Lecture 06
Network Analysis 1

Andrew Sonta

CIVIL 534: Computational systems thinking for sustainable engineering

02 April 2025

Outline

- Exam review
- Introduction to network analysis
- Adjacency matrices
- Network types
- Cycles
- Bipartite networks
- Network degree
- Paths, cut sets, and min-cut/max-flow

- Corresponding parts of Newman: Chapter 1; 6.1-6.4; 6.6; 6.10-6.11
- Assignment 2 out today, due May 2
- Milestone 1 due Friday

Course feedback

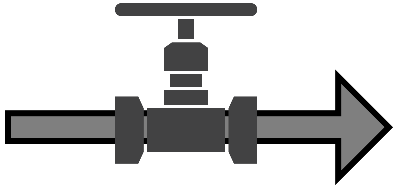
- Only 4/16 responses – shame!
- As a result… not much to report!

Systems thinking definitions

stock

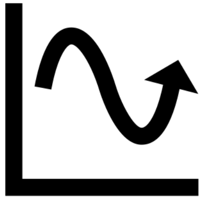
Elements that can be measured as quantities

flow



Flows cause changes in quantities of stocks

dynamics



Dynamics represent behaviors of stocks and flows over time

Network definitions

Objects: **nodes** **V**

Interactions: **edges** **E**

System: **network** **G(V,E)**

Network	**Nodes**	**Edges**/Links
Graph	Vertices	Edges
Social network	Actors	Ties
Physics network	Sites	Bonds

Refining definitions…

Systems thinking

- Refers to the description of systems as **stocks** and **flows**
- Focused on dynamics of how certain components of the system affect others
- First half of the course

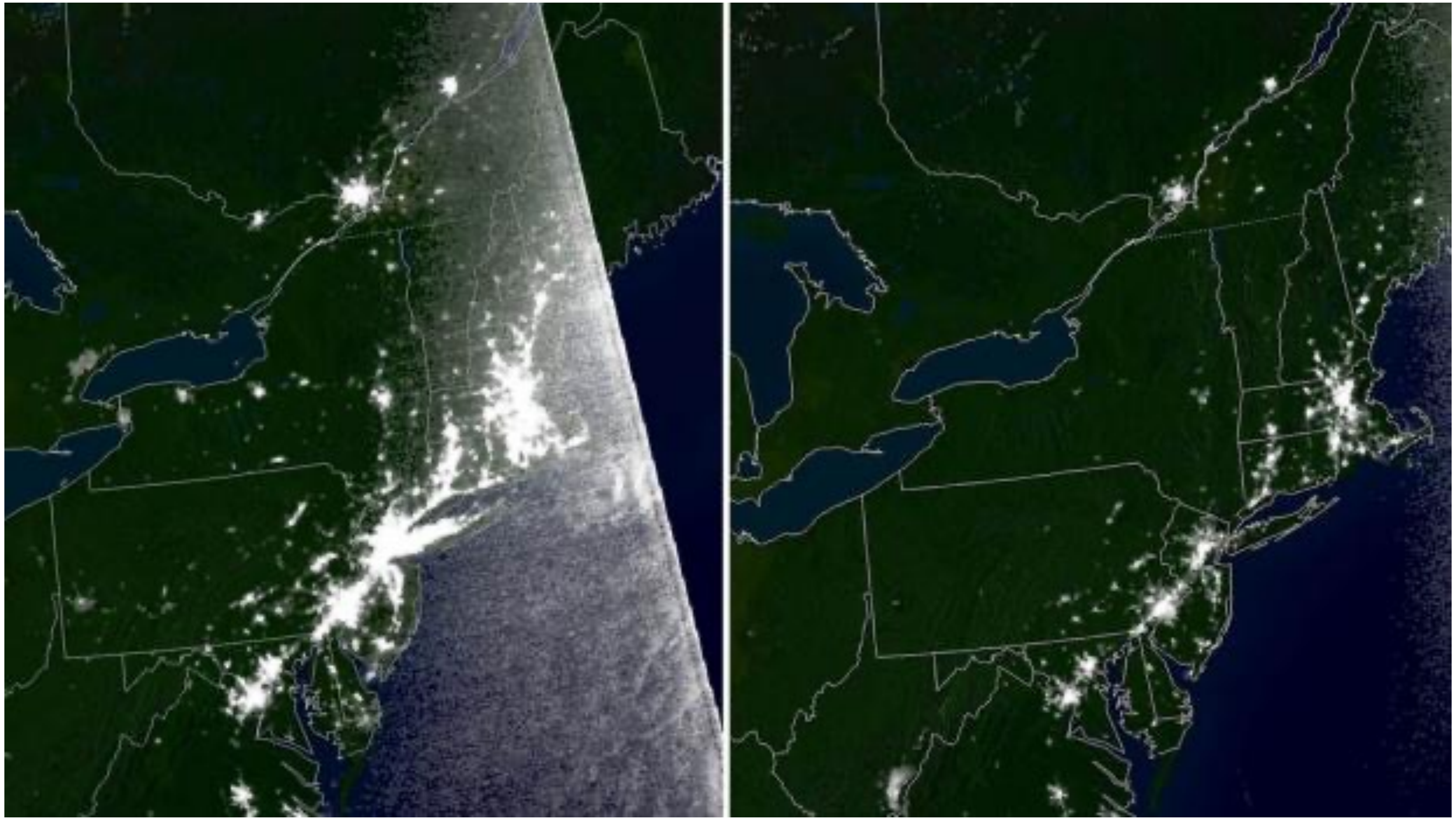
Network analysis

- Refers to the definition of a “system” as **nodes** and **edges**
- Focused on **mathematical structure** of interconnections in the system

Why do we study networks?

- Networks are everywhere
- Networks are a general language for describing complex systems
- Cities are full of complex systems:
  - Transportation networks
  - Electric grid
  - Human social networks
  - Networks of buildings?

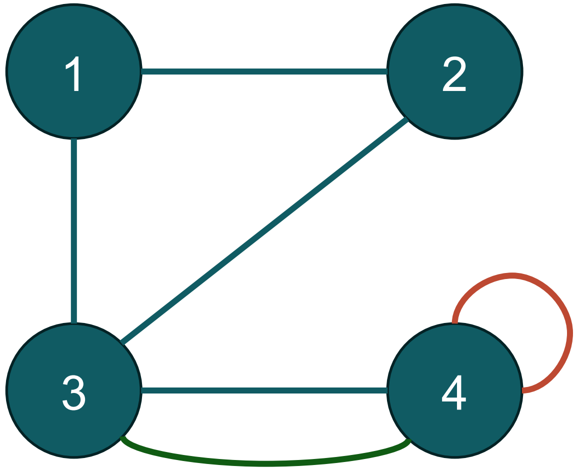
2003 Blackout in Northeastern US



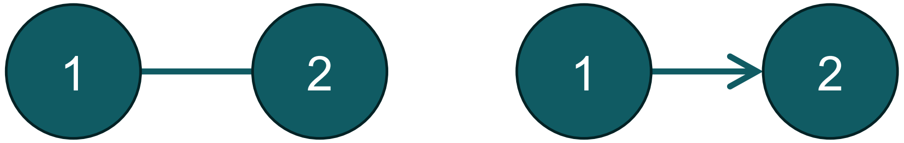
- A failure of one part of the system led to a cascade of multiple failures across the region
- Understanding network structure and behavior could have reduced the extent of the blackout

Network notation

- # of nodes = n
- # of edges = m
- Multi-edge
- Self-loop



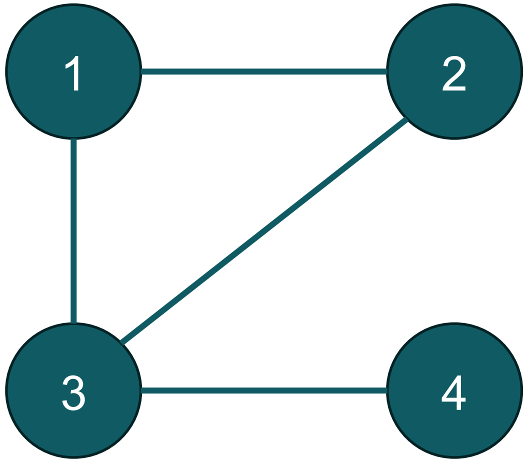
- Undirected network
- Directed network



Adjacency matrix

- Mathematical building block of network analysis
- Symmetric (n x n)
- Indexed by *i* (row) and *j* (column)
- For simple undirected:

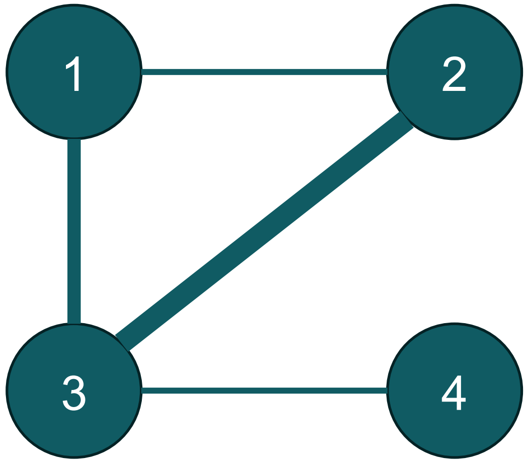
$$A_{i,j} = \begin{cases} 1 & \text{if edge} \\ 0 & \text{else} \end{cases}$$



$$A = \underbrace{\begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{j} \Bigg\} i$$

Weighted networks

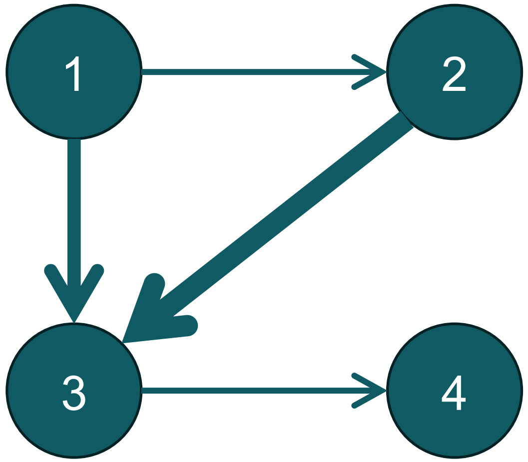
- When we assign ***values*** to edges
- Note: not the same as multi-edges, but we can think of them the same when values are integers
- Examples:
  - Strength of working relationship
  - Capacity in water pipes



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

Directed networks

- Edges have ***directionality***
- Can also have self-edges and multi-edges
- Examples:
  - Transport of people
  - Energy flow
- Adjacency matrix (no longer symmetric):



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

$$A_{i,j} = \begin{cases} 1 & \text{if edge from } j \to i \\ 0 & \text{else} \end{cases}$$

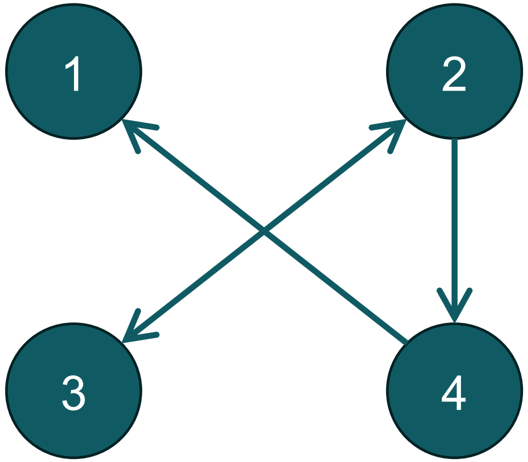
Warning: this convention is not universal!

Directed networks

- Good for representing complex and non-symmetric dynamics
- Some network analysis tools are not defined for directed networks
- Often we need to translate from a directed to an undirected network

Other representations

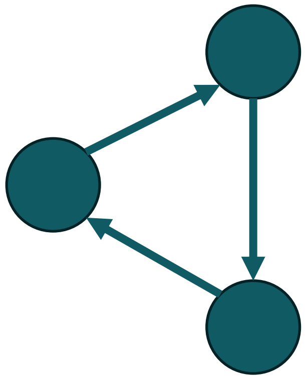
- **Edge list**
  - (2,3)
  - (2,4)
  - (3,2)
  - (4,1)
- **Adjacency list**
  - 1:
  - 2: 3, 4
  - 3: 2
  - 4: 1
- These are commonly used in computational network packages for saving network data



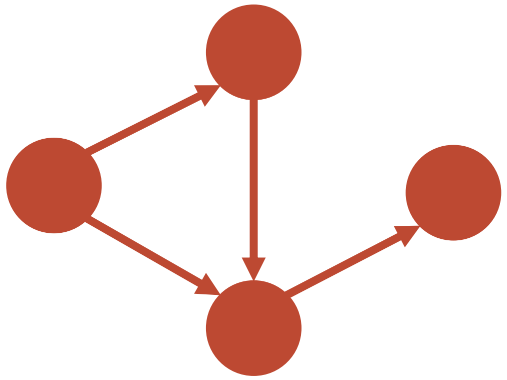
note: directed network!

Cycles and DAGs (Directed Acyclic Graphs)

- Cycle: closed loop of edges with arrows pointing the same way around the loop



- DAG: Graph/network with no cycles
  - Often denotes dependency or sequence
    - Time dependency
    - Causal relationship

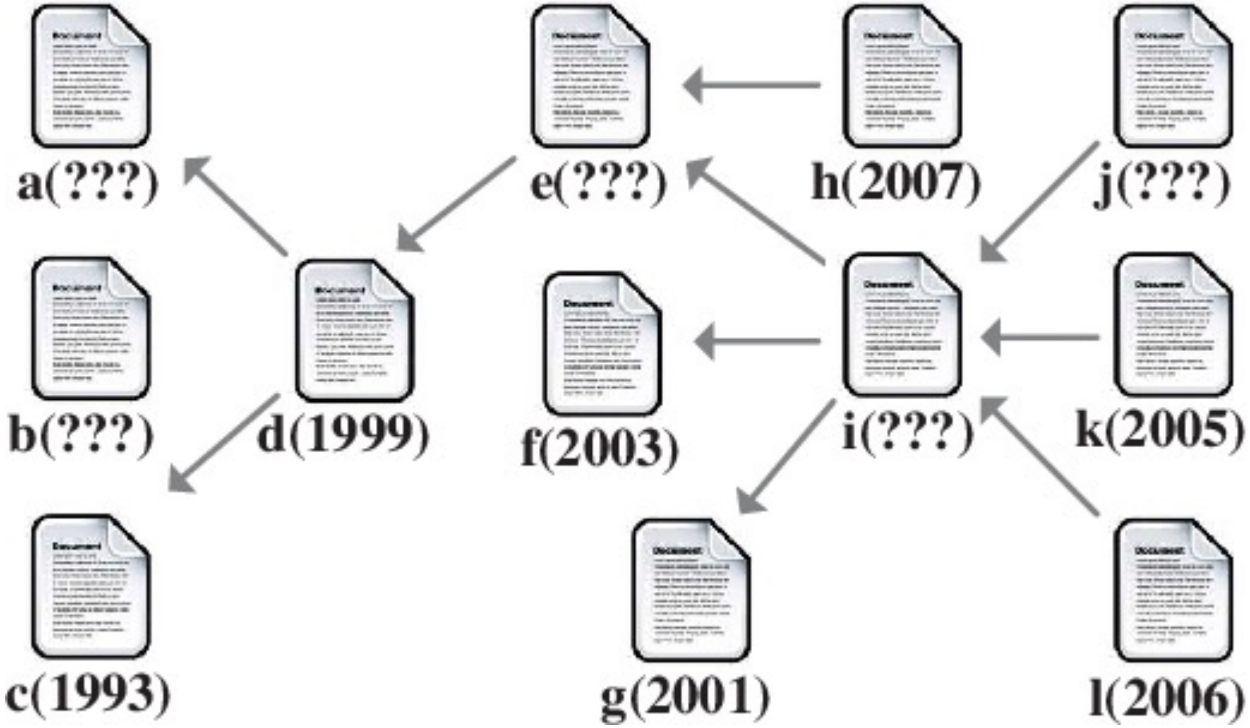


Examples of cycles or lack thereof?

- Water network: if no cycles, water that leaves a node never returns
- Causality: causal relationships are often defined as a DAG
- Others?

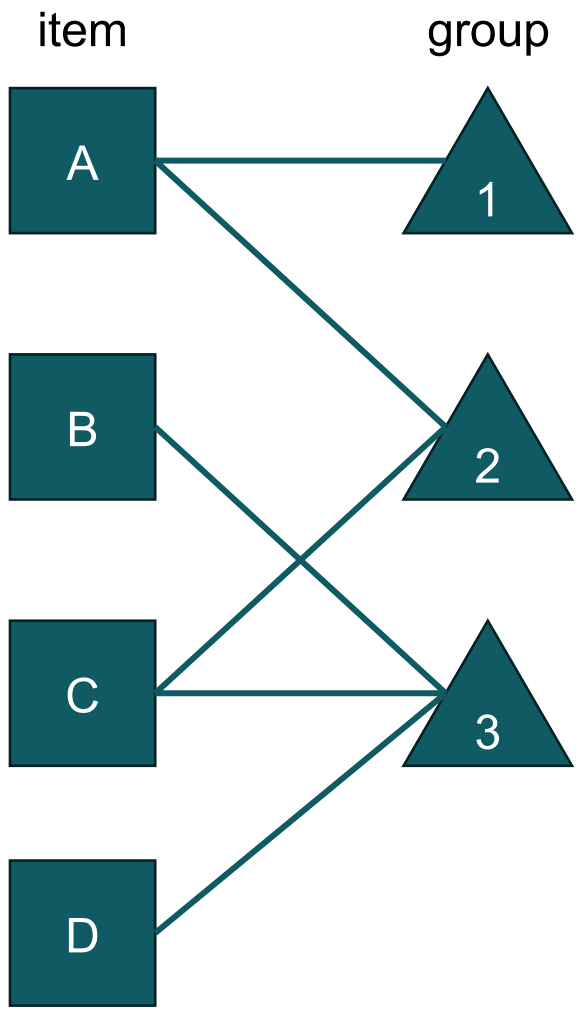
Comprehension check: What type of graph is this?

- Citation network showing which papers cite which papers



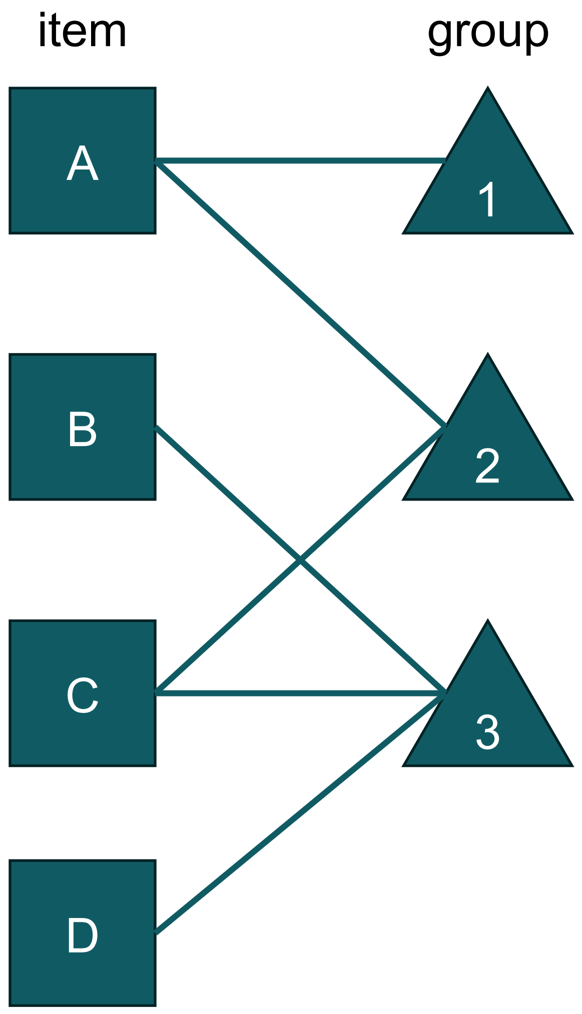
Bipartite networks

- Special class of networks with two types of nodes
- Examples
  - Product recommendation engines
  - Matching people in a ride-share
- Adjacency matrix of bipartite network is called the ***incidence matrix***
  - $B_{i,j} = \begin{cases} 1 & \text{if item } j \text{ belongs to } i \\ 0 & \text{else} \end{cases}$
  - In the example to the right, it would be 3x4



Projecting bipartite networks

- Can create an adjacency matrix for either items or groups (or whatever you call the two types)
- If projecting onto items (A-D), items are connected if they both belong to the same group(s)
- If projecting onto groups (1-3), groups are connected if they contain the same items
- Projections can have multi-edges



Network metric: Degree

- For undirected: Degree (k) of node x is the number of edges connected to node x
- Example:

$$A = \begin{bmatrix} 0 & 1 & 0 & 2 \\ 1 & 0 & 0 & 3 \\ 0 & 0 & 0 & 4 \\ 2 & 3 & 4 & 0 \end{bmatrix}$$

$$k_i = \sum_{j=1}^{n} A_{ij}$$

$$k_1 = 3$$

Network metric: Degree (directed network)

- We define "in-degree" and "out-degree" for each node
- Example:

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

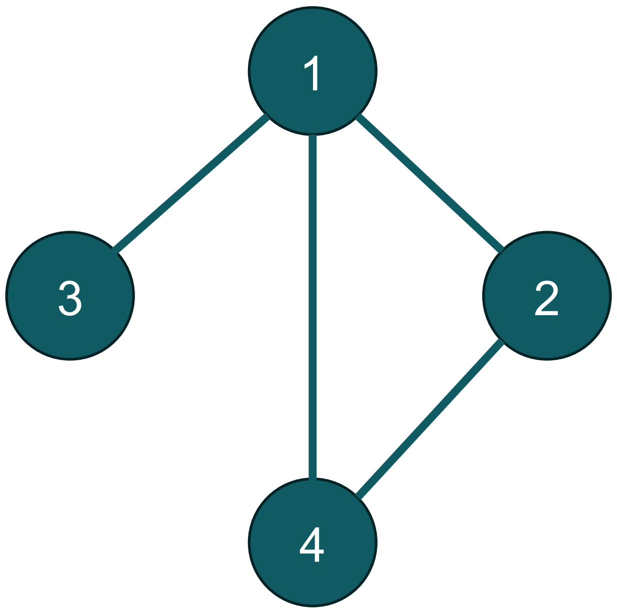
$$k_i^{\text{in}} = \sum_{j=1}^{n} A_{ij}$$

$$k_i^{\text{out}} = \sum_{i=1}^{n} A_{ij}$$

In this case: edges are ***from* j *to* i**

Network paths

- A path is a route that traverses a network (can be directed or undirected)
- How to calculate number of paths of a given length?

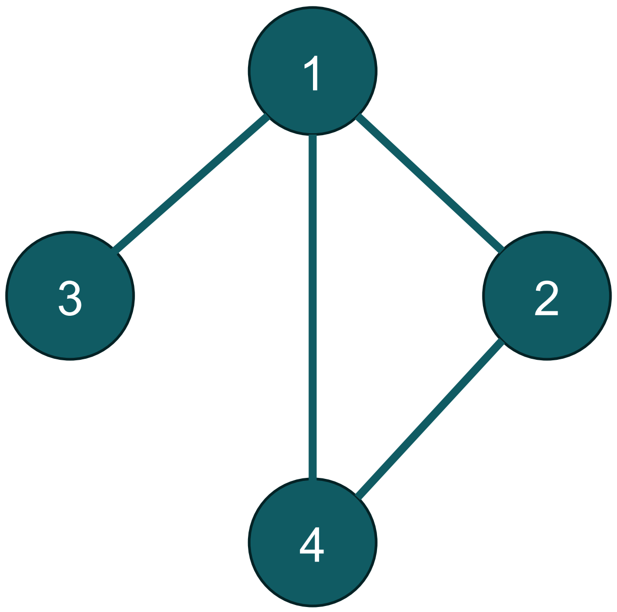


Network paths

- A path is a route that traverses a network (can be directed or undirected)
- How to calculate number of paths of a given length?
- For path from j → i of length 2:

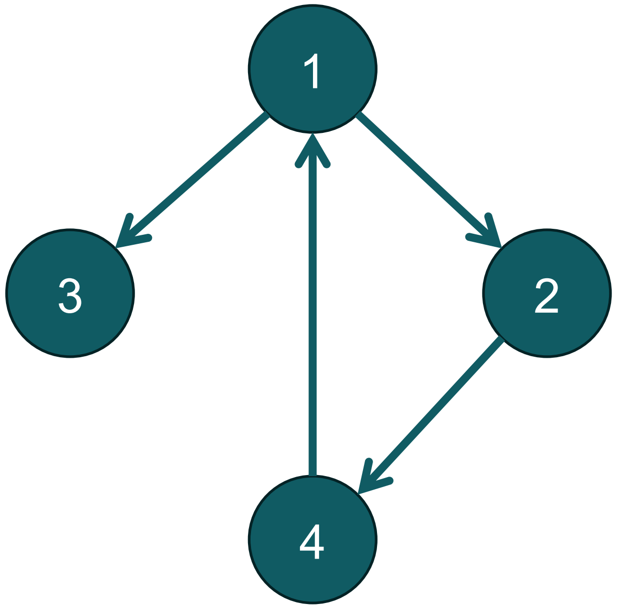
$$N_{ij}^{(2)} = \sum_{k=1}^{n} A_{ik} A_{kj} = \left[A^2\right]_{ij}$$

For any path length r: $N_{ij}^{(r)} = [A^r]_{ij}$



Network path example (directed)

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

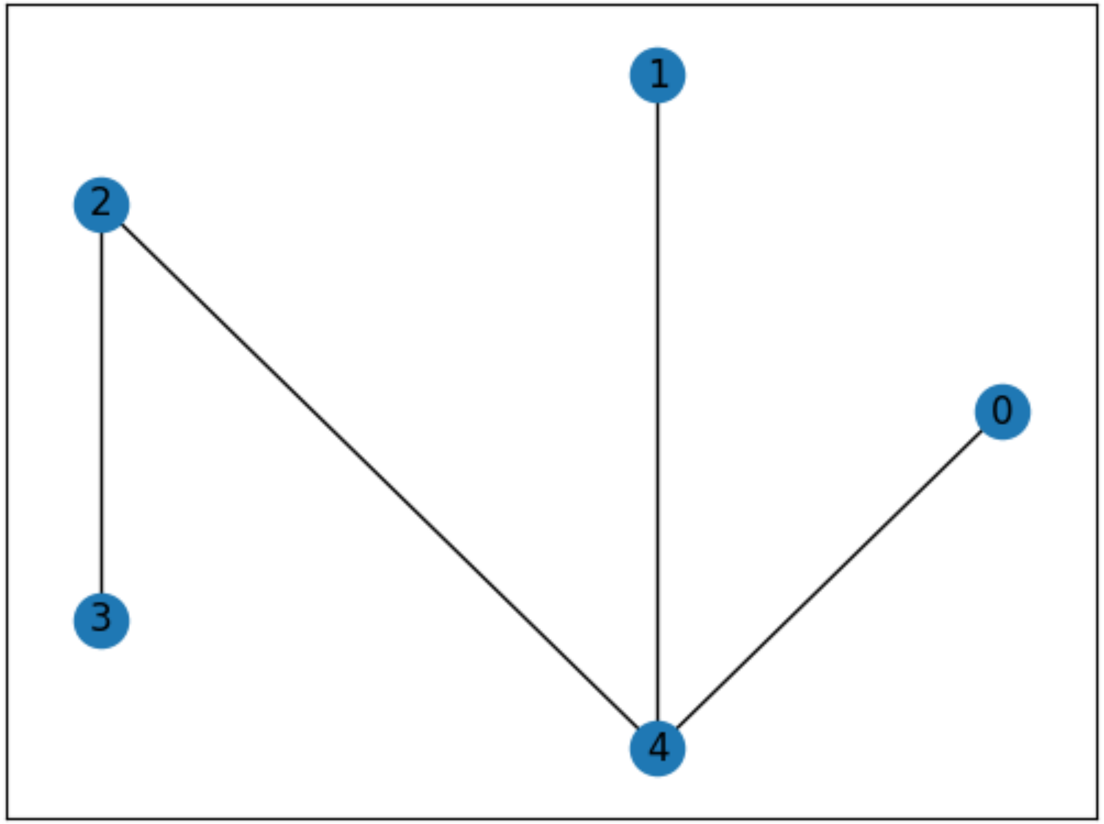


Diagonal of matrix: self-loops of given length
Sum of diagonals: Tr(A)

How Tr(A) change if it were undirected? Check in Python!

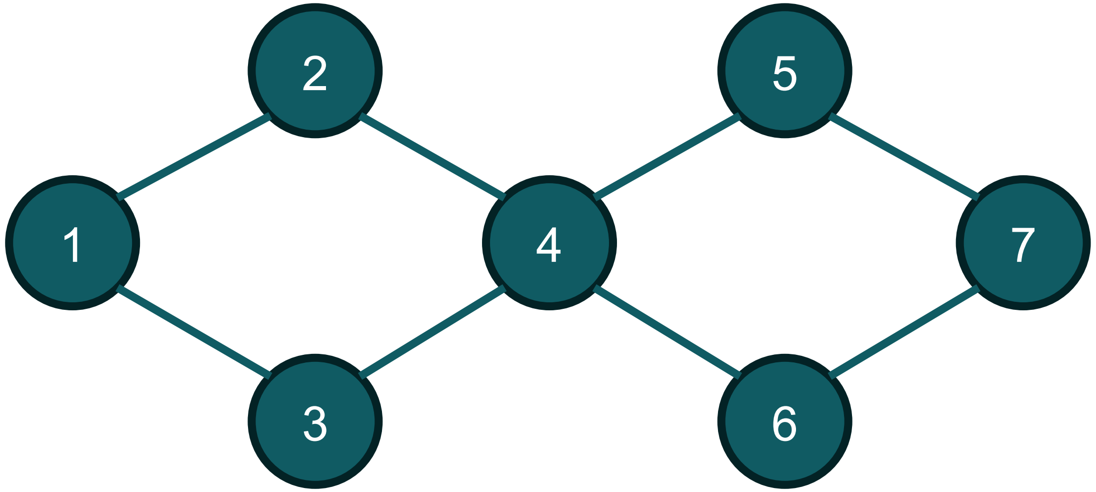
Application: Electric vehicles

- Suppose you have a simplified electric vehicle charging network and **you can add one link** connecting any two charging stations
- You decide to choose new edges that **maximize the total number of paths** to/from your charging stations
- Which link do you add if you want to **increase the total number of paths of length 2**?



Independent paths

- Edge independent: paths that do not share edges
- Node independent: paths that do not share nodes (except start/end)



How many edge-independent and node-independent paths from 1→7?

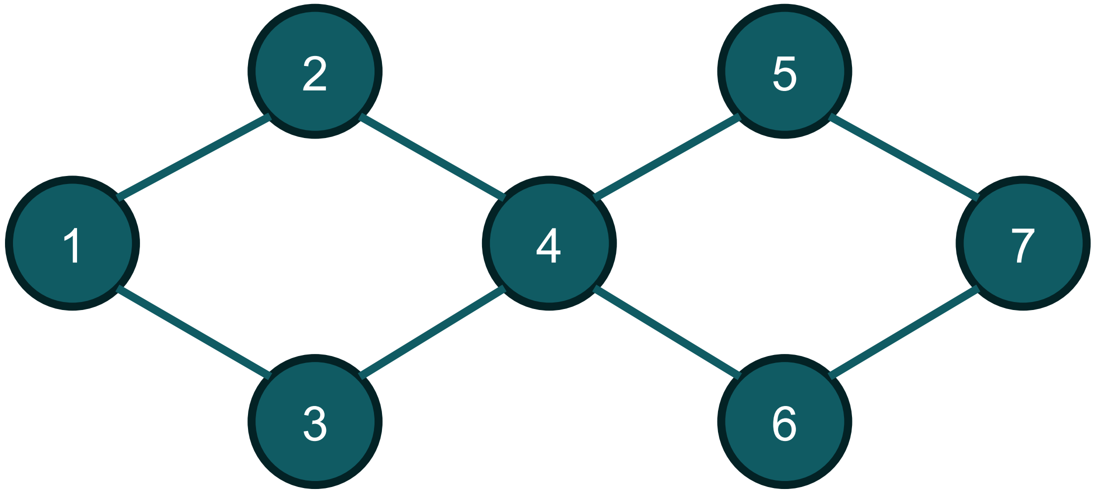
Independent paths

- Edge independent: paths that do not share edges
- Node independent: paths that do not share nodes (except start/end)

- The number of independent paths is also called the ***connectivity*** (edge-connectivity and node-connectivity)

Cut sets

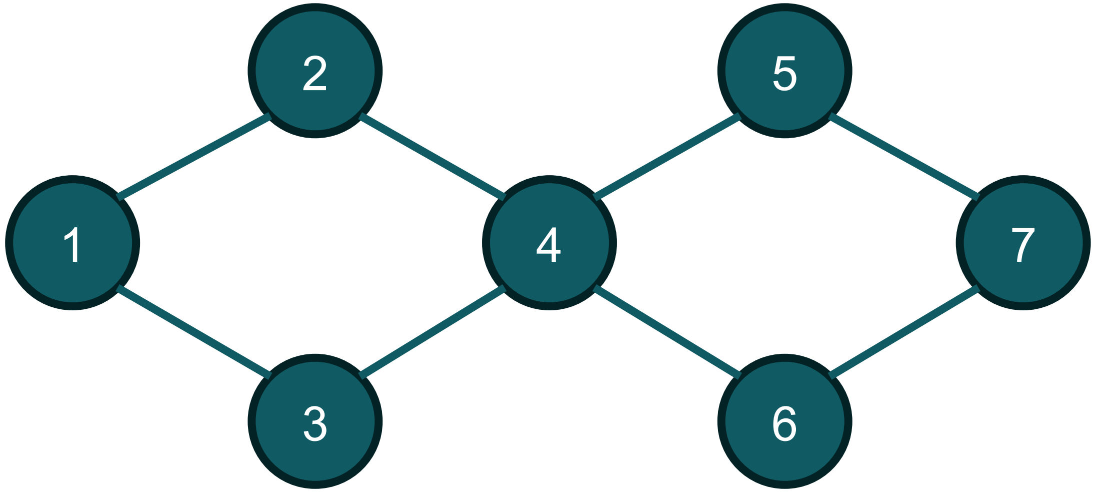
- **Node cut set**: set of nodes whose removal will disconnect a specified pair of nodes
- **Edge cut set**: set of edges whose removal will disconnect a specified pair of nodes
- **Min cut set** (node or edge): smallest cut set that will disconnect a specified pair of nodes



Min node cut set for this network? Min edge cut set?

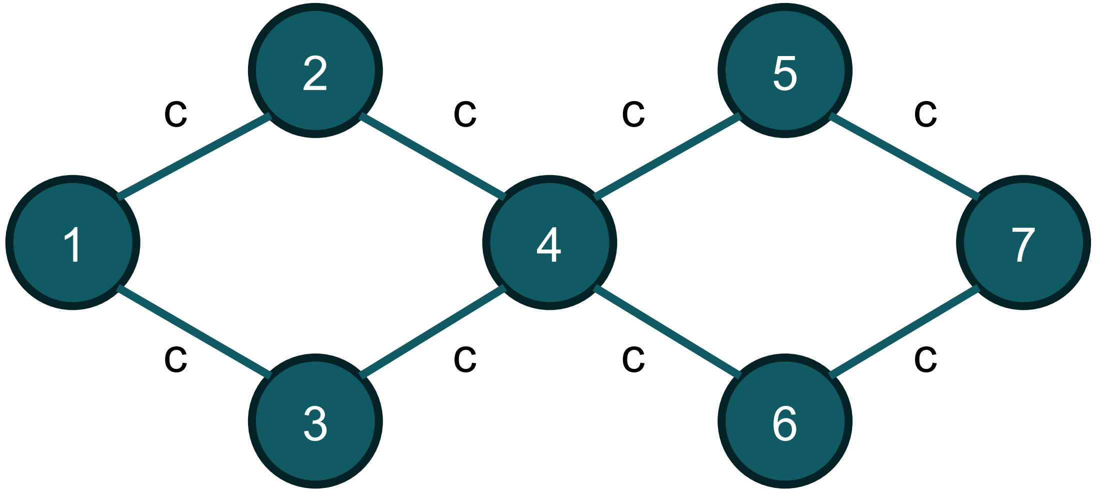
Menger's theorem

- The size of the minimum cut set is equal to the number of independent paths
- This holds for both nodes and edges



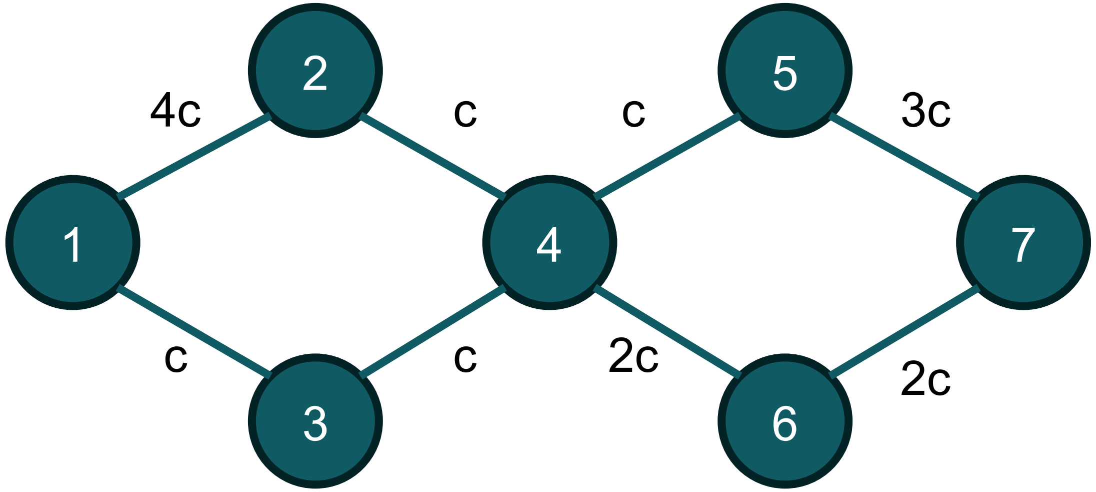
Edges: Minimum cuts and maximum flows

- The maximum flow between two nodes is equal to the size of the minimum edge cut set times capacity



Max-flow/min-cut for weighted networks

- Where is the minimum cut here?



Summary of path discussion

- For edges, we have three equal values:
  - Number of edge independent paths (edge connectivity)
  - Min edge cut set
  - Maximum flow through edges

Application example

- Suppose we have a water network as shown to the right. Solid lines are existing pipes and their capacity. Dashes lines show possible new pipes and their capacity (cost ~ capacity). You need to add new pipes to the network and ensure that flow from 1→7 has:
  - 2 node independent paths and 2 edge independent paths
  - Max-flow ≥ 2r
- What is the minimum cost solution?