Machine Learning with Graphs (MLG)

Structure of Graphs

Cheng-Te Li (李政德)

Institute of Data Science

National Cheng Kung University

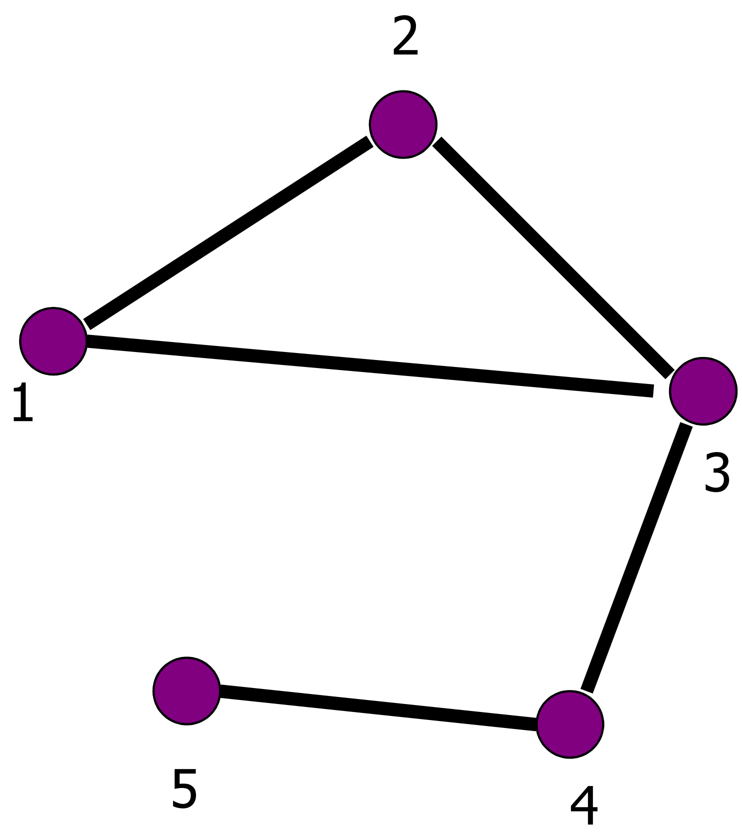
chengte@mail.ncku.edu.tw

Some Tutorials

- **NetworkX**
  - https://github.com/ericmjl/Network-Analysis-Made-Simple [Recommended!!]
  - https://github.com/CambridgeUniversityPress/FirstCourseNetworkScience
- **PyTorch Geometric**
  - A better choice, compared to DGL
  - Official site: https://github.com/rusty1s/pytorch_geometric
  - https://pytorch-geometric.readthedocs.io/en/latest/notes/introduction.html
  - https://www.pytorchtutorial.com/pytorch-geometric-for-gnn/
  - https://towardsdatascience.com/hands-on-graph-neural-networks-with-pytorch-pytorch-geometric-359487e221a8

Undirected Graph

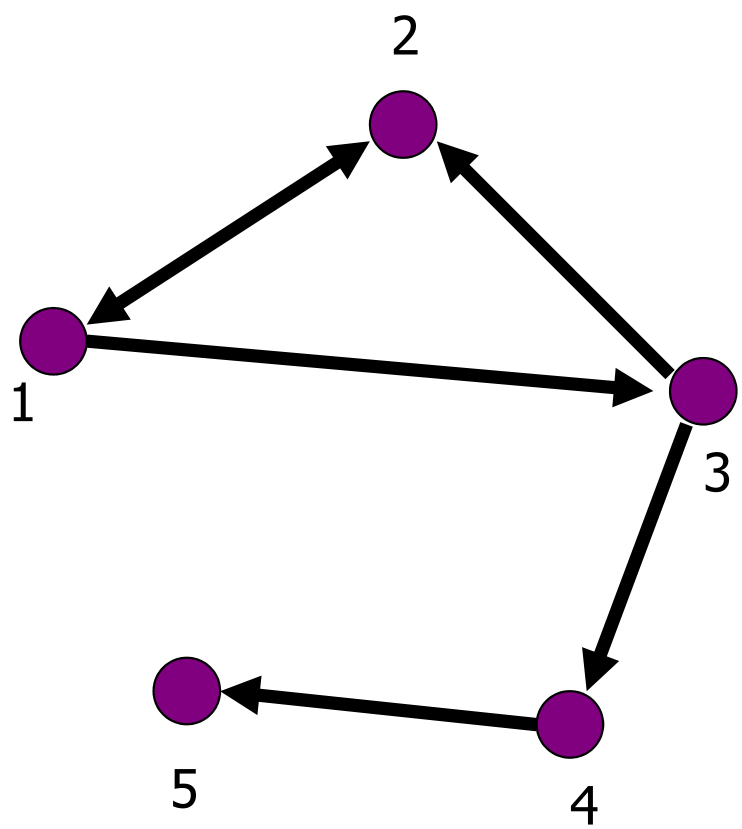
- Graph G=(V,E)
  - V = set of vertices (nodes)
  - E = set of edges



undirected graph
V = {1, 2, 3, 4, 5}
E={(1,2),(1,3),(2,3),(3,4),(4,5)}

Directed Graph

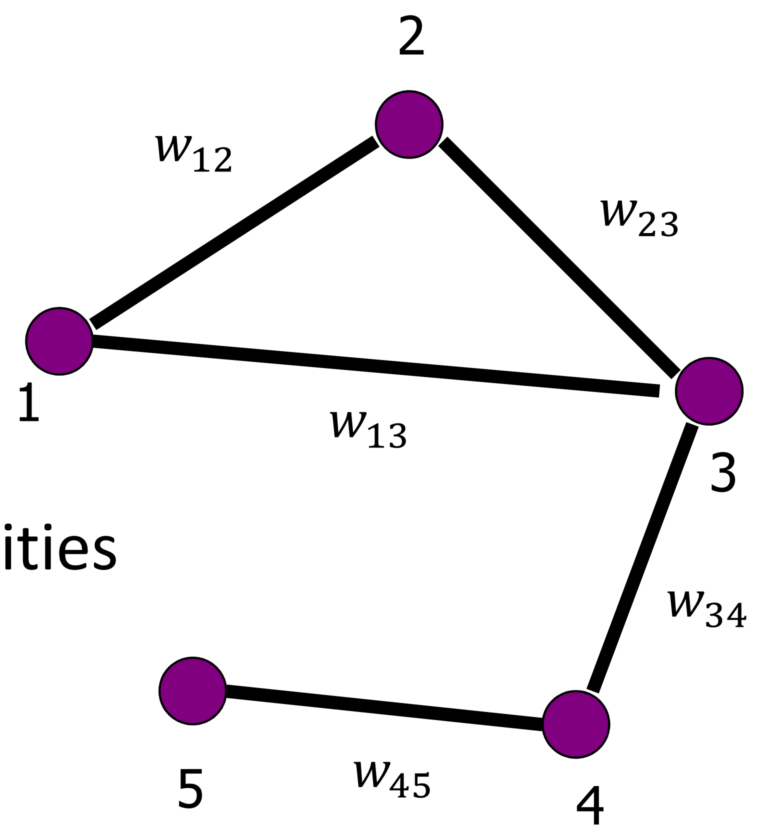
- Graph G=(V,E)
  - V = set of vertices (nodes)
  - E = set of edges



directed graph
V = {1, 2, 3, 4, 5}
E={‹1,2›, ‹2,1› ‹1,3›, ‹3,2›, ‹3,4›, ‹4,5›}

Weighted Graph

- Graph G=(V,E)
  - V = set of vertices (nodes)
  - E = set of edges and their weights

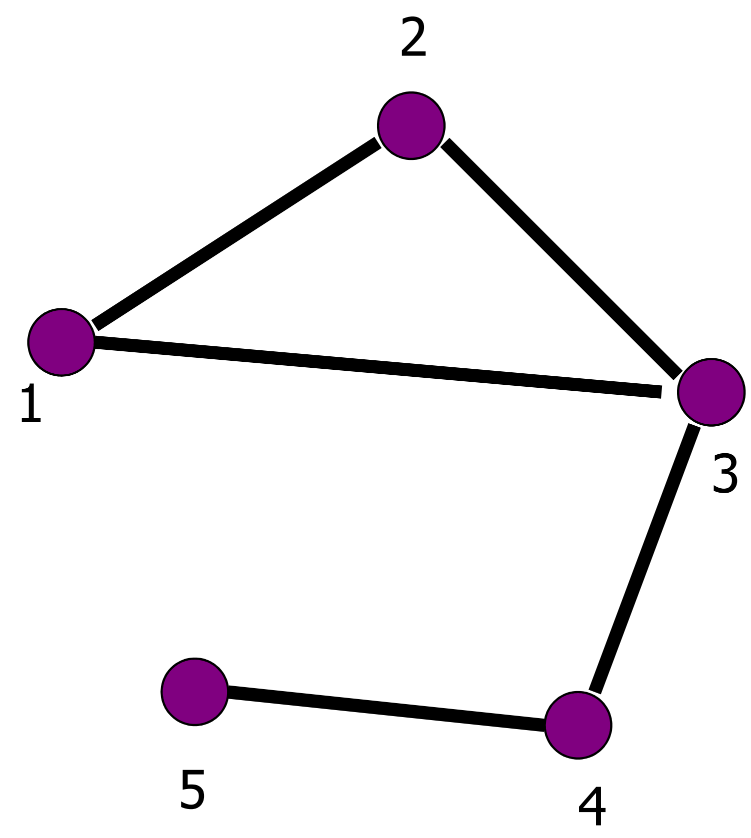


Weights can be either distances or similarities

Weighted graph
V = {1, 2, 3, 4, 5}
E = $\{(1,2,w_{12}),(1,3, w_{12}),(2,3, w_{12}),(3,4, w_{12}),(4,5, w_{12})\}$

Undirected graph

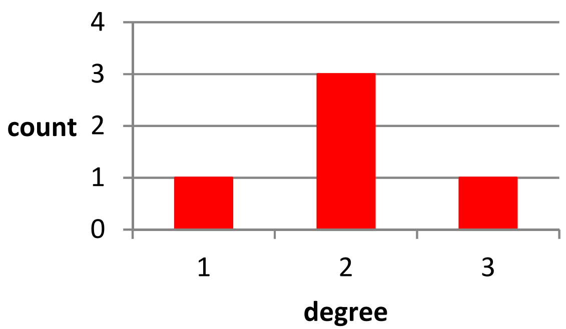
- Neighborhood $N(i)$ of node i
  - For any node i, in an undirected graph, the set of nodes it is connected to via an edge is called its neighborhood and is represented as $N(i)$



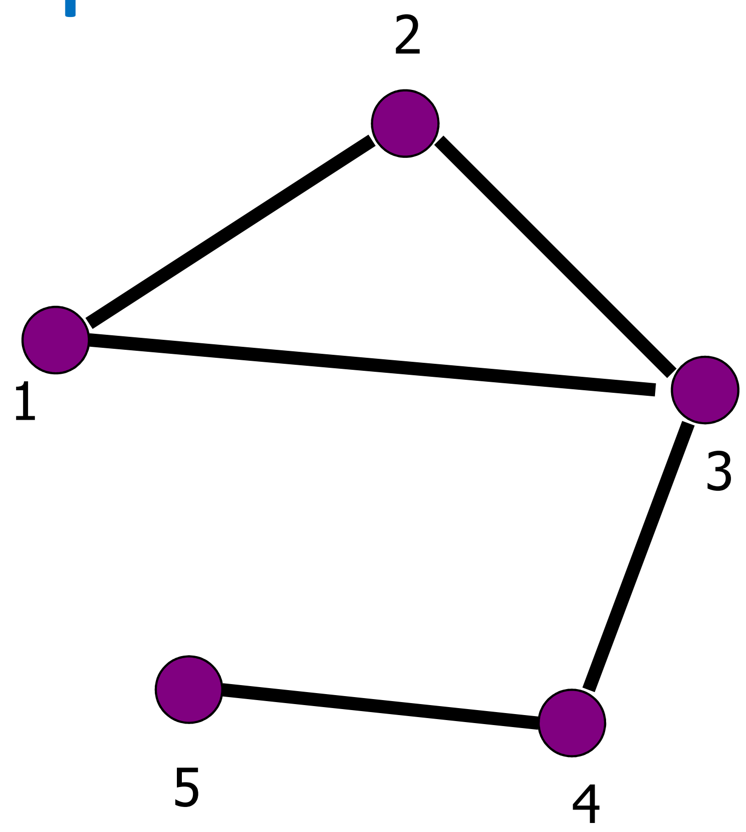
- degree $d(i)$ of node i
  - Size of $N(i)$
  - number of edges incident on i

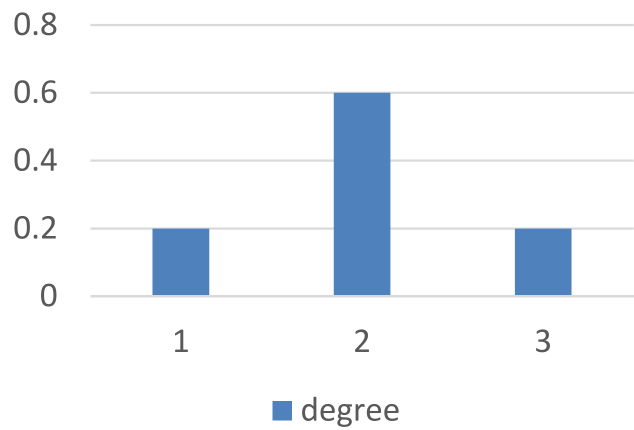
Undirected graph

- degree sequence
  - [d(1),d(2),d(3),d(4),d(5)]
  - [2,2,3,2,1]
- degree histogram
  - [(1:1),(2:3),(3,1)]



- degree distribution
  - [(1:0.2),(2:0.6),(3,0.2)]

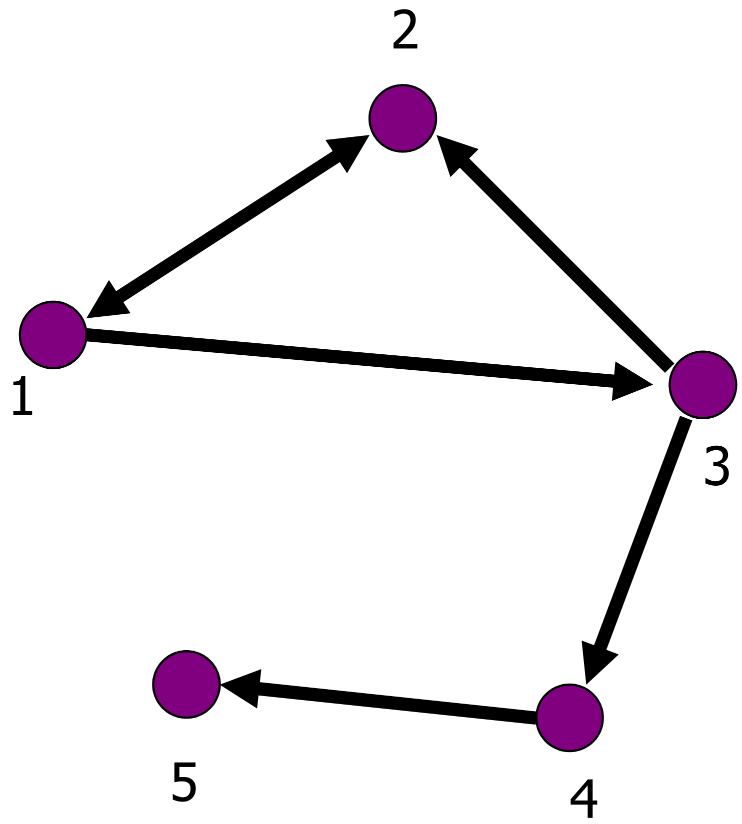




Directed Graph

In directed graphs we have incoming neighbors $N_{in}(v)$ (nodes that connect to v) and outgoing neighbors $N_{out}(v)$

- in-degree $d_{in}(i)$ of node i
  - number of edges incoming to node i
- out-degree $d_{out}(i)$ of node i
  - number of edges leaving node i



- in-degree sequence
  - [1,2,1,1,1]
- out-degree sequence
  - [2,1,2,1,0]
- in-degree histogram
  - [(1:4),(2:1)]
- out-degree histogram
  - [(0:1),(1:2),(2:2)]

Degree Distribution

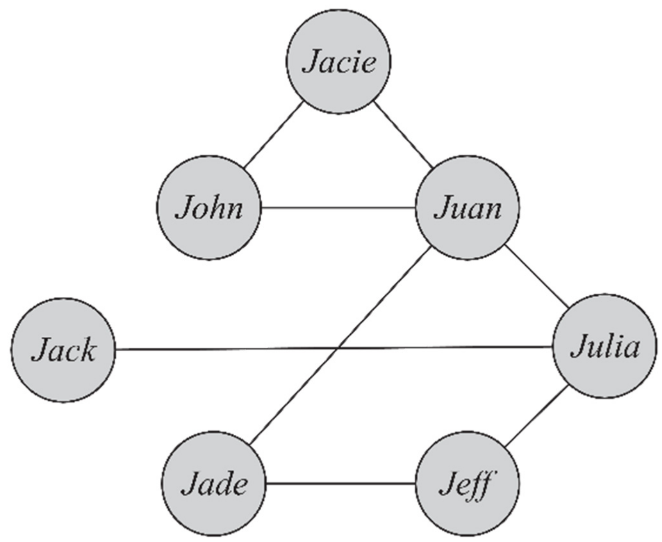
- When dealing with very large graphs, how nodes' degrees are distributed is an important concept to analyze and is called Degree Distribution

$$\pi(d) = \{d_1, d_2, \ldots, d_n\}$$

$$p_d = \frac{n_d}{n}$$

n_d is the number of nodes with degree d

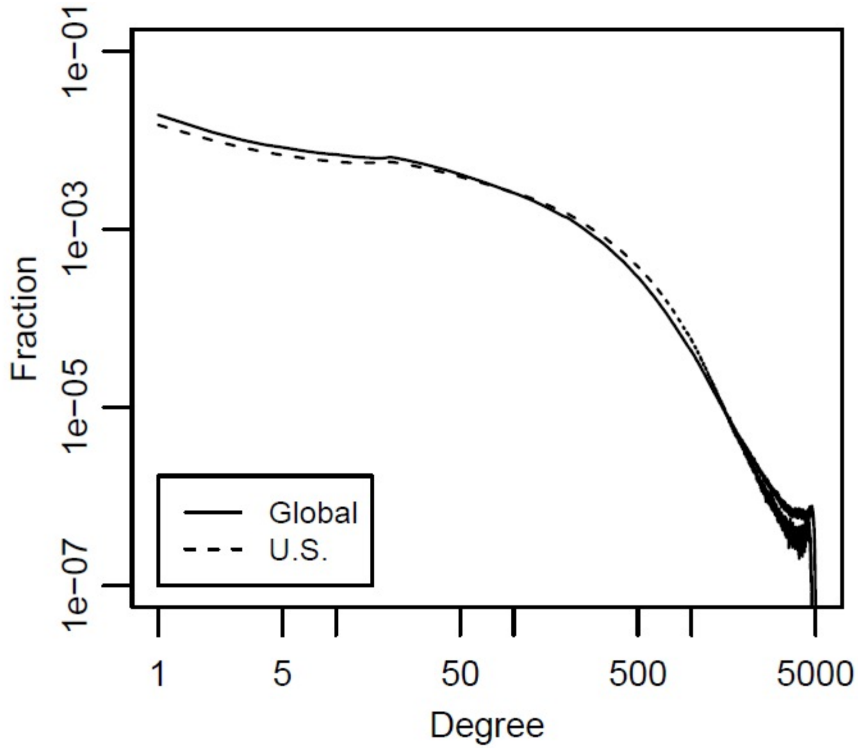
$$\sum_{d=0}^{\infty} p_d = 1$$



$$p_1 = \frac{1}{7}, p_2 = \frac{4}{7}, p_3 = \frac{1}{7}, p_4 = \frac{1}{7}$$

Degree Distribution Plot

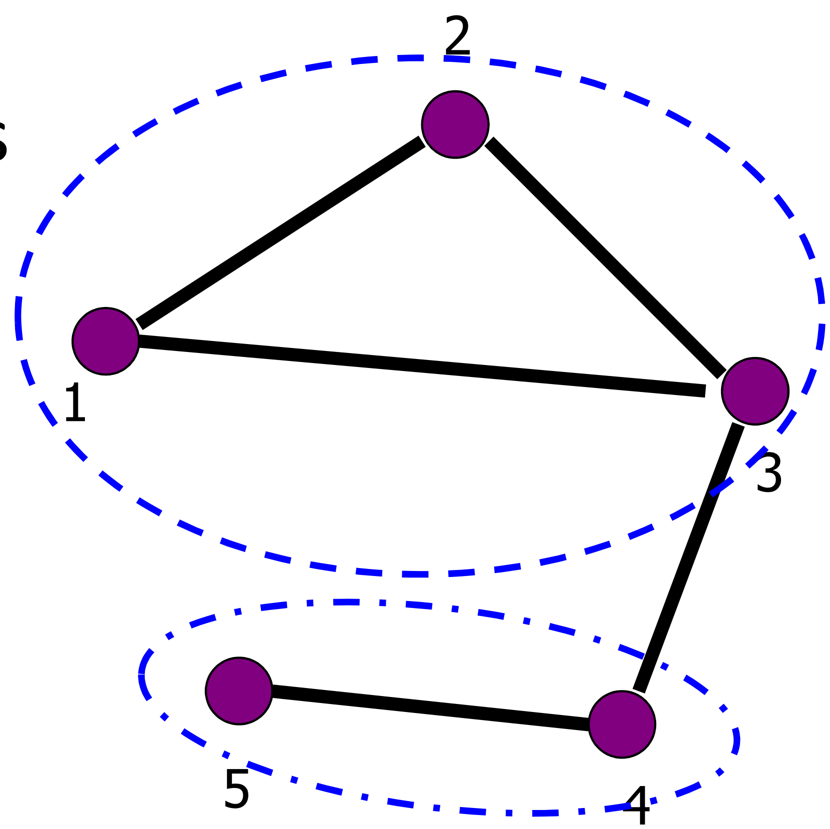
- The x-axis represents the degree and the y-axis represents the fraction of nodes having that degree
- On social networking sites
  - Many users with few connections
  - A handful of users with very large numbers of friends
- Power-Law Degree Distribution



Facebook Degree Distribution

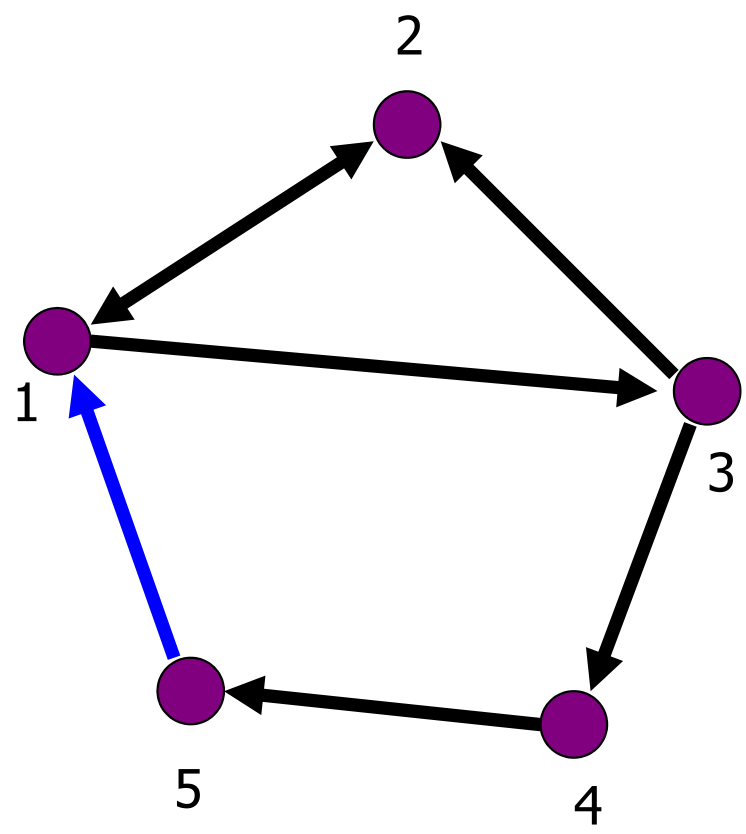
Connectivity for Undirected graph

- Connected graph: a graph where there every pair of nodes is connected through a path
- Disconnected graph: a graph that is not connected
- Connected Components: subsets of vertices that are connected



Connectivity for Directed Graph

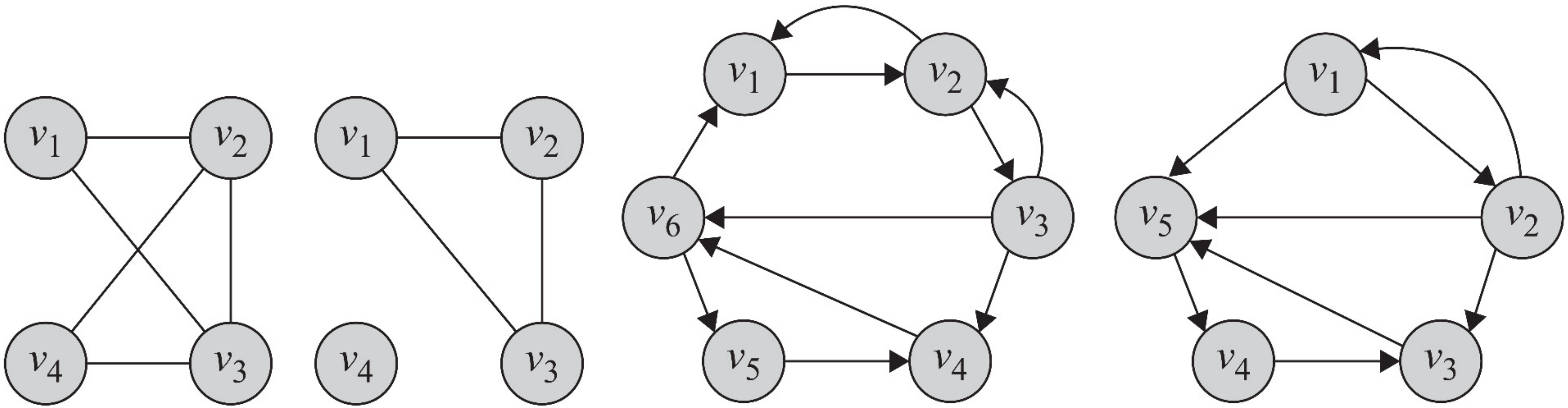
- Strongly connected graph: there exists a path from every i to every j
- Weakly connected graph: If its underlying undirected graph is connected



Connectivity

- A node v_i is connected to node v_j (or reachable from v_j) if it is adjacent to it or there exists a path from v_i to v_j
- A graph is connected, if there exists a path between any pair of nodes in it
  - In a directed graph, **a graph is strongly connected** if there exists a **directed path** between any pair of nodes
  - In a directed graph, **a graph is weakly connected** if there exists a **path** between any pair of nodes, without following the edge directions
- A graph is **disconnected,** if nodes are not connected

Connectivity

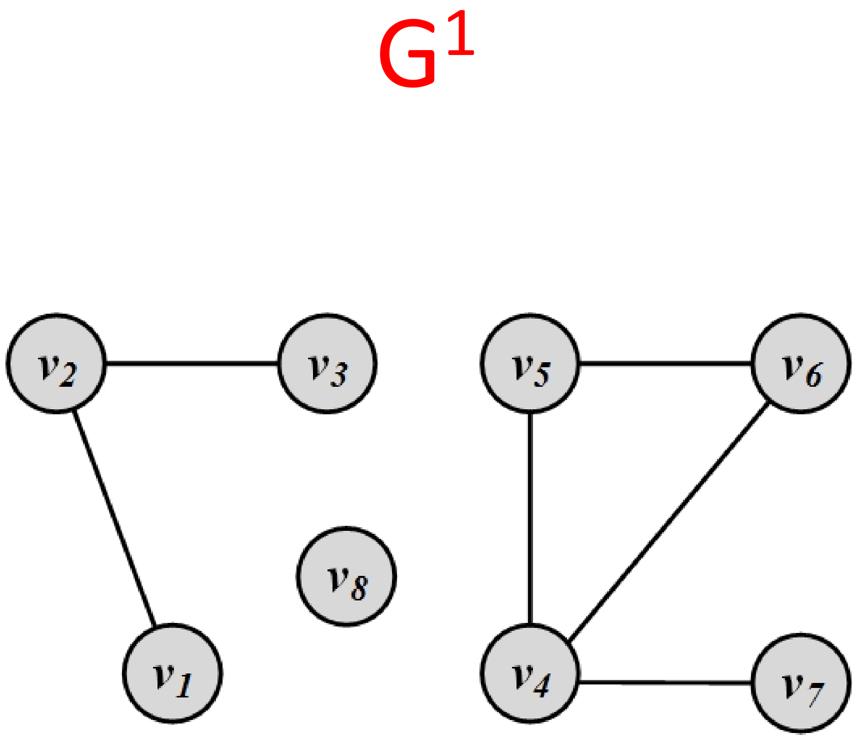


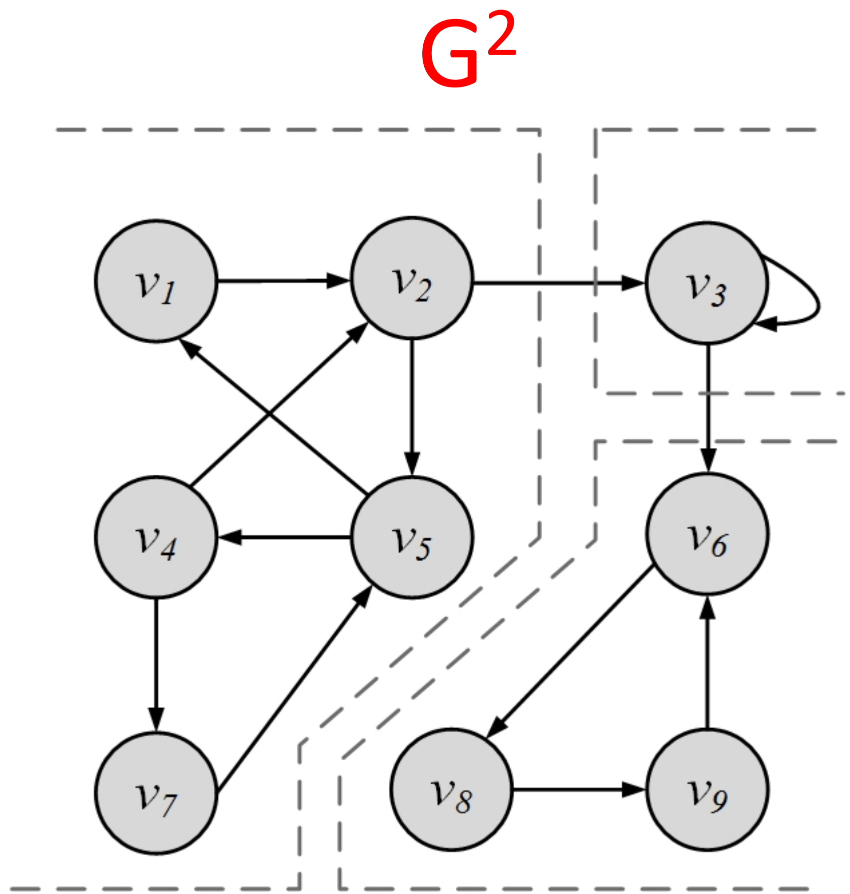
(a) Connected (b) Disconnected (c) Strongly connected (d) Weakly connected

Components

- A component in an undirected graph is a connected subgraph, i.e., there is a path between every pair of nodes inside the component
- In directed graphs, we have a strongly connected component when there is a path from u to v and one from v to u for every pair of nodes u and v
- The component is weakly connected if replacing directed edges with undirected edges results in a connected component

Components



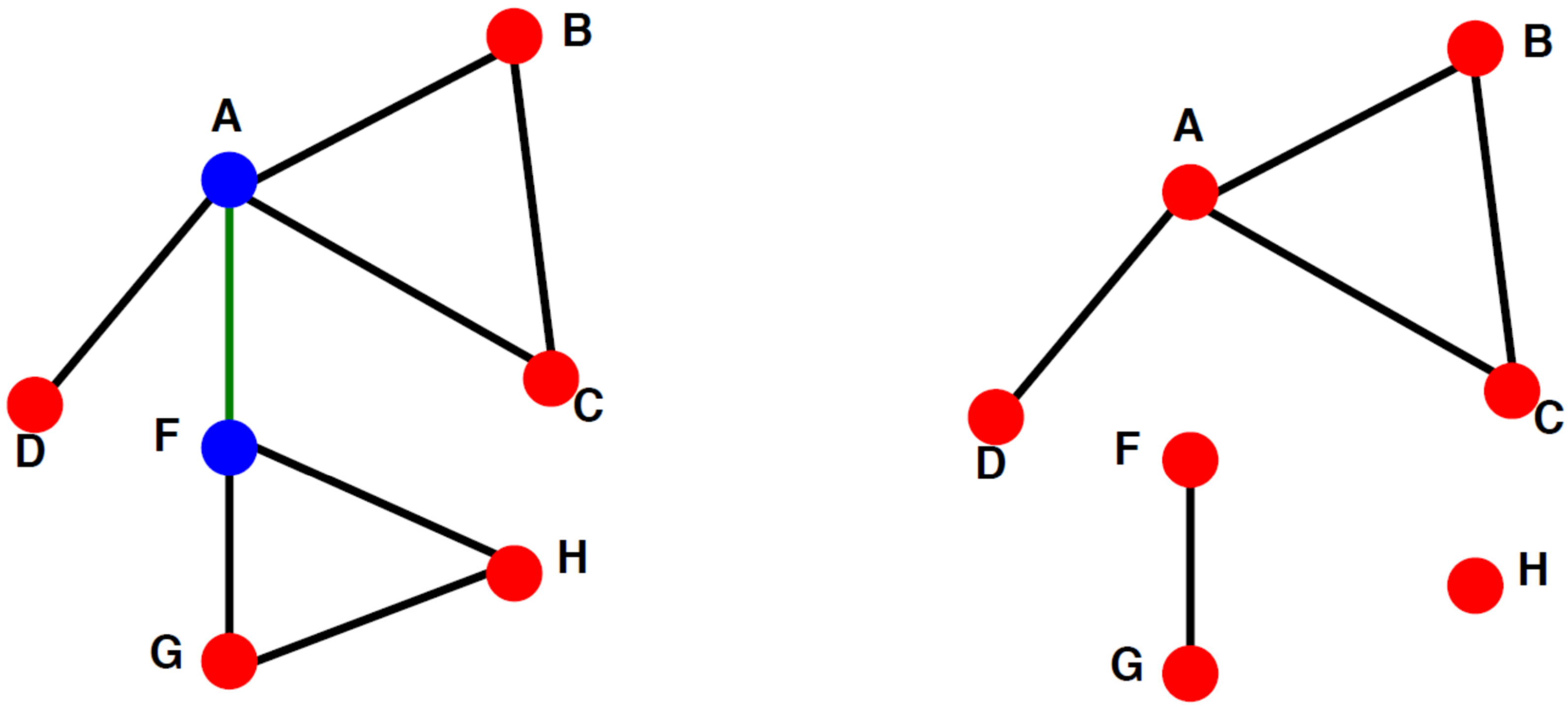


3 components

3 Strongly-connected components

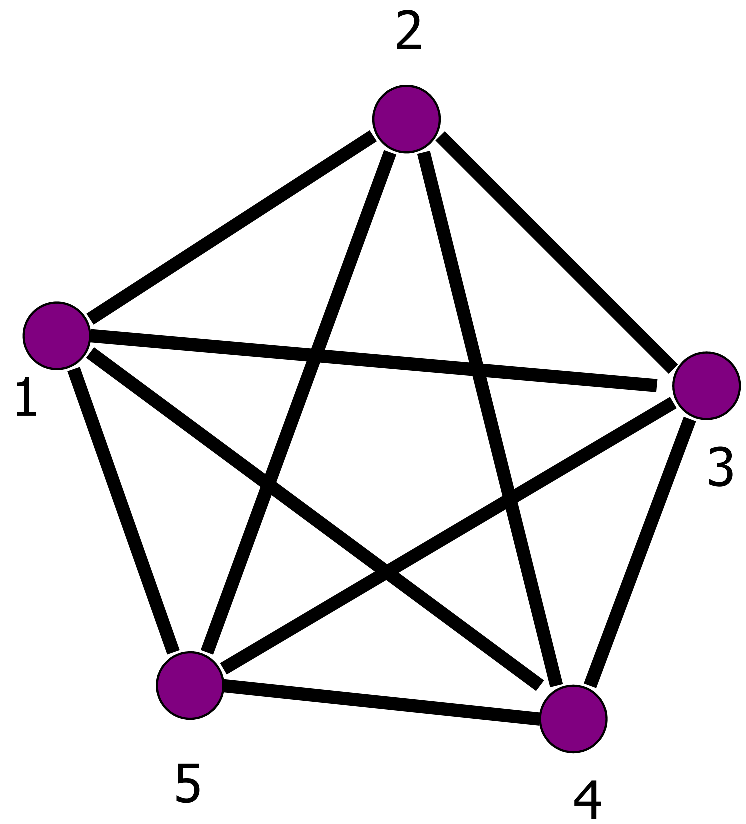
Articulation and Bridge

- Bridge edge: If we erase the edge, the graph becomes disconnected
- Articulation node: If we erase the node, the graph becomes disconnected



Fully Connected Graph

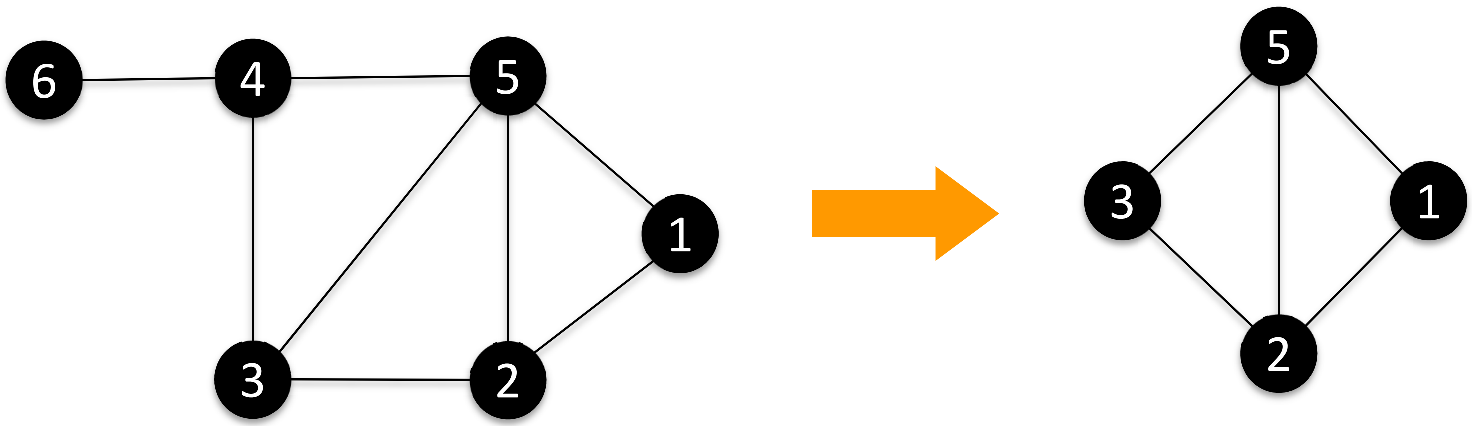
- Clique K_n
- A graph that contains all possible n(n-1)/2 edges



Subgraph

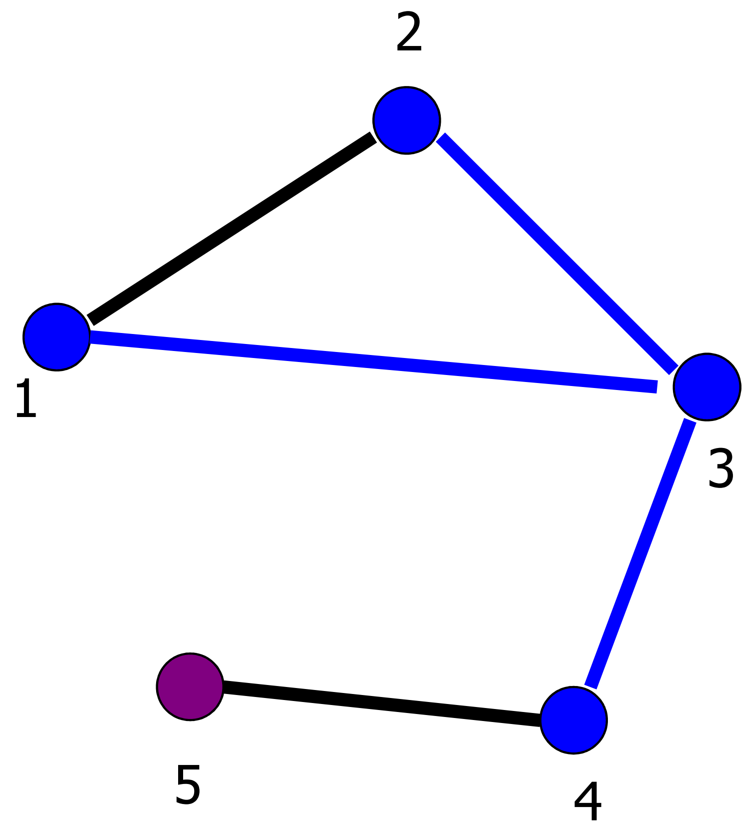
- Graph G can be represented as a pair $G(V, E)$
  - where V is the node set and E is the edge set
- $G'(V', E')$ is a subgraph of $G(V, E)$

$$V' \subseteq V$$
$$E' \subseteq E$$



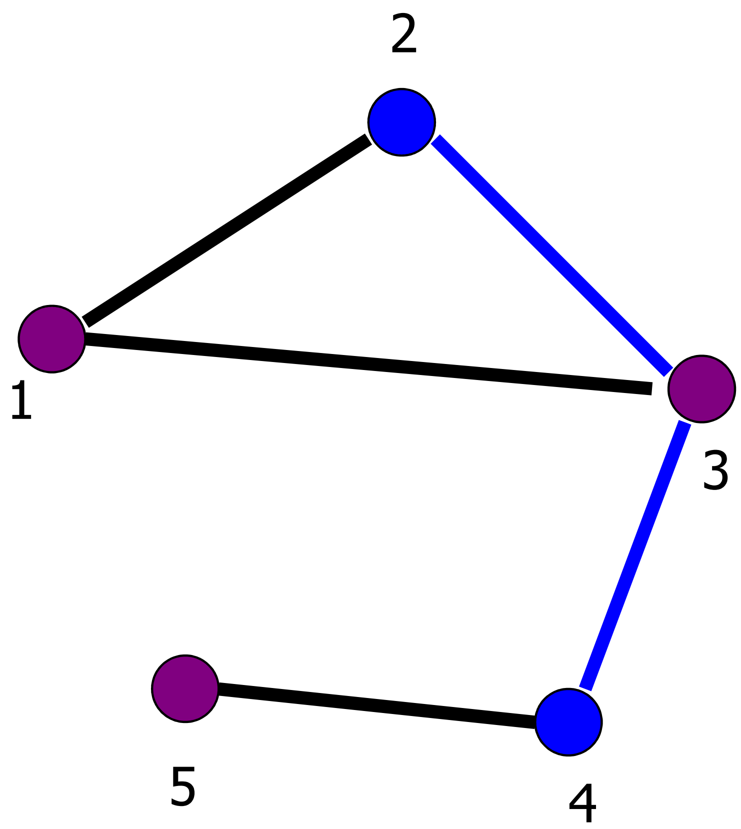
Subgraphs

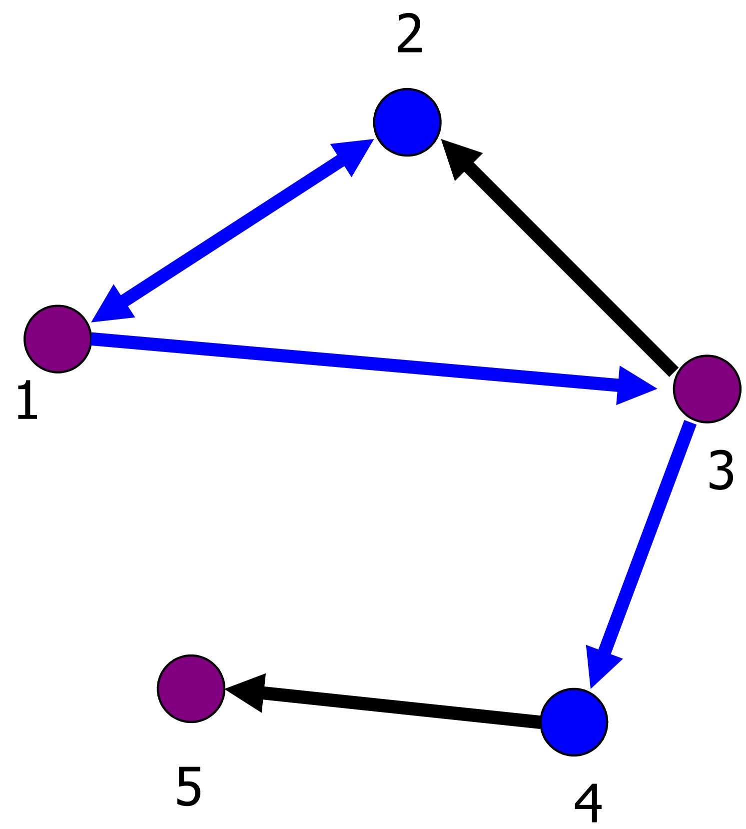
- Subgraph: Given V' $\subseteq$ V, and E' $\subseteq$ E, the graph G'=(V',E') is a subgraph of G
- Induced subgraph: Given V' $\subseteq$ V, let E' $\subseteq$ E is the set of all edges between the nodes in V'. The graph G'=(V',E'), is an induced subgraph of G



Paths

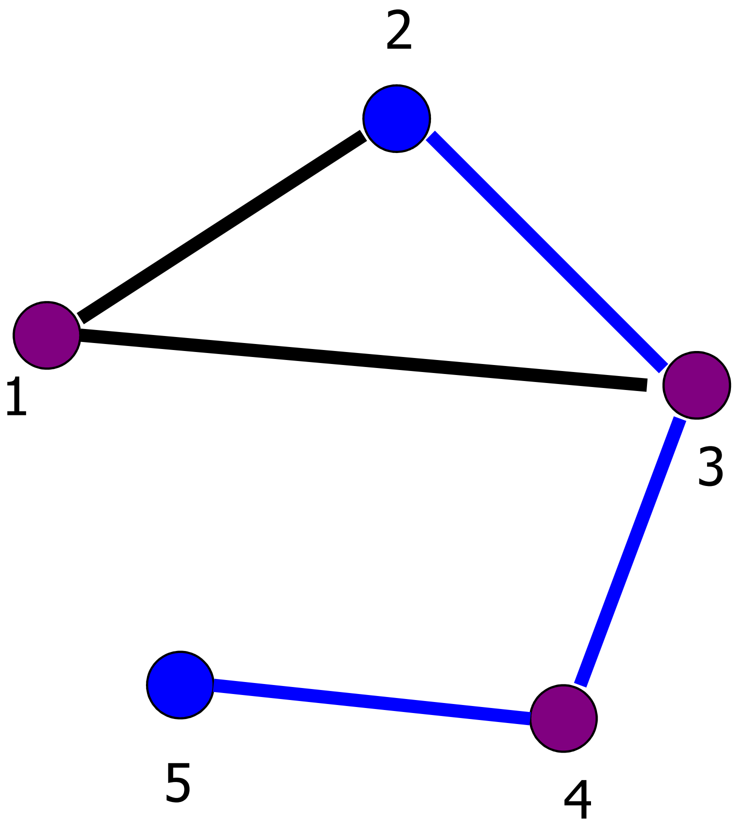
- Path from node i to node j: a sequence of edges (directed or undirected from node i to node j)
  - **Path length**: number of edges on the path
  - nodes i and j are connected if there exists a path from i to j
  - Cycle: a path that starts and ends at the same node

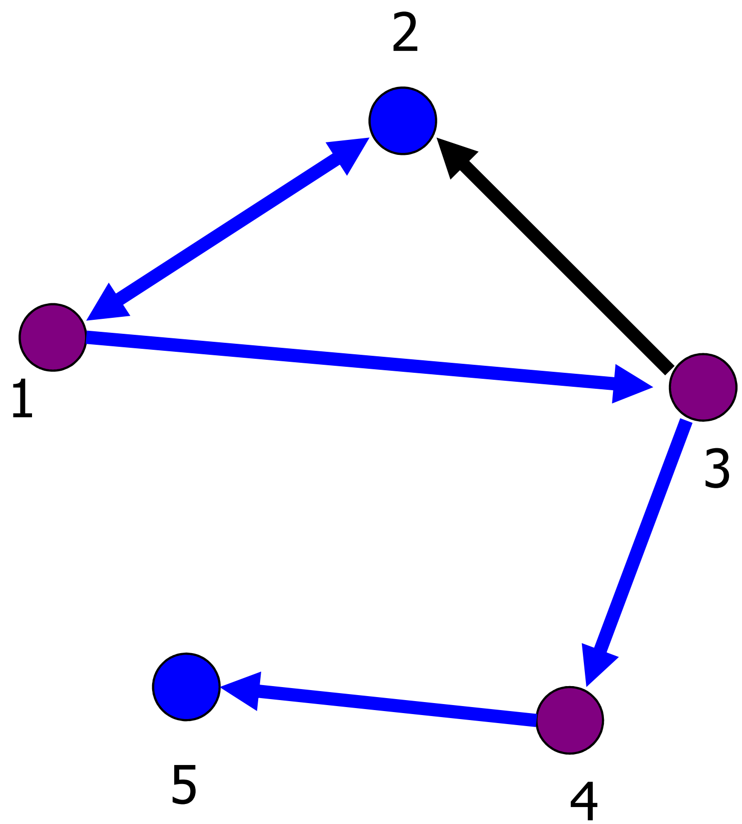




Diameter

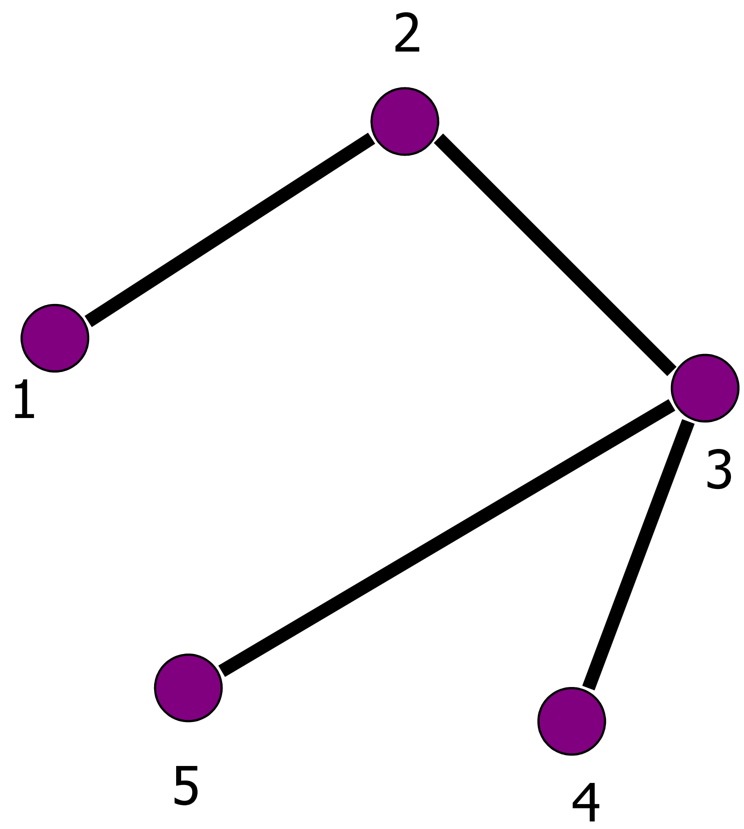
- The largest shortest path in the graph





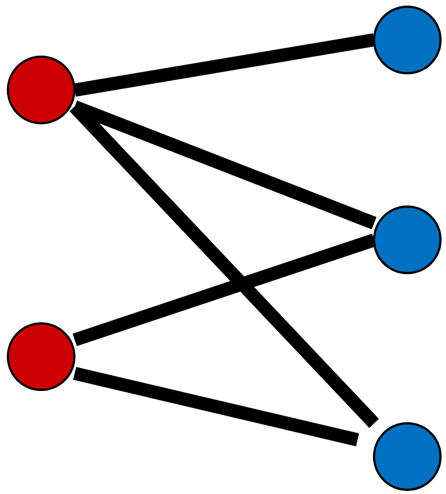
Trees

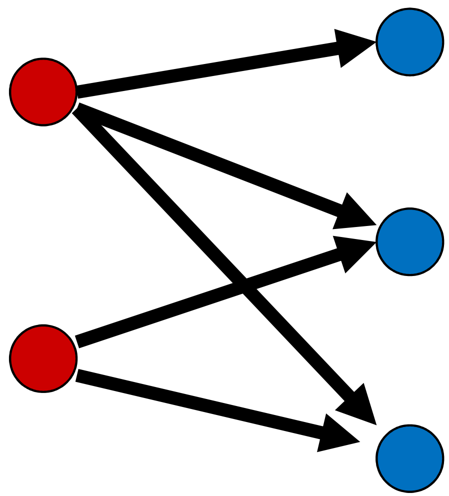
- Connected Undirected graphs without cycles



Bipartite Graphs

- Graphs where the set of nodes V can be partitioned into two sets L and R, such that there are edges only between nodes in L and R, and there is no edge within L or R

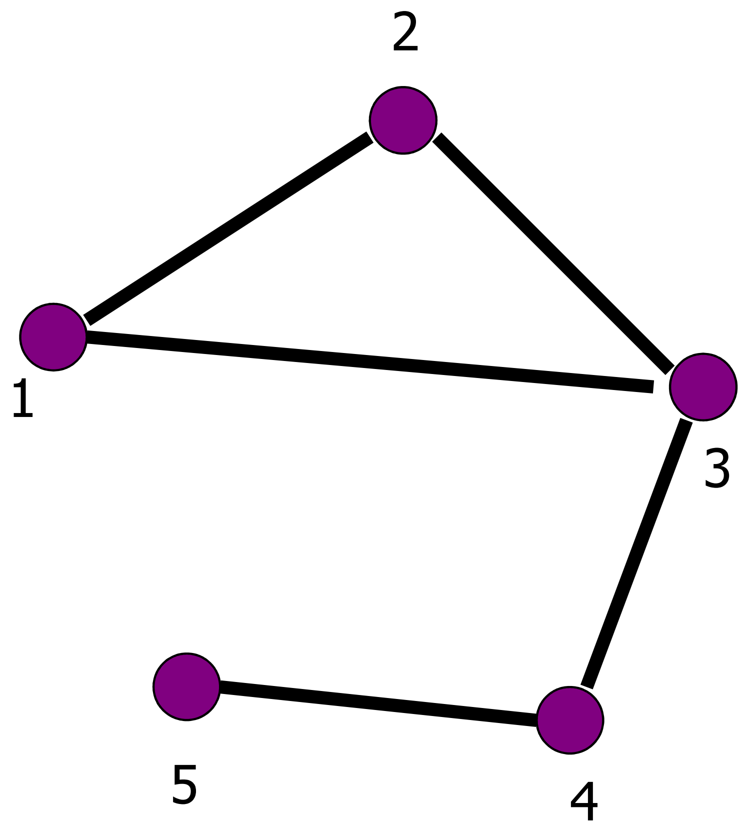




Graph Representation

- Adjacency Matrix
  - symmetric matrix for undirected graphs

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

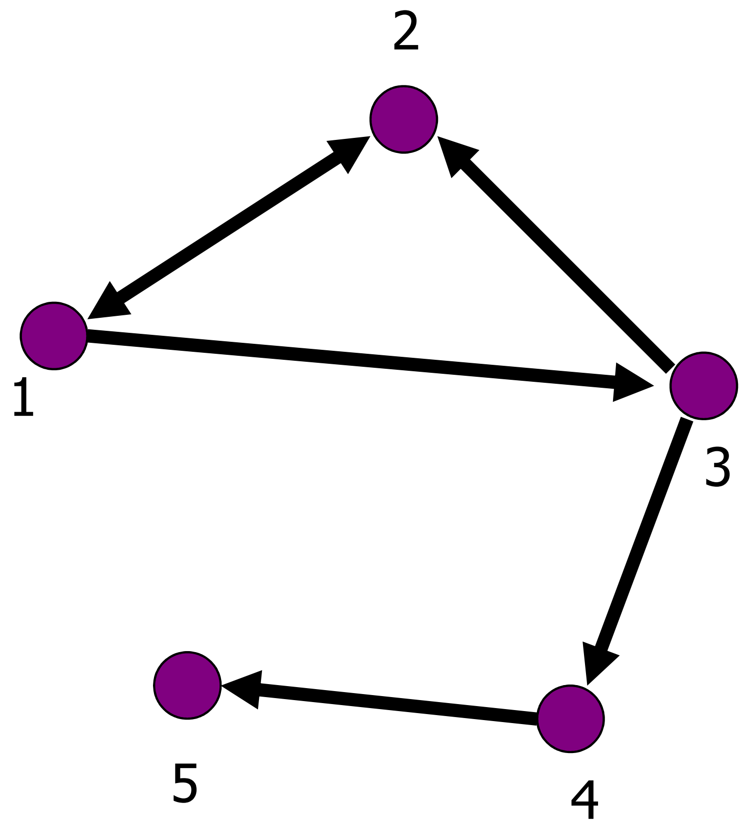


Graph Representation

However, social networks have very **sparse** Adjacency matrices

- Adjacency Matrix
  - unsymmetric matrix for directed graphs

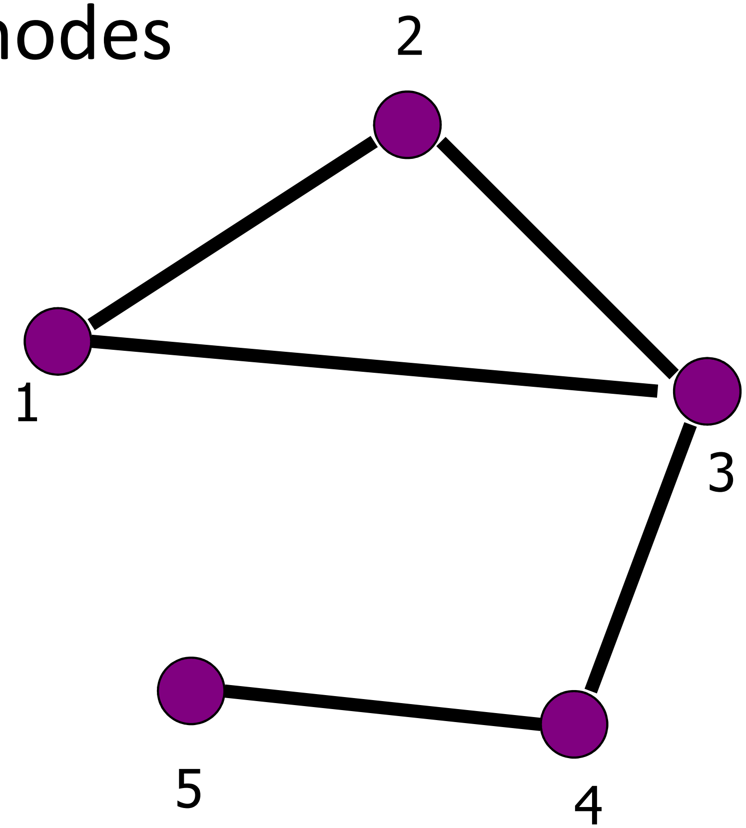
$$A = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$



Graph Representation

- Adjacency List
  - For each node in an undirected graph, keep a list with neighboring nodes

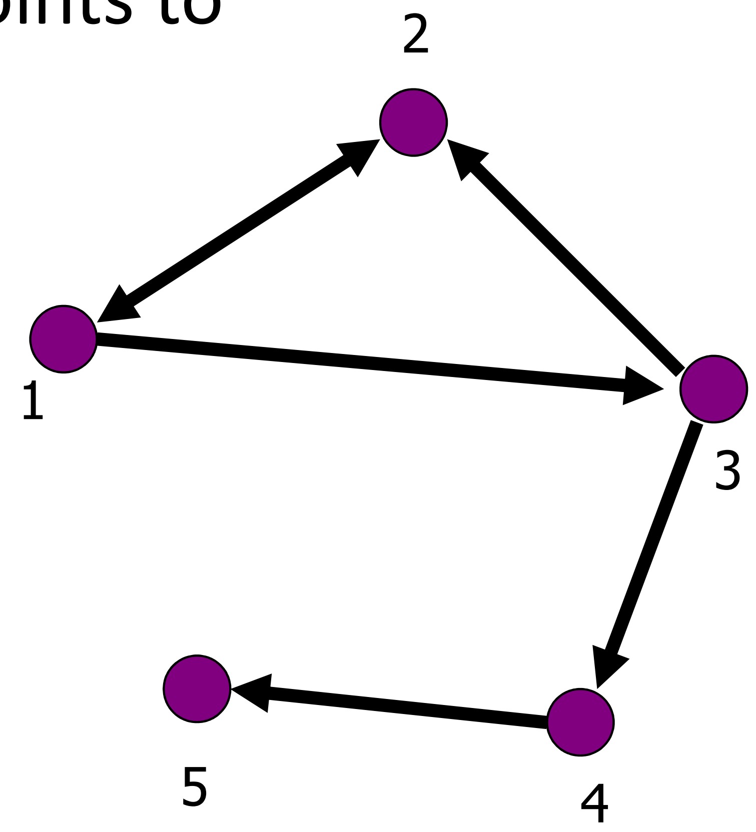
1: [2, 3]
2: [1, 3]
3: [1, 2, 4]
4: [3, 5]
5: [4]



Graph Representation

- Adjacency List
  - For each node in a directed graph, keep a list of the nodes it points to

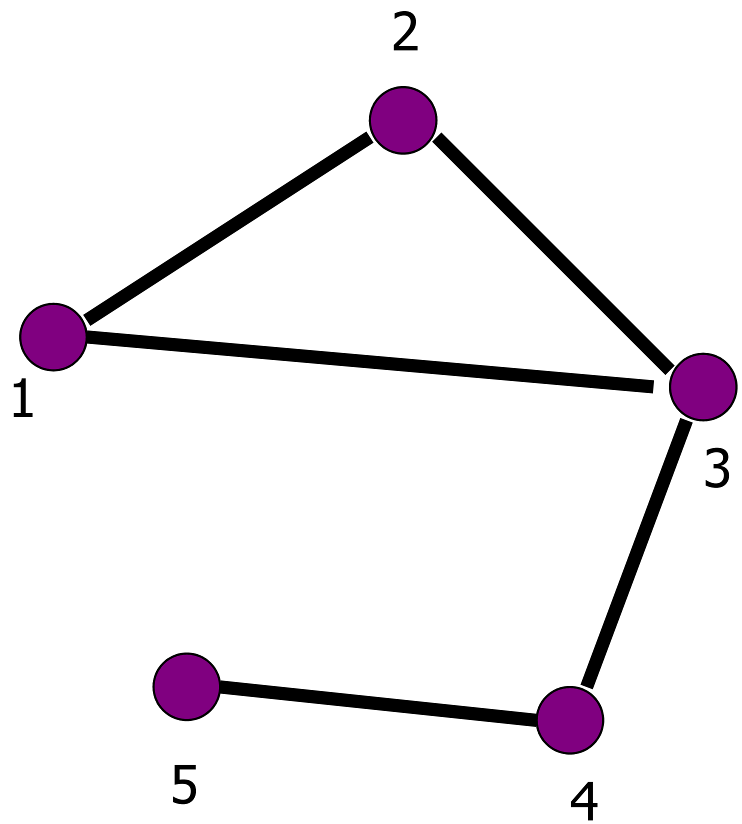
1: [2, 3]
2: [1]
3: [2, 4]
4: [5]
5: [null]



Graph Representation

- List of edges
  - Keep a list of all the edges in the undirected graph

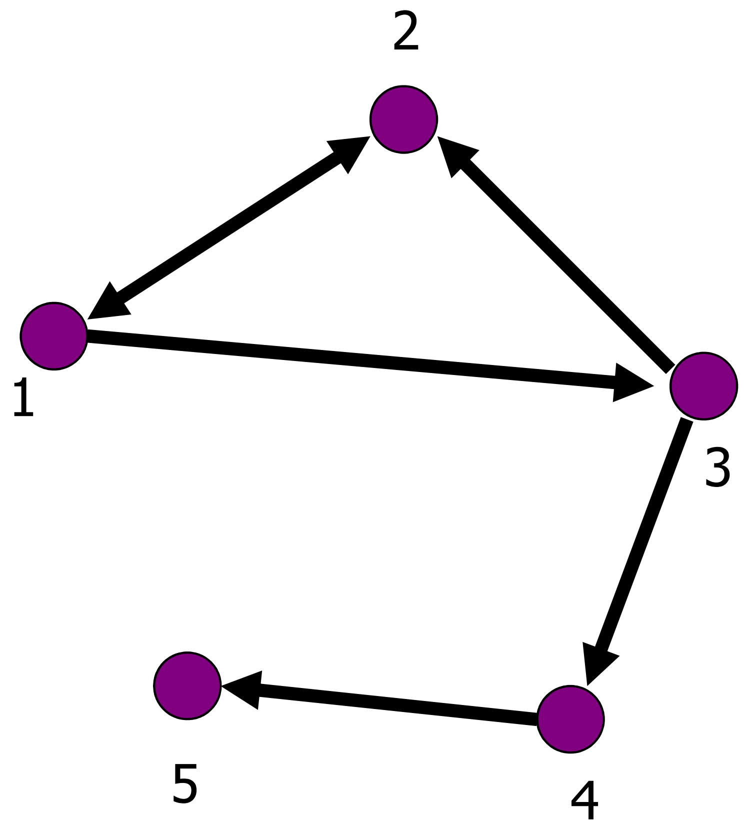
(1,2)
(2,3)
(1,3)
(3,4)
(4,5)



Graph Representation

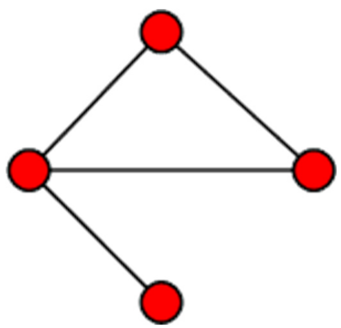
- List of edges
  - Keep a list of all the directed edges in the directed graph

(1,2)
(2,1)
(1,3)
(3,2)
(3,4)
(4,5)

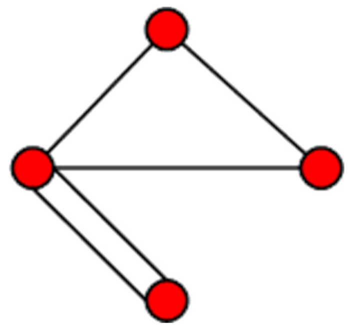


Simple Graphs and Multigraphs

- Simple graphs are graphs where only a single edge can be between any pair of nodes
- Multigraphs are graphs where you can have multiple edges between two nodes and loops



Simple graph



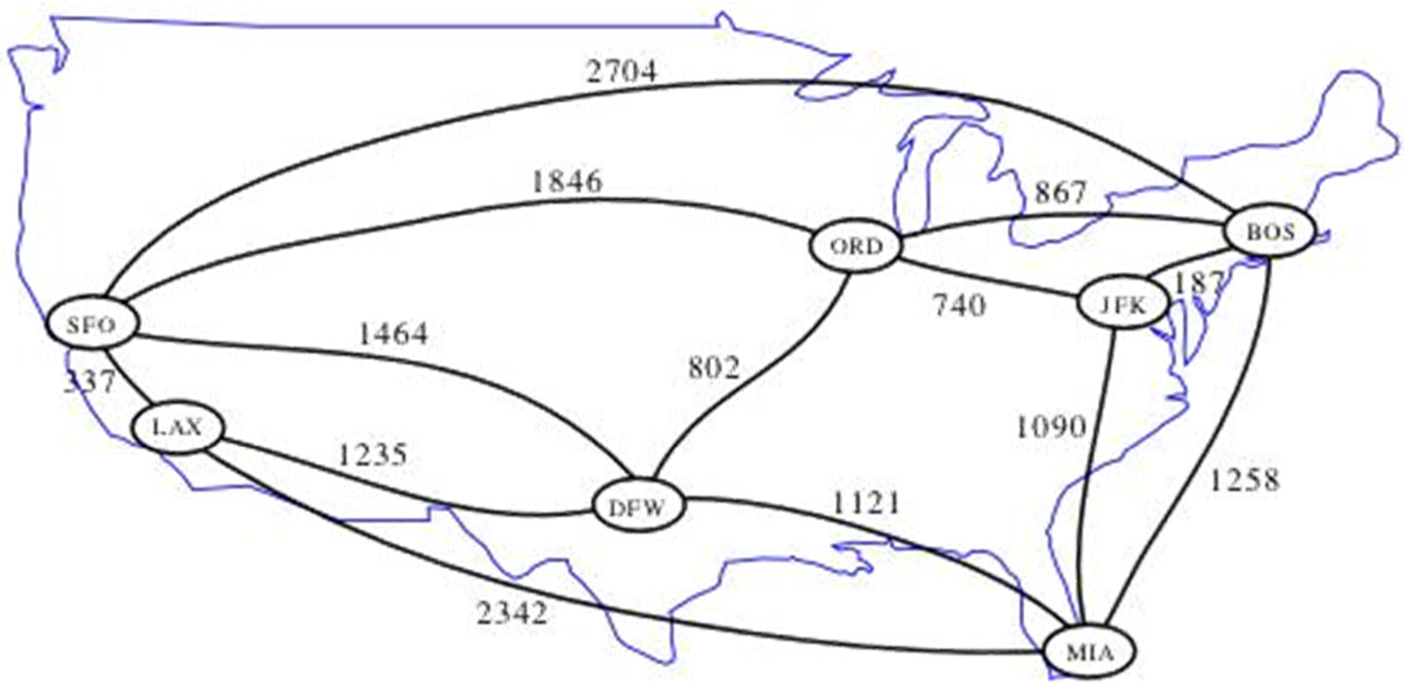
Multigraph

Weighted Graph

- A weighted graph $G(V, E, W)$ is one where edges are associated with weights

$$A_{ij} = \begin{cases} w_{ij} \text{ or } w(i,j), w \in R \\ 0, \text{there is no edge between } v_i \text{ and } v_j \end{cases}$$

- An example on US flight network:



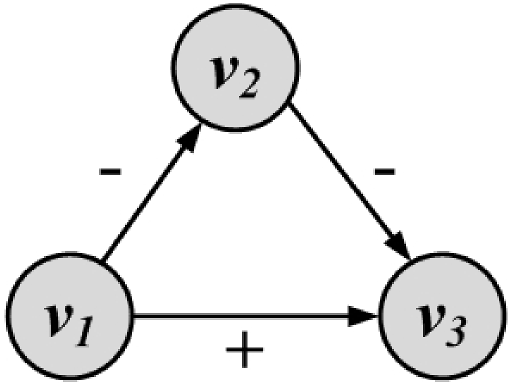
Edge Attributes

Possible options

- Weight (e.g. frequency of communication)
- Ranking (best friend, second best friend…)
- Type (friend, relative, co-worker)
- Sign: Friend vs. Foe, Trust vs. Distrust
- Properties depending on the structure of the rest of the graph: number of common friends

Signed Graph

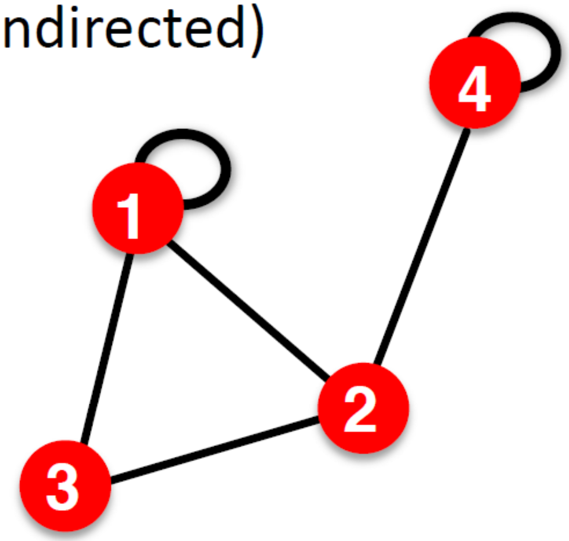
- When weights are binary (0/1, -1/1, +/-), we have a signed graph



- It is used to represent friends or foes
  - Trust/Distrust (Epinions), Support/Oppose (Wikipedia) Like/Dislike (YouTube),
- It is also used to represent social status
  - $A \xrightarrow{+} B$: B has higher status than A

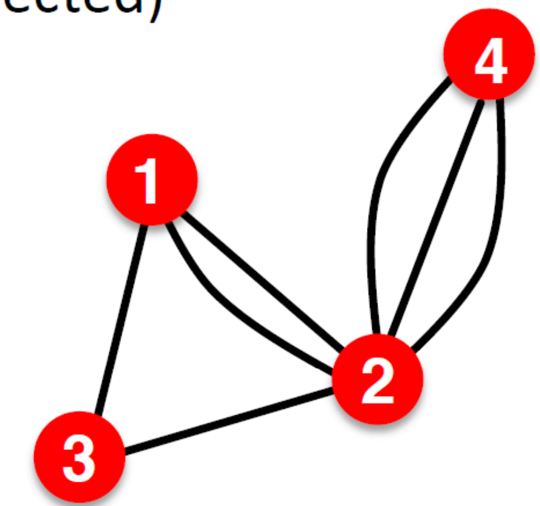
More Types of Graphs

- **Self-edges (self-loops)**
  (undirected)



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

- **Multigraph**
  (undirected)



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

Example Graphs

Email network >> directed multigraph with self-edges
Facebook friendships >> undirected, unweighted
Citation networks >> unweighted, directed, acyclic
Collaboration networks >> undirected multigraph or weighted graph
Mobile phone calls >> directed, (weighted?) multigraph
Protein Interactions >> undirected, unweighted with self-interactions