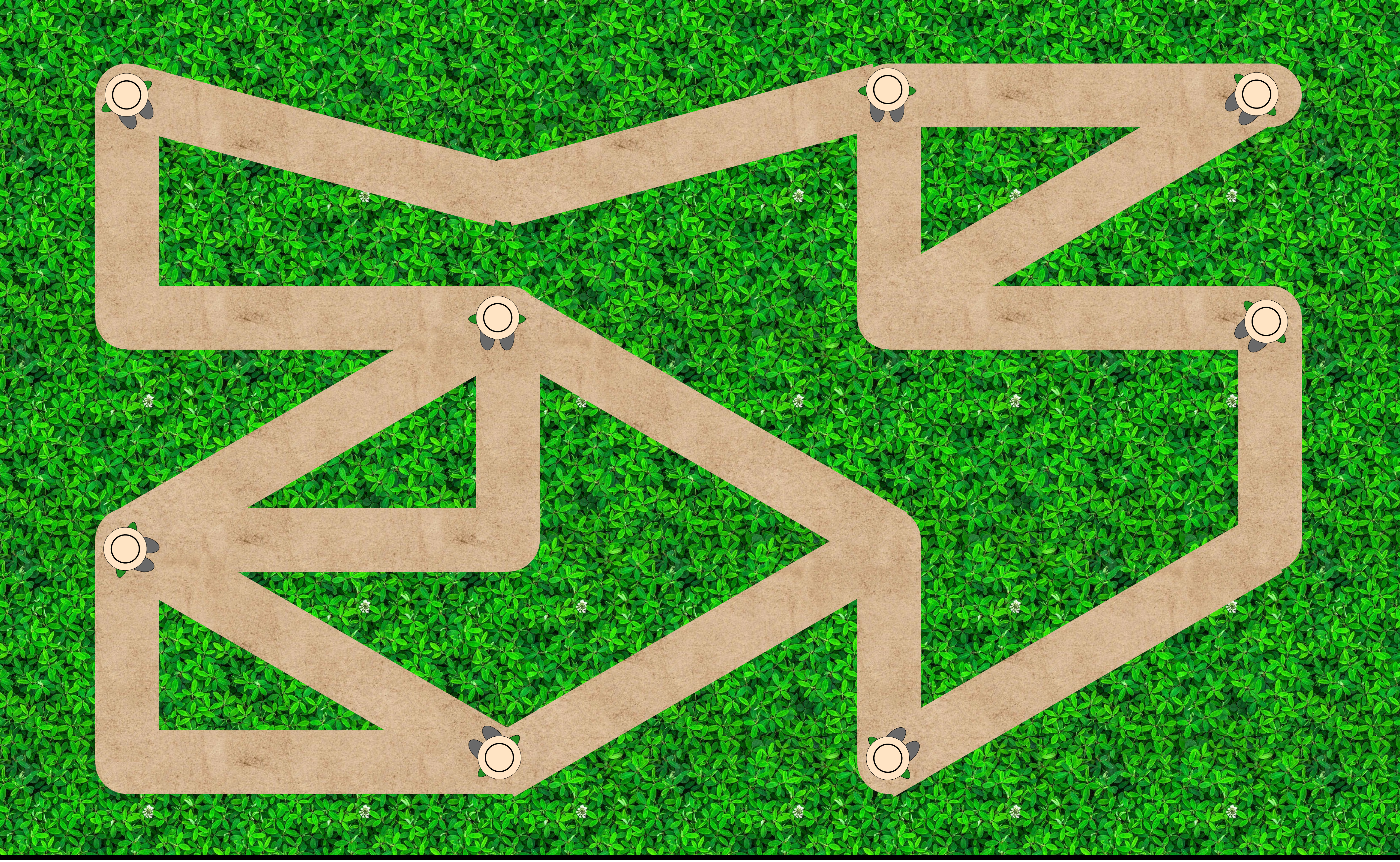
Two Motivating Problems



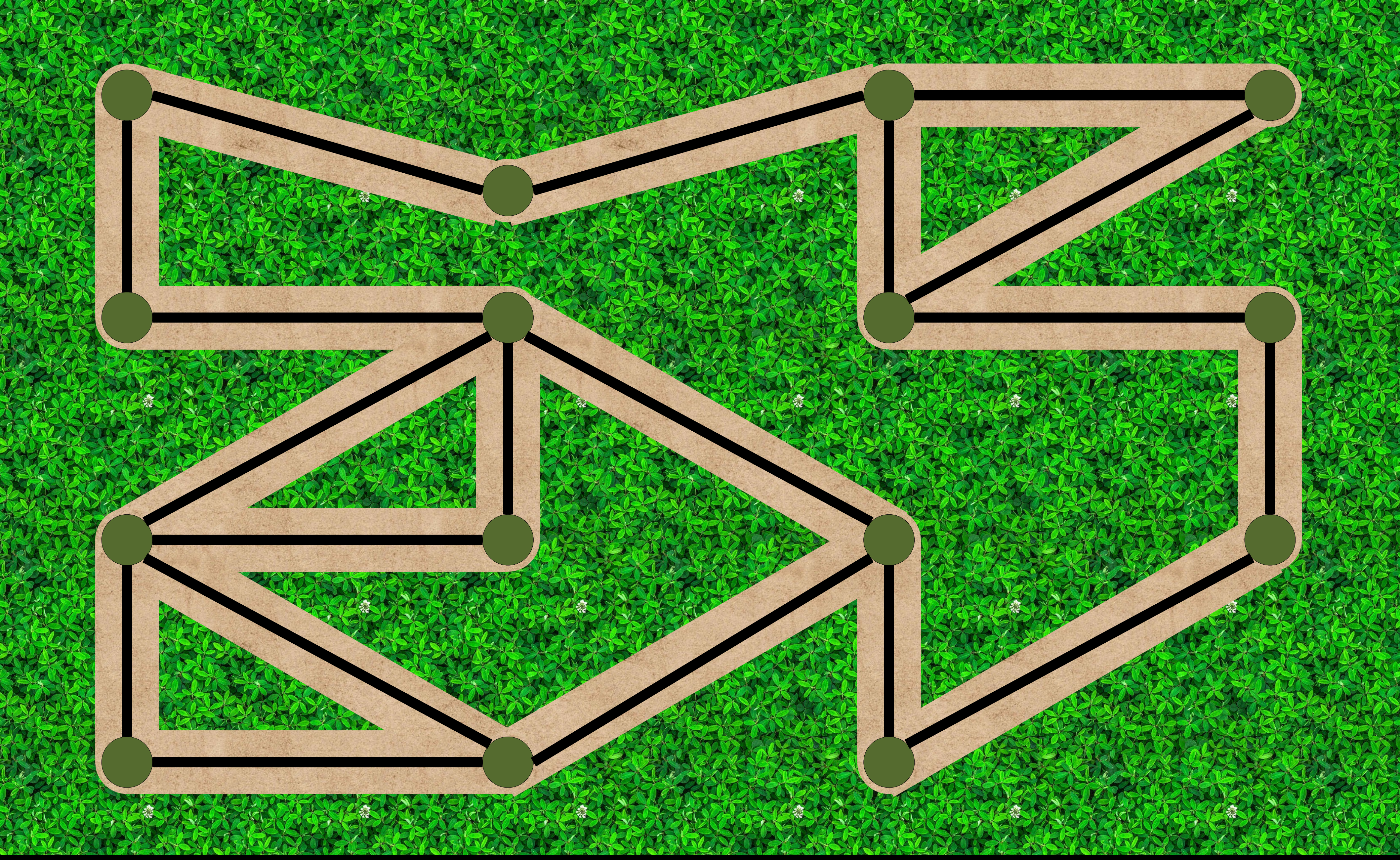
Place park rangers in these forest trails so that
a hiker anywhere on a trail can see a park ranger.



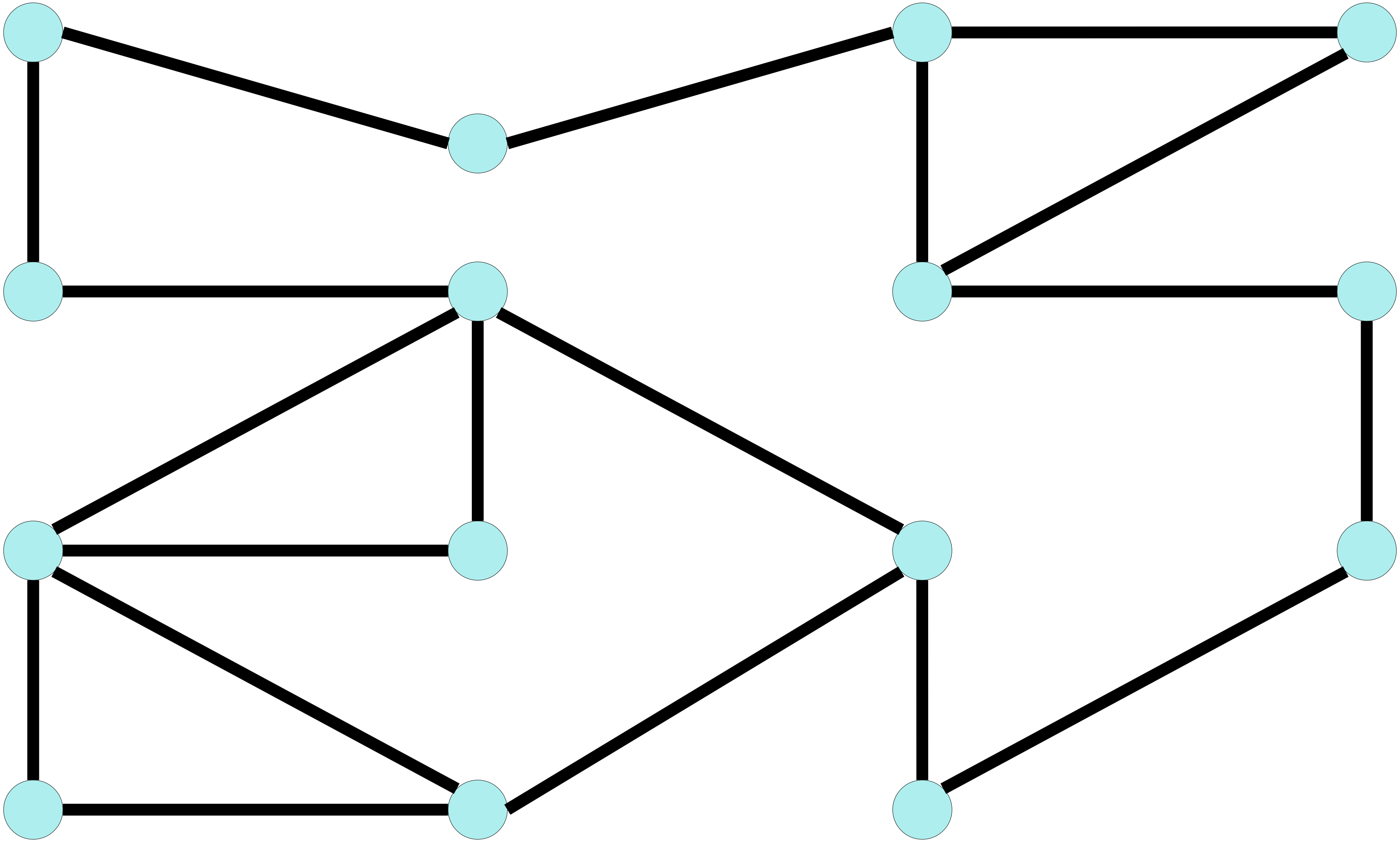
Place park rangers in these forest trails so that
a hiker anywhere on a trail can see a park ranger.



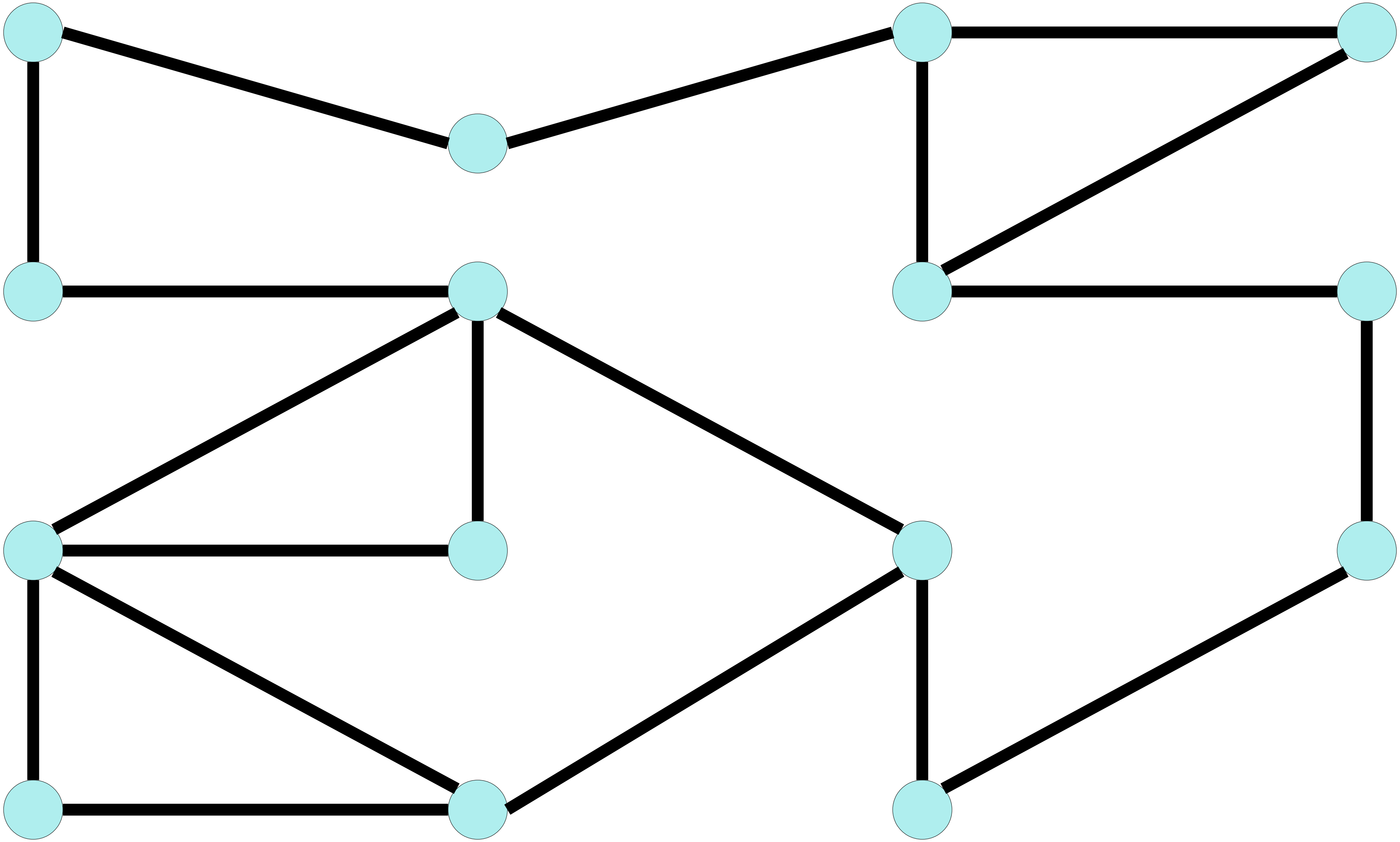
Place park rangers in these forest trails so that
a hiker anywhere on a trail can see a park ranger.



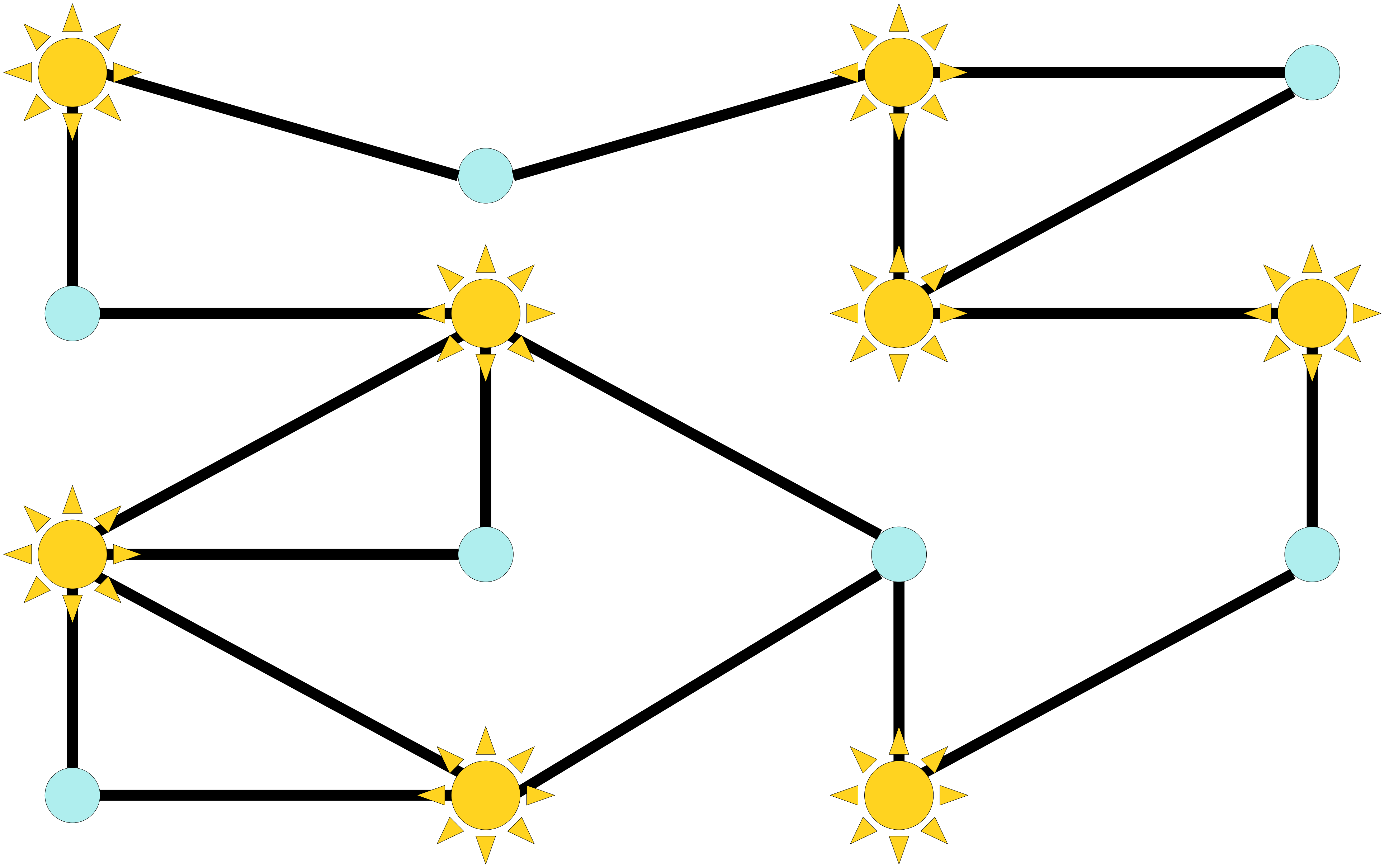
Place park rangers in these forest trails so that
a hiker anywhere on a trail can see a park ranger.



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



Choose at least one endpoint of each edge.



Choose at least one endpoint of each edge.

Vertex Covers

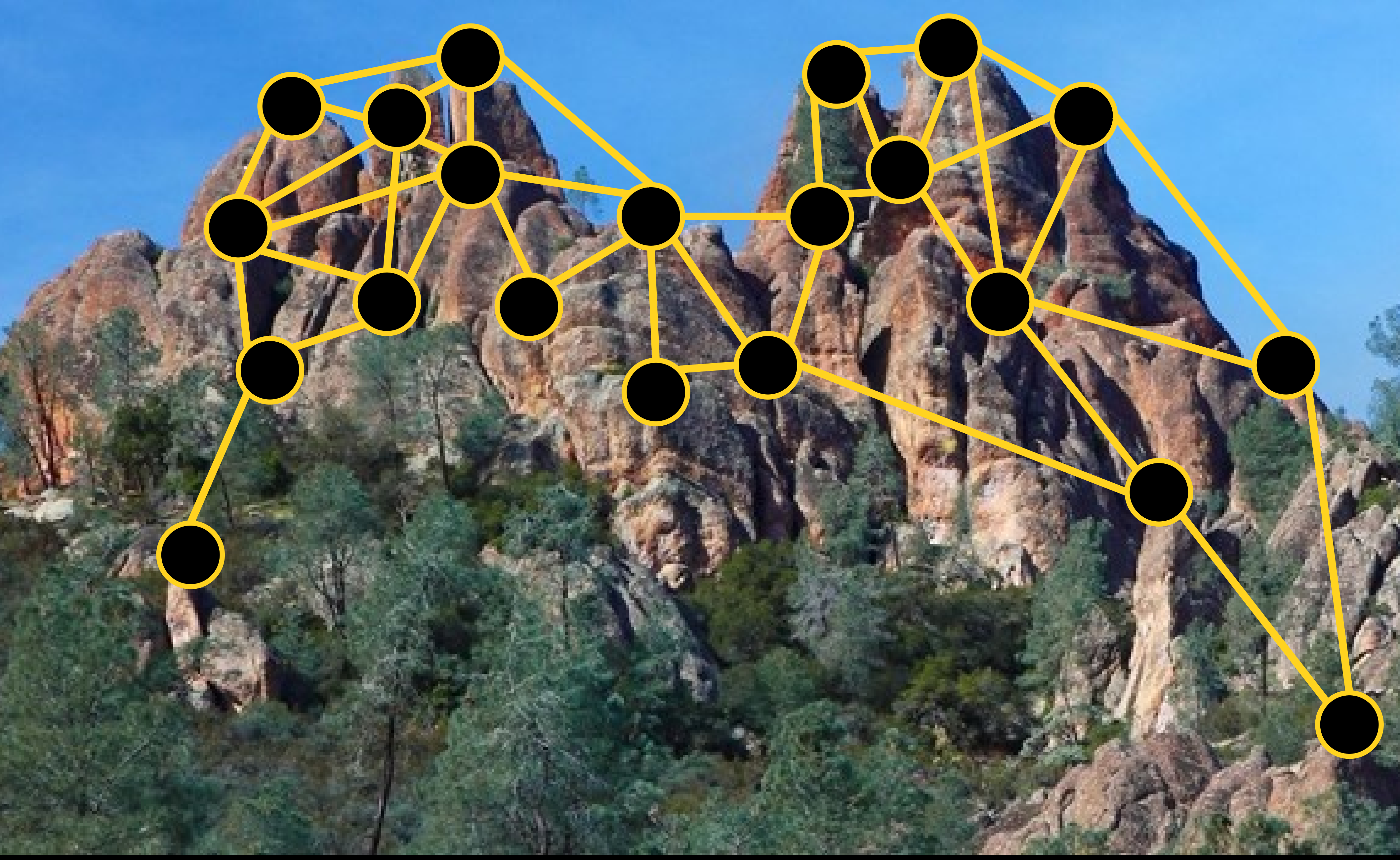
- Let $G = (V, E)$ be an undirected graph. A ***vertex cover*** of G is a set $C \subseteq V$ such that the following statement is true:

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

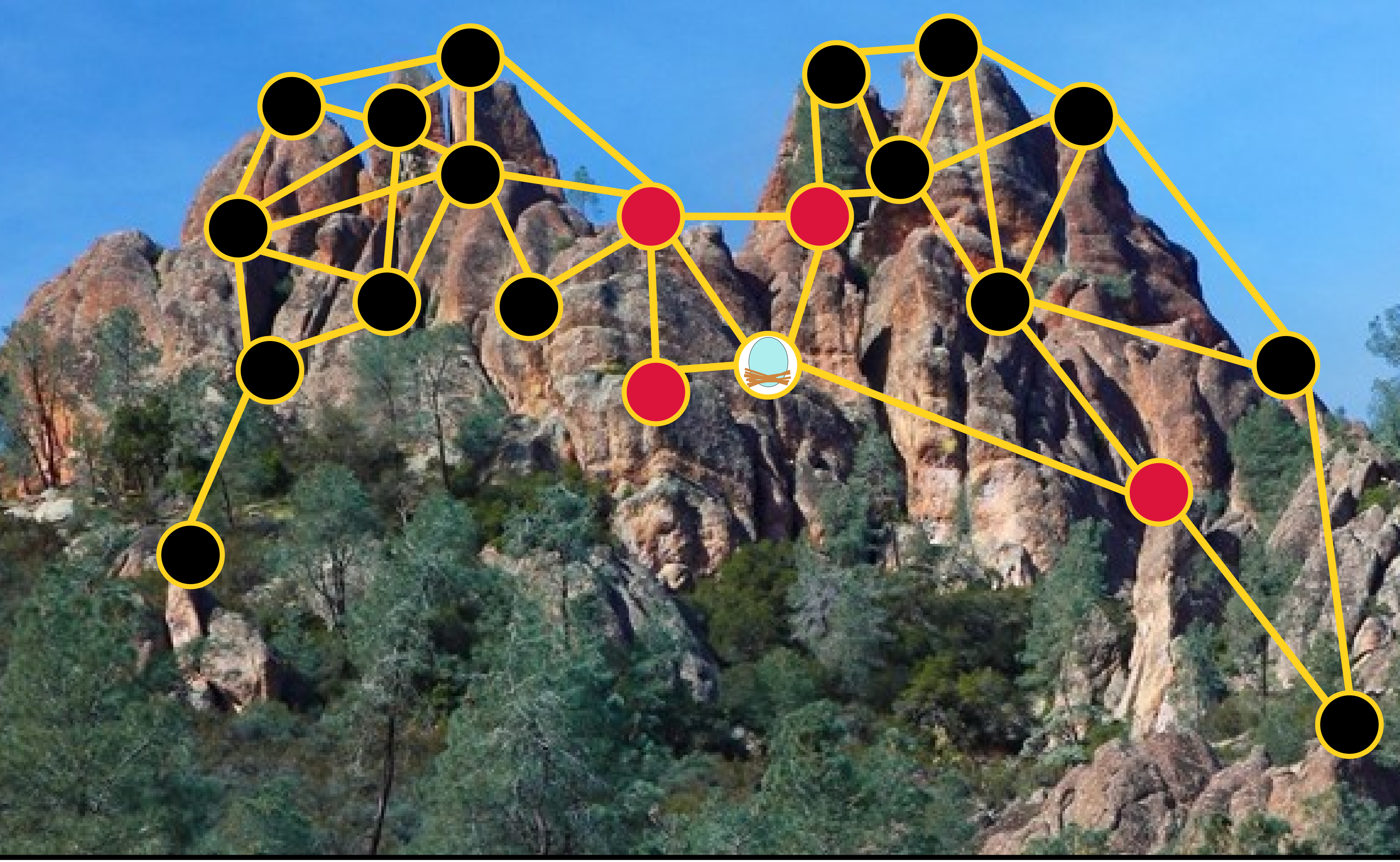
("Every edge has at least one endpoint in C.")

- 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.

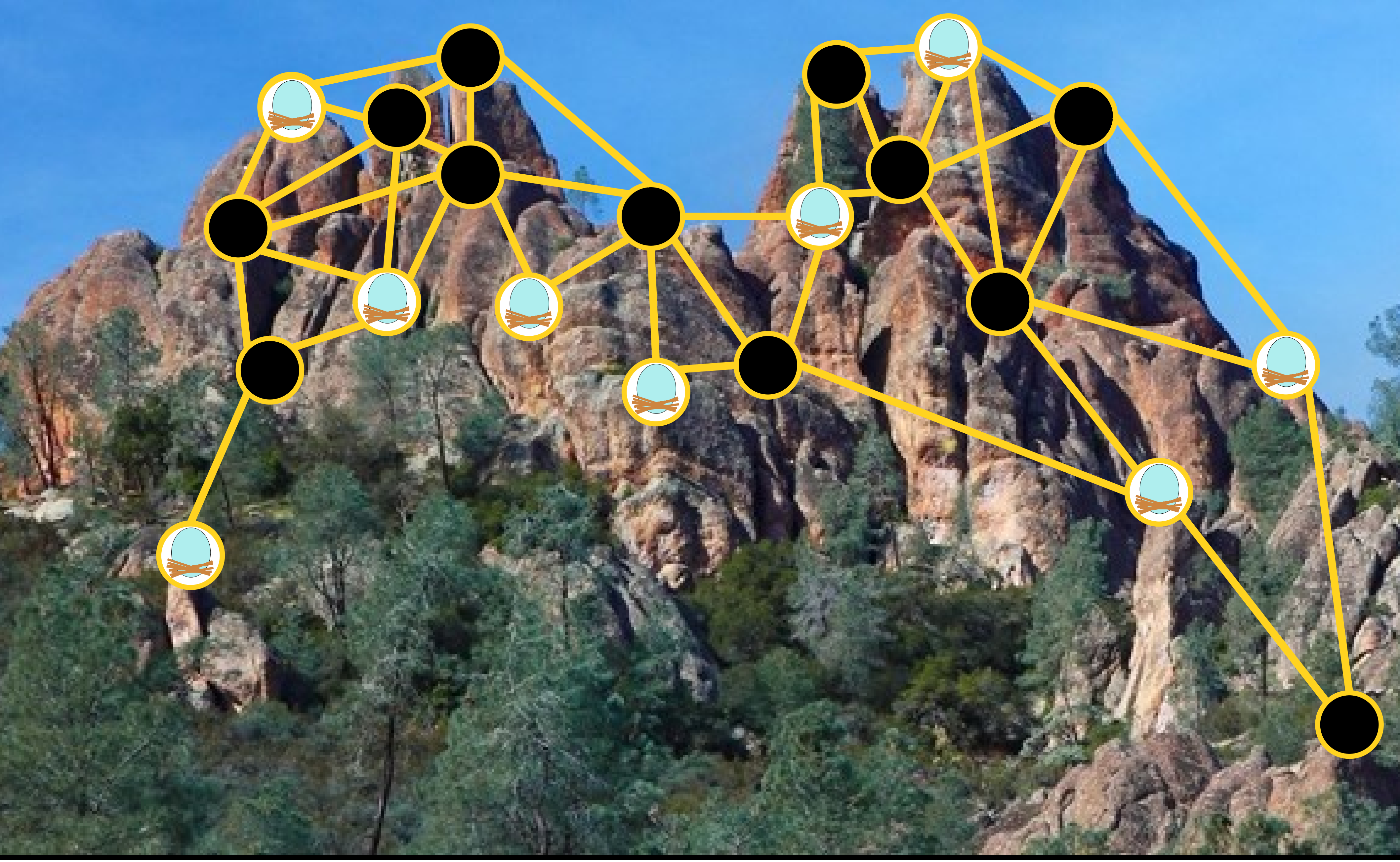
A Separate Motivating Problem



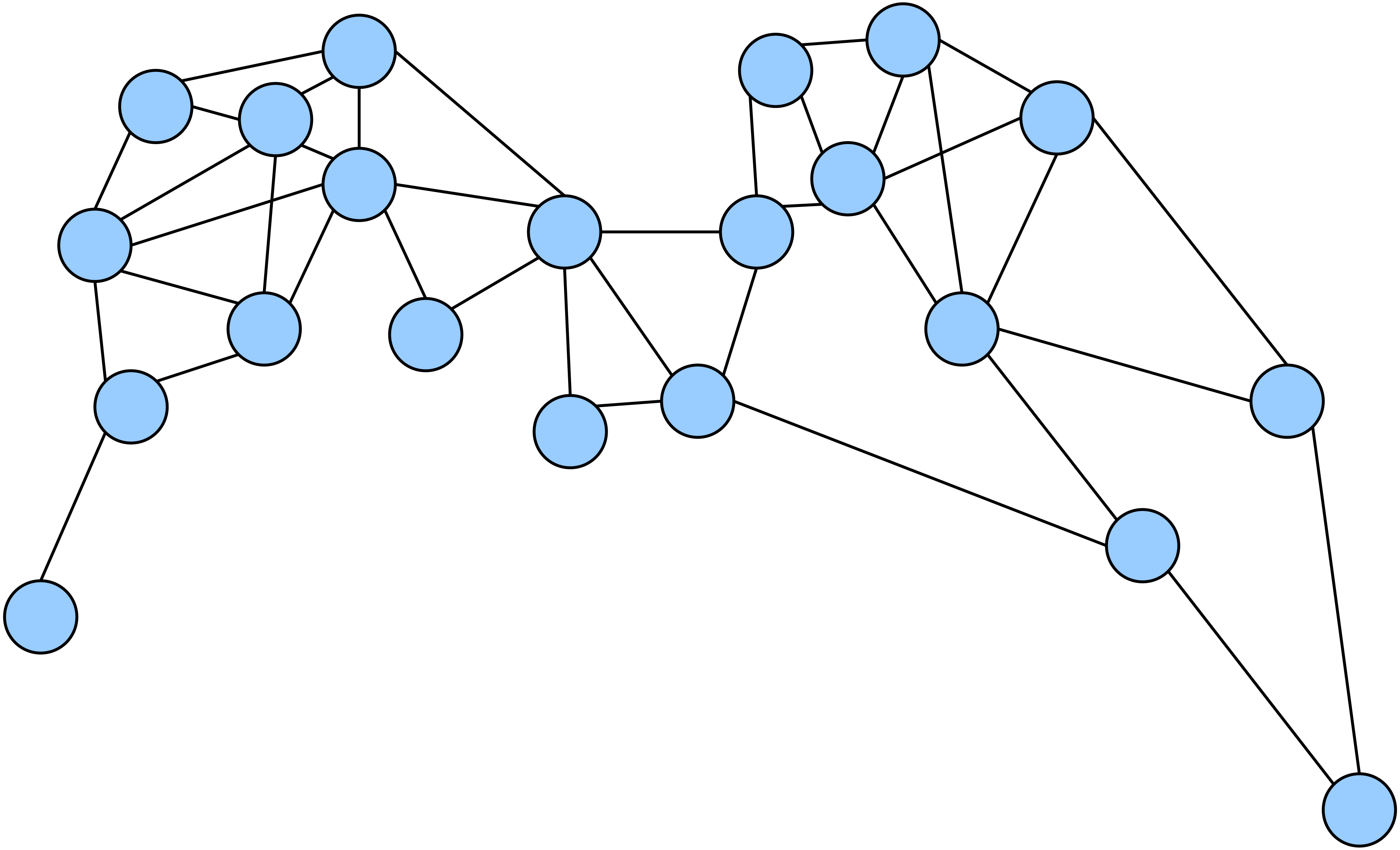
Set up nests for the California condor. Condors are territorial and won’t nest if they can see other condors.



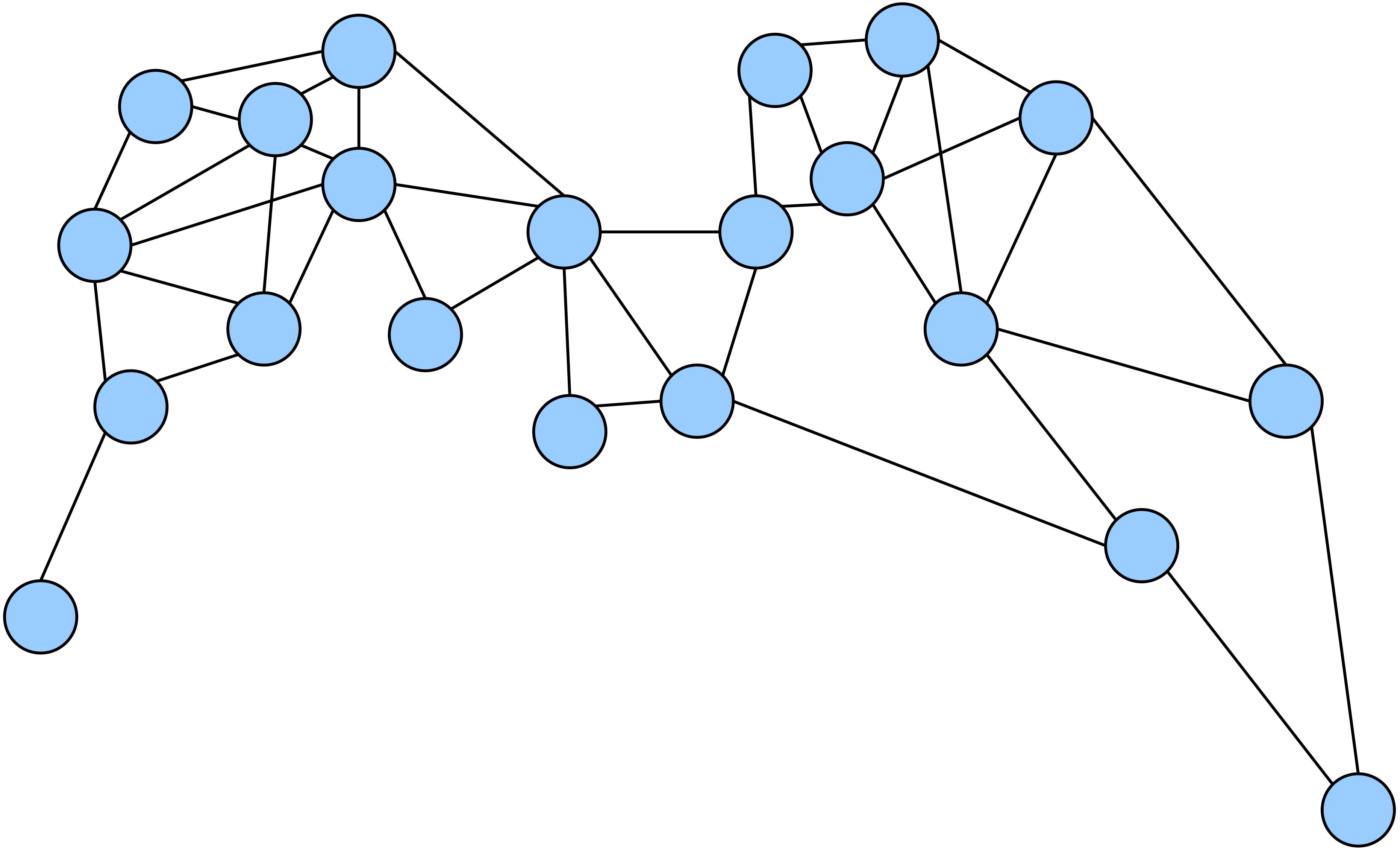
Set up nests for the California condor. Condors are territorial and won’t nest if they can see other condors.



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

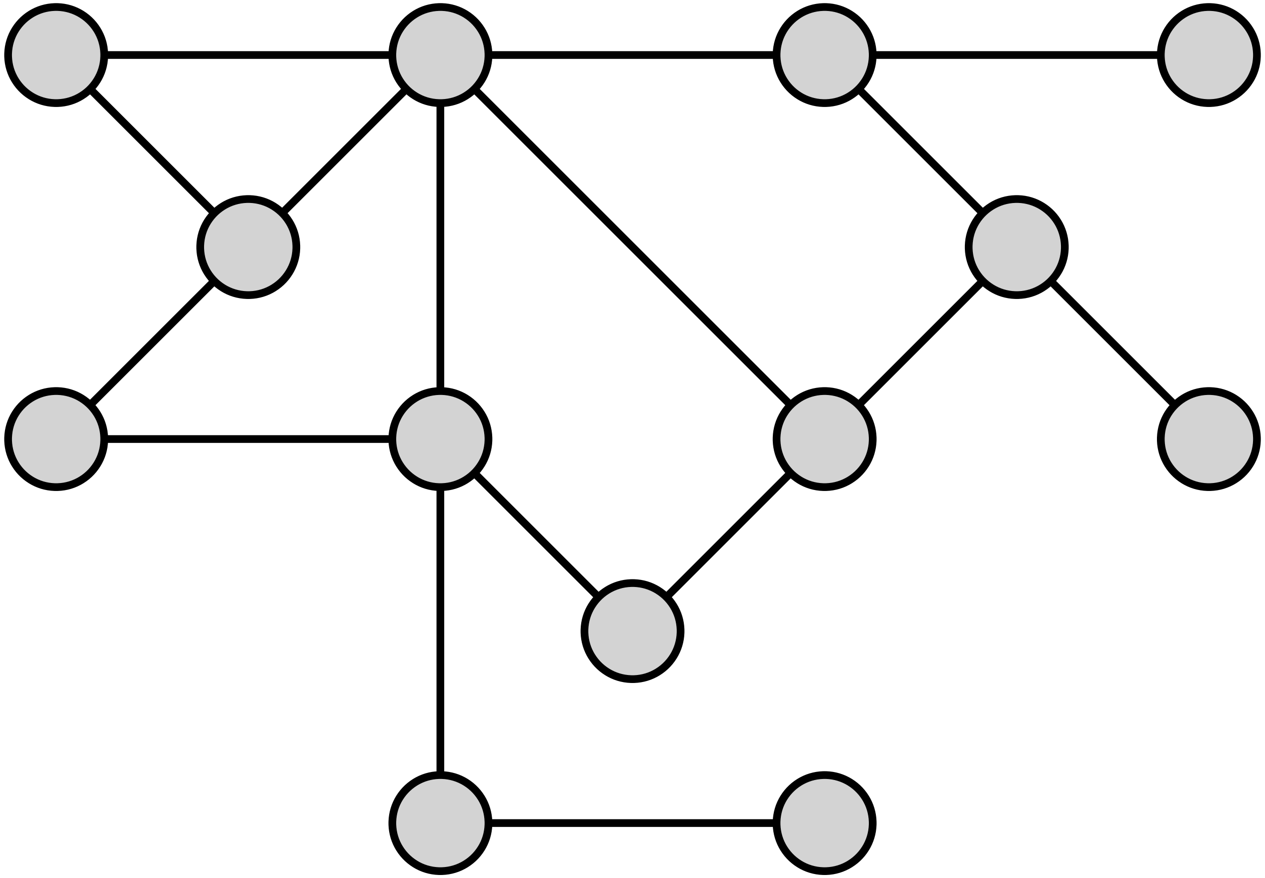
- If $G = (V, E)$ is an (undirected) graph, then an ***independent set*** in G is a set $I \subseteq V$ such that

$$\forall x \in I.\ \forall y \in I.\ \{x, y\} \notin E.$$

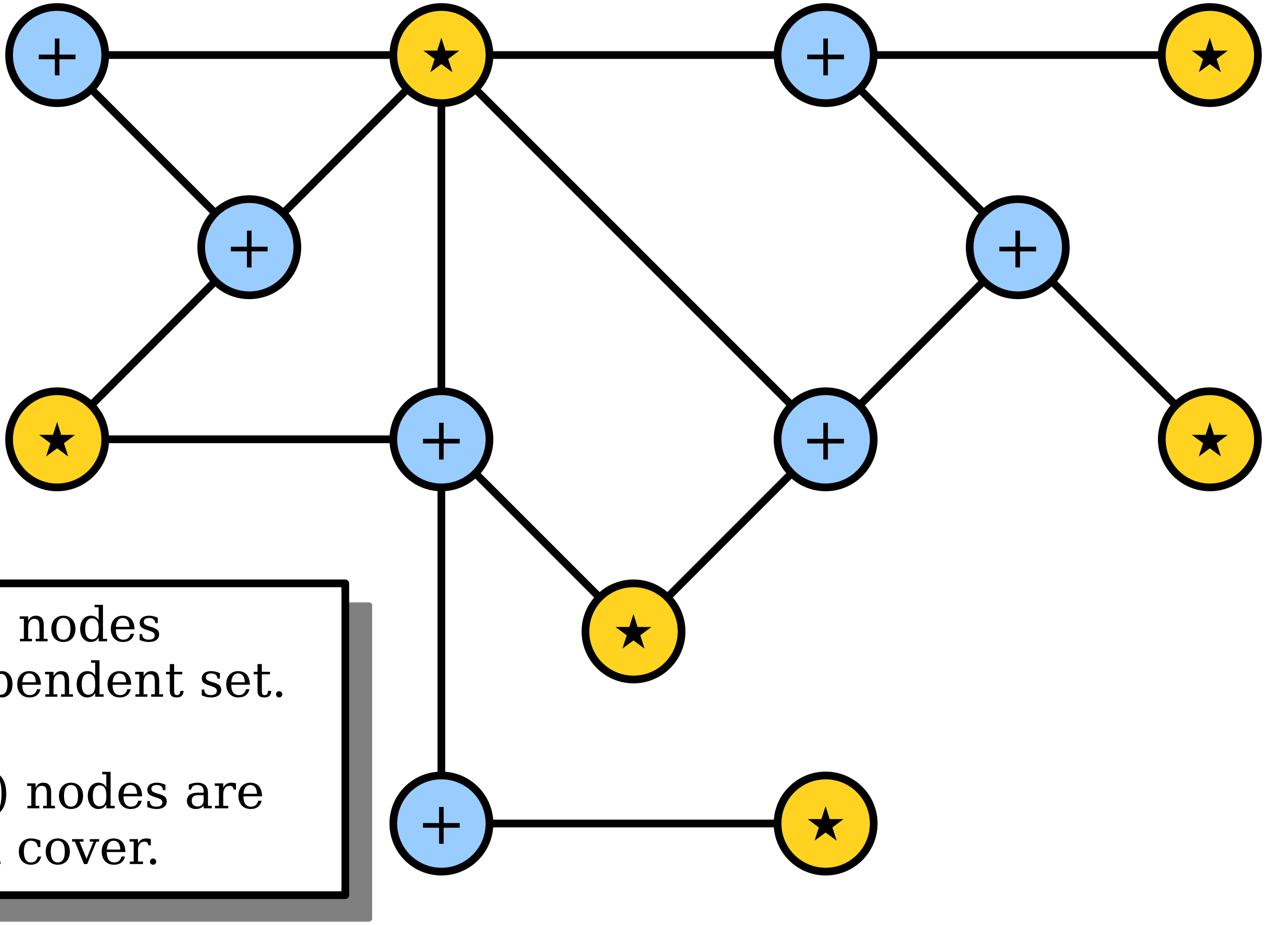
("No two nodes in I are adjacent.")

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

A Connection



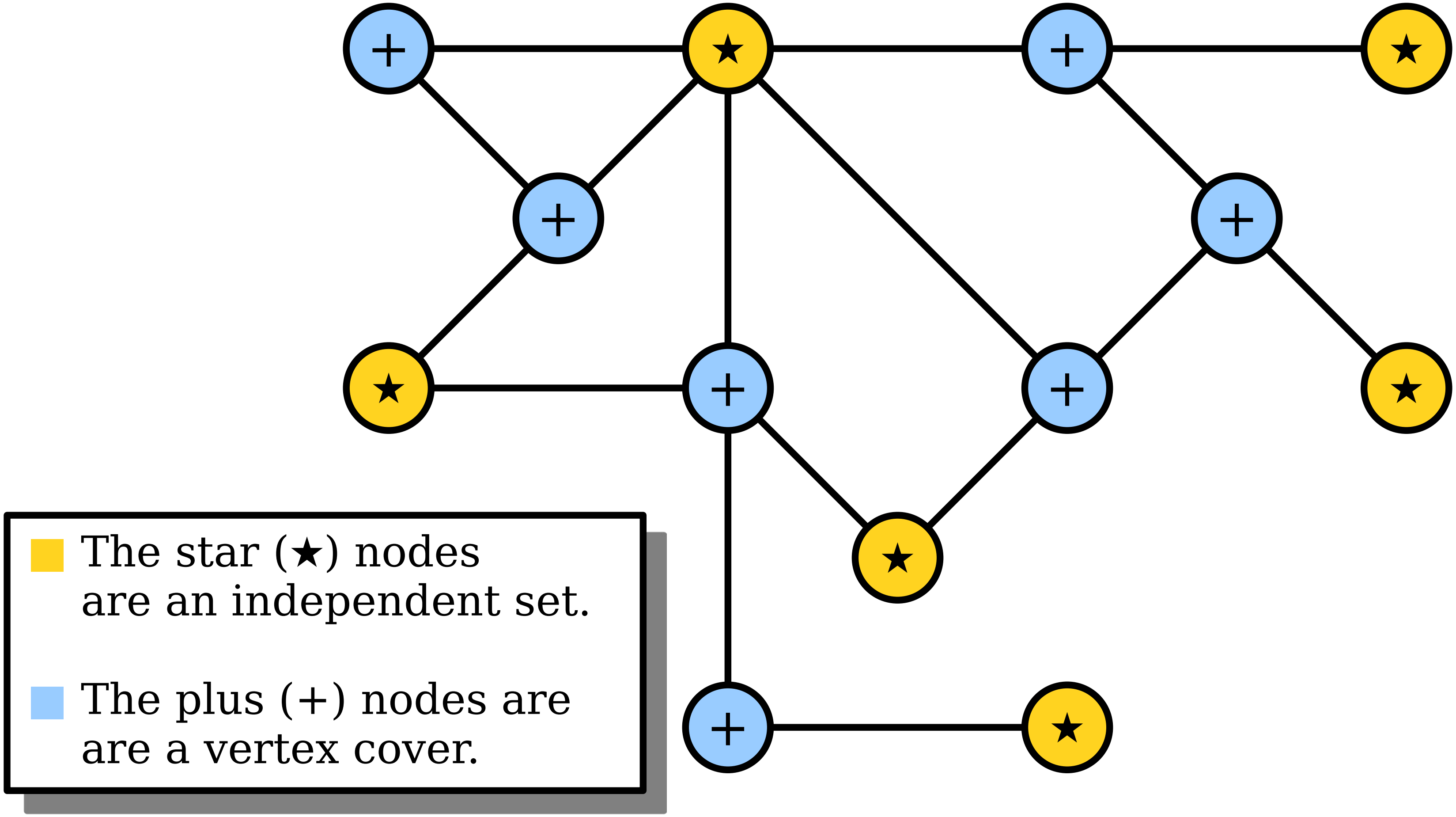
Independent sets and vertex covers are related.



🟨 The star (★) nodes
are an independent set.

🟦 The plus (+) nodes are
are a vertex cover.

Independent sets and vertex covers are related.



🟨 The star (★) nodes are an independent set.

🟦 The plus (+) nodes are are a vertex cover.

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

G is a graph.

C is a vertex cover of G.

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

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.

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.

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

$x \in V$ and $x \notin C$.

$y \in V$ and $y \notin C$.

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 in G.

What We're Assuming

G is a graph.

C is a vertex cover of G.

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

$x \in V$ and $x \notin C$.

$y \in V$ and $y \notin C$.

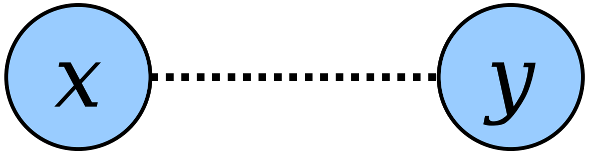
What We Need To Show

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

$\forall x \in V - C.$

$\forall y \in V - C.$

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



If this edge exists, at least one of x and y is 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 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 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.

See appendix for this proof!

Be sure to check this one out and follow through with the negations of the statements above.

Finding an IS or VC

- The previous theorem means that finding a large IS in a graph is equivalent to finding a small VC.
  - If you've found one, you've found the other!
- ***Open Problem:*** Design an algorithm that, given an *n*-node graph, finds either the largest IS or smallest VC "efficiently," where "efficiently" means "in time $O(n^k)$ for some $k \in \mathbb{N}$."
  - There's a $1,000,000 bounty on this problem – we'll see why in Week 10.

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

- ***Paths and Trails***
  - Walking from one point to another.
- ***Local Area Networks***
  - The building blocks of the internet.
- ***Trees***
  - A fundamental class of graphs.

Appendix

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.

Taking Negations

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

$$\begin{aligned}&\forall u \in V.\ \forall v \in V.\ (\{u, v\} \in E \to \\ &\qquad u \in C \quad \lor \quad v \in C \\ &)\end{aligned}$$

Taking Negations

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

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

- This says “there is an edge where both endpoints aren’t in *C*.”

Taking Negations

- What is the negation of this statement, which says “*V* – *C* is an independent set?”

$$\forall u \in V - C.\ \forall v \in V - C.\ \{u, v\} \notin E$$

Taking Negations

- What is the negation of this statement, which says “V – C is an independent set?”

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

- This says “there are two adjacent nodes 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 in G.

What We're Assuming

G is a graph.

C is a not a vertex cover of G.

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

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

What We Need To Show

$V - C$ is not an ind. set in G.

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

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

G is a graph.

C is a not a vertex cover of G.

$u \in V - C$.

$v \in V - C$.

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

What We Need To Show

$V - C$ is not an ind. set in G.

$\exists x \in V - C$.

$\exists y \in V - C$.

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

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.

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. ■