Introduction to Network Model

WAN DANG

Draft Edition

01/05/2026

Contents

Chapter 1

Motivation, and introduction to graph theory

1.1 Aspects of Networks

1.1.1 Overview: The Connected World

A network is a fundamentally defined as a pattern of interconnections among a set of objects. There has been a rising public interest in the complex "connectedness of modern society", driven by this idea.

Areas where network appear

1. Social networks: collections of social ties (e.g., friendships, business relationships) have increased in complexity due to technological advances like global communication and digital interaction, weakening traditional geographic limitations.
2. Information networks: the information consumed by people has a networked structure. Understanding any piece of information requires knowing how it refers to or is endorsed by other pieces within a large network of links.
3. Technological and economic systems: these systems rely on networks of enormous complexity, making their behavior hard to predict and susceptible to disruptions that can spread, turning localized breakdowns into cascading failures or financial crises.
4. Operational networks: many organizations depend on complex networks of people, resources, and processes to function effectively. Understanding and optimizing these networks is crucial for efficiency and resilience. Examples include networks of suppliers for global manufacturing, networks of users for websites, and networks of advertisers for media companies.

1.1.2 Structure and Behavior

Network structure

In its simplest form, a network is any group of objects where some pairs are connected by *link*. The objects are often shown as small circles, and the links connecting pairs are shown as lines between the circles. Such a representation is called a *graph* in mathematics, and the objects and links are called *nodes* (or vertices) and *edges*, respectively.

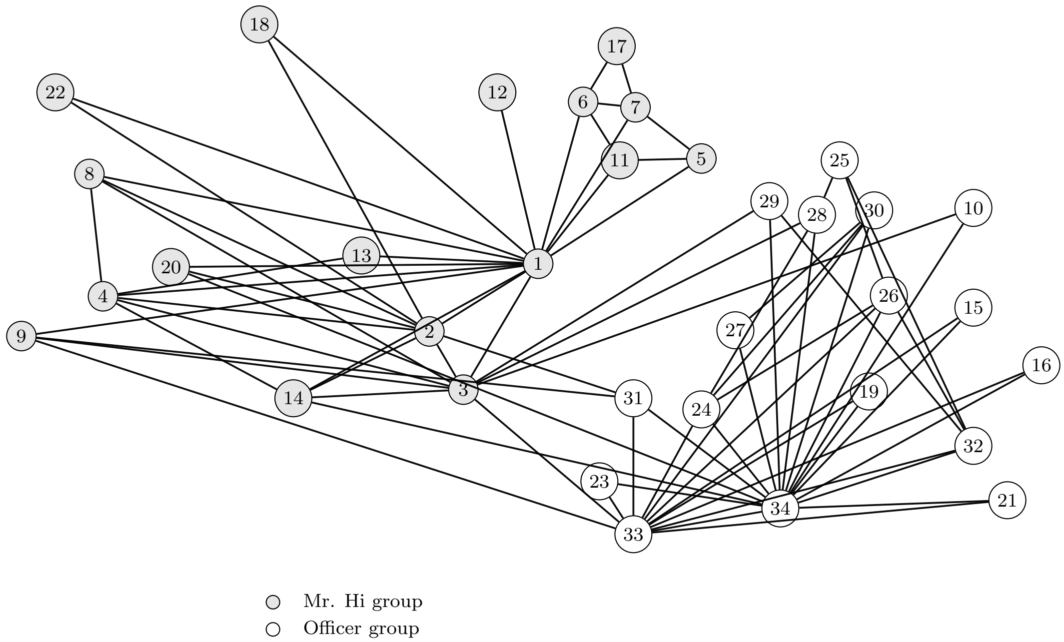


Figure 1.1: A social network of 34 members divided into two groups: Mr. Hi and Officer. Edges represent friendships between members.

Tip 1.1.1: Network Drawings

Networks often show great visual complexity, sometimes featuring central "cores", and sometimes splitting into multiple tightly-linked regions. Participants can be central or peripheral, or they may straddle the boundaries between tightly-linked regions. Understanding the structure of a network can provide insights into its function and behavior.

Behavior and dynamics

The "connectedness" of a complex system involves two related issues:

1. Connectedness of Structure: who is linked to whom.
2. Connectedness of Behavior: the fact that an individual's action have implicit consequences for everyone else in the system.

In network settings, individual actions must be evaluated with the expectation that the world — the interconnected system — will react. When individuals have incentives to achieve good outcomes, they must taks into account the strategic behavior of others when planning their own actions.

Tip 1.1.2: Aggregate Effects

When large groups are tightly interconnected, they respond in complex ways that are only visible at the population level, even if the underlying network is implicit (not directly observed). The rise in popularity of new products like YouTube or Flickr illustrates these collective feedback effects.

Confluence of ideas

Understanding highly connected systems requires combining ideas for reasoning about network structure, strategic behavior, and population feedback effects. This synthesis draws perspectives from several scientific disciplines:

- Computer Science, Applied Mathematics, and Operations Research: Provide the language to discuss the complexity of network structure and systems with interacting agents.
- Economics: Contributes models for the strategic behavior of interacting individuals.
- Sociology: Offers theoretical frameworks, particularly mathematical social network analysis, for the structure and dynamics of social groups.

1.2 Central Themes and Topics

1.2.1 Graph Theory (Theories of Structure)

Graph theory is defined as the study of network structure. Concepts drawn from social network analysis include:

Strong ties vs. Weak ties

- **Strong ties**: represent close and frequent social contacts and tend to be embedded in tightly-linked network regions.
- **Weak ties**: represent more casual and distinct social contacts and typically cross between tightly-linked regions. Weak ties can act as global "short-cuts", which rise to the phenomenon known as *six degrees of separation*.

Structural holes

These are gaps between parts of a network that interact very little, which suggests a strategic opportunity for navigating a social landscape. An individual who bridges a structural hole can access diverse information and resources, potentially gaining a competitive advantage.

Network Fissures

Networks can capture sources of conflict within a group. The theory of *structural balance* is used to reason about how conflicts or antagonism at a local level can cause fissures in a network structure.

1.2.2 Game Theory (Theories of Behavior)

Game theory provides a framework for reasoning about behavior when outcomes depend on the joint decisions made by all involved parties. The core framework is: individuals choose a *strategy* to maximize their own *payoff*, taking into account the strategies chosen by others.

Example 1.2.1: Driving Routes

The strategy is the choice of route, and the payoff is based on the resulting travel time, which is affected by traffic congestion from all drivers. A counter-intuitive outcome is *Braess's Paradox*, where adding resources to a network can actually decrease efficiency.

Example 1.2.2: Auctions

The strategy is how to bid, and the payoff is the difference between the value of the goods received and the price paid.

Equilibrium concepts are used to predict the outcome of strategic interactions. The most common is the *Nash equilibrium*, where no individual can improve their payoff by unilaterally changing their strategy.

1.2.3 Markets and Strategic interaction on Networks

Economics activity and trade naturally form networks where participants are linked by relationships (e.g., borrower-lender, trading partners).

- Network constraints: Network structures sometimes reflect constraints, such as institutional restrictions (regulations) or physical limitations (like geography, as seen in Medieval trade routes: Fig 1.2) which limit access between participants.
- Network position and power: The level of success for participants is affected by their positions in the network. Power depends on both the number of connections and the power of the other individuals connected to oneself.

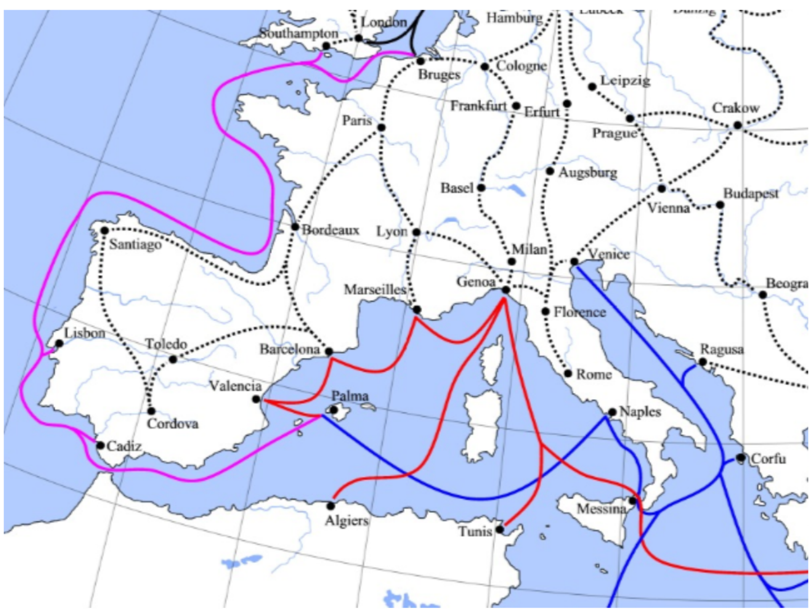


Figure 1.2: In some settings, such as this map of Medieval trade routes, physical networks constrain the patterns of interaction, giving certain participants an intrinsic economic advantage based on their network position.

1.2.4 Information Networks

Online information systems, such as the Web, possess a fundamental network structure.

- Community structure: link between Web pages reveal how they cluster into different communities. For instance, political blogs before the 2004 election separated into two distinct clusters corresponding to liberal and conservative perspectives.

- Prominence and ranking: Search engines (like Google) utilize network structure to evaluate page quality. Prominence is often recursively defined: a page is considered more prominent if it receives links from pages that are *themselves* prominent. This circular definition can be resolved during the concept of *equilibrium* in the link structure.

- Strategic interaction: The relationship between search engines and content creators is game-theoretic. Content creators constantly optimize their Web pages to achieve a high rank under the engines's current evaluation methods; thus, search methods must be developed considering these human feedback effects.

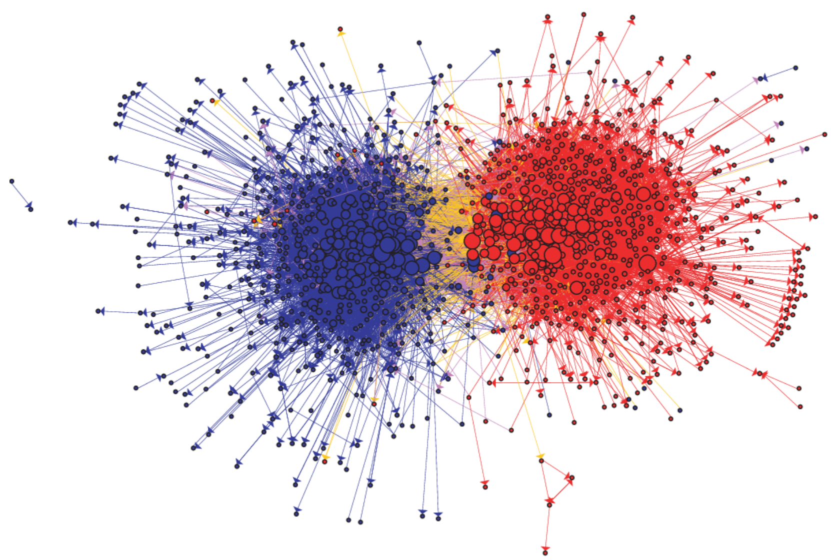


Figure 1.3: The links among Web pages can reveal densely-knit communities and prominent sites. In this case, the network structure of political blogs prior to the 2004 U.S. Presidential election reveals two natural and well-separated clusters.

1.2.5 Dynamics on Networks

Population Effects

Collective phenomena, such as the spread of new *social practices* (new beliefs, opinions, or technologies), following recurring patterns. We called that *mechanism of influence*, which describes how individuals change their behavior based on the actions of others in their network.

- Information: People may copy others because observed behavior conveys information. If many people use a product (like YouTube), it suggests they know something about its quality. This can lead to *information cascades*, where rational individuals follow the crowd, abandoning their private information.
- Direct benefit (network effects): There is a direct benefit to aligning one's behavior with others, regardless of whether the decision is optimal. For social media sites, value increases as more people join (more content, wider audience). *Network effects* amplify success, creating a "rich-get-richer" feedback process that is characteristic of the aggregate distribution of popularity.

Figure 1.4: Cascading adoption of a new technology or service (in this case, the social-networking site MySpace in 2005–2006) can be the result of individual incentives to use the most widespread technology — either based on the informational effects of seeing many other people adopt the technology, or the direct benefits of adopting what many others are already using.

Structural Effects

When individuals are motivated to adopt the behavior of their immediate neighbors, *cascading* effects can result, spreading outward from a small set of initial adopters.

- Diffusion barriers: The diffusion of technologies can be blocked by the boundary of a *densely-connected cluster* (a "closed community") that is highly resistant to outside influences. See Fig 1.5.

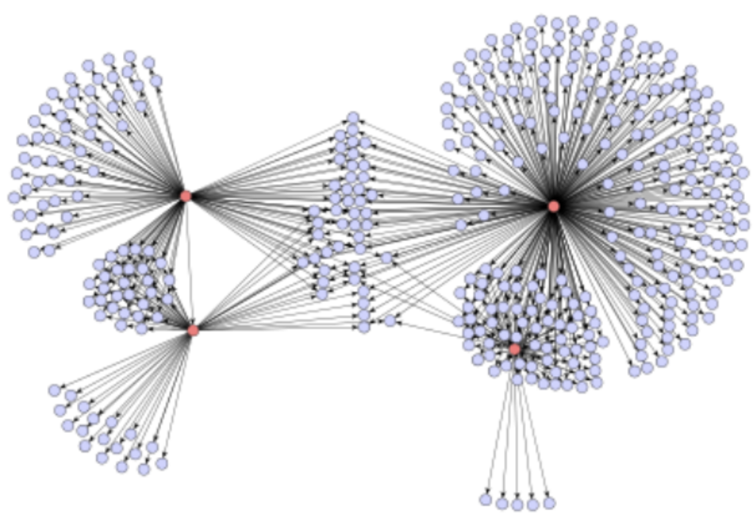


Figure 1.5: When people are influenced by the behaviors their neighbors in the network, the adoption of a new product or innovation can cascade through the network structure. Here, e-mail recommendations for a Japanese graphic novel spread in a kind of informational or social contagion.

- Social contagion: Cascading behavior is often referred to as *social contagion*, analogous to a biological epidemic. While social contagion involves decision-making and biological contagion involves pathogen transmission, the network-level dynamics are similar. See Fig 1.6.

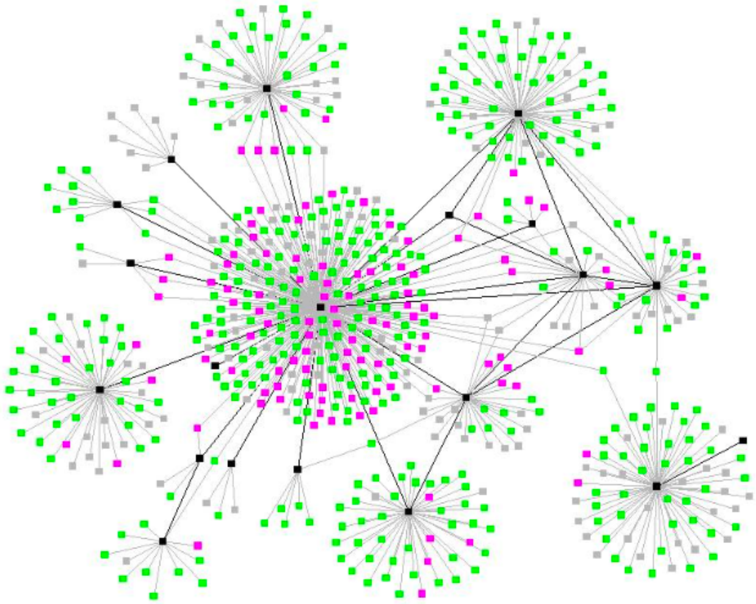


Figure 1.6: The spread of an epidemic disease (such as the tuberculosis outbreak shown here) is another form of cascading behavior in a network. The similarities and contrasts between biological and social contagion lead to interesting research questions.

1.2.6 Institutions and Aggregate Behavior

Institutions are the rules, conventions, or mechanisms designed by a society to synthesize individual actions into a pattern of aggregate behavior.

Example 1.2.3: Markets as Aggregators

Markets synthesize individuals' beliefs. For example, the price in a financial market aggregates beliefs about the value of assets. *Prediction markets* use a market mechanism where asset prices reflect an aggregate estimate for the probability of future events (e.g., elections).

Voting systems aggregate individual preferences over subjective choices. The process of producing a cumulative social preference from conflicting individual priorities is inherently difficult, as formalized by *Arrow's Impossibility Theorem*.

1.3 Introduction to Graph Theory

1.3.1 Basic Definitions and Components

The Graph Structure

A *graph* is a mathematical model for specifying relationships among a collection of items. It consists of two sets: $G(V, E)$.

- Vertices or nodes (V): the set of objects in the graph, often represented by small circles.
- Edges or links (E): connections between pairs of vertices. Edges are typically drawn as lines connecting the nodes.

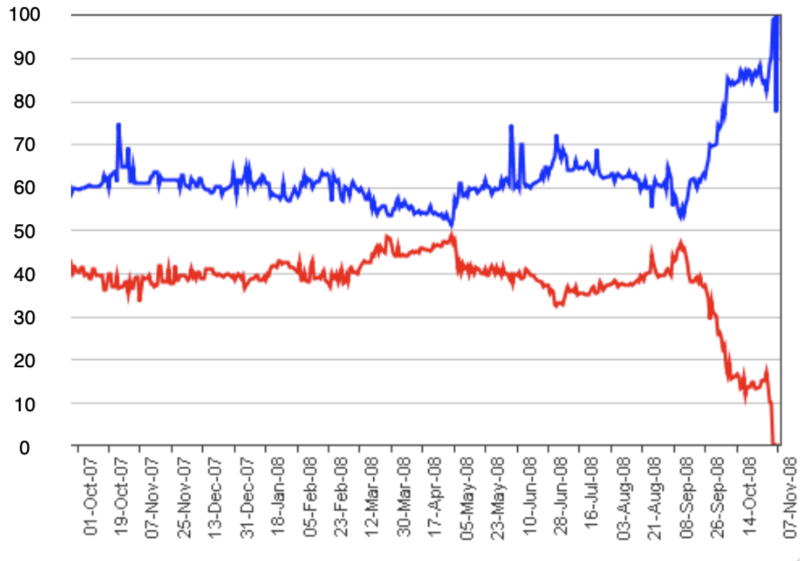


Figure 1.7: The plot here depicts the varying price over time for two assets that paid \$1 in the respective events that the Democratic or Republican nominee won the 2008 U.S. Presidential election.

Graph Types based on Edge direction

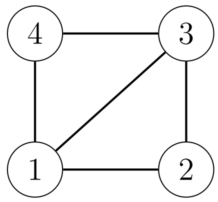
Graphs are categorized based on whether the relationships they model are symmetric or asymmetric. See table 1.1 for a summary.

Graph Type	Definition	Edge Pairs	Example
Undirected Graph	The relationship is symmetric; the edge simply connects the nodes to each other. Elements of E are unordered pairs.	$(u, v) = (v, u)$	Facebook friendships.
Directed Graph (Digraph)	The relationship is asymmetric, meaning a link goes from one node *to* another, and direction matters. Elements of E are ordered pairs (u, v). By convention, (u, v) points to v.	(u, v) is distinct from (v, u)	Twitter follower networks; who-calls-whom phone networks.

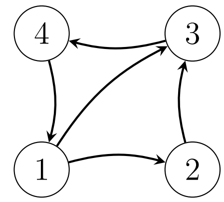
Table 1.1: Types of Graphs based on Edge Direction

Tip 1.3.1: Mutual edges

n directed graphs, it is possible for two nodes to have edges pointing to each other, known as *mutual edges*. This indicates a bidirectional relationship between the nodes, such as mutual friendships on social media platforms.



(a) Undirected graph on 4 nodes

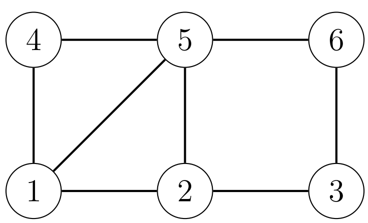


(b) Directed graph on 4 nodes

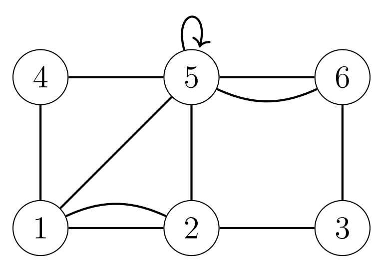
Figure 1.8: Examples of (a) undirected and (b) directed graphs on four nodes.

Graph Types based on Edge properties

- Simple graph: Graphs that do not contain self-loops (an edge connecting a node to itself) or multi-edges (multiple edges between the same two nodes). Most analysis focuses on simple graphs.
- Multigraph: Graphs that may contain self-loops and/or multi-edges. Multi-edges are often encoded as edge weights (counts).
- Weighted graph: Graphs where edges are labeled with numerical values. The length of a path in a weighted graph is the sum of weighted of the traversed edges.



(a) Simple graph on 6 nodes



(b) Multigraph on the same 6 nodes

Figure 1.9: A simple graph versus a multigraph on the same vertex set. The multigraph includes parallel edges and a self-loop.

Example 1.3.1: Network Representations as Graphs

Different types of networks can be represented using various graph structures. For instance, the World Wide Web is often modeled as a directed multigraph with loops, while citation networks are directed and acyclic. Collaboration networks are typically undirected and unweighted. Table 1.2 summarizes some common examples.

Network	**Graph Representations**
WWW	Directed multi-graph (with loops), unweighted
Citation Network	Directed, unweighted, acyclic
Collaboration Network	Undirected, unweighted
Mobile Phone Calls	Directed, weighted
Protein Interaction	Undirected multi-graph (with loops), unweighted

Table 1.2: Examples of networks and common graph representations.

1.3.2 Local Connectivity and Degree

Adjacency and Incidence

Two nodes are *adjacent* if they are directly connected by an edge. The set of nodes adjacent to a given node u is called the *neighborhood* of u, denoted $N(u)$.

An edge is *incident* to a node if the node is one of the edge's endpoints. For an edge (u, v), it is incident to both u and v. See Fig 1.10 for an illustration.

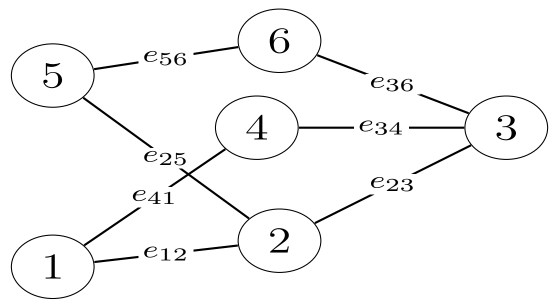


Figure 1.10: A simple undirected graph with vertex set $V = \{1, 2, 3, 4, 5, 6\}$ and edge set $E = \{e_{12}, e_{23}, e_{34}, e_{41}, e_{25}, e_{56}, e_{36}\}$. **Adjacency (vertex–vertex):** vertices 1 and 2 are adjacent because e_{12} joins them; vertices 1 and 3 are not adjacent because there is no edge between them. **Incidence (edge–vertex):** edge e_{12} is incident to vertices 1 and 2 (and is not incident to 3); the edges incident to vertex 2 are e_{12}, e_{23}, and e_{25}.

Vertex degrees

The *degree* (d_v) of a vertex v is its number of incident edges. High-degree vertices are often considered influential, central, or prominent. See Fig 1.11 for an illustration.

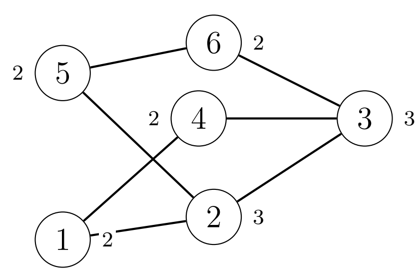


Figure 1.11: A simple undirected graph on $V = \{1, 2, 3, 4, 5, 6\}$ with vertex degrees shown. For example, $d(2) = 3$ because vertex 2 is adjacent to $\{1, 3, 5\}$, while $d(1) = 2$ because vertex 1 is adjacent to $\{2, 4\}$.

Definition 1.3.1: Neighborhood

The *neighborhood* $\mathcal{N}_i$ of a node i is the set of all its adjacent nodes.
The size of the neighborhood is equal to the degree, i.e., $|\mathcal{N}_i| = d_i$.

Degree formulas and properties

The following properties hold for an undirected graph:

Properties 1.3.1: Degree Properties

- Degree range: For any vertex v, $0 \le d_v \le n-1$, where n is the total number of vertices in the graph.
- Sum of degrees: The sum of the degrees of all vertices is equal to twice the number of edges:
$$\sum_{v=1}^{|V|} d_v = 2|E|.$$
- Odd degree count: The number of vertices with an odd degree must be an even number.

Proof. Consider a finite undirected graph $G = (V, E)$ (allowing neither direction nor multiple counting beyond multiplicity). Each edge $e = \{u, w\} \in E$ is incident to exactly two vertices, namely u and w.

Now count the number of *incidences* (vertex–edge pairs (v, e) where e is incident to v) in two ways:

- Fix a vertex v. There are exactly $d(v)$ edges incident to v, so the total number of incidences is $\sum_{v \in V} d(v)$.
- Fix an edge e. It contributes exactly 2 incidences (one for each endpoint), so the total number of incidences is $2|E|$.

Both expressions count the same set of incidences, hence

$$\sum_{v \in V} d(v) = 2|E|.$$

□

Degrees in Directed Graphs

In directed graphs (digraphs), vertices have two types of degrees:

- **In-degree** (d_v^i): the number of edges pointing *to* vertex v.
- **Out-degree** (d_v^o): the number of edges pointing *away* from vertex v.

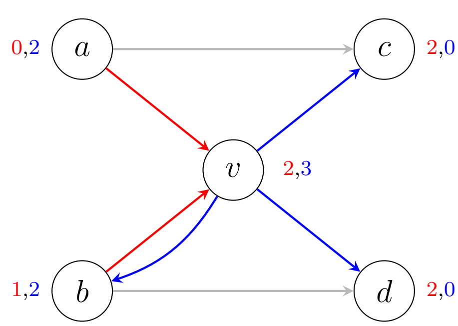


Figure 1.12: A directed graph where each vertex is annotated with its in-degree and out-degree as an ordered pair (d^-, d^+). Incoming edges are shown in red and outgoing edges in blue.

1.3.3 Global connectivity and Path

Movement in a Graph

Definition 1.3.2: Path, Walk, Cycle

A *path* in a graph is a consecutive sequence of distinct vertices $\{v_0, v_1, \dots, v_l\}$ such that v_i and v_{i+1} are adjacent. The length of the path is l.
A *walk* is similar to a path, but vertices and edges can be repeated. A walk that starts and ends at the same vertex is called a *closed walk*, or a *cycle* if no other vertices are repeated.

These definition generalize naturally to directed graphs.

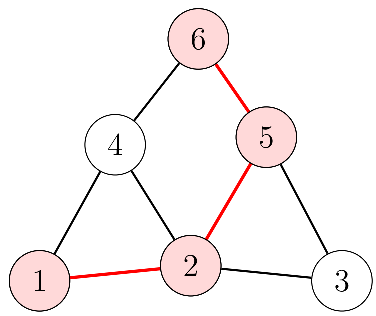


Figure 1.13: A graph with the path $1 \to 2 \to 5 \to 6$ highlighted in red.

Distance and Diameter

Definition 1.3.3: Distance and Diameter

The *distance* between two vertices u and v, denoted $d(u, v)$, is the length of the shortest path connecting them. If no path exists, the distance is defined to be infinite.
Diameter of a graph is the maximum distance between any pair of vertices in the graph:

$$\text{diameter}(G) = \max_{u,v \in V} d(u, v).$$

Efficient algorithms exist for computing shortest paths in both undirected and directed graphs, such as Dijkstra's algorithm for graphs with non-negative edge weights, Floyd-Warshall algorithm for dense graphs, and Bellman-Ford algorithm for graphs with negative edge weights.

Connectivity

Vertex v is *reachable* from vertex u if there exists a path from u to v.

Definition 1.3.4: Connected graphs (Undirected)

A graph is *connected* if every vertex is reachable from every other vertex. Removing a "bridge edge" (an edge whose removal increases the number of connected components) can disconnect the graph.

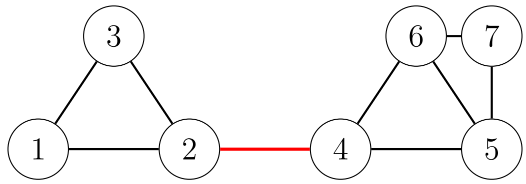


Figure 1.14: A connected undirected graph on 7 vertices. The red edge $(2, 4)$ is a *bridge*: removing it disconnects the graph into two components (the triangle on $\{1, 2, 3\}$ and the subgraph on $\{4, 5, 6, 7\}$).

Definition 1.3.5: Connected components

A *component* is a maximally connected subgraph. A maximal subgraph is one where adding any other vertex would ruin the connectivity. Disconnected graphs have two or more components.

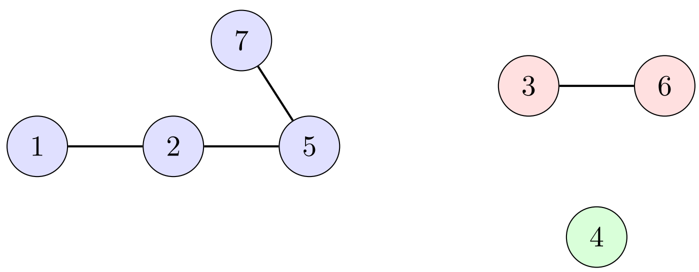


Figure 1.15: A disconnected graph with three connected components: $\{1, 2, 5, 7\}$, $\{3, 6\}$, and $\{4\}$. The subgraph on $\{3, 4, 6\}$ is not connected because vertex 4 is isolated from $\{3, 6\}$, while the subgraph on $\{1, 2, 5\}$ is not maximal since it can be enlarged to the connected component $\{1, 2, 5, 7\}$.

Disconnected graphs have 2 or more components, largest component often called giant component. Large real-world networks often exhibit one giant component.

Tip 1.3.2: Why do we expect to find a single giant component?

In many real-world networks, the presence of a single giant component can be attributed to the underlying connectivity patterns and the nature of interactions among nodes. Factors such as preferential attachment, where new nodes are more likely to connect to already well-connected nodes, and the small-world phenomenon, where most nodes can be reached from every other by a small number of steps, contribute to the formation of a giant component. Additionally, real-world networks often exhibit clustering and community structures that facilitate connectivity within the giant component.

Connectivity in Directed Graphs

Connectivity is more complex in directed graphs and has two main notions:

1. Strongly connected: A digraph is strongly connected if for every pair of vertices u and v, u is reachable from v (via a directed path) and v is reachable from u.

2. Weakly connected: A digraph is weakly connected if it remains connected after disregarding the edge directions (i.e., its underlying undirected graph is connected). Strongly connectivity implies weak connectivity.

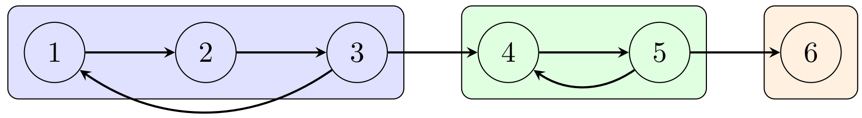
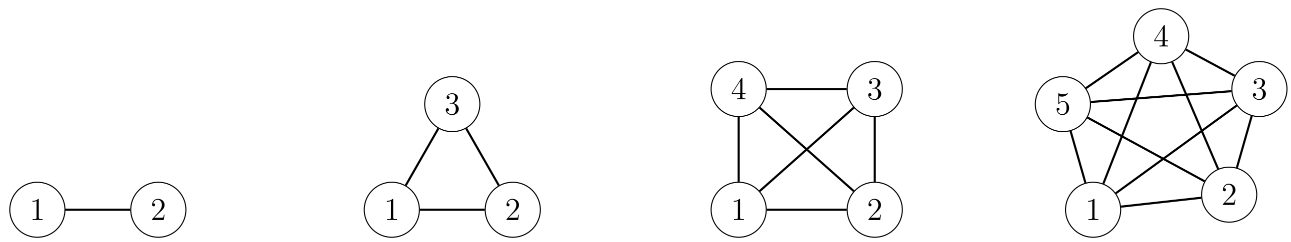


Figure 1.16: A directed graph that is *weakly connected* (ignoring arrow directions, all vertices lie in one connected component) but *not strongly connected* (there is no directed path from 6 back to earlier vertices, and no directed path from $\{4, 5\}$ back to $\{1, 2, 3\}$). The strongly connected components are highlighted: $\{1, 2, 3\}$, $\{4, 5\}$, and $\{6\}$.

1.3.4 Specialized Graph Structures

Complete Graphs and Cliques

A *complete graph* is a simple undirected graph in which every pair of distinct vertices is connected by a unique edge.



(a) K_2 (b) K_3 (c) K_4 (d) K_5

Figure 1.17: Complete graphs K_n for $n = 2, 3, 4, 5$.

Properties 1.3.2: Number of edges in complete graphs

The number of edges is equal to the number of vertex pairs:

$$\text{Number of edges in } K_n = \binom{n}{2} = \frac{n(n-1)}{2}$$

Proof. In the complete graph K_n, every pair of distinct vertices is connected by exactly one edge. Thus, counting edges is the same as counting unordered pairs of distinct vertices.

The number of ways to choose 2 vertices from n (order does not matter) is

$$\binom{n}{2} = \frac{n(n-1)}{2}.$$

Therefore $|E(K_n)| = \binom{n}{2} = \frac{n(n-1)}{2}$. □

A *clique* in a graph is a subset of vertices such that every two distinct vertices are adjacent. In other words, a clique is a complete subgraph. The size of the largest clique in a graph is called the *clique number*.

Regular graphs

A *regular graph* is a graph where every vertex has the same degree k. Such a graph is called a k-regular graph.

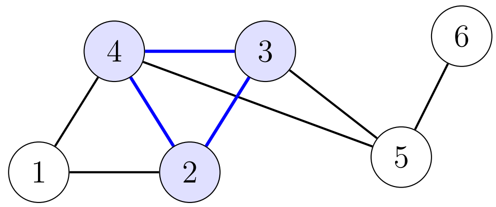
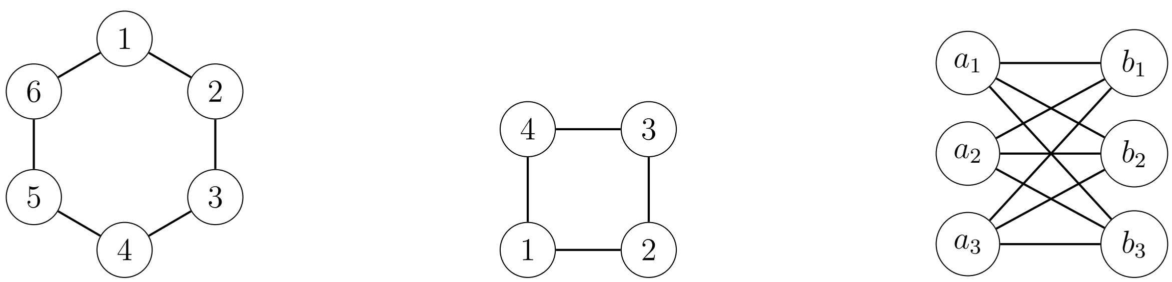


Figure 1.18: An example of a clique: vertices $\{2, 3, 4\}$ form a 3-clique (a K_3 subgraph), since every pair among them is connected by an edge.

The complete graph K_n is an example of a $(n-1)$-regular graph, since every vertex is connected to all other $n-1$ vertices. Cycles are examples of 2-regular graphs, as each vertex is connected to exactly two other vertices.



(a) C_6 (2-regular) (b) C_4 (2-regular) (c) $K_{3,3}$ (3-regular)

Figure 1.19: Examples of regular graphs: C_6 and C_4 are 2-regular; $K_{3,3}$ is 3-regular.

Regular graphs frequently arise in the study of crystal structures (physics/chemistry), pixel adjacency models in image processing (geo-spatial settings), and modeling opinion formation or information cycles.

Trees and Acyclic Graphs

Definition 1.3.6: Trees and Forests

A connected graph that is acyclic (contains no cycles) is called a *tree*.
A *forest* is a disjoint set of trees.

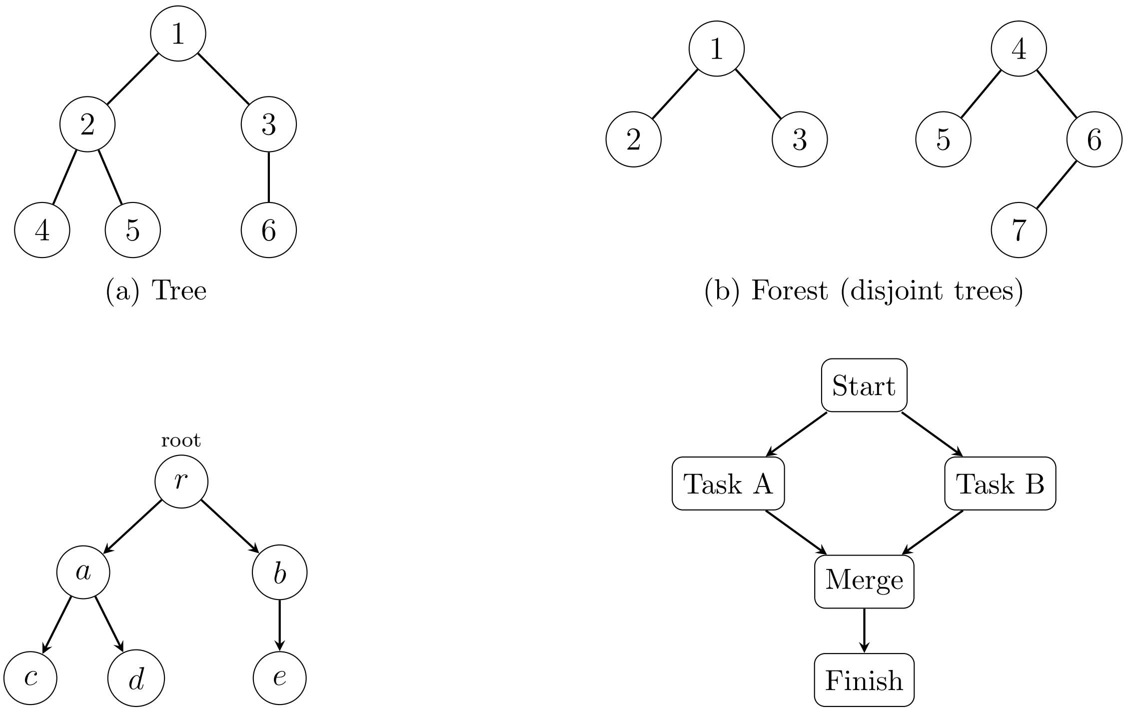
A *directed tree* (or *arborescence*) is a directed graph where there is a unique directed path from a designated root node to every other node.

Definition 1.3.7: Directed Acyclic Graph (DAG)

A *directed acyclic graph* (DAG) is a directed graph with no directed cycles. The underlying graph of a DAG need not be a tree. DAGs are commonly used to represent hierarchical structures, such as task scheduling or data processing pipelines.

Example 1.3.2: Hierarchical Structures as Trees and DAGs

Hierarchical structures, such as organizational charts or file directory systems, can be effectively modeled using trees and DAGs. In an organizational chart, each employee reports to a single



(a) Tree (b) Forest (disjoint trees)

(c) Directed tree (arborescence) (d) DAG (no directed cycles)

Figure 1.20: Examples: (a) a tree (connected and acyclic), (b) a forest (disjoint set of trees), (c) a directed tree/arborescence (unique directed path from root to each node), and (d) a DAG (directed acyclic graph).

(continued)

manager, forming a tree structure. In a file directory system, folders can contain subfolders and files, creating a DAG where cycles are avoided to prevent infinite loops in navigation.

Multiple Choice Questions

1. A network is described in the notes as a "pattern of interconnections among a set of objects." Which statement is *most consistent* with this definition?

   (A) Any collection of objects, regardless of whether any relationships are specified.

   (B) A set of objects together with a rule describing which pairs are connected (possibly implicitly).

   (C) Only systems where all objects are mutually connected (i.e., complete).

   (D) Only systems where connections represent physical links (e.g., wires, roads).

2. Which pairing best matches the notes' two notions of "connectedness" in complex systems?

   (A) Structure: payoff maximization   Behavior: shortest paths

   (B) Structure: who is linked to whom   Behavior: how actions have consequences through the system

   (C) Structure: diffusion   Behavior: clustering coefficient

   (D) Structure: equilibrium   Behavior: adjacency

3. In the "strong ties vs. weak ties" discussion,

which claim is the *most accurate*?

(A) Strong ties typically connect distant communities, acting as global shortcuts.

(B) Weak ties are usually embedded in tightly linked clusters and rarely bridge communities.

(C) Weak ties often span across tightly linked regions and can create "short-cuts" in the network.

(D) Strong ties reduce clustering by discouraging triangle formation.

4. The idea of a *structural hole* most directly emphasizes:

(A) A missing edge inside a clique, lowering the clique number.

(B) A gap between groups with little interaction, creating advantage for a broker who bridges them.

(C) A bridge edge whose removal increases the number of connected components.

(D) A node of degree 0 that isolates part of a graph.

5. "Network fissures" in the notes are primarily linked to which concept?

(A) Preferential attachment

(B) Structural balance and how local antagonisms can reshape global structure

(C) Dijkstra's algorithm and shortest paths

(D) Multigraph edge multiplicity

6. In the notes' game-theoretic framing, a *Nash equilibrium* is best described as:

(A) A strategy profile where everyone achieves the maximum possible payoff.

(B) A state where no one can improve their payoff by changing strategies together.

(C) A strategy profile where no individual can improve their payoff by unilaterally changing strategy.

(D) A state where total (social) payoff is minimized due to competition.

7. Braess's Paradox (as referenced in the driving-routes example) highlights a counterintuitive possibility:

(A) Adding a new connection can increase overall efficiency by lowering congestion everywhere.

(B) Adding resources to a network can *reduce* efficiency due to strategic route choices.

(C) Removing a bridge edge never changes shortest paths.

(D) Equilibrium flows always coincide with socially optimal flows.

8. In "Markets and Strategic Interaction on Networks," power is suggested to depend on:

(A) Only the number of one's connections (degree), independent of whom they connect to.

(B) Only geography or institutional constraints, not network position.

(C) Both the number of connections and the "power" of the nodes connected to oneself.

(D) Only whether the network is directed or undirected.

9. In information networks, "prominence and ranking" is described as *recursive*. Which option best captures this idea?

(A) A page is prominent if it has high out-degree, regardless of incoming links.

(B) A page is prominent if it links to many pages that link to it back (mutual edges).

(C) A page is prominent if it receives links from pages that are themselves prominent.

(D) A page is prominent if it lies in the largest connected component.

10. The notes contrast two reasons people may adopt what others adopt. Which pairing matches the notes?

    (A) Information cascades: direct benefits Network effects: inferring quality from others' actions
    (B) Information cascades: inferring information from others' actions Network effects: direct benefit from alignment
    (C) Information cascades: random imitation Network effects: purely biological transmission
    (D) Information cascades: shortest paths Network effects: graph diameter

11. A "rich-get-richer" feedback described in the notes is most naturally associated with:

    (A) Network effects amplifying success as more people join/adopt
    (B) Structural holes preventing information flow
    (C) The evenness of the odd-degree count
    (D) The definition of a multigraph

12. A *diffusion barrier* (structural effect) is best described as:

    (A) A region of the network with high degree that speeds adoption.
    (B) The boundary of a densely connected cluster that is resistant to outside influence.
    (C) A bridge edge whose removal always increases adoption speed.
    (D) A directed cycle that prevents convergence.

13. "Social contagion" is compared to biological epidemics mainly because:

    (A) Both are governed by identical microscopic rules (no decision-making differences).
    (B) Both involve pathogens; social contagion is just a metaphor with no structural similarities.
    (C) Network-level spreading patterns can look similar even if mechanisms differ.
    (D) Biological contagion only occurs on directed graphs, while social contagion only occurs on undirected graphs.

14. The notes describe institutions as aggregating individual actions. Which example matches that role most directly?

    (A) A clique aggregating all possible edges among its vertices
    (B) A prediction market price aggregating beliefs about future events
    (C) A bridge edge aggregating two components into one
    (D) A DAG aggregating tasks into a single cycle-free path

15. In the notes, Arrow's Impossibility Theorem is invoked to emphasize that:

    (A) Producing a consistent social ranking from individual preferences is inherently difficult.
    (B) Any voting rule yields the same outcome when the graph is connected.
    (C) Social preferences always form a tree.
    (D) Market prices cannot reflect aggregate information.

16. Which statement correctly distinguishes *undirected* vs. *directed* graphs as defined?

    (A) Undirected edges are ordered pairs; directed edges are unordered pairs.
    (B) In undirected graphs, $(u, v) = (v, u)$; in directed graphs, (u, v) and (v, u) are distinct.
    (C) Directed graphs cannot have mutual edges; undirected graphs can.
    (D) Directed graphs must be acyclic by definition.

17. Which choice best matches the notes' graph-type distinctions?

   (A) Simple graph: allows multi-edges and self-loops; multigraph: forbids both.

   (B) Weighted graph: edges carry numbers; path length is the sum of traversed edge weights.

   (C) Multigraph: forbids parallel edges but allows negative weights.

   (D) Simple graph: requires direction on every edge.

18. In an undirected graph, which statement is *necessarily* true given the definitions of adjacency/incidence/neighborhood?

   (A) If $v \in N(u)$ then the edge (u, v) is incident to u but not to v.

   (B) $|N(u)|$ equals the number of edges in the whole graph.

   (C) The neighborhood $N(u)$ is the set of vertices adjacent to u, and $|N(u)| = d(u)$.

   (D) Two vertices are adjacent if they share a common neighbor (even without an edge between them).

19. For a finite undirected graph, which combined statement is correct?

   (A) $\sum_{v \in V} d(v) = |E|$ and the number of odd-degree vertices is odd.

   (B) $\sum_{v \in V} d(v) = 2|E|$ and the number of odd-degree vertices is even.

   (C) $\sum_{v \in V} d(v) = 2|E|$ and the number of odd-degree vertices can be any number.

   (D) $\sum_{v \in V} d(v) = |V||E|$ and the number of odd-degree vertices is even.

20. In a directed graph with edge set E, which identity must hold?

   (A) $\sum_v d^-(v) = \sum_v d^+(v) = |V|$

   (B) $\sum_v d^-(v) = |E|$ and $\sum_v d^+(v) = |E|$

   (C) $\sum_v d^-(v) = 2|E|$ and $\sum_v d^+(v) = 2|E|$

   (D) $\sum_v d^-(v) = \sum_v d^+(v) = 2|E|$

21. Which statement best matches the notes' distinctions among walk/path/cycle and implications for distance/diameter?

   (A) A path may repeat vertices, but a walk may not.

   (B) A cycle is any closed walk, even if it repeats intermediate vertices.

   (C) Distance $d(u, v)$ is the length of a shortest path; if no path exists, $d(u, v) = \infty$ (so the diameter can become ∞ in a disconnected graph).

   (D) Diameter is defined as the minimum distance over all pairs.

22. In directed graphs, which statement about connectivity is correct?

   (A) Weakly connected implies strongly connected.

   (B) Strongly connected implies weakly connected, but not conversely.

   (C) Strong connectivity depends only on the underlying undirected graph.

   (D) Strongly connected components are defined by removing bridge edges.

23. Which option is correct about complete graphs, cliques, and regular graphs?

   (A) In K_n, $|E| = \binom{n}{2}$ and K_n is $(n-1)$-regular.

   (B) A clique is any connected subgraph; its size is the graph's diameter.

   (C) A k-regular graph must be complete whenever $k \geq 2$.

   (D) Cycles C_n are $(n-1)$-regular for $n \geq 3$.

24. Which statement best distinguishes trees, forests, arborescences, and DAGs as in the notes?

   (A) A DAG must have a unique directed path from a root to every node.

   (B) A tree is connected and acyclic; a forest is a disjoint set of trees.

(C) Any arborescence becomes a DAG only after ignoring edge directions.

(D) The underlying graph of a DAG must be a tree.

Chapter 2

Mathematics of Networks

2.1 Adjacency Matrix

The *adjacency matrix* A is the preferred mathematical representation of a network structure.

2.1.1 Undirected Graphs

Definition 2.1.1: Adjacency Matrix of an Undirected Graph

For a simple graph, the elements A_{ij} are defined as

$$A_{ij} = \begin{cases} 1 & \text{if there is an edge between vertices } i \text{ and } j, \\ 0 & \text{otherwise.} \end{cases}$$

Example 2.1.1: Adjacency Matrix of a Simple Graph

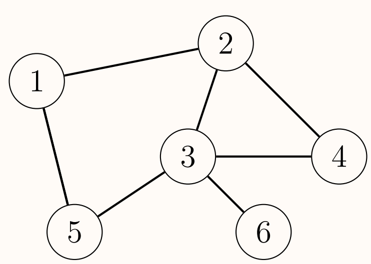


Figure 2.1: A simple graph on 6 nodes

The adjacency matrix for the simple graph in Figure 2.1 is

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

Properties 2.1.1: Adjacency Matrix of an Undirected Graph

The matrix is *symmetric* ($A_{ij} = A_{ji}$)
If there are no self-edges, the diagonal elements A_{ii} are all zero.

2.1.2 Handling Non-Simple Undirected Graphs

Multi-edges

A multi-edge is represented by setting the corresponding matrix element A_{ij} equal to the multiplicity of the edge.

Self-edges

A single self-edge for vertex i is represented by setting the diagonal element $A_{ii} = 2$.

Tip 2.1.1: Why is $A_{ii} = 2$ for a self-edge?

This is because a self-edge contributes 2 to the degree of the vertex (both endpoints of the edge are incident to the same vertex). Thus, to maintain the property that the degree of vertex i is given by the sum of the i-th row (or column) of the adjacency matrix, we set $A_{ii} = 2$ for a single self-edge.

Example 2.1.2: Adjacency Matrix of a Multigraph with Self-Edges

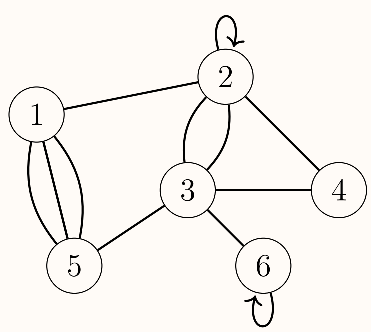


Figure 2.2: A multigraph on 6 nodes, with parallel edges between nodes 1 and 5, nodes 2 and 3, and self-loops on nodes 2 and 6.

The adjacency matrix for the multigraph in Figure 2.2 is

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 3 & 0 \\ 1 & 2 & 2 & 1 & 0 & 0 \\ 0 & 2 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 3 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 2 \end{bmatrix}$$

2.1.3 Directed Graphs (Digraphs)

Definition 2.1.2: Adjacency Matrix of a Directed Graph

The adjacency matrix A for a directed network uses a specific convention:

$$A_{ij} = \begin{cases} 1 & \text{if there is an edge from } j \text{ to } i, \\ 0 & \text{otherwise.} \end{cases}$$

Example 2.1.3: Adjacency Matrix of a Directed Graph

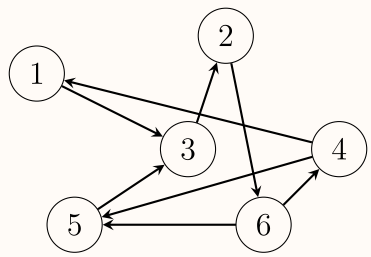


Figure 2.3: A simple directed graph on 6 nodes

The adjacency matrix for the directed graph in Figure 2.3 is:

$$A = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Properties 2.1.2: Adjacency Matrix of a Directed Graph

The matrix is generally *asymmetric*.
A single self-edge in a directed network is represented by $A_{ii} = 1$.
An undirected network can be viewed as a directed network where every undirected edge is replaced by two directed edges running in opposite directions, resulting in a symmetric A.

2.1.4 Weighted Networks

In a *weighted (or valued) network*, edges possess a strength, weight, or value (usually a real number). The elements of the adjacency matrix A_{ij} are set equal to the weights of the corresponding connections.

Example 2.1.4: Adjacency Matrix of a Weighted Network

The adjacency matrix

$$A = \begin{bmatrix} 0 & 2 & 1 \\ 2 & 0 & 0.5 \\ 1 & 0.5 & 0 \end{bmatrix}$$

represents a weighted network in which the connection between vertices 1 and 2 is twice as strong as that between 1 and 3, which in turn is twice as strong as that between 2 and 3.

2.2 Degree

2.2.1 Vertex Degree

Definition 2.2.1: Degree of a Vertex in an Undirected Graph

The *degree* k_i of vertex i is the number of edges connected to it.

Formula	Description
Degree in terms of A	For an undirected graph of n vertices: $$k_i = \sum_{j=1}^{n} A_{ij}.$$
Total edges	The number of ends of edges is equal to the sum of the degrees of all vertices: $$2m = \sum_{i=1}^{n} k_i.$$
Mean degree c	The mean degree c of a vertex is: $$c = \frac{1}{n} \sum_{i=1}^{n} k_i = \frac{2m}{n}.$$

Table 2.1: Basic degree and edge-count identities for an undirected graph.

Recap 2.2.1: Regular Graphs

As discussed in Section 1.3.4, a *regular graph* is a graph where every vertex has the same degree k. Such a graph is called a k-regular graph.

2.2.2 Connectance and Density

Definition 2.2.2: Connectance (Density) of a Graph

The *connectance* or *density* ρ of a graph is the fraction of all possible edges that are actually present.

$$\rho = \frac{m}{\binom{n}{2}} = \frac{2m}{n(n-1)} = \frac{c}{n-1}$$

Dense networks are networks where ρ tends to a constant as $n \to \infty$, while *sparse networks* are networks where $\rho \to 0$ as $n \to \infty$ (i.e., the mean degree c tends to a constant). Most real-world networks studied are considered sparse.

Definition 2.2.3: Occasional connectance

Occasional connectance is defined as

$$\rho = \frac{m}{n^2},$$

which for large networks differs from the above equation about a factor of 2. With that definition,

$$0 \leq \rho \leq \frac{1}{2}$$

2.2.3 Degrees in Directed Networks

In a directed network, each vertex has two degrees: *in-degree* and *out-degree*.

Measure	Definition	Formula
In-degree ($k_i^{\mathbf{in}}$)	Number of edges pointing *to* vertex i	$k_i^{\text{in}} = \sum_{j=1}^{n} A_{ij}$
Out-degree ($k_i^{\mathbf{out}}$)	Number of edges pointing *away* from vertex i	$k_i^{\text{out}} = \sum_{j=1}^{n} A_{ji}$

Table 2.2: In-degree and out-degree in a directed graph (adjacency matrix A).

Properties 2.2.1: Equality of In-Degree and Out-Degree Sums

The total number of edges m is equal to the sum of all in-degrees and the sum of all out-degrees. Consequently, the mean in-degree and mean out-degree are always equal

$$c^{\text{in}} = c^{\text{out}} = c$$

Therefore, the mean in-degree and out-degree are both equal to the mean degree c of the corresponding undirected graph obtained by ignoring edge directions:

$$c = \frac{m}{n}.$$

2.3 Connectivity

2.3.1 Connectedness and Components

Recap 2.3.1: Connected Graphs

From Section 1.3.3, a non-empty graph G is *connected* if any two of its vertices are linked by a path.

Proposition 2.3.1: Building a Connected Graph One Vertex at a Time

The vertices of a connected graph G can always be enumerated, say as $v_1, \ldots, v_n$, so that $G_i := G[v_1, \ldots, v_i]$ is connected for every i

Proof. Let $G = (V, E)$ be a connected graph with $|V| = n$.

We build an ordering $v_1, \ldots, v_n$ inductively. Pick any vertex and call it v_1. Then $G_1 = G[\{v_1\}]$ is connected.

Assume we have chosen distinct vertices $v_1, \ldots, v_i$ such that $G_i := G[\{v_1, \ldots, v_i\}]$ is connected, for some $i < n$. Let $S = \{v_1, \ldots, v_i\}$. Since $i < n$, we have $S \neq V$.

Because G is connected, there exists a path from S to any vertex in $V \setminus S$. In particular, there must be an edge with one endpoint in S and the other in $V \setminus S$ (otherwise no vertex outside S could be reached from S). Choose such an edge xy with $x \in S$ and $y \in V \setminus S$, and set $v_{i+1} = y$.

In the induced subgraph $G_{i+1} = G[S \cup \{v_{i+1}\}]$, the edge xv_{i+1} is present, so v_{i+1} is adjacent to a vertex in S. Hence G_{i+1} is connected.

By induction, we obtain an ordering $v_1, \ldots, v_n$ such that each G_i is connected. □

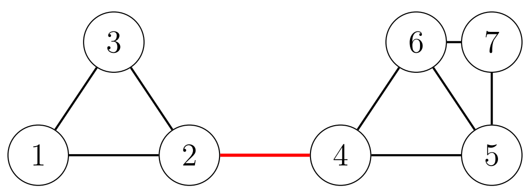


Figure 2.4: A connected undirected graph on 7 vertices. The red edge $(2, 4)$ is a *bridge*: removing it disconnects the graph into two components (the triangle on $\{1, 2, 3\}$ and the subgraph on $\{4, 5, 6, 7\}$).

Definition 2.3.1: Components of a Graph

A *component* of a graph is a maximal connected subgraph.

2.3.2 Separating Sets

A set $X \subseteq V \cup E$ *separates* sets A and B if every $A - B$ path contains an element of X. X is a *separating set* if it separates two vertices of $G - X$ in G.

Example 2.3.1: Separating Set

In $A - B - C - D$, take $X = \{B\}$. Then B is a separating set for A and D, because every $A - D$ path goes through B.

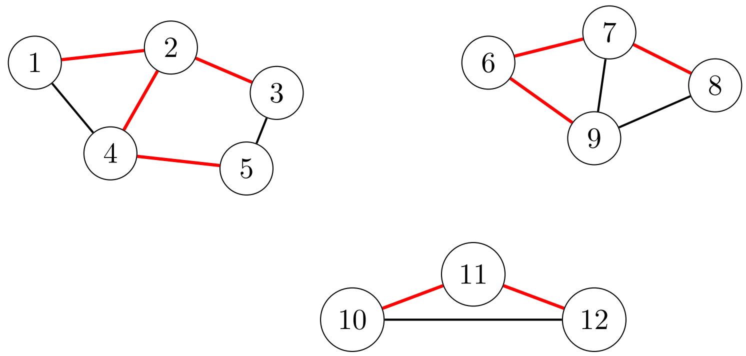


Figure 2.5: A graph with three connected components. In each component, the highlighted red edges form a *minimal connected spanning subgraph* (a spanning tree of that component): it connects all vertices in the component using the fewest possible edges (exactly $|V_{\text{comp}}| - 1$).

Definition 2.3.2: Cutvertex and Bridge

A *cutvertex* is a vertex whose removal increases the number of components of the graph. A *bridge* (or cut-edge) is an edge whose removal increases the number of components of the graph.

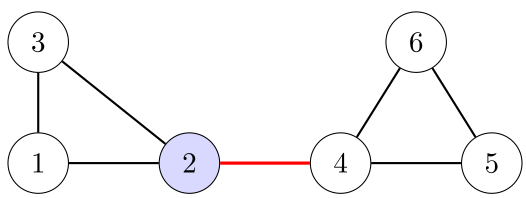


Figure 2.6: Example of a **cut-vertex** and a **bridge**. The blue vertex 2 is a cut-vertex: removing it disconnects the graph into two components. The red edge $(2, 4)$ is a bridge: removing it disconnects the graph (the two triangles separate).

2.3.3 Connectivity Measures

Definition 2.3.3: k-connected Graph

A graph G is k*-connected* if $|G| > k$ and $G - X$ is connected for every set $X \subseteq V$ with $|X| < k$.

Example 2.3.2: k-connected Graph

The graph in Figure 2.6 is 2-connected. Removing any single vertex leaves the graph connected. However, removing vertices 2 and 4 disconnects the graph.
A triangle $A - B - C$ (K_3) is 2-connected: removing one vertex leaves an edge (still connected).
A square $A - B - C - D$ with diagonals (K_4) is 3-connected: removing any two vertices still leaves a connected piece.

Its application includes network reliability analysis, where higher connectivity implies greater resilience to failures or attacks. It answers the questions: "How many nodes or edges can be removed before the network becomes disconnected?" and "What is the minimum number of nodes or edges whose removal would disconnect the network?"

Definition 2.3.4: Connectivity of a Graph

The greatest integer k such that G is k-connected is the *connectivity* $\kappa(G)$ of G. $\kappa(G) = 0$ if G is disconnected or K_1, and $\kappa(K_n) = n - 1$ for $n \geq 1$

Graph	$\boldsymbol{\kappa(G)}$	**Why**
K_1	0	single vertex (trivial)
K_3	2	remove ≤ 1 vertex $\to$ still connected
K_4	3	remove ≤ 2 vertices $\to$ still connected

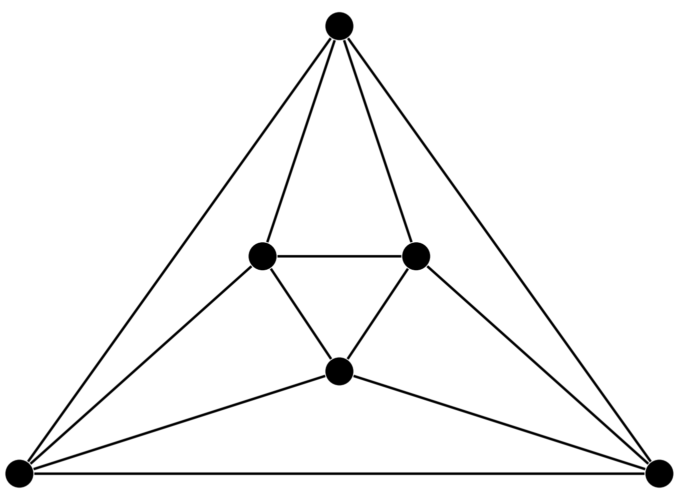
Table 2.3: Vertex connectivity $\kappa(G)$ for small complete graphs.

Definition 2.3.5: l-edge-connected Graph

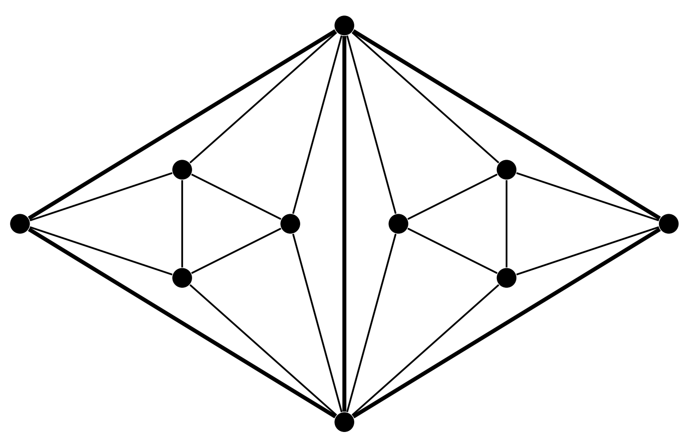
A graph G is l-edge-connected if $|G| > 1$ and $G - F$ is connected for every set $F \subseteq E$ of fewer than l edges.

Definition 2.3.6: Edge-Connectivity of a Graph

The greatest integer l such that G is l-edge-connected is the *edge-connectivity* $\lambda(G)$ of G. $\lambda(G) = 0$ if G is disconnected, and $\lambda(K_n) = n - 1$ for $n \geq 2$.



(a)



(b)

Figure 2.7: The octahedron G (left) with $\kappa(G) = \lambda(G) = 4$, and a graph H with $\kappa(H) = 2$ but $\lambda(H) = 4$.

Recap 2.3.2: Minimum degree

As discussed in Section 2.2, the minimum degree of a graph $\delta(G)$ is the smallest number of links any node has, calculated as

$$\delta(G) = \min_{v \in V} d(v)$$

Properties 2.3.1: Relations between Connectivity Measures

For every non-trivial graph G:

$$\kappa(G) \leq \lambda(G) \leq \delta(G)$$

This means high connectivity requires a large minimum degree.

Proof. Let $G = (V, E)$ be a non-trivial graph (so $|V| \geq 2$).

1) Proof that $\kappa(G) \leq \lambda(G)$. Let $F \subseteq E$ be a minimum edge cut, so $|F| = \lambda(G)$ and $G - F$ is disconnected. Let A and B be two vertex sets forming a partition of V into (at least) two components of $G - F$.

Define

$$S := \{ v \in V : v \text{ is incident to some edge in } F \}.$$

Every edge in F has two endpoints, so $|S| \leq |F| = \lambda(G)$.

Now consider $G - S$. If there were a path in $G - S$ from a vertex in A to a vertex in B, that path would also be present in $G - F$ (since it uses no edge of F) and would connect A to B, contradicting that $G - F$ is disconnected. Hence $G - S$ is disconnected. Therefore S is a vertex cut, so

$$\kappa(G) \leq |S| \leq \lambda(G).$$

2) Proof that $\lambda(G) \leq \delta(G)$. Let $v \in V$ be a vertex with minimum degree $d(v) = \delta(G)$. Remove all edges incident to v; call this set F_v. Then $|F_v| = d(v) = \delta(G)$ and in $G - F_v$ the vertex v becomes isolated, so $G - F_v$ is disconnected. Hence there exists an edge cut of size $\delta(G)$, implying

$$\lambda(G) \leq \delta(G).$$

Combining (1) and (2) gives $\kappa(G) \leq \lambda(G) \leq \delta(G)$. □

Example 2.3.3: Network reliability via $\kappa(G)$, $\lambda(G)$, and $\delta(G)$

Think of a network as a graph G whose vertices are *devices/locations* and whose edges are *communication links/roads.*

- **Vertex connectivity $\kappa(G)$:** the minimum number of *vertices* whose failure can disconnect the network. Interpreted as "how many routers (or stations) must fail before communication breaks."
- **Edge connectivity $\lambda(G)$:** the minimum number of *edges* whose failure can disconnect the network. Interpreted as "how many links (cables/roads) must fail before the network breaks."
- **Minimum degree $\delta(G)$:** the smallest number of incident edges among all vertices. Interpreted as a *local redundancy guarantee*: every node has at least $\delta(G)$ direct links.

Use cases.

- *Internet backbone design:* requiring $\kappa(G) \geq 2$ ensures there is no single router whose

(continued)

removal disconnects the network (i.e., there is no articulation point). The network can tolerate the failure of any one router without losing global connectivity.

- *Transportation planning:* if $\lambda(G) \geq 2$, then there is no single road/bridge whose closure disconnects the system (i.e., there is no bridge edge). At least two independent road links must fail before the network becomes disconnected, improving robustness against a single closure.

Theorem 2.3.1: Mader 1972

Every graph of average degree at least $4k$ has a k-connected subgraph.

Proof. Let $G = (V, E)$ be a graph with $|V| = n \geq 2$ and average degree

$$\bar{d}(G) = \frac{2|E|}{|V|} \geq 4k, \quad \text{so} \quad |E| \geq 2kn.$$

Step 1 (pruning to get minimum degree $\geq 2k$). Repeatedly delete any vertex of degree at most $2k-1$ (together with its incident edges). Suppose at some step we delete a vertex v from a current graph with n' vertices and m' edges. If $\deg(v) \leq 2k-1$, then the new graph has

$$n'' = n' - 1, \qquad m'' = m' - \deg(v) \geq m' - (2k-1).$$

If the current graph still satisfies $m' \geq 2kn'$, then

$$m'' \;\geq\; 2kn' - (2k-1) \;=\; 2k(n'-1) + 1 \;>\; 2k(n'-1) = 2kn''.$$

Hence the average degree after deletion remains *strictly larger* than $4k$. Therefore this deletion process cannot remove all vertices; it terminates with a nonempty subgraph $H \subseteq G$ such that

$$\delta(H) \geq 2k.$$

Step 2 (from $\delta(H) \geq 2k$ to a k-connected subgraph). We now use the following standard result (also due to Mader):

Lemma (Mader). If a graph H satisfies $\delta(H) \geq 2k$, then H contains a k-connected subgraph.

Applying the lemma to the subgraph H produced in Step 1 yields a k-connected subgraph of H, and hence of the original graph G. □

Example 2.3.4: Why Mader's theorem matters in network science

Interpretation. High *overall density* (large average degree) forces the existence of a *highly robust core*: a vertex-induced subnetwork that stays connected even after the removal of up to $k-1$ vertices.

Applications / uses.

- **Core robustness and resilience.** In infrastructure or communication networks, vertex connectivity captures tolerance to node failures. Mader's theorem guarantees that if the network is sufficiently dense on average, then there exists a subnetwork that can

(continued)

survive many node deletions without fragmenting. This supports the idea of a resilient "backbone" or "core" inside a large system.

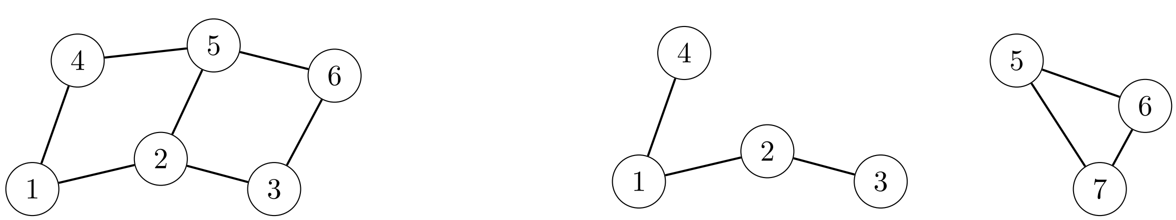
- **Justifying k-core / dense-subgraph heuristics.** Many network-science pipelines extract a dense core (e.g., k-core peeling) to find a stable, cohesive region. Mader provides a theoretical underpinning: density is not just a visual property; beyond a threshold it forces nontrivial structural robustness (a k-connected subgraph).
- **Robust routing and redundancy.** A k-connected subgraph has multiple vertex-disjoint routes between nodes (by Menger's theorem), which is exactly what you want for reliable routing or load balancing. Mader is a density-to-redundancy guarantee: enough edges imply a region with many independent alternative paths.
- **Network design and certification.** In design problems, it may be hard to certify global k-connectivity, but easier to measure average degree. Mader's theorem says: if you can ensure $\bar{d} \geq 4k$, then you can *certify existence* of a k-connected subnetwork, even if the whole network is not k-connected. This is useful when you only need a robust subnetwork for critical services (control plane, emergency response, etc.).
- **Algorithmic motivation (finding robust subgraphs).** The theorem motivates algorithms that search for highly connected subgraphs by first locating dense regions. In practice, one often: (i) prune low-degree vertices, (ii) extract dense components, then (iii) test/augment connectivity. Mader explains why this strategy can succeed when the input graph is globally dense.

2.4 Network Connectivity, Components, and Flow

2.4.1 Components

Recap 2.4.1: Connected and Disconnected Networks

A network is *connected* if there is a path between any pair of vertices. If not, it is *disconnected*. A *component* is a maximal connected subgraph of a disconnected graph.



(a) Connected graph (b) Disconnected graph

Figure 2.8: Examples of a connected and a disconnected undirected graph.

The adjacency matrix of a network with more than one component can be written in block diagonal form, meaning that the non-zero elements of the matrix are confined to square blocks along the diagonal of the matrix, with all other elements being zero:

$$A = \begin{pmatrix} \square & 0 & \cdots \\ 0 & \square & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix}$$

2.4.2 Components in Directed Networks

Definition 2.4.1: Strongly and Weakly Connected Directed Graphs

A directed graph is *strongly connected* if there is a directed path from any vertex to any other vertex. It is *weakly connected* if the underlying undirected graph (obtained by ignoring edge directions) is connected.

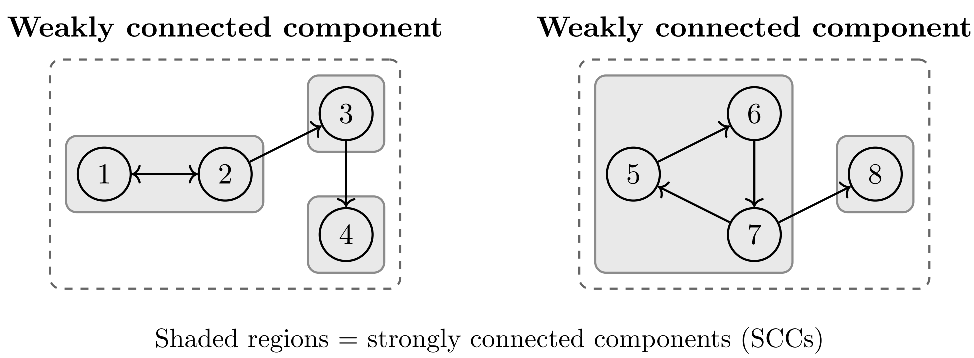


Figure 2.9: A directed network with two weakly connected components (dashed boxes), and five strongly connected components (shaded).

Definition 2.4.2: Out-Component and In-Component

In a directed graph, the *out-component* of a vertex v is the set of vertices reachable from v via directed paths. The *in-component* of v is the set of vertices from which v can be reached via directed paths.

2.4.3 Acyclic Directed Networks (DAGs)

Recap 2.4.2: Directed Acyclic Graphs (DAGs)

As discussed in Section 1.3.7, a *directed acyclic graph* (DAG) is a directed graph with no directed cycles (closed loops where arrows point the same way).

They contain no self-edge.

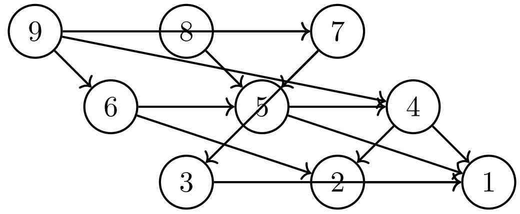


Figure 2.10: A 9-node DAG with edges directed from higher labels to lower labels.

Properties 2.4.1: Matrix Properties of DAGs

If the vertices are labeled in a specific order (such that all edges point from higher to lower labels), the adjacency matrix A will be *strictly upper triangular* (non-zero elements only above the diagonal).

The adjacency of the DAG in Figure 2.10 is:

$$A = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Properties 2.4.2: Eigenvalues of DAGs

A directed network is acyclic if and only if all eigenvalues of its adjacency matrix are zero. Such matrices are called *nilpotent matrices*.

Because A is upper triangular, $\lambda I - A$ is also upper triangular, with diagonal elements $\lambda - A_{ii} = \lambda$. Therefore, the characteristic polynomial[1] is

$$\chi_A(\lambda) = \det(\lambda I - A) = \prod_{i=1}^{9} \lambda = \lambda^9.$$

So the eigenvalues are all

$$\mathrm{spec}(A) = \{0, 0, 0, 0, 0, 0, 0, 0, 0\}.$$

[1]For any $n \times n$ matrix A, the characteristic polynomial of a matrix A is defined as $\chi_A(\lambda) = \det(\lambda I - A)$. It's a polynomial in λ (degree n).

2.4.4 Independent Paths and Cut Sets

Definition 2.4.3: Independent Paths

Two paths between vertices u and v are *independent* if they do not share any internal vertices (vertex-independent) or edges (edge-independent).

(a) (b)

Figure 2.11: Edge independent paths. (a) There are two edge-independent paths from A to B in this figure, as denoted by the arrows, but there is only one vertex-independent path, because all paths must pass through the center vertex C. (b) The edge-independent paths are not unique; there are two different ways of choosing two independent paths from A to B in this case.

Definition 2.4.4: Cut Set

Let $G = (V, E)$ be a graph and let $u, v \in V$ be distinct.

1. A u–v *vertex cut set* (or u–v *separator*) is a set $S \subseteq V \setminus \{u, v\}$ such that in the graph $G - S$ there is no path from u to v.
2. A u–v *edge cut set* is a set $F \subseteq E$ such that in the graph $G - F$ there is no path from u to v.

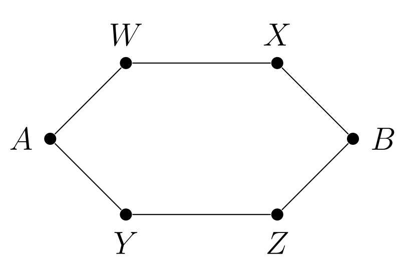
The u–v *vertex-connectivity* is

$$\kappa(u, v) = \min\{|S| : \ S \text{ is a } u\text{–}v \text{ vertex cut set}\},$$

and the u–v *edge-connectivity* is

$$\lambda(u, v) = \min\{|F| : \ F \text{ is a } u\text{–}v \text{ edge cut set}\}.$$

A *minimum cut set* is a cut set of size equal to the vertex- or edge-connectivity. A minimum cut set need not be unique. For instance, there is a variety of minimum vertex sets of size two between the vertices A and B in this network:



{W,Y}, {W,Z}, {X,Y}, and {X,Z} are all minimum cut sets for this network. Of course all the minimum cut sets must have the same size.

Theorem 2.4.1: Menger's Theorem (local form)

Let $G = (V, E)$ be a graph and let $u, v \in V$ be distinct.

1. Vertex version: The maximum number of pairwise vertex-independent u–v paths equals the minimum size of a u–v vertex cut set.
2. Edge version: The maximum number of pairwise edge-independent u–v paths equals the minimum size of a u–v edge cut set.

Corollary 1 ("No small cut" $\Rightarrow$ many independent paths)**.** *Fix $k \in \mathbb{N}$.*

1. *If there is no u–v vertex cut set of size $< k$, then there exist at least k vertex-independent u–v paths.*
2. *If there is no u–v edge cut set of size $< k$, then there exist at least k edge-independent u–v paths.*

Proof. This is immediate from Theorem 2.4.1: if the minimum cut size is $\geq k$, then the maximum number of independent paths (which equals that minimum) is also $\geq k$. $\square$

Remark 1 (Significance of Menger's Theorem)**.** *Menger's theorem is a fundamental connectivity principle. In the next section we introduce network flows, which (i) generalize the "counting" viewpoint to* weighted capacities*, and (ii) provide an algorithmic route to compute these quantities. In particular, Menger's theorem can be recovered as a unit-capacity special case of Max-Flow/Min-Cut (see Theorem 2.4.3).*

Definition 2.4.5: Flow network, flow value, and cuts

A *(directed) flow network* is a directed graph $N = (V, A)$ with a source $s \in V$, sink $t \in V$, and capacities $c : A \to \mathbb{R}_{\geq 0}$. An *$s$–$t$ flow* is a function $f : A \to \mathbb{R}_{\geq 0}$ satisfying:

$$0 \leq f(a) \leq c(a) \quad \text{for all } a \in A,$$

and (flow conservation) for every $x \in V \setminus \{s, t\}$,

$$\sum_{(y,x)\in A} f(y, x) = \sum_{(x,y)\in A} f(x, y).$$

The *value* of f is

$$\mathrm{val}(f) = \sum_{(s,y)\in A} f(s, y) - \sum_{(y,s)\in A} f(y, s).$$

A *cut* is a set $S \subseteq V$ with $s \in S$ and $t \notin S$. Let $\delta^+(S) = \{(x, y) \in A : \ x \in S, \ y \notin S\}$ be the set of arcs leaving S. The *capacity* of the cut is

$$c(\delta^+(S)) = \sum_{a\in\delta^+(S)} c(a).$$

Theorem 2.4.2: Max-Flow/Min-Cut

In any flow network (N, c) with source s and sink t,

$$\max\{\mathrm{val}(f) : f \text{ is an } s\text{–}t \text{ flow}\} \ = \ \min\{c(\delta^+(S)) : \ S \subseteq V, \ s \in S, \ t \notin S\}.$$

Remark 2 (Unweighted vs. weighted). *If all capacities are* 1 *("unweighted"), then the minimum cut capacity equals the* number *of arcs in a minimum cut. In general ("weighted"), the minimum cut is measured by the* sum of capacities.

Remark 3 (Integrality). *If all capacities $c(a)$ are integers, then there exists a maximum flow f that is integral on every arc. Consequently, a unit-capacity max flow can be decomposed into a collection of edge-disjoint s–t flow paths.*

Theorem 2.4.3: Menger via Max-Flow/Min-Cut

The two equalities in Theorem 2.4.1 follow from Max-Flow/Min-Cut (Theorem 2.4.2) by choosing appropriate unit-capacity networks.

Proof. We prove the edge and vertex versions separately.

(Edge version). Form a directed network $N = (V, A)$ from G by replacing each undirected edge $\{x, y\} \in E$ with two opposite arcs (x, y) and (y, x), each of capacity 1. Any u–v cut $(S, \overline{S})$ in N has capacity equal to the number of edges of G crossing between S and $\overline{S}$; thus minimum cut capacity equals the minimum size of a u–v edge cut set. By Theorem 2.4.2, this equals the maximum flow value. Using integrality and flow decomposition, a maximum integral flow of value r decomposes into r pairwise edge-disjoint directed u–v flow paths, which correspond to r edge-independent u–v paths in G. Hence

$$\max\{\#\text{ edge-independent } u\text{–}v \text{ paths}\} = \min\{\#\text{ edges in a } u\text{–}v \text{ edge cut set}\}.$$

(Vertex version). We encode "using a vertex at most once" by splitting vertices. Construct a directed network N' as follows: for each $x \in V \setminus \{u, v\}$ create two nodes x_{in} and x_{out} and add the *split arc* $x_{\text{in}} \to x_{\text{out}}$ with capacity 1. Keep u and v unsplit (or give their split arcs capacity $+\infty$). For every undirected edge $\{a, b\} \in E$, add arcs $a_{\text{out}} \to b_{\text{in}}$ and $b_{\text{out}} \to a_{\text{in}}$ with capacity $+\infty$ (any number $\geq |V|$ suffices).

Cuts correspond to vertex cut sets. In N', the only finite-capacity arcs are split arcs. Therefore any minimum u–v cut consists of some set of split arcs $\{x_{\text{in}} \to x_{\text{out}}\}$, and its capacity equals the number of such vertices x. Removing exactly those vertices in G disconnects u from v, so minimum cut capacity equals the minimum size of a u–v vertex cut set.

Flows correspond to internally vertex-disjoint paths. By Theorem 2.4.2, the maximum flow value equals the minimum cut capacity. With integer capacities, take an integral maximum flow and decompose it into u–v flow paths. Because each split arc has capacity 1, at most one unit of flow can pass through any internal vertex $x \neq u, v$, so the resulting u–v paths are internally vertex-disjoint in G. Hence

$$\max\{\#\text{ vertex-independent } u\text{–}v \text{ paths}\} = \min\{\#\text{ vertices in a } u\text{–}v \text{ vertex cut set}\}.$$

Combining the two parts yields Theorem 2.4.1. □

2.5 Specialized Structures

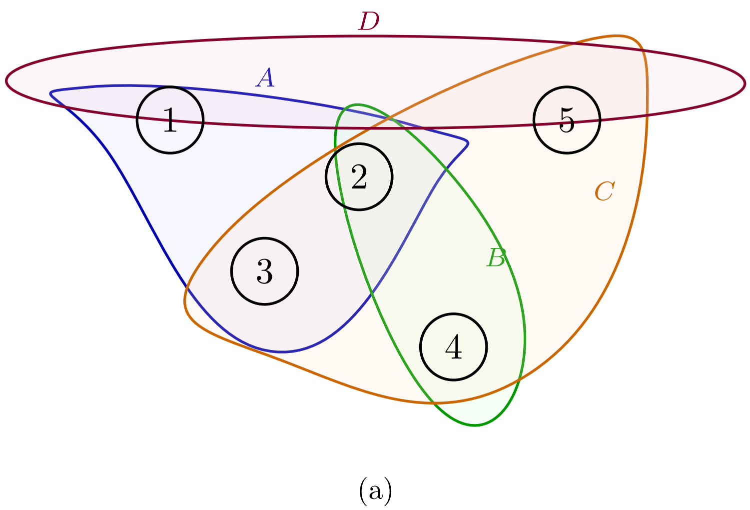
2.5.1 Hypergraphs and Bipartite Networks

Definition 2.5.1: Hypergraph

A *hypergraph* is a generalization of a graph where edges (called hyperedges) can connect any number of vertices, not just two. Formally, a hypergraph H is defined as a pair $H = (V, E)$, where V is a set of vertices and E is a set of non-empty subsets of V (the hyperedges).

Definition 2.5.2: Bipartite Graphs

A *bipartite graph* is a graph whose vertices can be divided into two disjoint sets U and V such that every edge connects a vertex in U to one in V. There are no edges between vertices within the same set.



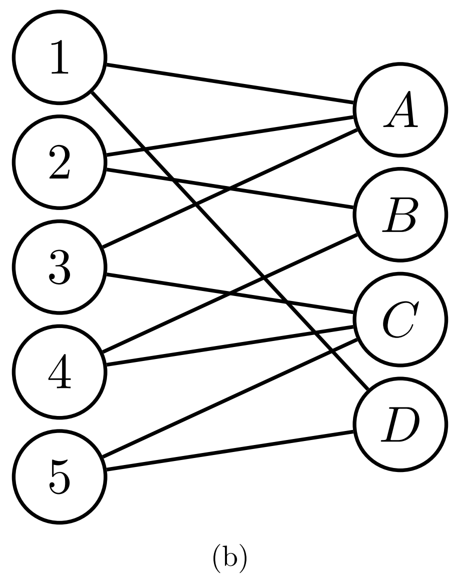


Figure 2.12: A hypergraph and corresponding bipartite graph. These two networks show the same information - the membership of five vertices in four different groups. (a) The hypergraph representation in which the groups are represented as hyperedges, denoted by the loops circling sets of vertices. (b) The bipartite representation in which we introduce four new vertices (open circles) representing the four groups, with edges connecting each vertex to the groups to which it belongs.

Properties 2.5.1: Incidence Matrix of bipartite graph

The equivalent of the adjacency matrix for a bipartite network is the *incidence matrix* B. If n is the number of participants and g is the number of groups, B is a $g \times n$ matrix defined as:

$$B_{ij} = \begin{cases} 1 & \text{if vertex } j \text{ belongs to group } i, \\ 0 & \text{otherwise.} \end{cases}$$

Example 2.5.1: Incidence Matrix Example

The 4×5 incidence matrix of the network in Figure 2.12 is:

$$B = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad (\text{rows } A, B, C, D, \text{ cols } 1, \dots, 5).$$

Definition 2.5.3: One-mode Projection

The *one-mode projection* of a bipartite graph onto one set of vertices (e.g., actors) is a graph where two vertices are connected if they share membership in at least one common group (e.g., appeared in the same film). The weight of the edge can represent the number of shared groups.

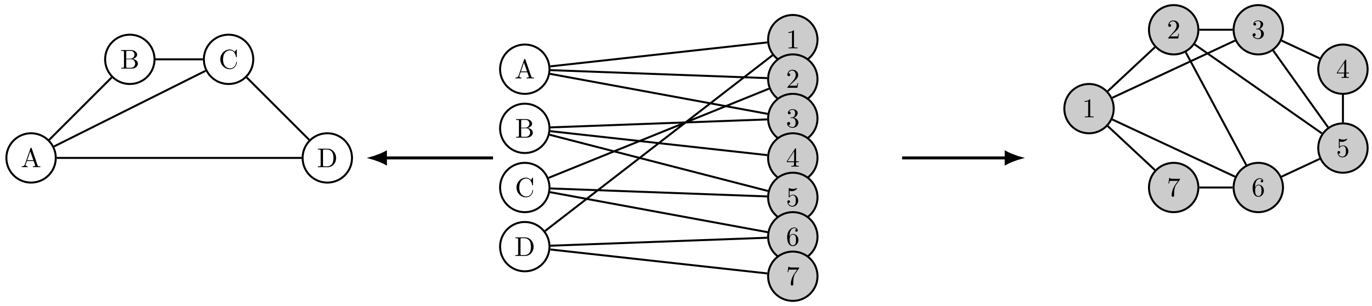


Figure 2.13: The two one-mode projections of a bipartite network. The central portion of this figure shows a bipartite network with four vertices of one type (open circles labeled A to D) and seven of another (filled circles, 1 to 7). The left and right show the one-mode projections onto the two sets of vertices.

Properties 2.5.2: Weighted Projection Matrix

The projection can be written in terms of the incidence matrix B.

$$P = B^{\intercal} B$$

- The off-diagonal element P_{ij} equals the number of common groups shared by i and j.
- The diagonal element P_{ii} equals the number of groups to which vertex i belongs.

The incidence matrix B can be used to derive the adjacency matrix A of the one-mode projection onto the set of vertices (participants) as:

$$A = B^{\intercal} B - D$$

where D is a diagonal matrix with the degrees of the vertices on the diagonal, accounting for self-loops that arise from the multiplication.

2.5.2 Trees and Planar Networks

Recap 2.5.1: Trees and Forests

As discussed in Section 1.3.6, a *tree* is a connected graph with no cycles. A *forest* is a disjoint union of trees (i.e., an acyclic graph that may be disconnected).

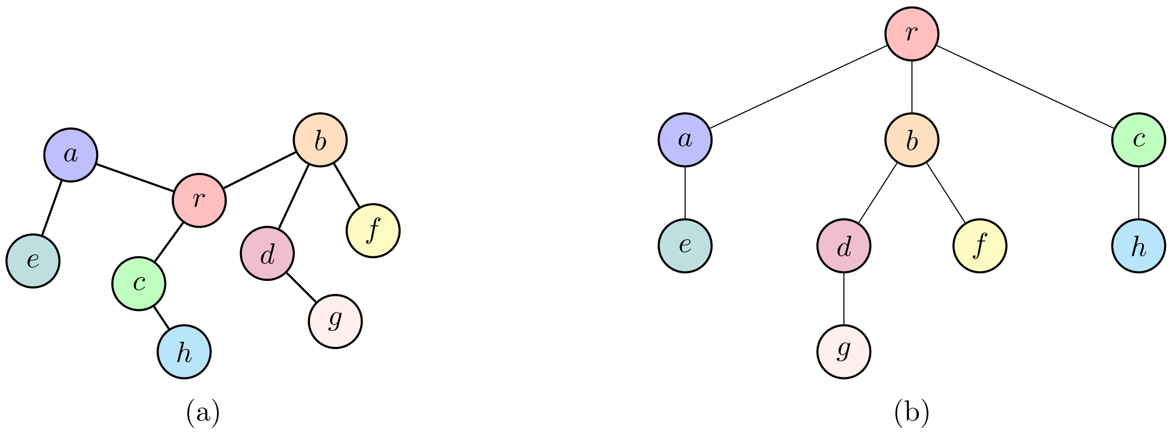


Figure 2.14: Two sketches of the same tree. The two panels here show two different depictions of a tree, a network with no closed loops. In (a) the vertices are positioned on the page in any convenient position. In (b) the tree is a laid out in a "rooted" fashion, with a root node at the top and branches leading down to "leaves" at the bottom.

Properties 2.5.3: Properties of Trees

There is exactly one path between any pair of vertices. A tree with n vertices always has exactly $m = n - 1$ edges.

Trees are often drawn in a *rooted* manner, with a *root node* at the top and a branching structure going down. The vertices at the bottom that are connected to only one other vertex are called *leaves*.

Definition 2.5.4: Planar Graph

A graph is *planar* if it can be drawn on a plane without any edges crossing each other.

Definition 2.5.5: Expansion / subdivision (homeomorphic copy)

A graph H is an *expansion* (also called a *subdivision*) of a graph G if H can be obtained from G by repeatedly *subdividing edges*: replacing an edge $\{x, y\}$ by a path $x - w_1 - \cdots - w_r - y$ whose internal vertices $w_1, \ldots, w_r$ are new and have degree 2 in the resulting graph. Equivalently, H contains a copy of G in which every edge of G is replaced by an internally vertex-disjoint path.

Lemma 2.5.1: Planarity is preserved under subgraphs

If G is planar and H is a subgraph of G, then H is planar.

Figure 2.15: Two drawings of a planar graph. (a) A small planar graph with four vertices and six edges. It is self-evident that the graph is planar, since in this depiction it has no edge that cross. (b) The same graph redrawn with two of its edges crossing. Even though the edges cross, the graph is still planar - a graph is planar if it *can* be drawn without crossing edges.

Proof. Take a planar drawing of G. Deleting vertices/edges yields a drawing of H with no crossings. □

Lemma 2.5.2: Planarity is preserved under subdivisions

A graph G is planar if and only if any (equivalently, every) subdivision of G is planar.

Proof. (⇒) Given a planar drawing of G, subdividing an edge $\{x, y\}$ means placing the new degree-2 vertex along the drawn curve for that edge; no crossings are introduced.

(⇐) Suppose a subdivision H of G is planar. In a planar drawing of H, every vertex created by subdividing has degree 2. *Suppress* (smooth) each degree-2 vertex by replacing the two incident edge-arcs with a single arc. Repeating this removes all subdivision vertices and produces a planar drawing of G. □

Lemma 2.5.3: Nonplanarity of K_5 and $K_{3,3}$

The graphs K_5 and $K_{3,3}$ are nonplanar.

Proof. We use standard planar edge bounds derived from Euler's formula.

(1) If G is a simple planar graph with $n \geq 3$ vertices, then $m \leq 3n - 6$. Indeed, in a planar embedding each face has boundary length at least 3, so $3f \leq 2m$. Euler's formula gives $n - m + f = 2$, hence

$$2 = n - m + f \leq n - m + \frac{2m}{3} = n - \frac{m}{3} \quad \Rightarrow \quad m \leq 3n - 6.$$

(2) If G is a simple planar *bipartite* graph with $n \geq 3$, then $m \leq 2n - 4$. Now every face has boundary length at least 4, so $4f \leq 2m$, and similarly

$$2 = n - m + f \leq n - m + \frac{m}{2} = n - \frac{m}{2} \quad \Rightarrow \quad m \leq 2n - 4.$$

Now K_5 has $n = 5$ and $m = 10$, but $3n - 6 = 9$, so $10 > 9$ and K_5 is not planar. Also $K_{3,3}$ is bipartite with $n = 6$ and $m = 9$, but $2n - 4 = 8$, so $9 > 8$ and $K_{3,3}$ is not planar. □

Theorem 2.5.1: Kuratowski's Theorem

A (finite) graph G is non-planar if and only if it contains a subgraph that is an expansion (subdivision) of K_5 or of the utility graph $UG = K_{3,3}$.

Proof. We prove both directions.

(⇐) If G contains a subdivision of K_5 or $K_{3,3}$, then G is nonplanar. Assume G contains a subgraph H that is a subdivision of K_5 or $K_{3,3}$. By Lemma 2.5.3, K_5 and $K_{3,3}$ are nonplanar. By Lemma 2.5.2, any subdivision of a nonplanar graph is nonplanar, hence H is nonplanar. If G were planar, then its subgraph H would be planar by Lemma 2.5.1, contradiction. Therefore G is nonplanar.

(⇒) If G is nonplanar, then G contains a subdivision of K_5 or $K_{3,3}$. Let G be nonplanar. Choose a *minimal* nonplanar subgraph $H \subseteq G$: nonplanar, but every proper subgraph of H is planar (minimal with respect to $|V| + |E|$).

Step 1: H has minimum degree at least 3. If H had a vertex x of degree 0 or 1, removing x would not affect connectivity between any other vertices, hence $H - x$ would remain nonplanar, contradicting minimality. If x had degree 2, suppress x (replace its two incident edges by one edge). This operation preserves planarity/nonplanarity by Lemma 2.5.2 (it is the reverse of subdividing). Thus we would obtain a smaller nonplanar graph, again contradicting minimality. Hence $\delta(H) \geq 3$.

Step 2: H is 2-connected. If H had a cut-vertex c, then H could be written as the union of two subgraphs H_1, H_2 intersecting only at c. Since each H_i is a proper subgraph of H, both are planar by minimality. Moreover, planar drawings of H_1 and H_2 can be arranged so that the vertex c lies on the outer face in both drawings; then gluing the drawings at c yields a planar drawing of H, contradiction. So H has no cut-vertex.

Step 3: H contains a Kuratowski subdivision. This is the heart of Kuratowski's theorem. One classical route is to analyze a cycle C in H and the *C-bridges* (components of $H - V(C)$ together with their attachment edges to C), using the minimality of H to force a very constrained attachment pattern:

- Because $H - e$ is planar for every edge e (minimality), in any planar drawing of $H - e$ the endpoints of e must lie on a common face boundary; otherwise e could be added without crossings. This forces multiple internally disjoint u–v connections along the boundary cycle.
- Using $\delta(H) \geq 3$ and 2-connectivity, one shows that there exists a cycle C in H with at least three bridges whose attachment vertices on C *interlace* around the cycle (their endpoints alternate along C). Two interlacing bridges force a crossing in every drawing unless one bridge is routed through the other, which is prevented by minimality.
- From such an interlacing configuration one can extract either (i) five branch vertices each pair connected by internally disjoint paths (a subdivision of K_5), or (ii) two sets of three branch vertices with all cross connections by internally disjoint paths (a subdivision of $K_{3,3}$).

Concretely, the branch vertices are chosen among the attachment points on C plus one vertex in a bridge, and the required paths are built from (a) subpaths of C between successive attachments and (b) internally disjoint paths inside bridges. The interlacing guarantees the "all-to-all" routing needed for K_5 or $K_{3,3}$, while minimality prevents shortcuts that would avoid the forced structure. Therefore H contains a subgraph that is a subdivision of K_5 or $K_{3,3}$.

Since $H \subseteq G$, the same subdivision appears as a subgraph of G. This completes the proof. □

2.6 The Graph Laplacian and Dynamic Processes

2.6.1 The Graph Laplacian

Definition 2.6.1: Graph Laplacian

The *graph Laplacian* $\mathbf{L}$ is a symmetric matrix closely related to the adjacency matrix. It is central to modeling diffusion and random walks on networks.

$$\mathbf{L} = \mathbf{D} - \mathbf{A}$$

where $\mathbf{D}$ is the diagonal matrix of vertex degrees ($D_{ii} = k_i$), and $\mathbf{A}$ is the adjacency matrix.

Written out in full, the elements of the Laplacian matrix are

$$L_{ij} = \begin{cases} k_i & \text{if } i = j, \\ -1 & \text{if } i \neq j \text{ and there is an edge } (i,j), \\ 0 & \text{otherwise.} \end{cases}$$

so it has the degrees of the vertices down its diagonal and a -1 element for every edge.

Tip 2.6.1: Diffusion interpretation of the graph Laplacian

The time evolution of the amount of a commodity ψ_i at vertex i undergoing diffusion is modeled by:

$$\frac{\mathrm{d}\psi}{\mathrm{d}t} + C\mathbf{L}\psi = 0$$

where $C > 0$ is a diffusion constant. The Laplacian encodes the net flow of the commodity out of each vertex to its neighbors.

We can solve the diffusion equation by writing the vector ψ as a linear combination of the eigenvector $\mathbf{v}_i$ of the Laplacian thus:

$$\psi(t) = \sum_i a_i(t)\mathbf{v}_i$$

with the coefficients $a_i(t)$ varying over time. Substituting this from into the diffusion equation and making use of $\mathrm{L}\mathbf{v}_i = \lambda_i \mathbf{v}_i$, where λ_i is the eigenvalue corresponding to the eigenvector $\mathbf{v}_i$, we get

$$\sum_i \left(\frac{\mathrm{d}a_i}{\mathrm{d}t} + C\lambda_i a_i \right) \mathbf{v}_i = 0$$

for all i, which has the solution

$$a_i(t) = a_i(0)e^{-C\lambda_i t}$$

Given an initial condition for the system, as specified by the quantities $a_i(0)$, therefore, we can solve for the state at any later time, provided we know the eigenvalues and eigenvectors of the graph Laplacian.

2.6.2 Laplacian Eigenvalues

The Laplacian is a symmetric matrix with real eigenvalues λ_i.

Properties 2.6.1: Properties of Laplacian Eigenvalues

Let $G = (V, E)$ be an undirected graph with $n = |V|$ and Laplacian $\mathbf{L} = \mathbf{D} - \mathbf{A}$.

(1) Non-negativity: All eigenvalues of $\mathbf{L}$ are non-negative ($\lambda_i \geq 0$).

(2) Lowest eigenvalue: The lowest eigenvalue is always zero, $\lambda_1 = 0$, with eigenvector $\mathbf{1} = (1, 1, \ldots, 1)^\top$.

(3) Number of zero eigenvalues: The number of zero eigenvalues equals the number of connected components (c) in the network.

(4) Algebraic connectivity: The second eigenvalue λ_2 is the *algebraic connectivity.* It is non-zero if and only if the network is connected.

Proof. **(1)** For any $\mathbf{x} \in \mathbb{R}^n$,

$$\mathbf{x}^\top \mathbf{L}\mathbf{x} = \mathbf{x}^\top (\mathbf{D} - \mathbf{A})\mathbf{x} = \sum_{(i,j)\in E} (x_i - x_j)^2 \ \geq \ 0,$$

so $\mathbf{L}$ is positive semidefinite and all eigenvalues satisfy $\lambda_i \geq 0$.

(2) Each row of $\mathbf{L}$ sums to 0, hence $\mathbf{L}\mathbf{1} = \mathbf{0}$. Therefore 0 is an eigenvalue with eigenvector $\mathbf{1}$, and by (1) it is the smallest.

(3) Suppose G has connected components $C_1, \ldots, C_c$. For each r, let $\mathbf{1}_{C_r}$ be the indicator vector of C_r. If $i \in C_r$, then all neighbors of i lie in C_r, so $(\mathbf{L}\mathbf{1}_{C_r})_i = d_i \cdot 1 - \sum_{j\sim i} 1 = 0$; if $i \notin C_r$, both terms are 0. Thus $\mathbf{L}\mathbf{1}_{C_r} = \mathbf{0}$, and the vectors $\mathbf{1}_{C_1}, \ldots, \mathbf{1}_{C_c}$ are linearly independent, so $\dim\ker(\mathbf{L}) \geq c$.

Conversely, if $\mathbf{L}\mathbf{x} = \mathbf{0}$ then

$$0 = \mathbf{x}^\top \mathbf{L}\mathbf{x} = \sum_{(i,j)\in E} (x_i - x_j)^2,$$

so $x_i = x_j$ for every edge $(i, j) \in E$. Hence $\mathbf{x}$ is constant on each connected component, so the solution space has dimension at most c. Therefore $\dim\ker(\mathbf{L}) = c$, i.e., the multiplicity of eigenvalue 0 is c.

(4) If G is disconnected then $c \geq 2$, so by (3) there are at least two zero eigenvalues, hence $\lambda_2 = 0$. If G is connected then $c = 1$, so by (3) the zero eigenvalue has multiplicity 1; thus the next eigenvalue must be strictly positive, i.e. $\lambda_2 > 0$. □

Example 2.6.1

Consider the graph with two components: a single edge 1–2 and an isolated vertex 3. Then

$$\mathbf{L} = \begin{bmatrix} 1 & -1 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

whose eigenvalues are $\{0, 0, 2\}$. There are $c = 2$ components and exactly two zero eigenvalues, so $\lambda_2 = 0$ as expected for a disconnected graph.

2.6.3 Random Walks

Definition 2.6.2: Random Walk on a Graph

A *random walk* is a path where, at each step, an edge is chosen uniformly at random from those attached to the current vertex.

Properties 2.6.2: Random Walk Transition Matrix

The *transition matrix* $\mathbf{P}$ of a random walk on an undirected graph is given by:

$$\mathbf{P} = \mathbf{D}^{-1}\mathbf{A}$$

where $\mathbf{D}$ is the diagonal degree matrix and $\mathbf{A}$ is the adjacency matrix.

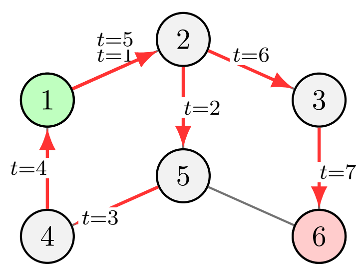


Figure 2.16: Example of a random walk on an undirected graph. The highlighted arrows show one realization: $1 \to 2 \to 5 \to 4 \to 1 \to 2 \to 3 \to 6$.

Example 2.6.2: Random walk and its transition matrix

Consider the undirected graph in figure 2.16 with vertex set $V = \{1, 2, 3, 4, 5, 6\}$ and edge set

$$E = \{\{1, 2\}, \{2, 3\}, \{1, 4\}, \{4, 5\}, \{5, 6\}, \{2, 5\}, \{3, 6\}\}.$$

A *simple random walk* on this graph moves from the current vertex to a uniformly random neighbor. Hence the transition probabilities are

$$P_{ij} = \Pr(X_{t+1} = j \mid X_t = i) = \begin{cases} \frac{1}{\deg(i)} & \text{if } \{i, j\} \in E, \\ 0 & \text{otherwise.} \end{cases}$$

Adjacency and degree matrices. With the vertex order $(1, 2, 3, 4, 5, 6)$, the adjacency matrix is

$$A = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}.$$

The degrees are $\deg(1) = 2, \deg(2) = 3, \deg(3) = 2, \deg(4) = 2, \deg(5) = 3, \deg(6) = 2$, so

$$D = \operatorname{diag}(2, 3, 2, 2, 3, 2), \qquad D^{-1} = \operatorname{diag}\left(\tfrac{1}{2}, \tfrac{1}{3}, \tfrac{1}{2}, \tfrac{1}{2}, \tfrac{1}{3}, \tfrac{1}{2}\right).$$

(continued)

Computing $P = D^{-1}A$. Left-multiplying by D^{-1} scales row i of A by $1/\deg(i)$. Therefore

$$P = D^{-1}A = \begin{bmatrix} 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 \\ \frac{1}{3} & 0 & \frac{1}{3} & 0 & \frac{1}{3} & 0 \\ 0 & \frac{1}{2} & 0 & 0 & 0 & \frac{1}{2} \\ \frac{1}{2} & 0 & 0 & 0 & \frac{1}{2} & 0 \\ 0 & \frac{1}{3} & 0 & \frac{1}{3} & 0 & \frac{1}{3} \\ 0 & 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 \end{bmatrix}.$$

For instance, vertex 2 has three neighbors $\{1, 3, 5\}$, so the second row is $\left[\frac{1}{3}, 0, \frac{1}{3}, 0, \frac{1}{3}, 0\right]$.

Random walks are fundamental stochastic processes on networks, modeling diffusion, search, and navigation. The transition matrix encodes the one-step dynamics, and its powers give multi-step transition probabilities. Random walks underpin algorithms like PageRank and are closely linked to spectral properties of the graph Laplacian.

Properties 2.6.3: Stationary distribution of a random walk (Long-time limit)

On a connected network, the probability p_i that a random walk is found at vertex i is proportional to its degree:

$$p_i = \frac{k_i}{\sum_j k_j} = \frac{k_i}{2m}$$

where k_i is the degree of vertex i and m is the total number of edges in the network.

Proof. Let $G = (V, E)$ be a connected undirected graph with adjacency matrix A, degrees k_i, and transition matrix for the simple random walk

$$P = D^{-1}A, \qquad P_{ij} = \begin{cases} \frac{1}{k_i} & \text{if } \{i, j\} \in E, \\ 0 & \text{otherwise,} \end{cases}$$

where $D = \operatorname{diag}(k_1, \ldots, k_n)$. Define

$$p_i = \frac{k_i}{\sum_j k_j} = \frac{k_i}{2m}, \qquad \text{and let } \mathbf{p} = (p_1, \ldots, p_n).$$

Step 1: Detailed balance. For any edge $\{i, j\} \in E$ we have

$$p_i P_{ij} = \frac{k_i}{2m} \cdot \frac{1}{k_i} = \frac{1}{2m} = \frac{k_j}{2m} \cdot \frac{1}{k_j} = p_j P_{ji}.$$

If $\{i, j\} \notin E$, then $P_{ij} = P_{ji} = 0$ and the equality still holds. Hence

$$p_i P_{ij} = p_j P_{ji} \quad \text{for all } i, j,$$

so the chain is reversible with respect to $\mathbf{p}$.

Step 2: Stationarity. Using detailed balance,

$$(\mathbf{p}P)_j = \sum_i p_i P_{ij} = \sum_i p_j P_{ji} = p_j \sum_i P_{ji} = p_j,$$

because each row of P sums to 1 (equivalently, $\sum_i P_{ji} = 1$ when summing over the incoming index for fixed j). Therefore $\mathbf{p}P = \mathbf{p}$, so $\mathbf{p}$ is a stationary distribution.

Step 3: Long-time limit. Since G is connected, the random walk Markov chain is irreducible. If in addition the walk is aperiodic (e.g. the graph is not bipartite, or we use a lazy random walk), then the distribution converges to the unique stationary distribution $\mathbf{p}$ as $t \to \infty$. In any case, $\mathbf{p}$ is the stationary distribution, and it satisfies $p_i \propto k_i$. □

Definition 2.6.3: First Passage Time

The mean first passage time τ for a random walk from vertex u to vertex v can be calculated using the inverse of the reduced Laplacian $\mathbf{L}'$.

$$\tau_{uv} = \frac{1}{p_v}\left((\mathbf{L}')^{-1}_{vv} - (\mathbf{L}')^{-1}_{uv}\right)$$

where p_v is the stationary probability of vertex v.

2.6.4 Resistor Networks

The flow of current in a network where edges are identical resistors and vertices are junctions is governed by Kirchhoff's current law, which can be expressed using the Laplacian

$$\mathbf{L}\mathbf{V} = R\mathbf{I}$$

where $\mathbf{V}$ is the voltage vector, R is the resistance, and $\mathbf{I}$ is the vector of injected external currents. Since $\mathbf{L}$ is singular (zero eigenvalue), this equation is typically solved using the inverse of the reduced Laplacian $\mathbf{L}'$.

Multiple Choice Questions

1. For a simple undirected graph (no multi-edges, no self-edges), which condition is BOTH necessary and sufficient for a matrix A to be its adjacency matrix?

   (A) A is symmetric and has all diagonal entries equal to 1

   (B) A is symmetric and has all diagonal entries equal to 0

   (C) A is skew-symmetric and has all diagonal entries equal to 0

   (D) A is symmetric and each row sums to 1

2. In the "non-simple undirected" convention used in the notes, a single self-edge at vertex i is encoded by

   (A) $A_{ii} = 1$ because it is one edge

   (B) $A_{ii} = 2$ because it contributes 2 to the degree

   (C) $A_{ii} = \deg(i)$ because the diagonal stores degrees

   (D) $A_{ii} = 0$ because self-edges are excluded from adjacency matrices

3. Consider an undirected multigraph where between vertices 2 and 5 there are exactly 4 parallel edges and no self-edges. Which entry must reflect this?

   (A) $A_{25} = 1$ and $A_{52} = 1$

   (B) $A_{25} = 4$ and $A_{52} = 4$

   (C) $A_{25} = 4$ and $A_{52} = 0$

(D) $A_{22} = 4$ and $A_{55} = 4$

4. A common convention for directed graphs is $A_{ij} = 1$ if there is an edge $i \to j$. In the file's convention instead, $A_{ij} = 1$ means

   (A) an edge $i \to j$

   (B) an edge $j \to i$

   (C) an undirected edge $\{i, j\}$

   (D) a 2-step path $j \to \cdots \to i$

5. Under the file's directed convention ($A_{ij} = 1$ if $j \to i$), the out-degree of vertex i is computed as

   (A) $\sum_{j=1}^{n} A_{ij}$

   (B) $\sum_{j=1}^{n} A_{ji}$

   (C) A_{ii}

   (D) $\sum_{j=1}^{n} |A_{ij} - A_{ji}|$

6. In a weighted undirected network with adjacency matrix A, which statement is most accurate?

   (A) $A_{ij} \in \{0, 1\}$ always

   (B) A_{ij} stores edge multiplicity but not strength

   (C) A_{ij} equals the weight/strength of edge $\{i, j\}$ (and 0 if absent)

   (D) weights must appear only on the diagonal

7. Suppose A is a 5×5 adjacency matrix of a simple undirected graph. If the sum of all entries of A is 24, how many edges does the graph have?

   (A) 12

   (B) 24

   (C) 6

   (D) 48

8. Let A be the adjacency matrix of a simple undirected graph. Which quantity equals the degree k_i of vertex i?

   (A) the ith diagonal entry A_{ii}

   (B) the sum of the ith row of A

   (C) the sum of the ith row of A^2

   (D) the determinant of A

9. Consider a simple undirected graph. If A is its adjacency matrix and $D = \mathrm{diag}(k_1, \ldots, k_n)$, which matrix identity is always true?

   (A) $A = D - A$

   (B) $A = D^{-1}A$

   (C) $D_{ii} = \sum_j A_{ij}$

   (D) $D = A^{-1}$

10. A symmetric 0-1 matrix with a nonzero diagonal entry can be the adjacency matrix of

    (A) a simple graph (by definition)

    (B) a multigraph with self-edges (under the file's undirected convention)

    (C) a directed acyclic graph (always)

    (D) a bipartite graph (always)

11. In a simple undirected graph with n vertices and m edges, the mean degree c satisfies

    (A) $c = \dfrac{m}{n}$

    (B) $c = \dfrac{2m}{n}$

    (C) $c = \dfrac{n}{2m}$

    (D) $c = \dfrac{2m}{n(n-1)}$

12. For a simple undirected graph, the connectance (density) ρ is

    (A) $\rho = \dfrac{m}{n^2}$

    (B) $\rho = \dfrac{2m}{n(n-1)}$

    (C) $\rho = \dfrac{n(n-1)}{2m}$

    (D) $\rho = \dfrac{c}{n}$

13. Suppose $n = 100$ and the mean degree is $c = 6$. Using $\rho = \dfrac{c}{n-1}$, which is closest to ρ?

(A) 0.0600
(B) 0.0303
(C) 0.0061
(D) 0.0006

14. A sequence of graphs is called *sparse* (as $n \to \infty$) if
    (A) $\rho \to 1$
    (B) $\rho \to 0$ (equivalently mean degree tends to a constant)
    (C) $m \sim n^2$
    (D) the graph is disconnected for all n

15. In a directed network, the file states that the sum of all in-degrees equals
    (A) 0
    (B) the number of vertices n
    (C) the number of edges m
    (D) twice the number of edges $2m$

16. Which statement is always true for any directed graph (with the usual counting of directed edges)?
    (A) average in-degree equals average out-degree
    (B) average in-degree equals average total degree in the underlying undirected graph
    (C) in-degree sequence must match out-degree sequence entrywise
    (D) the adjacency matrix must be symmetric

17. A *component* of an undirected graph is best described as
    (A) any connected subgraph
    (B) a maximal connected subgraph
    (C) a subgraph with minimum degree at least 2
    (D) a spanning tree

18. Which removal can turn a connected graph into a disconnected graph?
    (A) removing any vertex
    (B) removing any edge
    (C) removing a cutvertex
    (D) removing any two edges

19. A *bridge* is an edge whose removal
    (A) decreases the number of components
    (B) increases the number of components
    (C) makes the graph complete
    (D) always decreases the minimum degree

20. A graph is k-connected (vertex-connectivity notion) if
    (A) removing any k edges keeps it connected
    (B) removing fewer than k vertices never disconnects it (and $|G| > k$)
    (C) it contains a clique of size k
    (D) its minimum degree is exactly k

21. The vertex connectivity $\kappa(G)$ is
    (A) the smallest degree in G
    (B) the largest k for which G is k-connected
    (C) the number of components in G
    (D) the number of bridges in G

22. The edge-connectivity $\lambda(G)$ is
    (A) the largest ℓ such that removing fewer than ℓ edges keeps G connected
    (B) the number of edges in a spanning tree
    (C) always equal to $\kappa(G)$
    (D) always equal to $2m/n$

23. For every non-trivial graph G, which inequality chain is guaranteed?
    (A) $\delta(G) \le \lambda(G) \le \kappa(G)$
    (B) $\kappa(G) \le \lambda(G) \le \delta(G)$
    (C) $\lambda(G) \le \kappa(G) \le \delta(G)$
    (D) $\kappa(G) = \lambda(G) = \delta(G)$

24. If $\delta(G) = 1$ for a connected graph G with at least 2 vertices, what can you conclude for sure?
    (A) $\kappa(G) \geq 2$
    (B) $\lambda(G) \geq 2$
    (C) $\lambda(G) \leq 1$
    (D) G is complete

25. Which is a correct interpretation of $\kappa(G) \geq 2$ in network reliability language?
    (A) no single edge failure can disconnect the network
    (B) no single vertex failure can disconnect the network
    (C) every vertex has degree exactly 2
    (D) the graph has exactly two components

26. Mader's theorem (as stated) guarantees a k-connected subgraph if the average degree satisfies
    (A) $\bar{d}(G) \geq 2k$
    (B) $\bar{d}(G) \geq 3k$
    (C) $\bar{d}(G) \geq 4k$
    (D) $\bar{d}(G) \geq k+1$

27. In the pruning step used in the proof sketch of Mader's theorem, vertices are repeatedly deleted while they have degree at most
    (A) $k-1$
    (B) $2k-1$
    (C) $4k-1$
    (D) $k+1$

28. A directed graph is *weakly connected* if
    (A) it has a directed path between every ordered pair of vertices
    (B) it becomes connected after ignoring edge directions
    (C) every vertex has positive in-degree
    (D) it has no directed cycles

29. A directed graph is *strongly connected* if
    (A) the underlying undirected graph is connected
    (B) every vertex has the same in-degree as out-degree
    (C) there is a directed path from any vertex to any other vertex
    (D) its adjacency matrix is symmetric

30. The out-component of a vertex v in a digraph is the set of vertices
    (A) that can reach v
    (B) reachable from v by directed paths
    (C) adjacent to v by an incoming edge
    (D) in the same weakly connected component as v

31. A directed acyclic graph (DAG) must satisfy
    (A) it has no self-edges and no directed cycles
    (B) it is strongly connected
    (C) its adjacency matrix is symmetric
    (D) every vertex has out-degree at least 1

32. If vertices of a DAG are labeled so every edge points from higher label to lower label, then its adjacency matrix is
    (A) strictly lower triangular
    (B) strictly upper triangular
    (C) diagonal
    (D) symmetric

33. The file states: a directed network is acyclic if and only if all eigenvalues of its adjacency matrix are
    (A) positive
    (B) equal to 1
    (C) equal to 0
    (D) purely imaginary

34. Two u–v paths are *vertex-independent* if they
    (A) share no edges at all

(B) share no internal vertices (they may share endpoints)

(C) have the same length

(D) lie in different components

35. A u–v vertex cut set S is a subset of

(A) V that must include u and v

(B) $V \setminus \{u, v\}$ whose removal destroys all u–v paths

(C) E whose removal destroys all u–v paths

(D) V of size exactly $\kappa(G)$

36. The local form of Menger's theorem states that the maximum number of pairwise edge-independent u–v paths equals

(A) $\deg(u) + \deg(v)$

(B) the minimum size of a u–v edge cut set

(C) $\kappa(G)$

(D) the number of components of G

37. If there is no u–v vertex cut set of size $< k$, then you can conclude

(A) there are at least k vertex-independent u–v paths

(B) there are exactly k vertex-independent u–v paths

(C) there are at most k vertex-independent u–v paths

(D) u and v are adjacent

38. In a flow network, flow conservation is required at

(A) every vertex, including s and t

(B) s only

(C) t only

(D) every vertex except s and t

39. The value of an s–t flow f is computed as

(A) total inflow to t minus total outflow from t

(B) total outflow from s minus total inflow to s

(C) sum of capacities of all arcs

(D) number of vertices in the network

40. For a cut S with $s \in S$ and $t \notin S$, the cut capacity is

(A) $\sum_{a \in \delta^+(S)} f(a)$

(B) $\sum_{a \in \delta^+(S)} c(a)$

(C) $\sum_{a \in \delta^-(S)} c(a)$ only

(D) the number of vertices in S

41. Max-Flow/Min-Cut states that

(A) every cut has capacity equal to the maximum flow

(B) the maximum flow value equals the minimum cut capacity

(C) the minimum flow value equals the maximum cut capacity

(D) maximum flow exists only when the graph is undirected

42. The integrality remark implies: if all capacities are integers, then there exists a maximum flow with

(A) irrational arc values

(B) all arc flows integral

(C) all arc flows equal to capacities

(D) exactly one s–t path used

43. In the construction proving Menger via Max-Flow/Min-Cut (vertex version), why "split" each internal vertex x into $x_{in} \to x_{out}$ with capacity 1?

(A) to force paths to be edge-disjoint

(B) to enforce that at most one unit of flow uses x (vertex-disjointness)

(C) to make the network undirected

(D) to ensure the adjacency matrix is upper triangular

44. A hypergraph differs from a (simple) graph because its edges
   (A) must be directed
   (B) may connect any number of vertices (as subsets)
   (C) must have weights
   (D) cannot share vertices

45. A graph is bipartite if and only if
   (A) its adjacency matrix is diagonal
   (B) its vertices can be partitioned into U, V with edges only between U and V
   (C) it has no cycles
   (D) it is planar

46. In a bipartite "groups vs participants" representation, the incidence matrix B is sized
   (A) $n \times n$
   (B) $g \times n$
   (C) $n \times g$
   (D) $g \times g$

47. In the one-mode projection onto participants, the matrix $P = B^\top B$ has the property that for $i \neq j$, P_{ij} equals
   (A) the number of groups participant i belongs to
   (B) the number of participants in group j
   (C) the number of groups shared by participants i and j
   (D) 1 if i and j share at least one group, else 0 (always unweighted)

48. Why does the formula $A = B^\top B - D$ appear for the adjacency matrix of the (weighted) projection?
   (A) to remove negative weights
   (B) to remove diagonal self-loops created by $B^\top B$
   (C) to symmetrize $B^\top B$
   (D) to normalize by degrees so rows sum to 1

49. Which statement characterizes a tree on n vertices?
   (A) it has exactly n edges
   (B) it has no cycles and is connected
   (C) it is disconnected but acyclic
   (D) it is planar and bipartite

50. A tree with n vertices has exactly
   (A) $n + 1$ edges
   (B) n edges
   (C) $n - 1$ edges
   (D) $2n - 2$ edges

51. Planarity is preserved under taking subgraphs means:
   (A) any supergraph of a planar graph is planar
   (B) any subgraph of a planar graph is planar
   (C) any planar graph must be a tree
   (D) any planar graph must be bipartite

52. A simple planar graph with $n \geq 3$ vertices must satisfy the edge bound
   (A) $m \leq 2n - 4$
   (B) $m \leq 3n - 6$
   (C) $m \leq n - 1$
   (D) $m \leq n(n - 1)$

53. A simple planar *bipartite* graph with $n \geq 3$ vertices must satisfy
   (A) $m \leq 3n - 6$
   (B) $m \leq 2n - 4$
   (C) $m \leq n - 1$
   (D) $m \leq n + 2$

54. Kuratowski's theorem says G is nonplanar iff it contains a subgraph that is a subdivision (expansion) of
   (A) K_4 or C_5

(B) K_5 or $K_{3,3}$

(C) $K_{2,3}$ or $K_{3,4}$

(D) C_3 or C_4

55. For an undirected graph, the Laplacian is defined as

(A) $L = A - D$

(B) $L = D - A$

(C) $L = D^{-1}A$

(D) $L = A^2$

56. For $i \neq j$, the Laplacian entry L_{ij} equals -1 exactly when

(A) $i = j$

(B) $\{i, j\} \in E$

(C) vertex i has degree 1

(D) there is a 2-step path from i to j

57. Which vector is always an eigenvector of L with eigenvalue 0 (for any undirected graph)?

(A) the degree vector $(k_1, \dots, k_n)^\top$

(B) the all-ones vector $\mathbf{1} = (1, \dots, 1)^\top$

(C) any standard basis vector e_i

(D) the indicator of a single edge

58. The multiplicity of eigenvalue 0 of the Laplacian equals

(A) the number of edges

(B) the number of vertices

(C) the number of connected components

(D) the minimum degree

59. The algebraic connectivity is

(A) λ_1 (smallest Laplacian eigenvalue)

(B) λ_2 (second-smallest Laplacian eigenvalue)

(C) the largest Laplacian eigenvalue

(D) $\det(L)$

60. For an undirected graph random walk, the transition matrix is

(A) $P = AD^{-1}$

(B) $P = D^{-1}A$

(C) $P = D - A$

(D) $P = A^\top A$

61. In a simple random walk on an undirected graph, if $\deg(i) = 5$ and $\{i, j\} \in E$, then $P_{ij} =$

(A) 5

(B) 1

(C) 1/5

(D) depends on $\deg(j)$

62. On a connected undirected graph, the stationary distribution π_i of the simple random walk is proportional to

(A) $1/k_i$

(B) k_i

(C) k_i^2

(D) the number of triangles containing i

Free-Response Questions

1. **Inference (modeling choice).** You are given data where each "event" can involve 2–50 people (e.g., meeting attendance), and you care about (i) who co-attends events and (ii) which events are "hubs."

   (a) Argue (with 2–3 concrete reasons) when a *hypergraph* representation is more faithful than a simple projected graph on people.

   (b) Explain what information is lost when you replace a hypergraph by a one-mode projection.

(c) Propose one statistic that is meaningful on the hypergraph but can become misleading after projection.

2. **Inference (directed connectivity).** In a directed citation network, explain (carefully) when it makes more sense to analyze *weakly connected components* vs *strongly connected components.*

   (a) Give a scenario where WCCs answer a natural "reachability" question but SCCs do not.

   (b) Give a scenario where SCCs capture a "feedback/reciprocity" phenomenon but WCCs hide it.

3. **Inference (robustness).** Explain the practical difference between $\kappa(G)$ and $\lambda(G)$ in network reliability.

   (a) Give an example (described in words) of a network where $\lambda(G)$ is large but $\kappa(G)$ is small.

   (b) What does this imply about whether you should harden routers (vertices) or cables (edges)?

4. **Inference (acyclicity test).** The notes mention a connection between DAGs and eigenvalues of the adjacency matrix.

   (a) Explain why "all eigenvalues are 0" is a strong signature of acyclicity in the *unweighted* DAG setting.

   (b) Describe one way this criterion could fail or become subtle if you change assumptions (e.g., weights, different adjacency convention, or numerical computation).

5. **Inference (random walk intuition).** On a connected undirected graph, the stationary distribution of the simple random walk satisfies $\pi_i \propto k_i$.

   (a) Give an intuitive explanation for why high-degree vertices get higher stationary probability.

   (b) If you *add one edge* between two low-degree vertices, do you expect the walk to mix faster or slower? Justify using concepts like bottlenecks/bridges or algebraic connectivity.

6. **Proof (handshake lemma and mean degree).** Let G be a simple undirected graph with n vertices and m edges and degrees $k_1, \ldots, k_n$.

   (a) Prove $\sum_{i=1}^{n} k_i = 2m$.

   (b) Deduce that the mean degree $c = \frac{1}{n}\sum_{i=1}^{n} k_i$ equals $\frac{2m}{n}$.

   (c) Explain in one sentence why the same "$2m$" statement can fail *as written* for directed graphs (and how it must be modified).

7. **Proof (connectance formula).** For a simple undirected graph on n vertices with m edges, define connectance (density) ρ as the fraction of possible edges that are present.

   (a) Show that the number of possible edges is $\binom{n}{2}$.

   (b) Prove $\rho = \dfrac{m}{\binom{n}{2}} = \dfrac{2m}{n(n-1)}$.

(c) Using your result from the previous proof, show $\rho = \dfrac{c}{n-1}$.

8. **Proof (Laplacian zero eigenvalue).** Let $L = D - A$ be the Laplacian of an undirected graph.

   (a) Prove that $L\mathbf{1} = \mathbf{0}$ where $\mathbf{1} = (1, \ldots, 1)^\top$.

   (b) Conclude that 0 is always an eigenvalue of L.

   (c) Explain what property of L (rows sum to 0) you used, and why that property holds from the definition $L = D - A$.

9. **Proof (stationary distribution for random walk).** Let G be a connected undirected graph with adjacency matrix A, degree matrix D, and random-walk transition matrix $P = D^{-1}A$.

   (a) Define $\pi_i = \dfrac{k_i}{\sum_j k_j}$. Show that π is a probability vector.

   (b) Prove the detailed balance condition $\pi_i P_{ij} = \pi_j P_{ji}$ for adjacent vertices $i \sim j$.

   (c) Deduce from detailed balance that $\pi^\top P = \pi^\top$ (i.e., π is stationary).

10. **Proof (planar edge bound).** Let G be a simple connected planar graph with $n \geq 3$ vertices and m edges.

    (a) State Euler's formula relating n, m, and the number of faces f.

    (b) Argue that every face has boundary length at least 3 (why?).

    (c) Use double-counting of edge-face incidences to show $3f \leq 2m$.

    (d) Combine the above to conclude $m \leq 3n - 6$.

11. **Computation (undirected multigraph adjacency + Laplacian).** Consider an undirected multigraph on vertices $\{1, 2, 3, 4, 5, 6\}$ with: two parallel edges between 1 and 2; one edge between 2 and 3; three parallel edges between 3 and 4; one edge between 4 and 5; one edge between 5 and 6; and one self-loop at vertex 3.

    (a) Write the adjacency matrix A using the convention that a self-loop contributes $A_{33} = 2$.

    (b) Compute the degree sequence $(k_1, \ldots, k_6)$ and the total number of edges m.

    (c) Form the degree matrix D and Laplacian $L = D - A$.

    (d) Compute the number of connected components from L (explain your reasoning).

12. **Computation (directed degrees + SCC/WCC).** Consider the directed graph on vertices $\{1, 2, 3, 4, 5, 6, 7\}$ with directed edges:

$$1 \to 2,\ 2 \to 3,\ 3 \to 1,\ 3 \to 4,\ 4 \to 5,\ 5 \to 4,\ 5 \to 6,\ 6 \to 7,\ 7 \to 6.$$

    (a) Using the convention $A_{ij} = 1$ if $j \to i$, write the adjacency matrix A.

    (b) Compute the in-degree and out-degree of each vertex from A (state clearly which sums you use).

(c) List all strongly connected components (SCCs).

(d) List all weakly connected components (WCCs).

13. **Computation (paths via matrix powers).** Let G be the simple undirected graph on $\{1, 2, 3, 4, 5, 6\}$ with edges

$$\{1, 2\}, \{1, 3\}, \{2, 3\}, \{2, 4\}, \{3, 5\}, \{4, 5\}, \{4, 6\}, \{5, 6\}.$$

(a) Write the adjacency matrix A (in the vertex order $1, 2, 3, 4, 5, 6$).

(b) Compute $(A^2)_{1,6}$ and interpret it as a number of length-2 paths.

(c) Compute $(A^3)_{1,6}$ and interpret it as a number of length-3 walks.

(d) Explain why $(A^2)_{i,i}$ counts (something) related to degree/2-cycles, and state what it equals here for $i = 2$.

14. **Computation (cutvertices and bridges).** Consider the undirected graph with vertex set $\{1, 2, 3, 4, 5, 6, 7, 8\}$ and edges

$$\{1, 2\}, \{2, 3\}, \{3, 4\}, \{4, 1\} \quad \text{(a 4-cycle)}$$

plus a "tail" $\{4, 5\}, \{5, 6\}, \{6, 7\}, \{6, 8\}$.

(a) Identify all bridges and justify each choice.

(b) Identify all cutvertices and justify each choice.

(c) Compute $\lambda(G)$ and $\kappa(G)$ (give short arguments, not just answers).

(d) If you add the edge $\{5, 7\}$, recompute which edges remain bridges and discuss how $\lambda(G)$ changes.

15. **Computation (random walk transition + stationary + 2-step probability).** Let G be the undirected graph on $\{1, 2, 3, 4, 5\}$ with edges

$$\{1, 2\}, \{1, 3\}, \{2, 3\}, \{2, 4\}, \{3, 5\}, \{4, 5\}.$$

(a) Compute A, D, and the random-walk transition matrix $P = D^{-1}A$.

(b) Compute the stationary distribution π and verify $\sum_i \pi_i = 1$.

(c) Starting at vertex 1, compute $\mathbb{P}(X_2 = 5)$ (probability to be at 5 after 2 steps).

(d) Determine whether the chain is bipartite from the graph structure, and explain what that implies about convergence (qualitatively).

16. **Computation (max flow / min cut).** Consider the flow network with vertices $\{s, a, b, c, t\}$ and directed edges with capacities:

$$s \to a : 7, \quad s \to b : 4, \quad a \to b : 3, \quad a \to c : 5, \quad b \to c : 6, \quad b \to t : 5, \quad c \to t : 8.$$

(a) Compute a maximum s–t flow (show at least two augmenting steps if you use Ford–Fulkerson).

(b) State the value of your max flow.

(c) Find a minimum cut and compute its capacity.

(d) Verify Max-Flow/Min-Cut by matching your answers from (b) and (c).

17. **Computation (vertex-disjoint paths via splitting construction).** Let G be the undirected graph on vertices $\{u, 1, 2, 3, 4, v\}$ with edges:

$$\{u, 1\}, \{u, 2\}, \{1, 3\}, \{2, 3\}, \{1, 4\}, \{2, 4\}, \{3, v\}, \{4, v\}.$$

(a) Convert this to a directed flow network to compute the maximum number of internally vertex-disjoint u–v paths (use vertex splitting with capacity 1 on internal vertices).

(b) Determine the maximum number of vertex-disjoint u–v paths.

(c) Exhibit that many disjoint paths explicitly in the original graph.

(d) Find a minimum u–v vertex cut set and verify Menger's theorem locally.

18. **Computation (bipartite incidence + one-mode projection).** A bipartite "group–participant" system has 4 groups $g_1, \ldots, g_4$ and 6 participants $p_1, \ldots, p_6$ with memberships:

$$g_1 : \{p_1, p_2, p_3\}, \quad g_2 : \{p_2, p_3, p_4\}, \quad g_3 : \{p_1, p_4, p_5\}, \quad g_4 : \{p_3, p_5, p_6\}.$$

(a) Write the incidence matrix B (rows = groups, columns = participants).

(b) Compute $P = B^\top B$.

(c) Form the projected (weighted) adjacency matrix $A_{\text{proj}} = P - D$ where $D = \text{diag}(P)$.

(d) Identify the pair of participants with the strongest tie in the projection and interpret the weight.

19. **Computation (planarity test using bounds + obstruction reasoning).** Let H be the graph obtained from $K_{3,3}$ by adding one extra edge between two vertices on the same side of the bipartition.

(a) Compute n and m for H.

(b) Use the planar bound $m \leq 3n-6$ to decide whether this bound alone can certify nonplanarity.

(c) Explain why H is nonplanar anyway (give a subgraph/subdivision argument).

(d) If you delete exactly one edge from H, is it guaranteed planar? Give a justification (not just a guess).

20. **Computation (Laplacian spectrum / algebraic connectivity comparison).** Consider the two graphs on vertices $\{1, 2, 3, 4\}$:

$$G_1 = P_4 : \{1, 2\}, \{2, 3\}, \{3, 4\} \quad \text{(path)}$$

$$G_2 = C_4 : \{1, 2\}, \{2, 3\}, \{3, 4\}, \{4, 1\} \quad \text{(cycle)}.$$

(a) Compute L_1 and L_2.

(b) Compute the eigenvalues of L_1 and L_2 (exact values).

(c) Identify λ_2 (algebraic connectivity) for each graph.

(d) Using λ_2, explain which graph is "more connected" in the spectral sense and why that matches intuition.