Lecture 1 - Introduction to Graph Theory

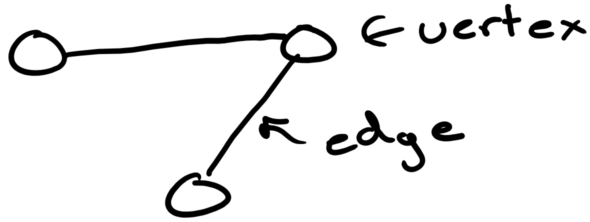
Sunday, January 11, 2026 5:42 PM

Things Slota hates**: dongles**

Lecture topics:
- What is a graph?
- History of graph theory
- Basic definitions and terminology
- Graph classes and basic topologies

Q: What is a graph?

- Discrete objects (vertices)
- Some modeled interaction (edge)



Real graphs: occur everywhere

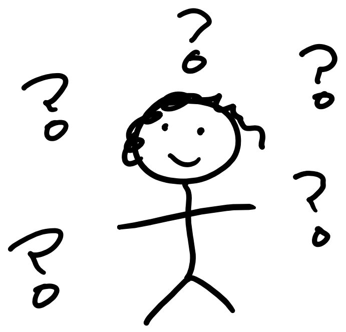
Biology: PPI networks

Sociology: social networks

Technology: Internet

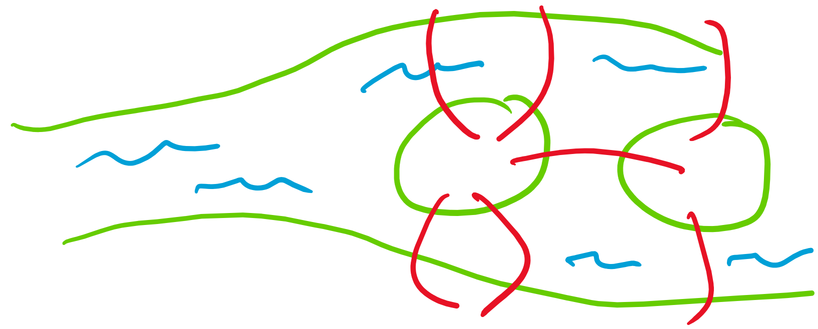
History of Graph Theory

100s of years ago:



Euler

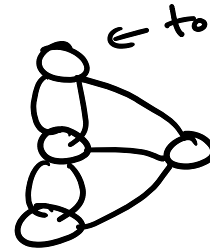
Bridges of Königsberg



Euler's Q: Can I start on one bank of the river, travers each bridge exactly once, and return back to my starting point?

Solution: invent graph theory

→ graph model



Answer: Not possible

(Euler Tour)

⇒ Common theme: real graphs motivating theoretic study

---

② basic definitions

Our basic definitions

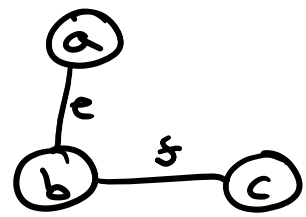
A <u>graph</u> is a tuple of

<u>vertices</u> and <u>edges</u>

(vertex set) (edge set)

$$G = \{V(G), E(G)\}$$

G:



$V(G) = \{a, b, c\}$

$E(G) = \{e = (a,b), f = (b,c)\}$

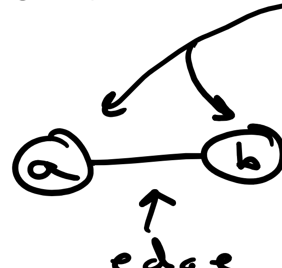
note how we label edges

$|V(G)| = 3$

cardinality of $V(G)$

More terminology

An edge in G has two <u>endpoints</u>

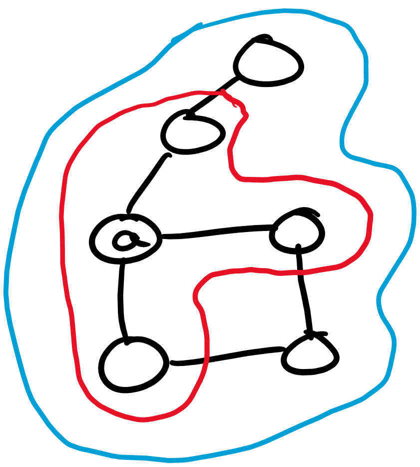


That edge is <u>incident</u>

That edge is <u>incident</u>
on those endpoints

Those two endpoints
are <u>adjacent</u>

Those two endpoints
are <u>neighbors</u>



1-hop neighborhood of a

2-hop neighborhood of a

→ note: a is not always included

→ key idea: definitions are often field/application specific

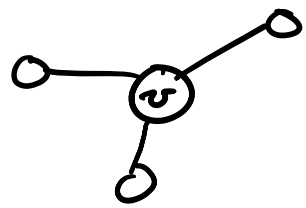
$N(v)$ = the neighborhood of v

→ set of vertices that v is adjacent to

The <u>degree</u> of some vertex v
is the number of edges incident

on v

$d(v)$ = degree of v



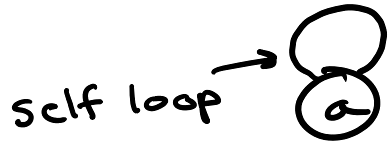
$d(v) = 3$

$d(v) \stackrel{?}{=} |N(v)|$

(depends on <u>simple</u> vs. <u>multi</u> graphs)

<u>Graph class:</u> a set of all possible graphs having some set of properties

<u>simple</u> graphs: graphs with no self loops or multi-edges



<u>loopy</u> graphs: can have self loops

<u>multi</u>-graphs: can have multi-edges

loopy multi-graphs: can have self loops and can have multi-edges

Finite graphs: have a fixed/finite vertex and edge cardinality

---

Graph order and size

$$\text{order} = |V(G)| = n = |V|$$

$$\text{size} = |E(G)| = m = |E|$$

used for simplicity if G is not ambiguous

if $|V| = 0$ and $|E| = 0$

→ null graph

if $|V| = 1$ and $|E| = 0$

→ trivial graph

if $|V| \geq 1$ and $|E| = 0$

→ empty graph

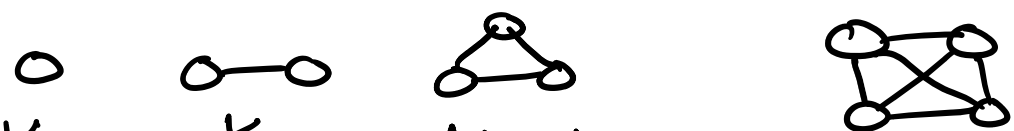
Note: these graphs often serve as a basis for inductive proofs

Note: These graphs often serve as a basis for inductive proofs

---

Basic graph configurations

(simple)
clique K_n: a graph on n vertices where all vertices are pairwise adjacent

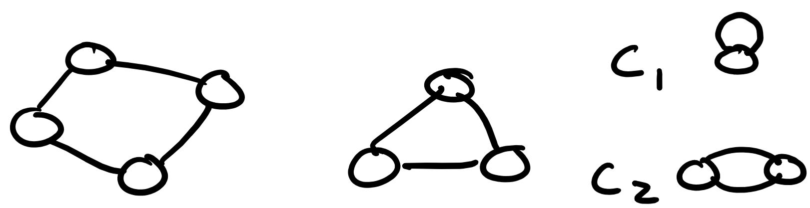


K_1 K_2 K_3 = triangle K_4

(each vertex/edge appears once)
path P_n:

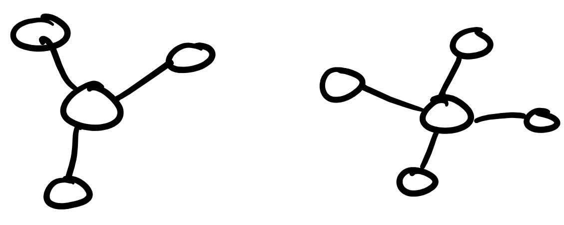
P_4 P_3

cycle C_n:



C_4 C_3 C_1 C_2

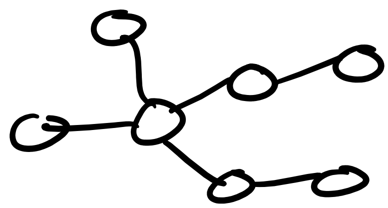
Star graphs:



S_4 = claw S_5

tree graphs: a <u>connected</u> and

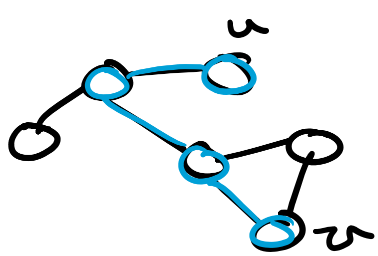
Tree graph: a connected and acyclic undirected simple graph



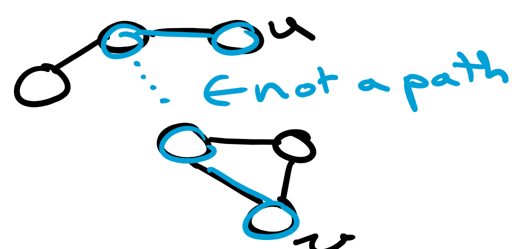
Connected graph G:

$\forall u,v \in V(G): \exists\ u,v\text{-path}$

for all ↗   in ↑   there exists ↑   a path from u to v ↑

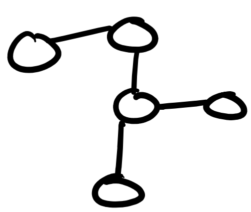


G is connected

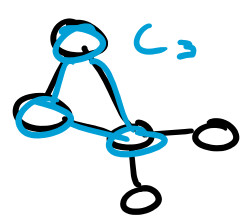


G' is disconnected

acyclic graph: contain no cycles



T is acyclic

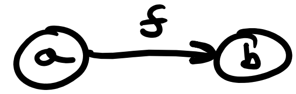


T' is cyclic

undirected graph: edges have no

<u>Undirected</u> graph: edges have no associated direction

undirected



directed

<u>directed</u> graph: all edges have some explicit direction

$f = (a, b)$