

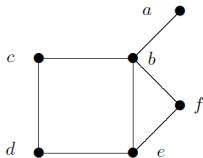
A complex network diagram with numerous nodes and edges. Nodes are represented by circles of various sizes and colors, including grey, yellow, green, blue, orange, and pink. Some nodes are highlighted with larger, semi-transparent circles of the same color. The edges are thin grey lines connecting the nodes, forming a dense web of connections. The diagram is centered on a white background.

Lecture 4 · Graph Theory III

Networks, Crowds and Markets

Warm Up

Given the following graph:



Determine:

- a) the average degree of the graph.
- b) the degree distribution.
- c) its adjacency matrix.

Average Degree

In a simple graph, the vector $A\mathbf{1}$ gives the degrees. Thus $\frac{1}{N}\mathbf{1}^\top A\mathbf{1} = \frac{1}{N}\sum_{i,j=1}^N A_{ij}$ gives the **average degree**. How about other cases?

a. Undirected

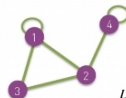


$$A_{ij} = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

$$A_{ii} = 0 \quad A_{ij} = A_{ji}$$

$$L = \frac{1}{2} \sum_{i,j=1}^N A_{ij} \quad \langle k \rangle = \frac{2L}{N}$$

b. Self-loops

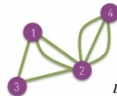


$$A_{ij} = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix}$$

$$\exists i, A_{ii} \neq 0 \quad A_{ij} = A_{ji}$$

$$L = \frac{1}{2} \sum_{i,j=1, i \neq j}^N A_{ij} + \sum_{i=1}^N A_{ii} \quad ?$$

c. Multigraph
(undirected)

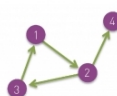


$$A_{ij} = \begin{pmatrix} 0 & 2 & 1 & 0 \\ 2 & 0 & 1 & 3 \\ 1 & 1 & 0 & 0 \\ 0 & 3 & 0 & 0 \end{pmatrix}$$

$$A_{ii} = 0 \quad A_{ij} \neq A_{ji}$$

$$L = \frac{1}{2} \sum_{i,j=1}^N A_{ij} \quad \langle k \rangle = \frac{2L}{N}$$

d. Directed



$$A_{ij} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

$$A_{ij} \neq A_{ji}$$

$$L = \sum_{i,j=1}^N A_{ij} \quad \langle k \rangle = \frac{L}{N}$$

Barabási denotes the average degree by $\langle k \rangle$. We do not follow this convention.

Today's Lecture

1. Distances in a graph
 - 1.1 Path / Shortest path / Distance
 - 1.2 Breadth First Search
 - 1.3 Diameter / Local Diameter
 - 1.4 Eccentricity
2. Connectivity
 - 2.1 Definition
 - 2.2 Bridge
3. Trees, Regular Graphs
4. Introduction to centrality measures
 - ▶ Degree centrality
 - ▶ Closeness centrality

Distances in graphs

Path

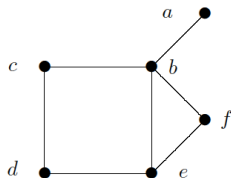
Definition

A **walk** in an undirected graph $G = (V, E)$ is a sequence of vertices (v_0, \dots, v_ℓ) s.t. each consecutive pair $v_{i-1}v_i$ is an edge in E .

A **path** in an undirected graph $G = (V, E)$ is a sequence of **distinct** vertices (v_0, \dots, v_ℓ) s.t. each consecutive pair $v_{i-1}v_i$ is an edge in E .

Length: the number of edges in the sequence, here ℓ .

In a weighted graph we could take the sum of the weights of the edges on the path; **path weight**.

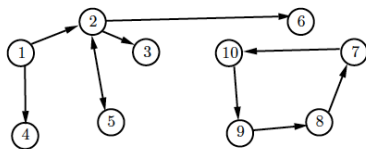


Examples: $p_1 = abfe$, $p_2 = cdef$,
 $p_3 = defba$

Directed Path

Definition

A **directed path** in a directed graph G is a finite sequence of edges $e_i = (x_i, y_i)$ such that for each $i \geq 1$, $x_{i+1} = y_i$.



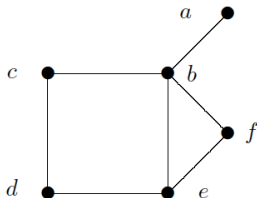
Examples:

$$p_1 = \overline{126}, p_2 = \overline{523}$$

Shortest Path

Definition

A **shortest path** in a graph G between two nodes u, v is any path that connects u with v having minimum length.



abe is the shortest path between a and e

$abcd$ and $abed$ are both shortest path between a and d

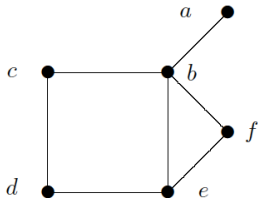
In NetworkX:

```
path = nx.shortest_path(G, source=1, target=3)
print("Shortest path:", path)
length = nx.shortest_path_length(G, source=1, target=3)
print("Path length:", length)
```


Distance

Definition

The **distance** $d_G(u, v)$ in a graph G between two nodes u, v is the length of any shortest path connecting u with v or **infinite** if there is no such path.



$$d(a, e) = 2$$

Properties of the graph distance

Let $G = (V, E)$ be a graph. The distance $d(u, v)$ between nodes u, v has the following properties:

- $d(u, v) \geq 0$, and $d(u, v) = 0 \iff u = v$.
- $d(u, v) = d(v, u)$ (symmetry).
- $d(u, v) \leq d(u, w) + d(w, v)$ (triangle inequality).
- If there is no path between u and v , then $d(u, v) = \infty$.

Connectedness: G is **connected** if $d(u, v) < \infty$ for every pair $u, v \in V$.

Extension: In weighted graphs with nonnegative edge weights, $d(u, v)$ is the minimum total weight of any path between u and v .

Diameter of a graph

Diameter and the Moore bound

Definition

The **diameter** $\text{diam}(G)$ of a graph $G = (V, E)$ is the maximum distance between two vertices:

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

Diameter and the Moore bound

Definition

The **diameter** $\text{diam}(G)$ of a graph $G = (V, E)$ is the maximum distance between two vertices:

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

Theorem (Moore Bound)

$$N \leq 1 + \Delta \cdot \sum_{k=0}^{D-1} (\Delta - 1)^k.$$

Diameter and the Moore bound

Definition

The **diameter** $\text{diam}(G)$ of a graph $G = (V, E)$ is the maximum distance between two vertices:

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

Theorem (Moore Bound)

$$N \leq 1 + \Delta \cdot \sum_{k=0}^{D-1} (\Delta - 1)^k.$$

Why is this useful?

It shows that, if degrees are bounded, you cannot have both a very small diameter and a very large N .

Small-world networks keep diameters small using **hubs** or **shortcuts**.

Proof: Moore Bound

Fix a vertex v . Since the diameter is D , every vertex lies within distance D of v . To get the maximal N given the constraints:

- Distance 0: only v .
- Distance 1: at most Δ neighbors of v .
- Distance 2: each neighbor adds at most $\Delta - 1$ new vertices, giving at most $\Delta(\Delta - 1)$.
- Distance i : at most $\Delta(\Delta - 1)^{i-1}$ vertices.

Summing up to distance D ,

$$N \leq 1 + \Delta \sum_{i=0}^{D-1} (\Delta - 1)^i.$$

When $\Delta = 2$, the graph is a path or cycle, giving $N \leq 2D + 1$. □

Breadth-First Search (BFS)

Definition

The **Breadth-First Search** explores a graph starting from a root node v :

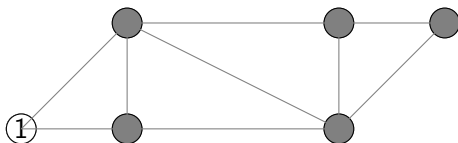
- Discover all neighbors of v first.
- Then, in order, discover neighbors of those neighbors, and so on.

Breadth-First Search (BFS)

Definition

The **Breadth-First Search** explores a graph starting from a root node v :

- Discover all neighbors of v first.
- Then, in order, discover neighbors of those neighbors, and so on.

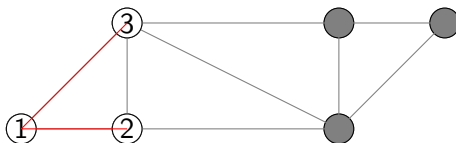


Breadth-First Search (BFS)

Definition

The **Breadth-First Search** explores a graph starting from a root node v :

- Discover all neighbors of v first.
- Then, in order, discover neighbors of those neighbors, and so on.

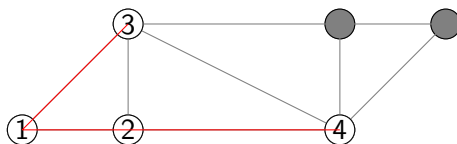


Breadth-First Search (BFS)

Definition

The **Breadth-First Search** explores a graph starting from a root node v :

- Discover all neighbors of v first.
- Then, in order, discover neighbors of those neighbors, and so on.

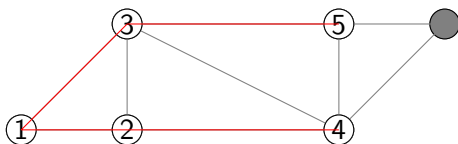


Breadth-First Search (BFS)

Definition

The **Breadth-First Search** explores a graph starting from a root node v :

- Discover all neighbors of v first.
- Then, in order, discover neighbors of those neighbors, and so on.

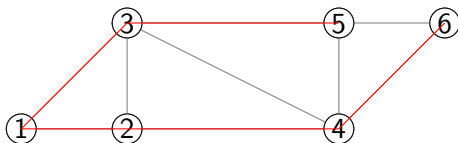


Breadth-First Search (BFS)

Definition

The **Breadth-First Search** explores a graph starting from a root node v :

- Discover all neighbors of v first.
- Then, in order, discover neighbors of those neighbors, and so on.

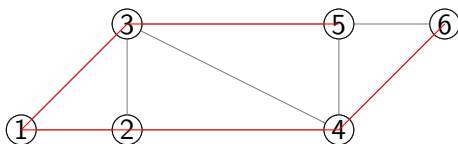


Breadth-First Search (BFS)

Definition

The **Breadth-First Search** explores a graph starting from a root node v :

- Discover all neighbors of v first.
- Then, in order, discover neighbors of those neighbors, and so on.



BFS naturally finds shortest paths from the root to all other vertices.

- First add all nodes at distance 1 from v .
- Then add all nodes at distance 2, etc. . .

The marked path 1–2–4–6 is a shortest path from 1 to 6 (length 3).

BFS Bookkeeping for Shortest Paths

Goal (unweighted graphs, source s): compute

- $d[v]$ = distance from s to v (in edges),
- $\sigma[v]$ = number of shortest $s \rightarrow v$ paths,
- $\text{Pred}[v]$ = predecessors of v on shortest $s \rightarrow v$ paths.

Initialization:

- For all v : $d[v] = \infty$, $\sigma[v] = 0$, $\text{Pred}[v] = \emptyset$.
- Set $d[s] = 0$, $\sigma[s] = 1$, push s in a queue Q .

BFS loop (standard queue):

- While Q not empty:
 - ▶ Pop v from Q .
 - ▶ For each neighbor w of v :
 - ▶ If $d[w] = \infty$ then
 $d[w] = d[v] + 1$; $\sigma[w] = \sigma[v]$; $\text{Pred}[w] = \{v\}$; push w .
 - ▶ Else if $d[w] = d[v] + 1$ then
 $\sigma[w] \leftarrow \sigma[w] + \sigma[v]$; add v to $\text{Pred}[w]$.

Graph Eccentricity

Definition

The **eccentricity of a vertex** u is: $\varepsilon(u) = \max_{v \in V} d(u, v)$.

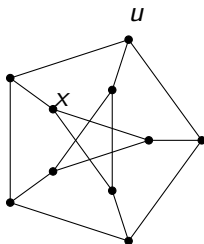
The **eccentricity of a graph** is: $\text{ecc}(G) = \min_{u \in V} \varepsilon(u)$.

(every minimizer is in some sense central)

- Captures the distance from the “most central” vertex to the farthest node.
- **Applications:**
 - ▶ In communication networks: optimal placement of a hub or server.
 - ▶ In social networks: identifying the most central or influential actors.
 - ▶ In epidemics: best/worst nodes to start monitoring or intervention.

Exercise 1

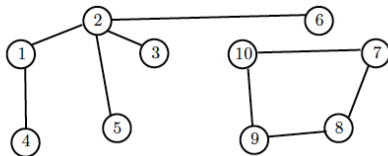
Given the following graph:



- a) Determine the shortest path and length between nodes u and x .
- b) Determine its diameter.
- c) Determine the eccentricity of node u
- d) Which is the minimum number of edges that we have to add to the graph so that its eccentricity is 1?

Connectivity

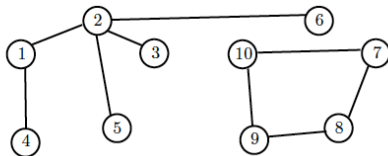
Recall: Connectivity



Definition

Two nodes u, v are **connected** in G if there exists a path connecting both nodes. A graph is **connected** if every two nodes are connected.

Recall: Connectivity



Definition

Two nodes u, v are **connected** in G if there exists a path connecting both nodes. A graph is **connected** if every two nodes are connected.

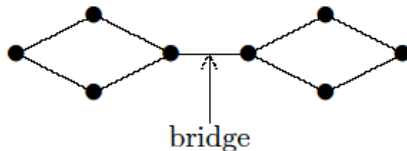
Definition

The **connected components** G are the maximal connected subgraphs.

Bridge

Definition

A **bridge** in a graph G is any edge such that when removing it from the graph, the number of connected components is increased.



(More) special graphs

Special Graphs: Trees

Definition

A tree is a **connected** and **acyclic** graph

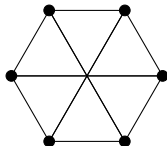
Paths P_n , Stars S_n are trees

- In a tree, every pair of vertices is joined by a unique path.
- The number of edges is one less than the number of vertices.
- Every connected graph contains a tree as a subgraph with the same vertex set (**spanning tree**).

Trees appear everywhere

Internet routing (spanning trees avoid cycles); **Data structures** in computer science (binary search trees, decision trees); **Phylogenetics** in biology (evolutionary trees); **Epidemics** and **rumor spreading** (branching processes); **Hierarchical clustering** (data science).

Special Graphs: Regular Graphs



Definition

A graph is called **r -regular** if every node has exactly degree r .

- Example: a cycle C_N is 2-regular.
- Example: a complete graph K_N is $(N - 1)$ -regular.

For an r -regular graph with N vertices:

$$L = \frac{Nr}{2}.$$

(try to prove it; see also Exercise C below)

Exercise A: Complements

- a) Show that the complement of an r -regular graph on N nodes is $(N - r - 1)$ -regular.
- b) Is the complement of a bipartite graph always bipartite? Give a counterexample if not.
- c) Give an example of a 4-node graph G such that both G and its complement are connected.

Exercise B: Connectivity

- a) Show that if a graph G is not connected, then its complement \overline{G} must be connected.
- b) Consider a tree T (connected, acyclic). Is its complement always connected? Give an example.

Exercise C: Adjacency matrix

Let A be the adjacency matrix of a graph G .

a) Show that G is r -regular if and only if

$$A\mathbf{1}_N = r\mathbf{1}_N,$$

where $\mathbf{1}_N$ is the all-ones vector in \mathbb{R}^N .

b) Suppose

$$I + A + A^2 + A^3 = J,$$

where J is the all-ones matrix. What does this imply about the diameter of G ?

Centrality: Motivation

Why centrality?

Which is the most important node in a network?

- In a social network: the most influential person.
 - ▶ Advertisers buy access to central nodes.
- In trade or finance: the most systemic firm or bank.
 - ▶ Lehman Brothers was “central” in interbank lending.
- In transport: the airport whose closure causes the largest disruption.

Challenge: “importance” is not unique. Different aspects motivate different measures:

- Many neighbours → **Degree centrality**.
- Close to everyone → **Closeness centrality**.
- On many shortest paths → **Betweenness centrality**.

Why centrality? Motivating examples

Different networks \Rightarrow different notions of importance

- **Social media (Twitter/X):** A user with millions of followers is central by *degree*. Another user might have fewer followers but be the main source of breaking news retweets \Rightarrow central by *betweenness*.
- **Transport networks:** Heathrow or Atlanta airports are central because they connect many international routes. Closure disrupts global traffic. \Rightarrow degree & betweenness both matter.
- **Electric power grids:** Centrality can mean carrying the largest electrical load (flow-based), or being geographically close to all others (closeness).
- **Economics and finance:** A highly connected bank in the interbank lending network may be systemically important (degree). A bank that connects otherwise disjoint clusters may trigger contagion (betweenness).

Centrality measures (overview)

- **Degree centrality:** $\deg(v)$.
- **Closeness centrality:** inverse average distance from v to all others.
- **Betweenness centrality:** share of shortest paths that go through v .
- **Eigenvector centrality:** v is important if its neighbours are important.
- **PageRank:** adapted to directed graphs (web, citation networks), weighting incoming links by their sources.

Degree centrality

Degree centrality

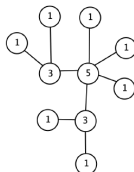
Definition

The degree centrality of a node v is its degree:

$$C_{\text{deg}}(v) = \text{deg}(v).$$

Interpretation:

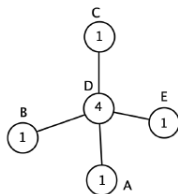
- High degree node can directly influence/reach many others.
- In undirected networks: count of adjacent edges.
- In directed networks: sometimes split into in-degree and out-degree centrality.



Degree via adjacency matrix

Let A_G be the $N \times N$ adjacency matrix of G . For $\mathbf{1} = (1, \dots, 1)^T$,

$$w = A_G \mathbf{1} = \begin{pmatrix} \deg(1) \\ \deg(2) \\ \vdots \\ \deg(N) \end{pmatrix}.$$



Normalised degree centrality

To compare networks of different sizes, normalise by $N - 1$:

$$C'_{\text{deg}}(v) = \frac{\text{deg}(v)}{N - 1} \in [0, 1].$$

Interpretation: fraction of all possible nodes to which v is directly connected.