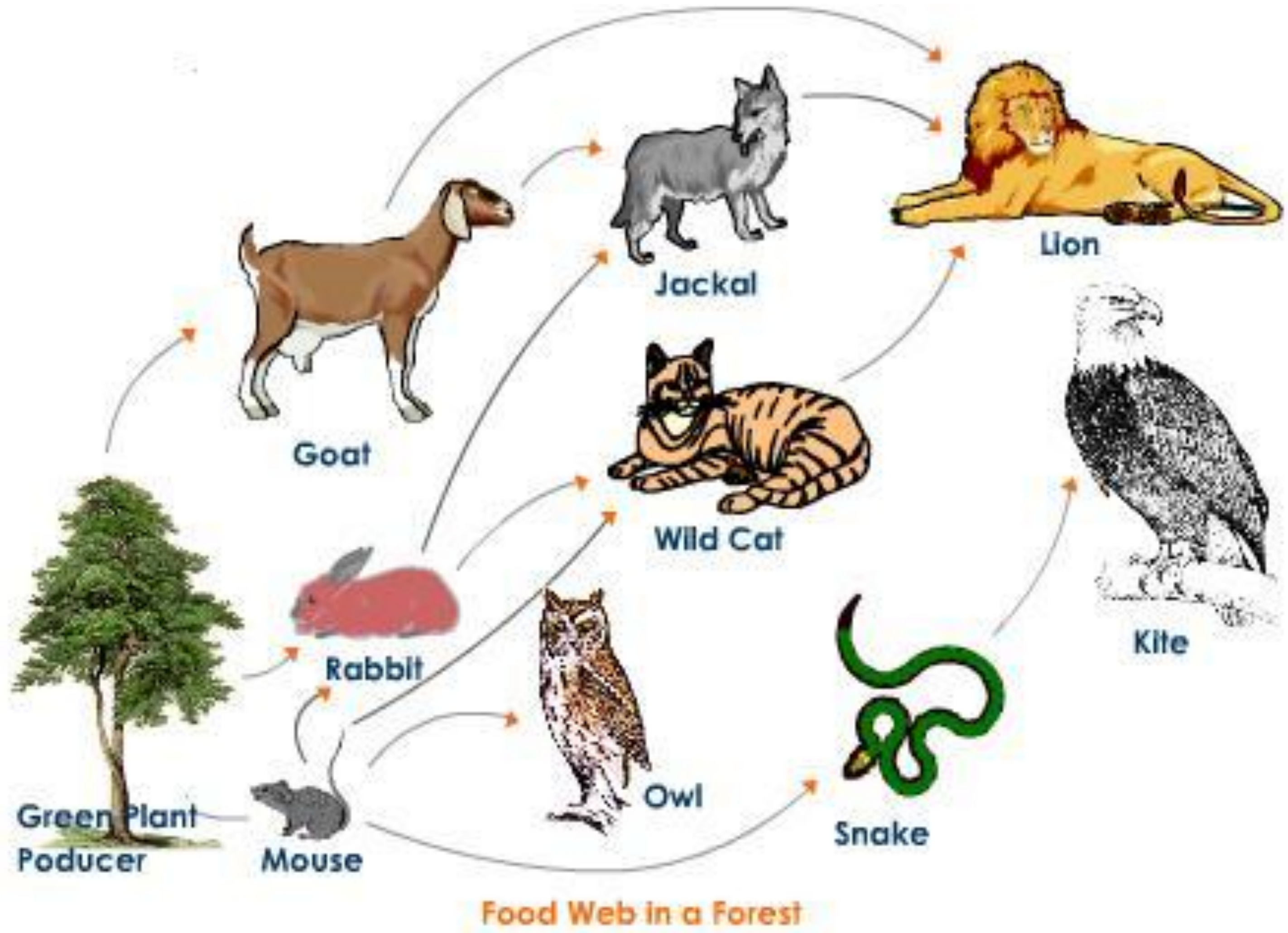
CS 579: Online Social Network Analysis

Chapter 2 Graph Essentials

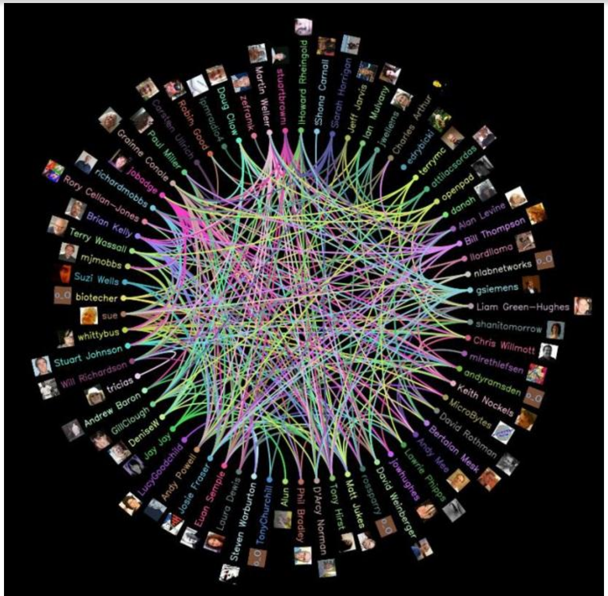
Kai Shu

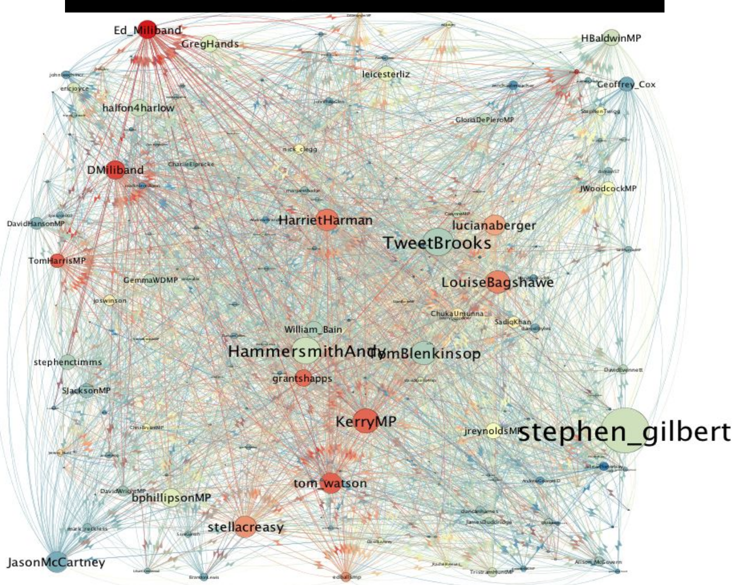
Spring 2023

Our World is a Connected World: Food Web

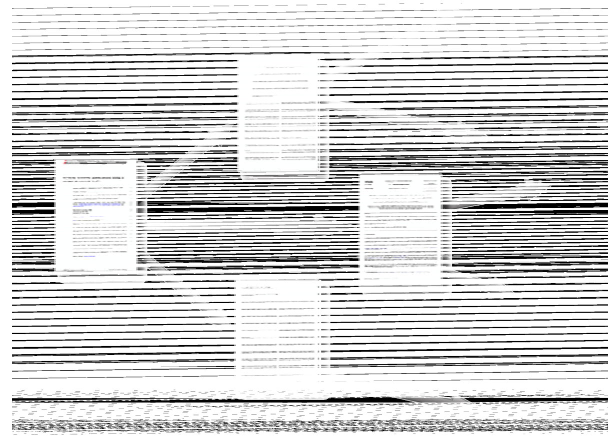


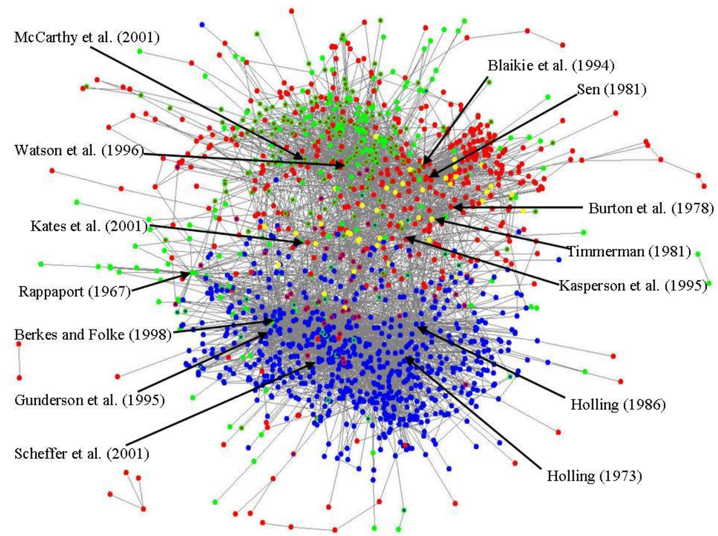
Networks are Pervasive





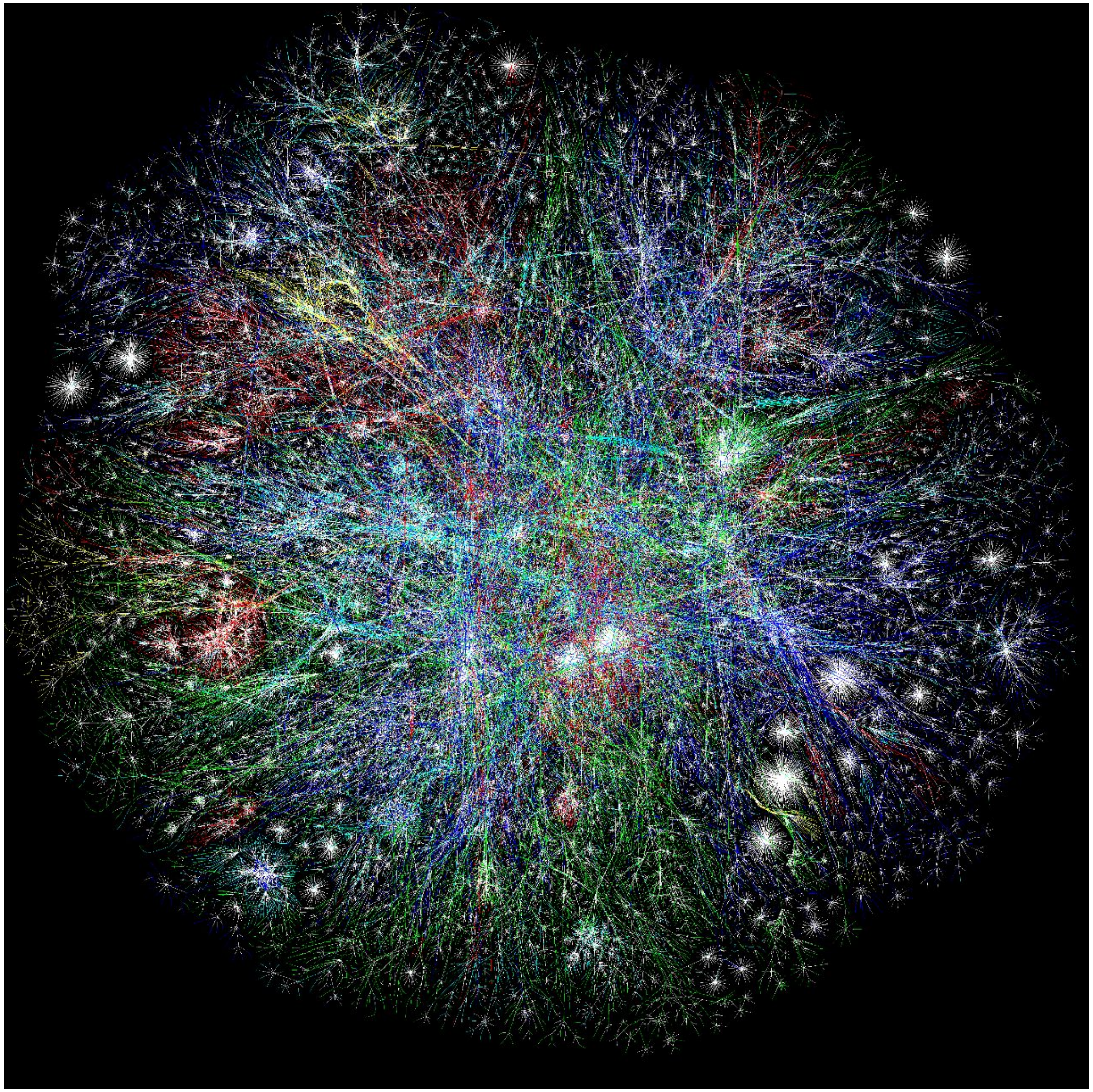
Twitter Networks



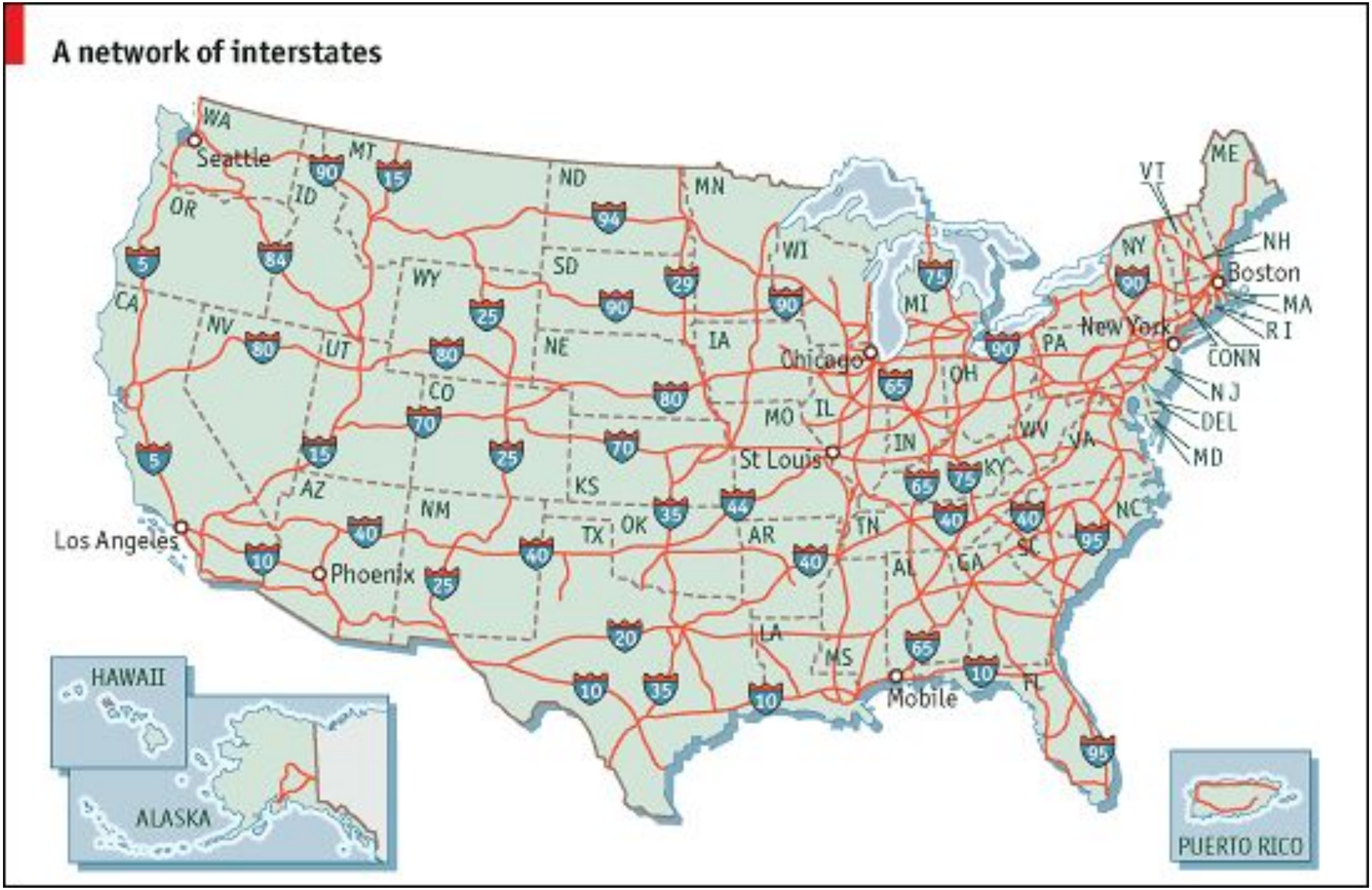


Citation Networks

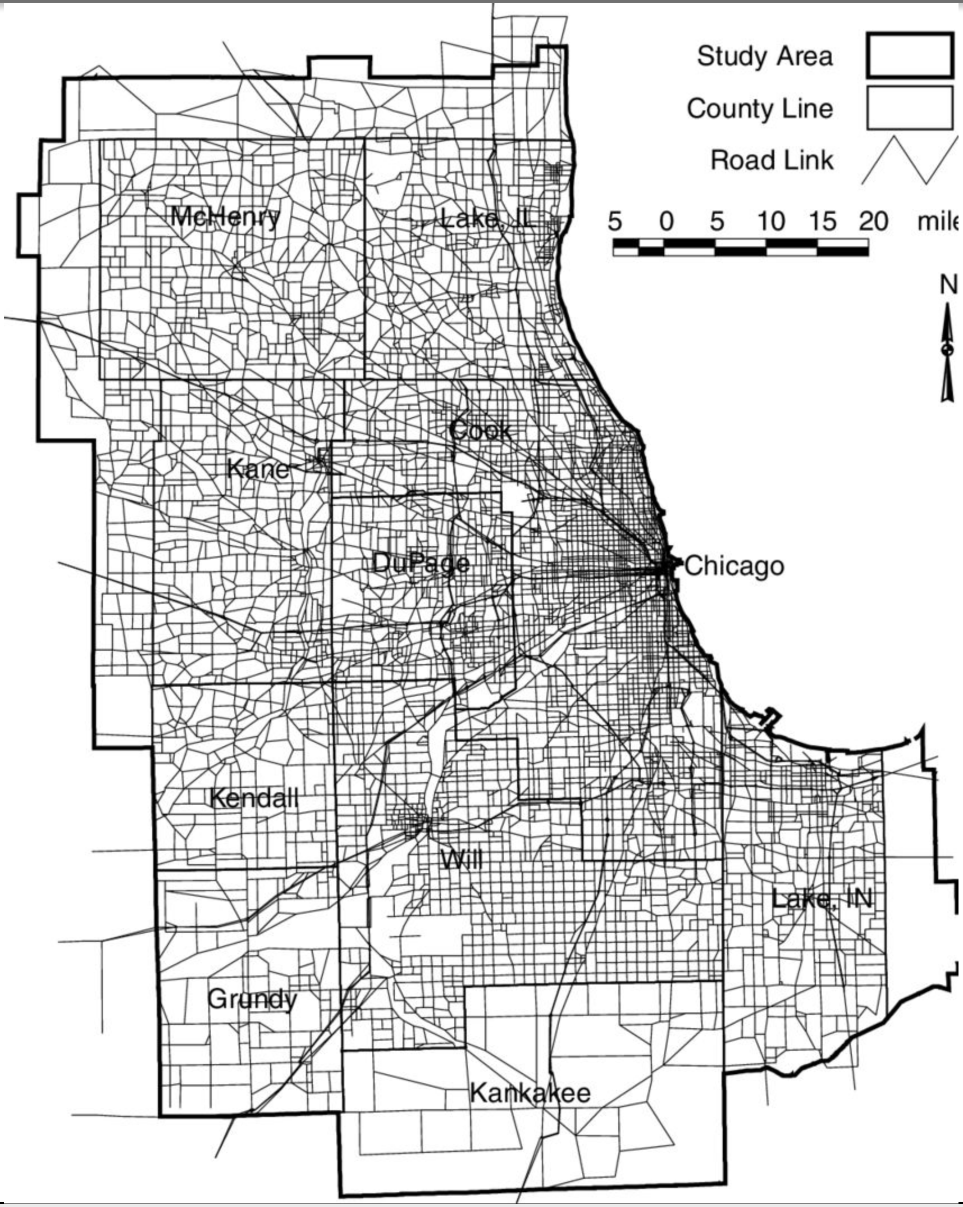
Internet is a Vast Network



US Interstate Highways Form a Network



Close to Home: Chicago Road Network



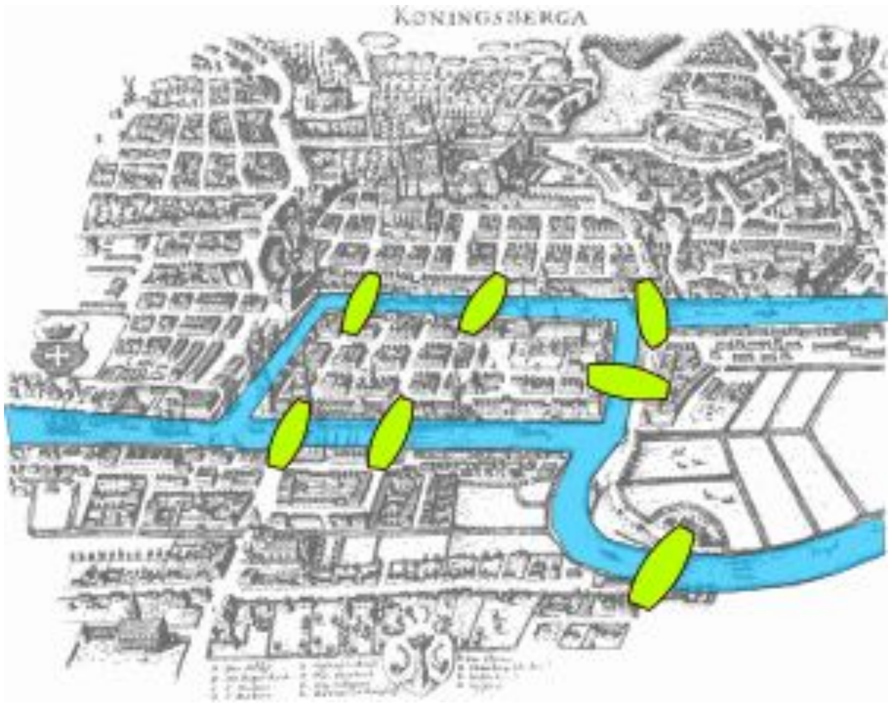
Social Networks and Social Network Analysis

- A social network
  - A network where elements have a social structure
    - A set of actors (such as individuals or organizations)
    - A set of ties (connections between individuals)
- Examples of social networks:
  - Our family networks, our friendship networks, our colleagues, etc.
- To analyze these networks, we use Social Network Analysis (SNA)
- Social Network Analysis is an interdisciplinary field from social sciences, statistics, physics, graph theory, complex networks, computer science, …

Graph Basics

Bridges of Konigsberg – A First Network in Math

- There are 2 islands and 7 bridges that connect the islands and the mainland
- Find a path that *crosses each bridge exactly once*

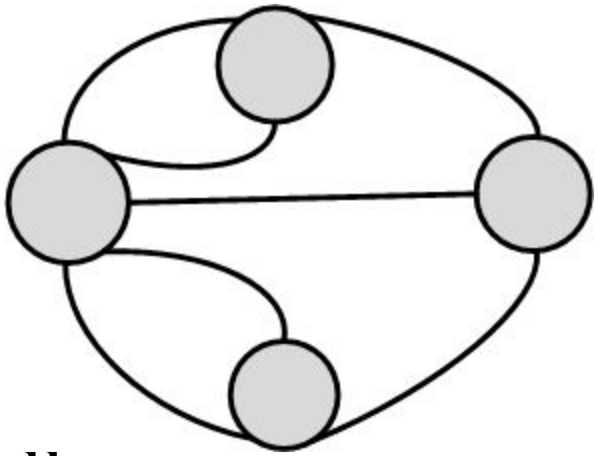


City Map (source: Wikipedia)

How to represent bridges and the land divided by them?

We need an abstract form, or representation

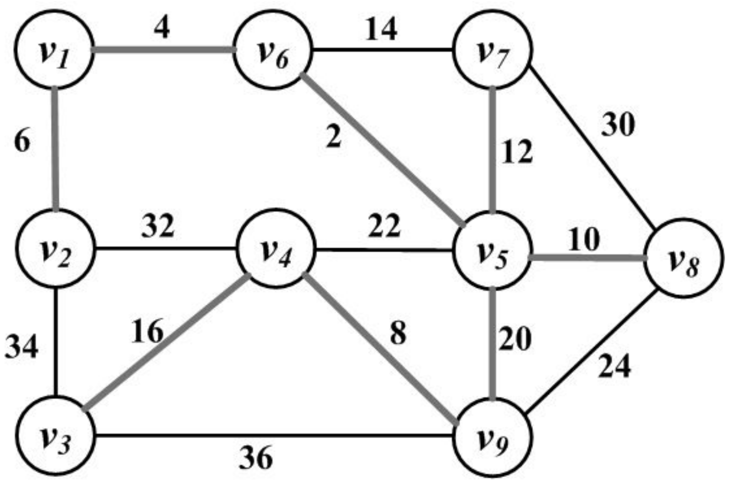
Modeling the Problem in Graph Theory



- The key to solving this problem is an ingenious graph representation
  - What should be nodes and edges?
- Euler proved that except for the starting and ending point of a walk, one has to enter and leave all other nodes, thus these nodes should have an even number of bridges connected to them
- This property does not hold in this problem

Networks

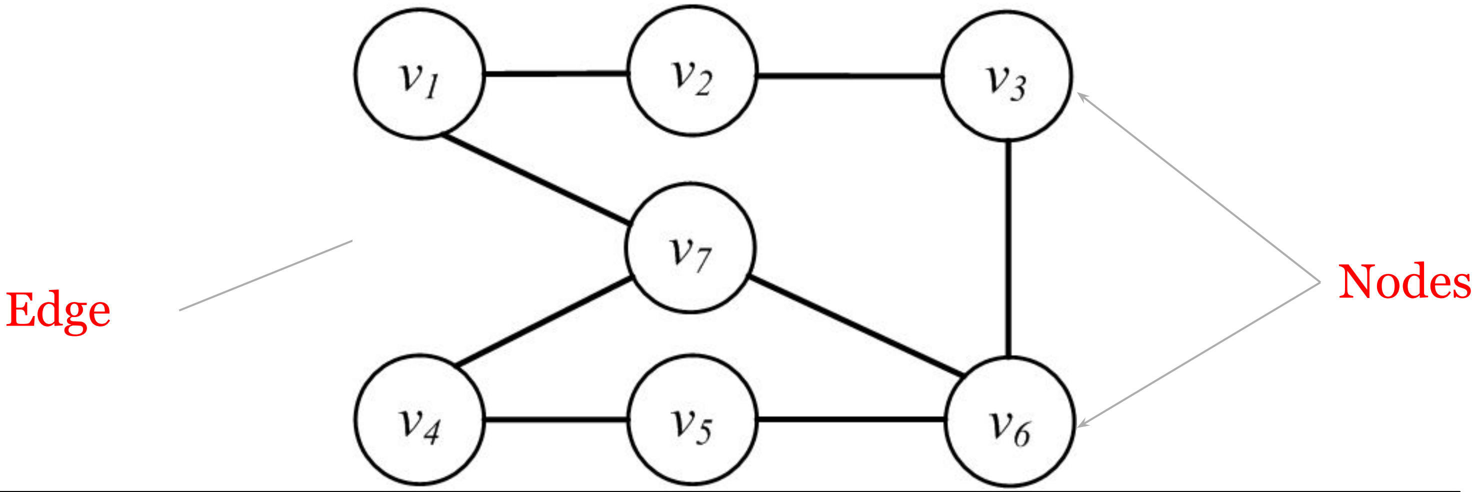
- A network can be represented as a graph
  - Elements of the network have meanings
- Network problems can usually be represented in terms of graph theory
- Twitter example:
  - Given a piece of information, a *network* of individuals, and the *cost* to propagate information among any connected pair, find the minimum *cost* to disseminate the information from a given pair of users



Nodes and Edges

A network is a graph, or a collection of *points* connected by *lines*

- Points are referred to as **nodes, actors,** or **vertices**
- Connections are referred to as **edges** or **ties**



Nodes or Actors

- In a *friendship* graph, *nodes* are people and any pair of people connected denotes the *friendship* between them
- Depending on the context, these nodes are called nodes, or actors
  - In a web graph, “nodes” represent sites and the connection between nodes indicates web-links between them
  - In a social setting, these nodes are called actors

$$V = \{v_1, v_2, \ldots, v_n\}$$

  - The size of the graph is $|V| = \mathbf{n}$

Edges or Ties

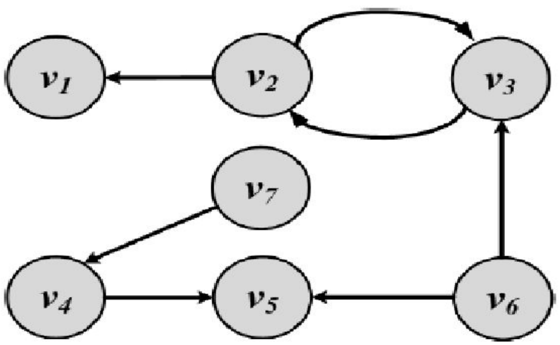
- Edges connect nodes and are also known as **ties** or **relationships**
- In a social setting, where *nodes* represent **social entities** such as people, *edges* indicate internode relationships and are therefore known as *relationships* or (social) *ties*

$$E = \{e_1, e_2, \ldots, e_m\}$$

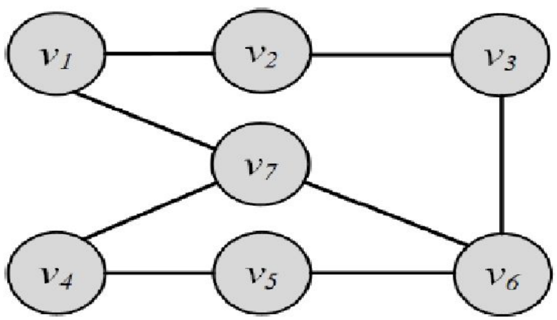
- Number is edges (size of the edge-set) is denoted as |E|=**m**

Directed Edges and Directed Graphs

- Edges can have directions. A directed edge is sometimes called an *arc*



(a) Directed Graph



(b) Undirected Graph

- Edges are represented using their end-points (or end-nodes), **e(v2,v1)**. In undirected graphs, both representations are the same, i.e., **e(v1,v2)=e(v2,v1)**

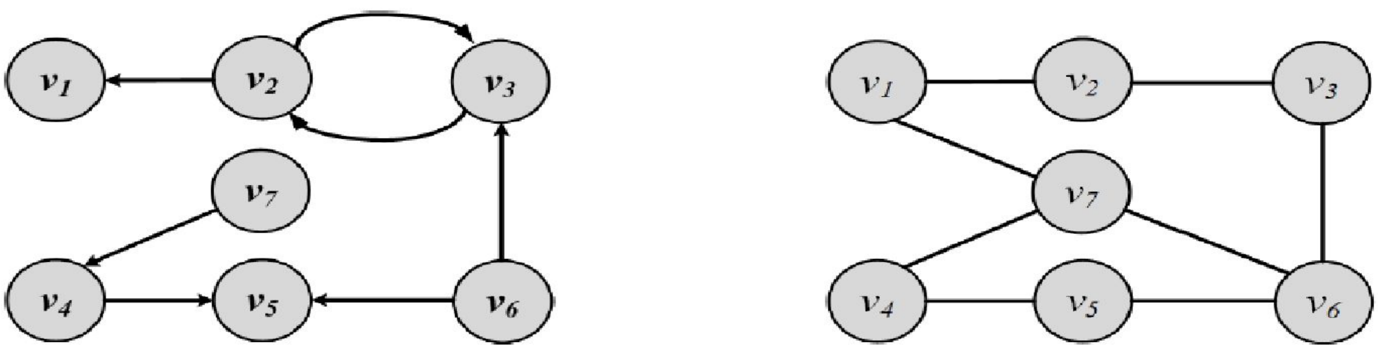
Neighborhood and Degree (In-degree and out-degree)

- For any node *v*, the set of nodes it is connected to via an edge is called its neighborhood and is represented as ***N(v)***

- The number of edges connected to one node is the **degree** of that node (the size of its neighborhood)
  - Degree of a node *i* is usually presented using notation d_i
  - In case of directed graphs,
    - d_i^{in} In-degrees is the number of edges pointing towards a node
    - d_i^{out} Out-degree is the number of edges pointing away from a node

Let’s look at some examples

- N(v1) = ?
- degree of node v1?
- in-degree and out-degree of v1?



(a) Directed Graph (b) Undirected Graph

Degree and Degree Distribution

- **Theorem 1.** The summation of degrees in an *undirected* graph is twice the number of edges

$$\sum_i d_i = 2|E|$$

- **Lemma 1.** The number of nodes with odd degree is even
- **Lemma 2.** In any *directed* graph, the summation of in-degrees is equal to the summation of out-degrees,

$$\sum_i d_i^{out} = \sum_j d_j^{in}$$

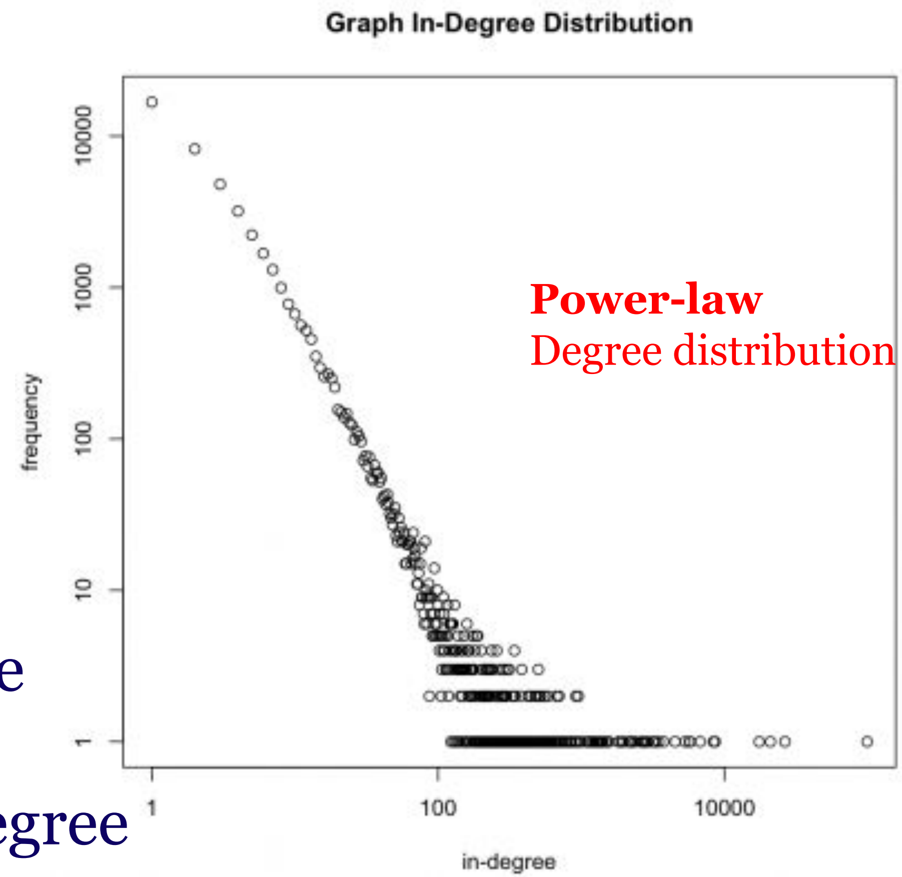
Degree Distribution

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

- Degree sequence

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

- Let's revisit Slide 17(a)



Degree distribution histogram

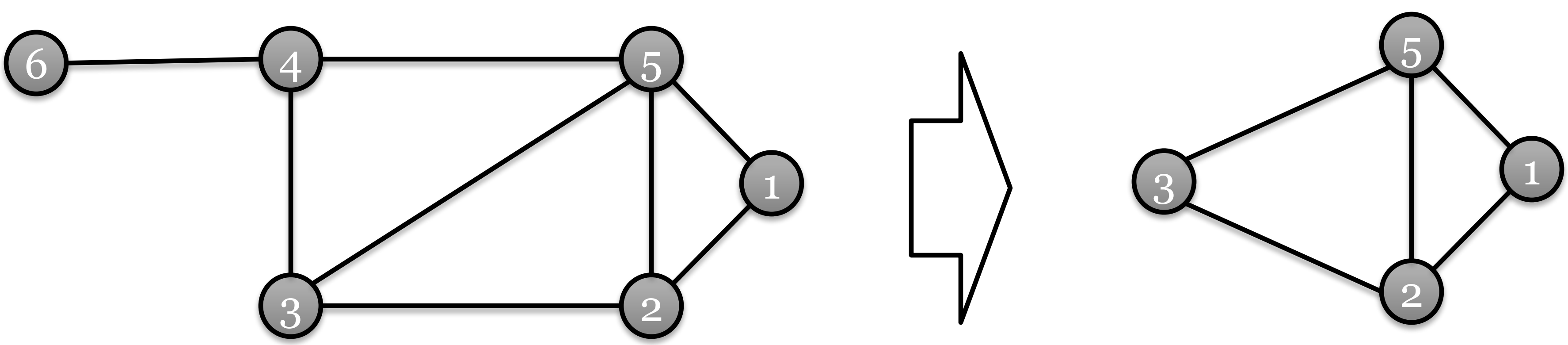
– The x-axis represents the degree and the y-axis represents the number of nodes having that degree

Subgraph

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

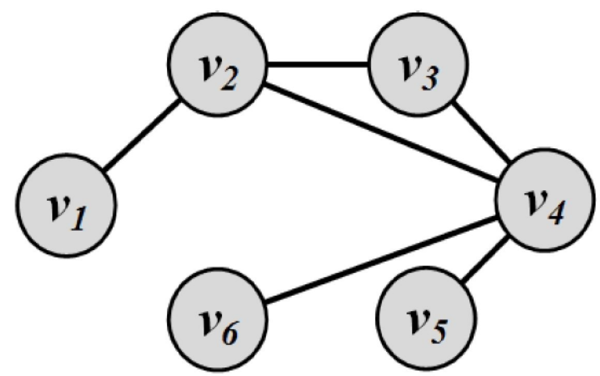
- G'(V',E') is a subgraph of G(V,E)

$$V' \subseteq V,$$
$$E' \subseteq (V' \times V') \cap E$$



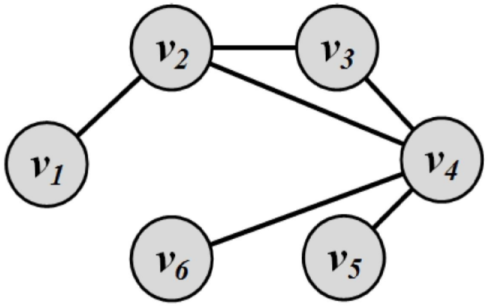
Graph Representation

- **Adjacency Matrix**
- **Adjacency List**
- **Edge List**



Graph Representation

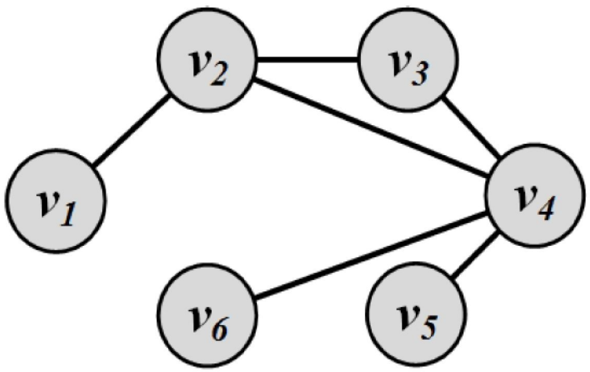
- Graph representation is straightforward and intuitive, but it cannot be effectively manipulated using mathematical and computational tools



- We are seeking representations that can store these **two sets (V, E)** in a way such that they
  – Do not lose information
  – Can be processed easily by computers
  – Can have mathematical methods to be applied easily

Adjacency Matrix

$$A_{ij} = \begin{cases} 1, \text{ if there is an edge between nodes } v_i \text{ and } v_j \\ 0, \text{ otherwise} \end{cases}$$



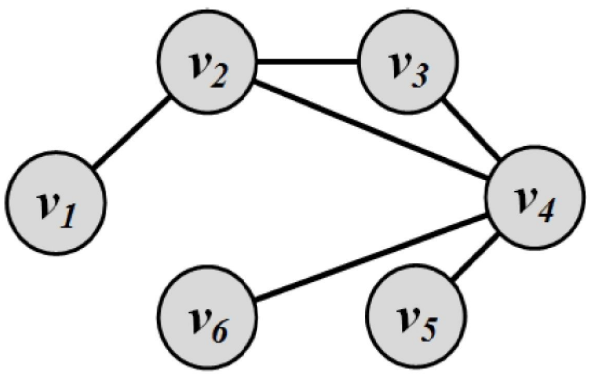
	v_1	v_2	v_3	v_4	v_5	v_6
v_1	0	1	0	0	0	0
v_2	1	0	1	1	0	0
v_3	0	1	0	1	0	0
v_4	0	1	1	0	1	1
v_5	0	0	0	1	0	0
v_6	0	0	0	1	0	0

- Diagonal Entries are self-links or loops

Social media networks have very sparse adjacency matrices

Adjacency List

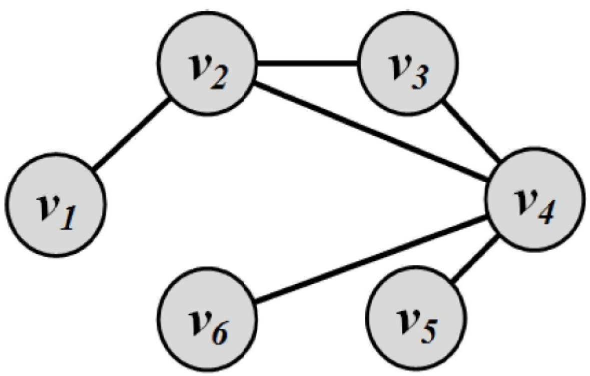
- In an adjacency list for every node, we maintain a list of all the nodes that it is connected to
- The list is usually **sorted** based on the node order or other preferences



Node	Connected To
v_1	v_2
v_2	v_1 , v_3 , v_4
v_3	v_2 , v_4
v_4	v_2 , v_3 , v_5 , v_6
v_5	v_4
v_6	v_4

Edge List

- In this representation, each element is an edge and is usually represented as (u, v), denoting that node u is connected to node v via an edge



(v_1,v_2)
(v_2,v_3)
(v_2,v_4)
(v_3,v_4)
(v_4,v_5)
(v_4,v_6)

Types of Graphs

- **Null, Empty, Directed/Undirected/Mixed, Simple/Multigraph, Weighted, Webgraph, Signed Graph**

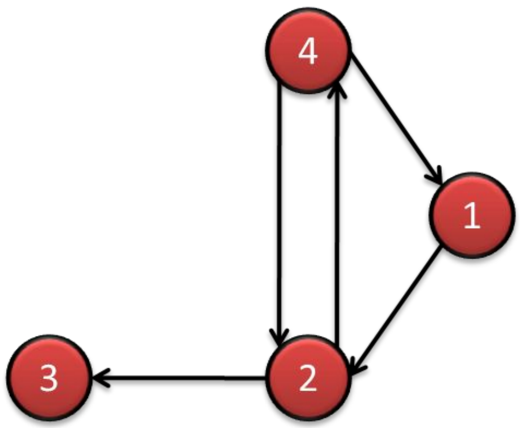
Null Graph and Empty Graph

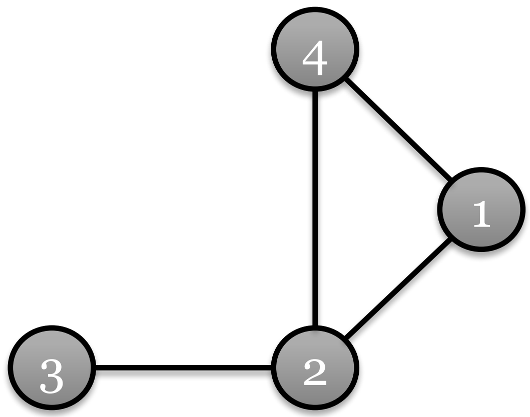
- A **null graph** is one where the node set is empty (there are no nodes)
  - Since there are no nodes, there are also no edges

$$G(V, E),\ \ V = E = \varnothing$$

- An **empty graph** or **edge-less graph** is one where the edge set is empty, $E = \varnothing$
  - The node set can be non-empty.
  - A null-graph is an empty graph.

Directed/Undirected/Mixed Graphs

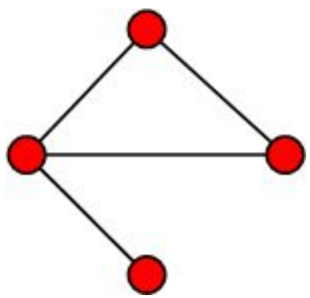




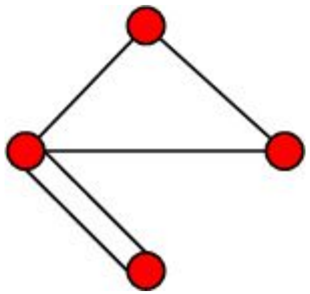
- The adjacency matrix for directed graphs is not symmetric ($A \neq A^T$)
  - ($A_{ij} \neq A_{ji}$) is not true for all pairs i,j
- The adjacency matrix for undirected graphs is symmetric ($A = A^T$)

Simple Graphs and Multigraphs

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



Simple graph



Multigraph

- The adjacency matrix for multigraphs can include numbers larger than one, say 3, indicating that multiple (3) edges can exist between nodes

Weighted Graph

- A weighted graph is one where edges are associated with **weights**
  – For example, a graph could represent a map where nodes are cities and edges are routes between them
  – The weight associated with each edge could represent the distance between these cities

G(V, E, W)

$$A_{ij} = \begin{cases} w, w \in R \\ 0, \text{There is no edge between} \end{cases}$$