Graphs

CS 103ACE Day 6 – 4/26/24

Agenda:

- Review graph definitions
- Understand how to apply first-order definitions on graphs

Announcements

- Take care of yourself! I'm here to support you this weekend :)
- Sign up for an optional 1:1 at calendly.com/103ace/week4 (or let me know if you can't make any of the times)
- More resources on the course website: set-builder notation guide, lectures 0-5 symbols reference

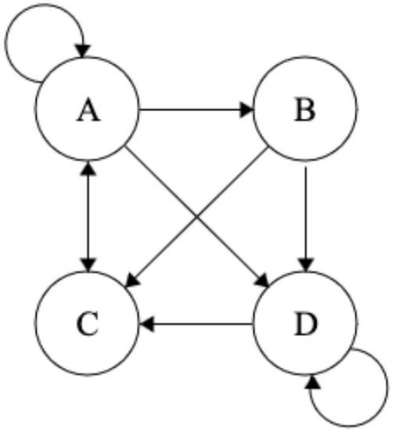
Midterm 1 Upcoming Events

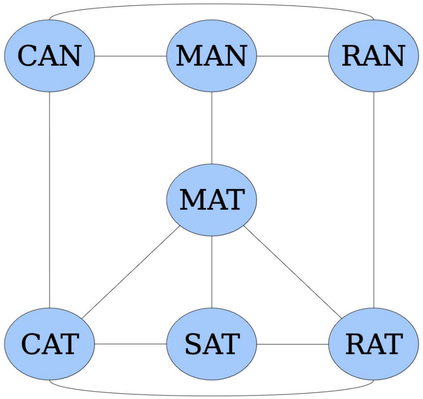
- **Friday**: ACE review session 1
- **Saturday**: no ACE events, but feel free to Slack / email / post on Ed with any questions as you review!
- **Sunday**: ACE review session 2
- **Monday**: ACE section and office hours, Stanley's Q&A

- Things to try to do this weekend:
  - Make a notes sheet
  - Take one of the practice exams on paper in a test-like environment
- Tip: use Tuesday for handling other things in your life, chilling, and managing test anxiety!

Graphs

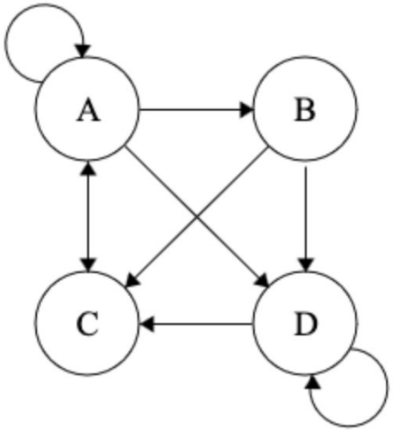
Graphs represent **nodes** (or **vertices**) connected by **edges**.

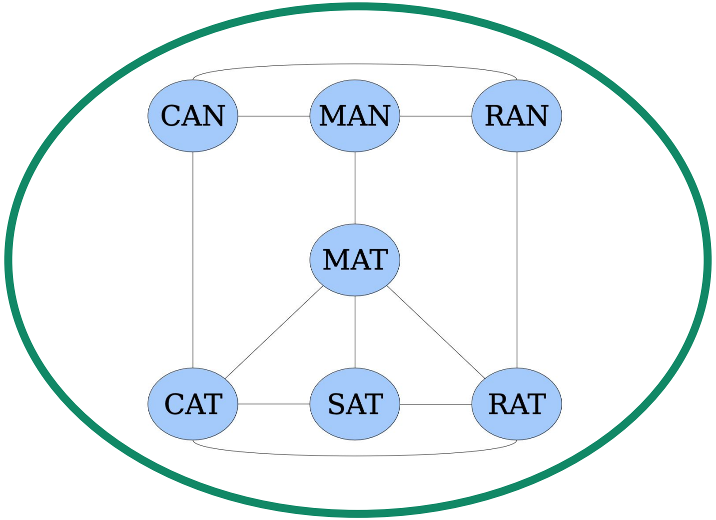




Graphs

Graphs represent **nodes** (or **vertices**) connected by **edges**.





We will mostly be talking about **undirected graphs**,
which have symmetric edges and don't allow self-loops.

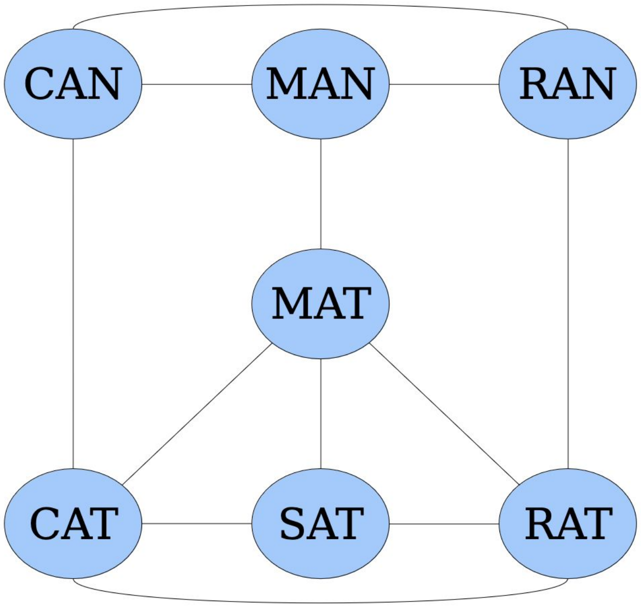
Graphs Concept Check

An undirected graph is formalized as **(V, E)** where:

- V is the set of all vertices
- E is the set of all edges
- An edge is an unordered pair of two vertices

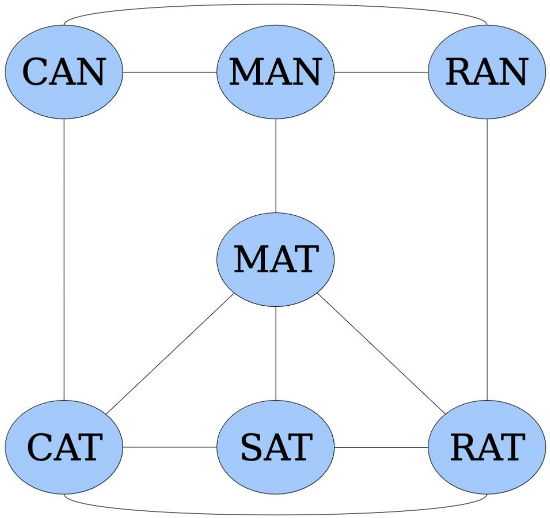
What does |V| mean?
What is |V| for this graph?

What does |E| mean?
What is |E| for this graph?



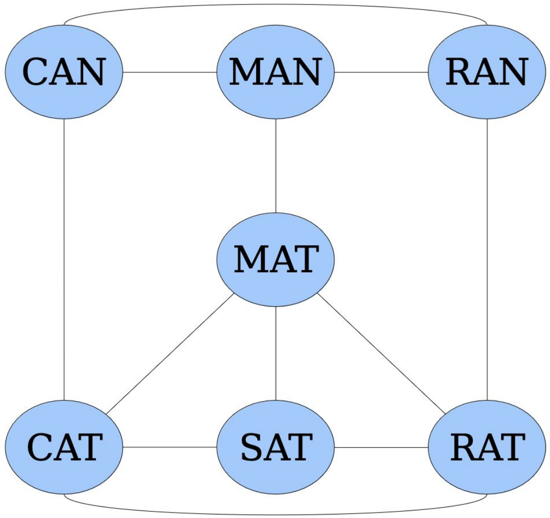
Traversing Graphs

- Two nodes u, v in a graph G = (V, E) are **adjacent** if {u, v} ∈ E
- **Walk**: a list of nodes where each node is adjacent to the next
  - N nodes → walk is length N-1
- Which of these are valid walks?
  - CAN, CAT, SAT, RAT, RAN, CAN
  - CAN
  - RAT, MAN



Traversing Graphs

- Two nodes u, v in a graph G = (V, E) are **adjacent** if {u, v} ∈ E
- **Walk**: a list of nodes where each node is adjacent to the next
  - N nodes → walk is length N-1
- Special types of walk!
  - **Closed Walk**: walk back to the same node you started with
  - e.g. CAN, CAT, SAT, RAT, RAN, CAN

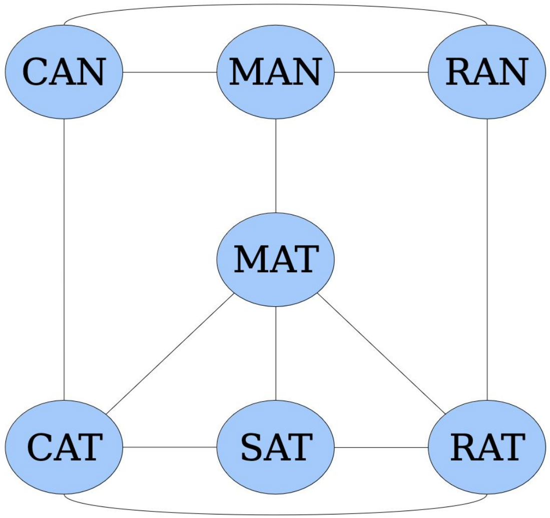


Traversing Graphs

- Two nodes u, v in a graph G = (V, E) are **adjacent** if {u, v} ∈ E
- **Walk**: a list of nodes where each node is adjacent to the next
  - N nodes → walk is length N-1
- Special types of walk!
  - **Closed Walk**: walk back to the same node you started with
  - **Path:** walk with no repeats
  - **Cycle**: a closed walk with no repeats of nodes or edges, except it goes back to the same node it started with (think “closed path”)

Traversing Graphs

Can you come up with a length 7 cycle?

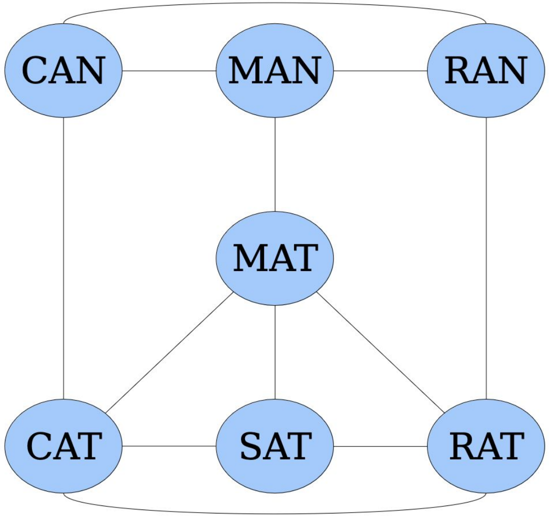


Graph Connectedness

- A node v is **reachable** from a node u if there is a path from u to v
- A graph is **connected** if any node is reachable from any other
- A **connected component** is a set of nodes that are all reachable from each other

Is this graph connected?

How many connected components are there?



Vertex Covers

A **vertex cover** C is a subset of nodes such that:

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

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

How would we show something is not a vertex cover?

Independent Sets

An **independent set** I is a subset of nodes such that

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

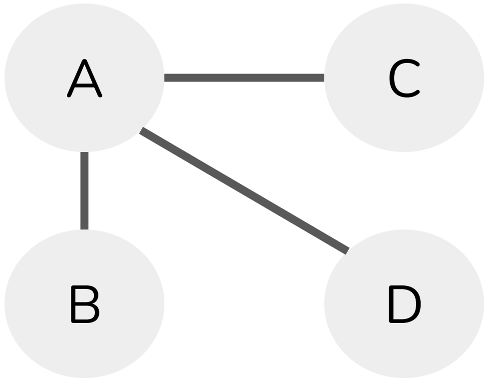
("No two nodes in I are adjacent.")

How would we show something is not a vertex cover?

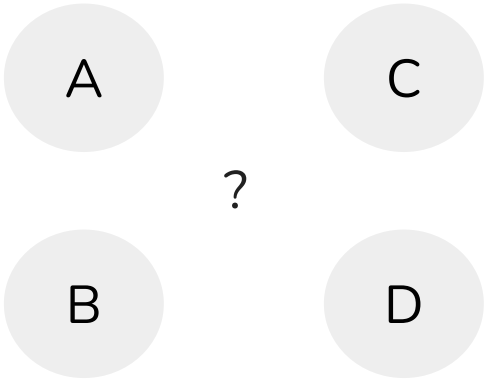
Complements

A graph $\mathbf{G}$'s **complement** $\mathbf{G^C}$ adds edges between any nodes that didn't have an edge in G, and removes all the original edges.

What's the complement of this graph?



$\mathbf{G}$

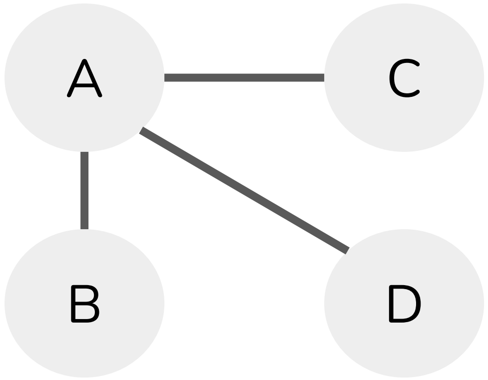


$\mathbf{G^C}$

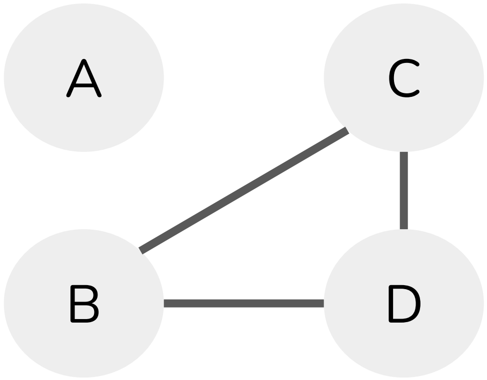
Complements

A graph $\mathbf{G}$'s **complement** $\mathbf{G^C}$ adds edges between any nodes that didn't have an edge in G, and removes all the original edges.

What's the complement of this graph?



$\mathbf{G}$



$\mathbf{G^C}$

Post-section recommendations

- Review for the midterm, and keep me in the loop!