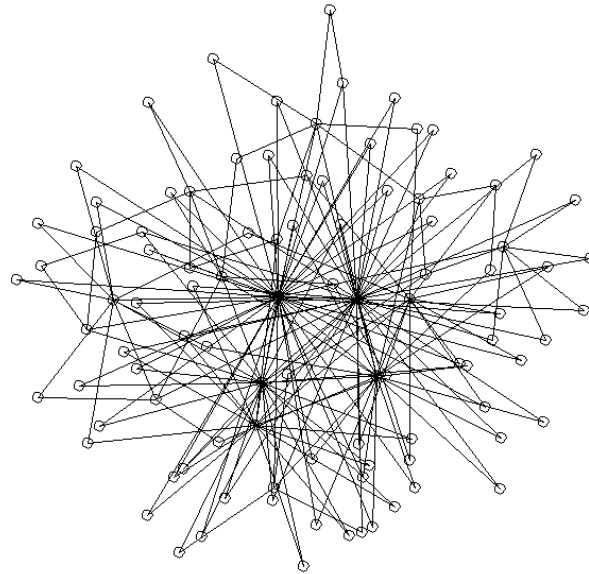
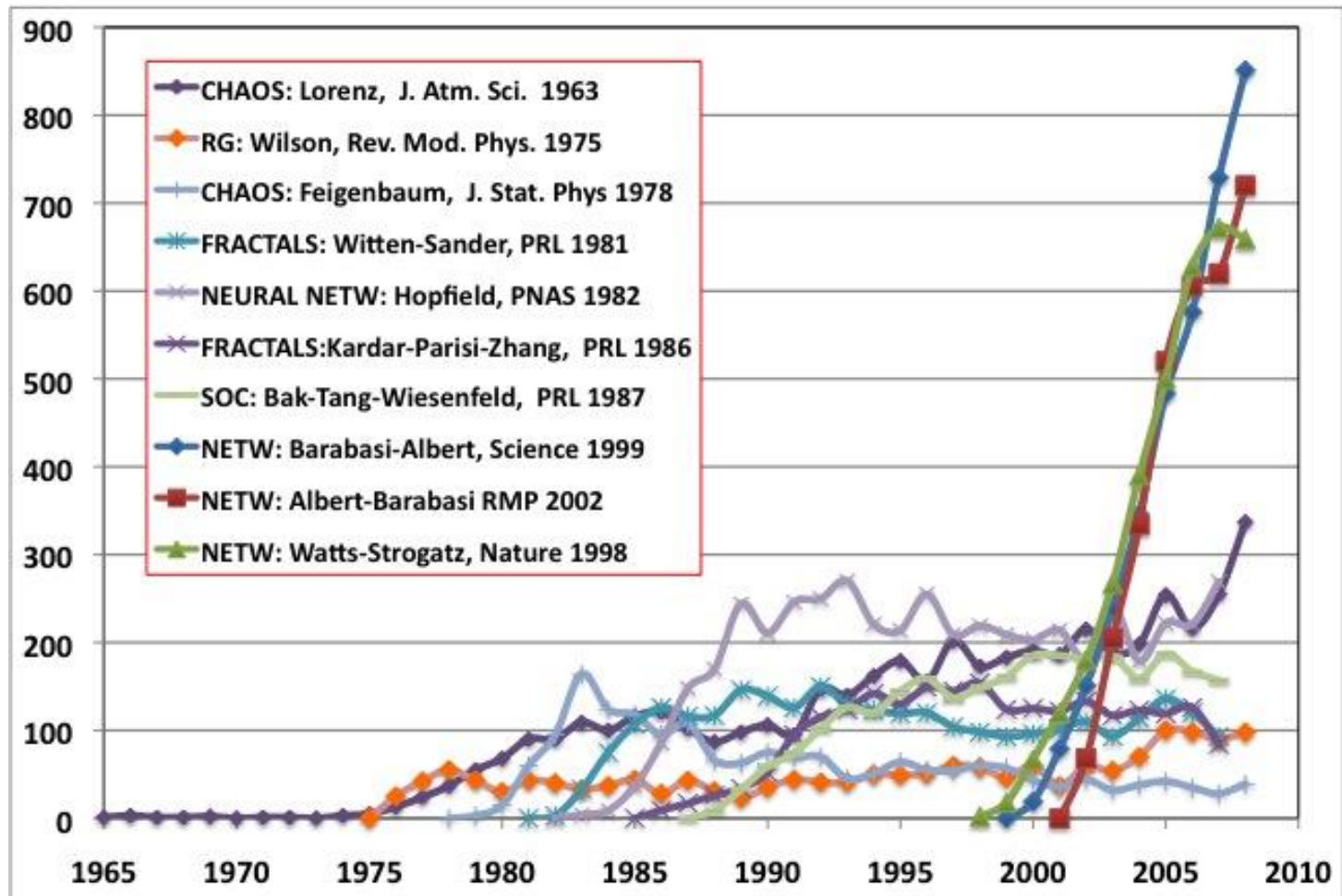


# Fundamentals of Graph Theory



**Hiroki Sayama**  
sayama@binghamton.edu

# Rise of Network Science



(From Barabasi 2010)

# Network models

---

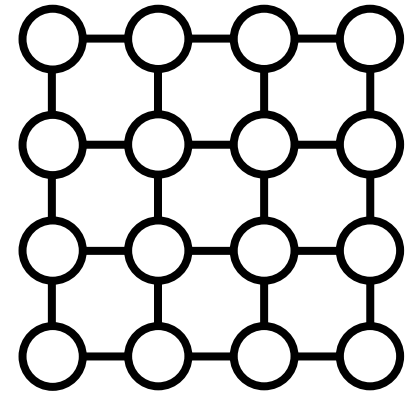
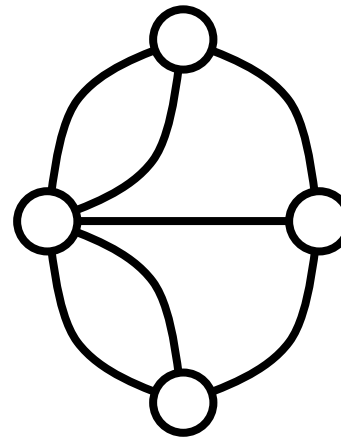
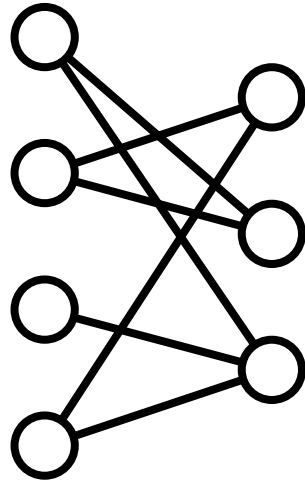
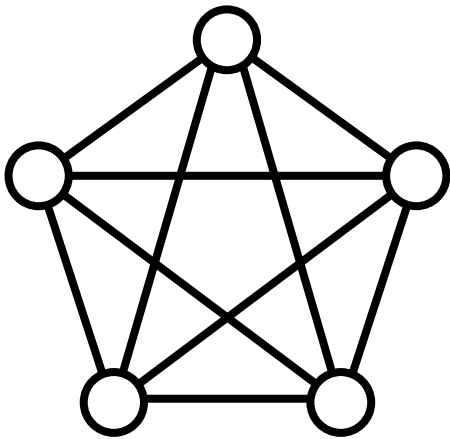
- Many discrete parts involved
  - Classic mean-field models not appropriate
- Parts are not uniformly connected
  - CA or cont. field models not appropriate
- Parts may dynamically increase / decrease in number
  - This deviates from typical dynamical systems assumptions

# Fundamentals of Graph Theory

# Graph?

---

- Mathematical structure that consists of "nodes" (or vertexes) and "edges" (or links) that make connection between nodes

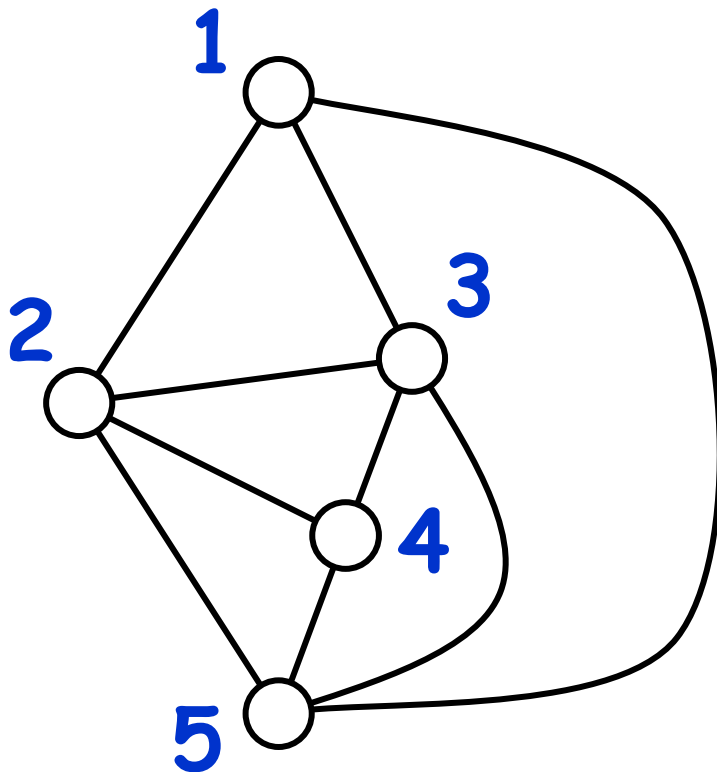


# Graph = Network

---

- $G(V, E)$ : graph (network)

$V$ : vertices (nodes),  $E$ : edges (links)



Nodes = 1, 2, 3, 4, 5

Links =

1  $\leftrightarrow$  2, 1  $\leftrightarrow$  3, 1  $\leftrightarrow$  5,  
2  $\leftrightarrow$  3, 2  $\leftrightarrow$  4, 2  $\leftrightarrow$  5,  
3  $\leftrightarrow$  4, 3  $\leftrightarrow$  5, 4  $\leftrightarrow$  5

(Nodes may have states;  
links may have directions  
and weights)

# Representation of a network

---

- **Adjacency matrix:**

A matrix with rows and columns labeled by nodes, where element  $a_{ij}$  shows the number of links going from node  $i$  to node  $j$   
(becomes symmetric for undirected graph)

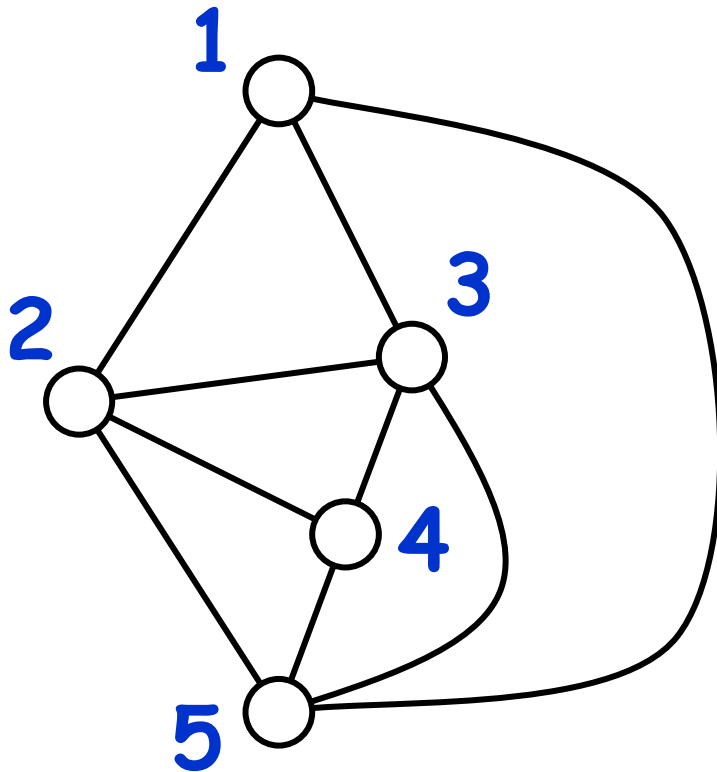
- **Adjacency list:**

A list of links whose element " $i \rightarrow j$ " shows a link going from node  $i$  to node  $j$   
(also represented as " $i \rightarrow \{j_1, j_2, j_3, \dots\}$ ")

# Exercise

---

- Represent the following network in:



- Adjacency matrix

- Adjacency list



# Properties of matrix description

- Labels of nodes and links are no more than a set of symbolic identifications that may be in any arbitrary order

→ Re-labeling of nodes / links doesn't affect graph topology

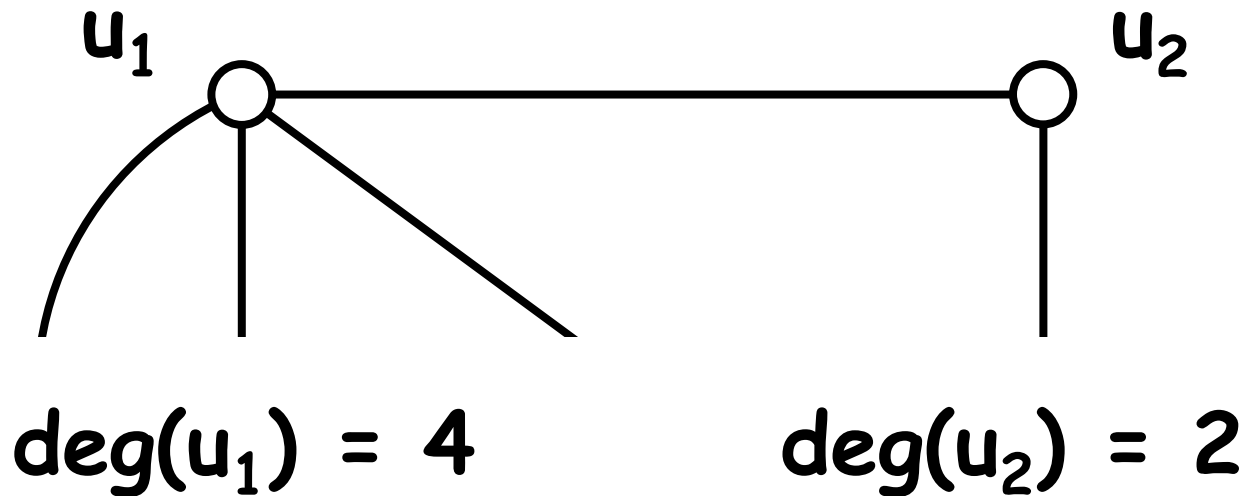
→ Permutation of rows / columns of the matrix doesn't affect graph topology

(For adjacency matrices, the same permutation must be applied for both rows and columns)

# Degree of a node

---

- A degree of node  $u$ ,  $\text{deg}(u)$ , is the number of links connected to  $u$



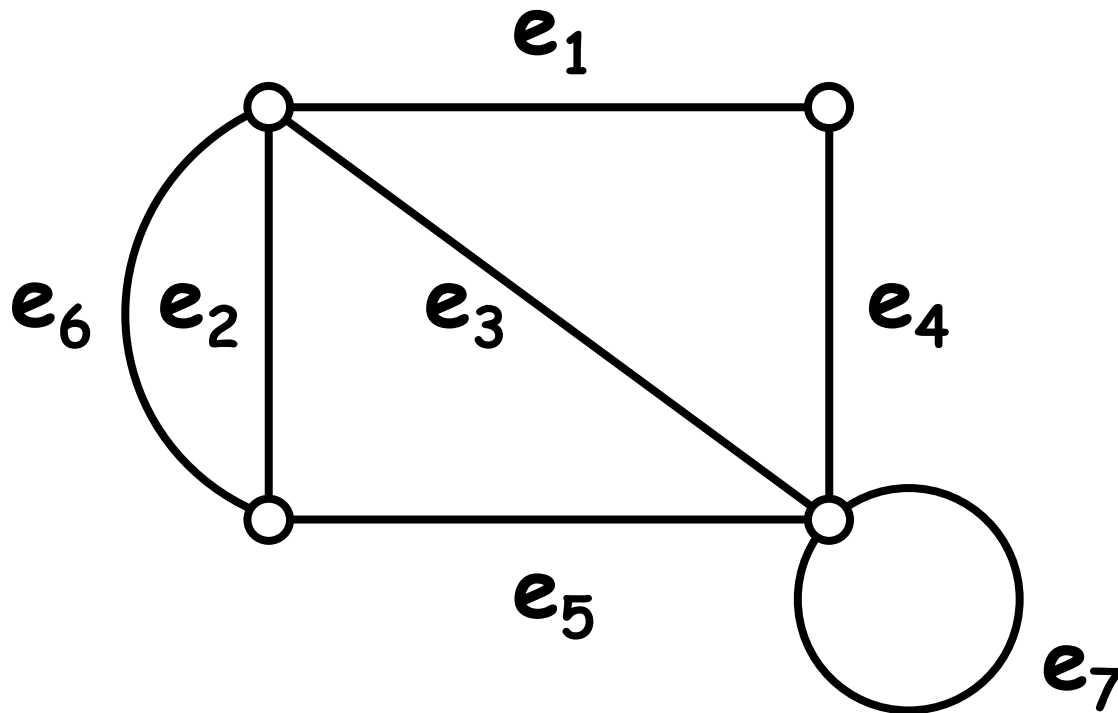
# Walk

---

- A list of links that are sequentially connected to form a continuous route
- In particular,
  - Trail** = a walk that does not go through the same **link** more than once
  - Path** = a walk that does not go through the same **node** more than once
  - Cycle** = a walk that **starts and ends at the same node** and that does not go through the same **node** on its way

# Exercise

Classify the following walks into trail, path, cycle, or other

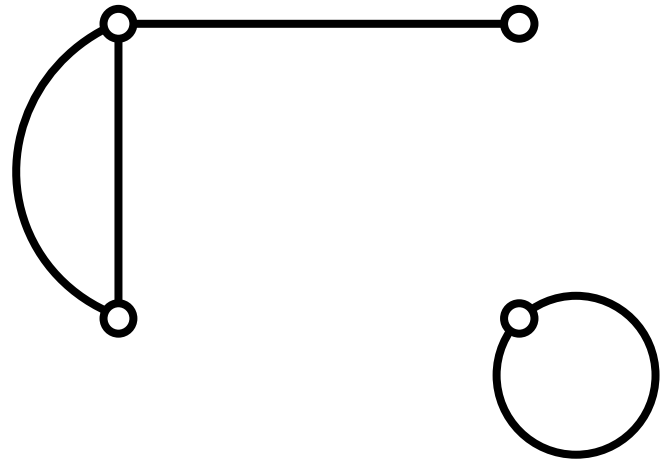
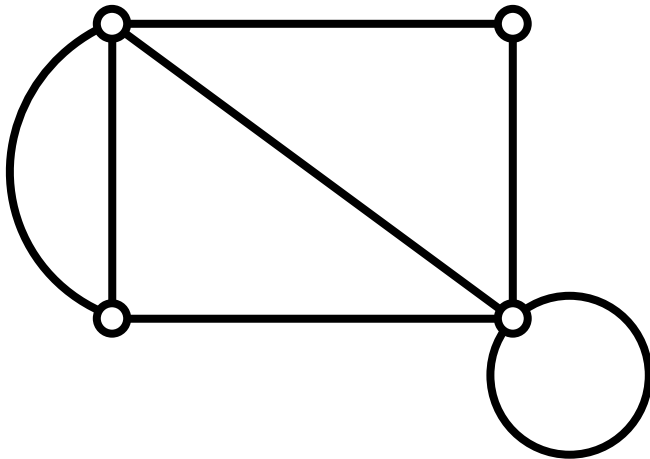


- $e_1, e_4, e_7, e_7, e_3, e_6$
- $e_6, e_5, e_3, e_1, e_4$
- $e_1, e_6, e_5$

# Connected graph

---

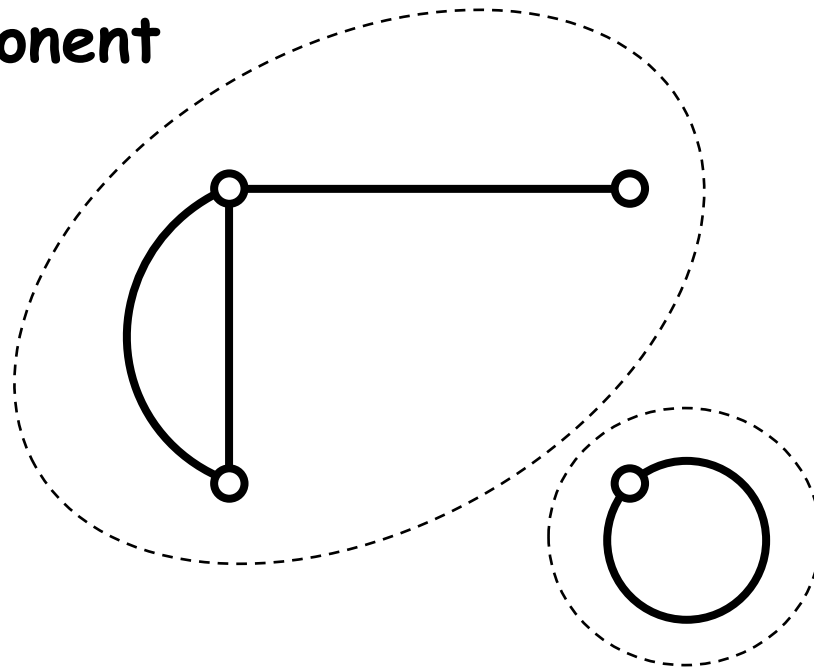
- A graph in which there is a path between any pair of nodes



# Connected components

---

Connected  
component



Connected  
component

Number of  
connected  
components  
= 2

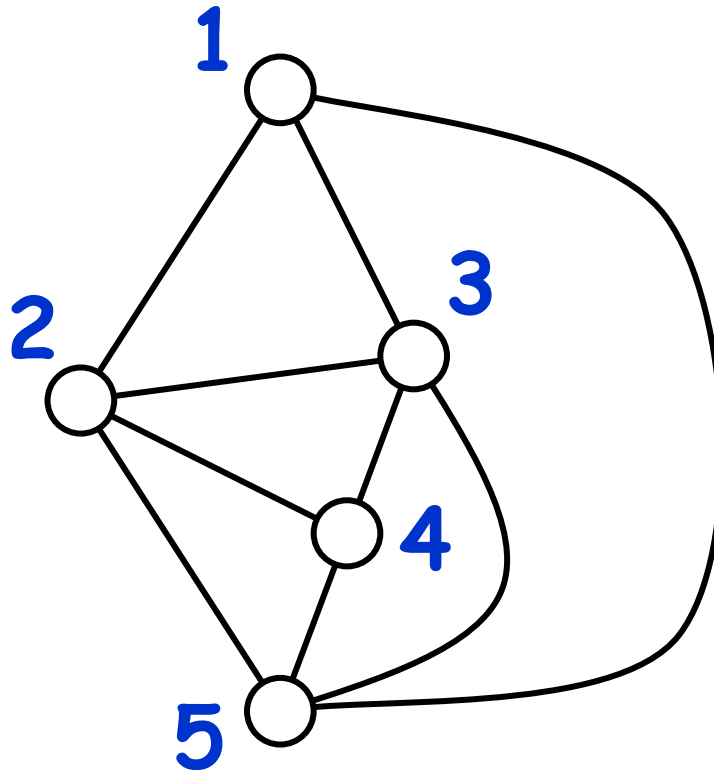
# Subgraph, induced subgraph

- A graph  $G'(V', E')$  is a **subgraph** of another graph  $G(V, E)$  if  $V'$  and  $E'$  are subsets of  $V$  and  $E$ , respectively
- $G'(V', E')$  is particularly called an **induced subgraph** of  $G(V, E)$  when  $E'$  is a set of all edges whose endpoints are both in  $V'$

# Exercise

---

- Find the subgraph of the following graph induced by nodes 1, 2 and 5

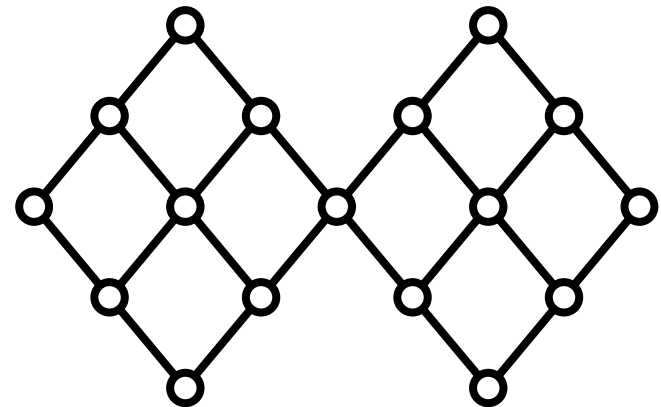
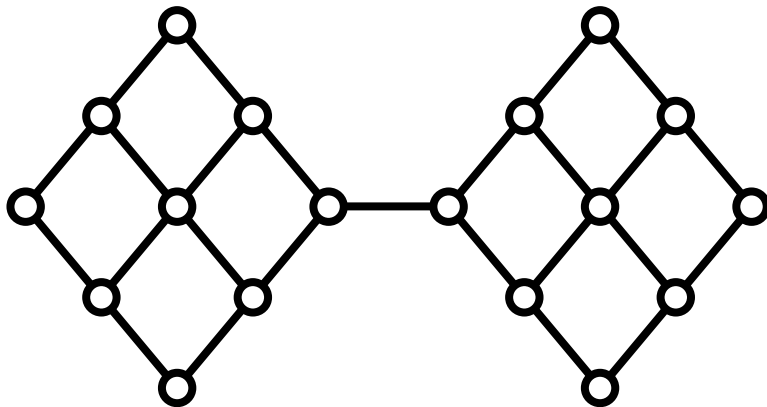




# Bridge, cut point

---

- A **bridge** is an edge whose removal would increase the # of connected components
- A **cut point** is a node whose removal would increase the # of connected components



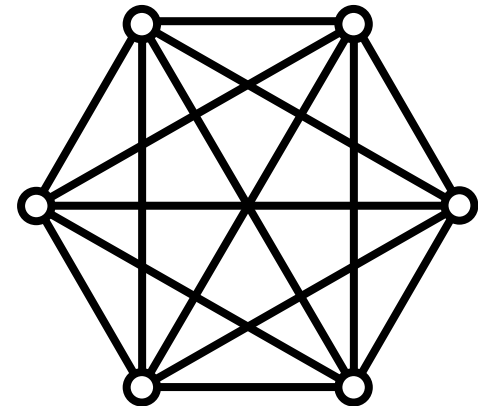
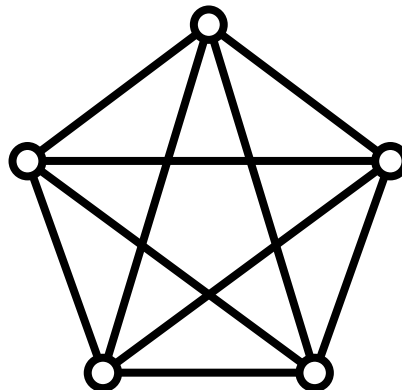
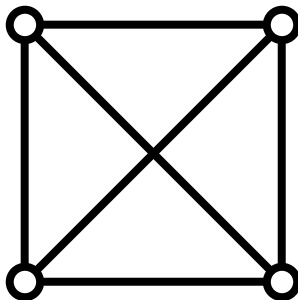
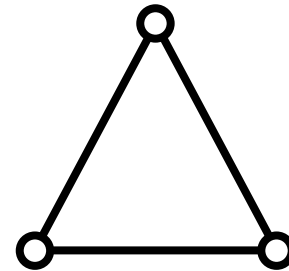
# Graphs of characteristic shapes

- Complete graph
- Regular graph
- Bipartite graph, n-partite graph
- Complete n-partite graph
- Planar graph
- Dual graph
- etc...

# Complete graph

---

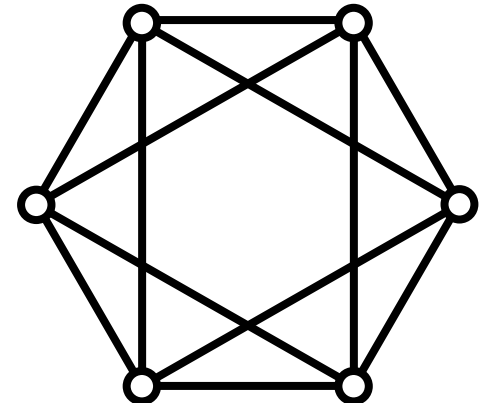
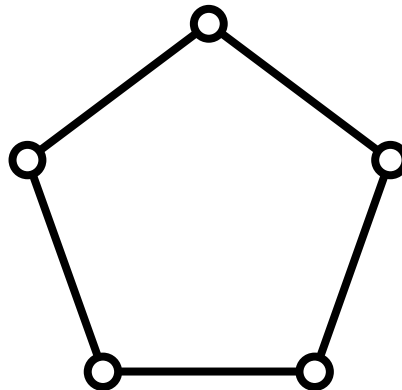
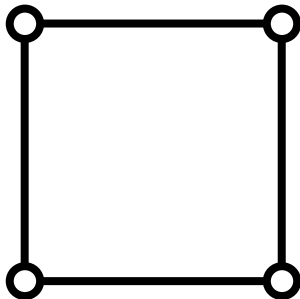
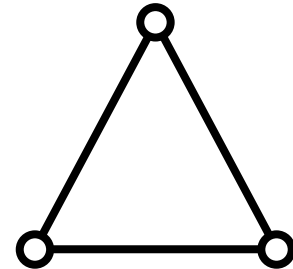
- A graph in which any pair of nodes are connected (often written as  $K_1$ ,  $K_2$ , ...)



# Regular graph

---

- A graph in which all nodes have the same degree (often called  $k$ -regular graph with degree  $k$ )



# Exercise

---

- How many links does a complete graph with  $n$  nodes have?
- How many links does a  $k$ -regular graph with  $n$  nodes have?

# Exercise

---

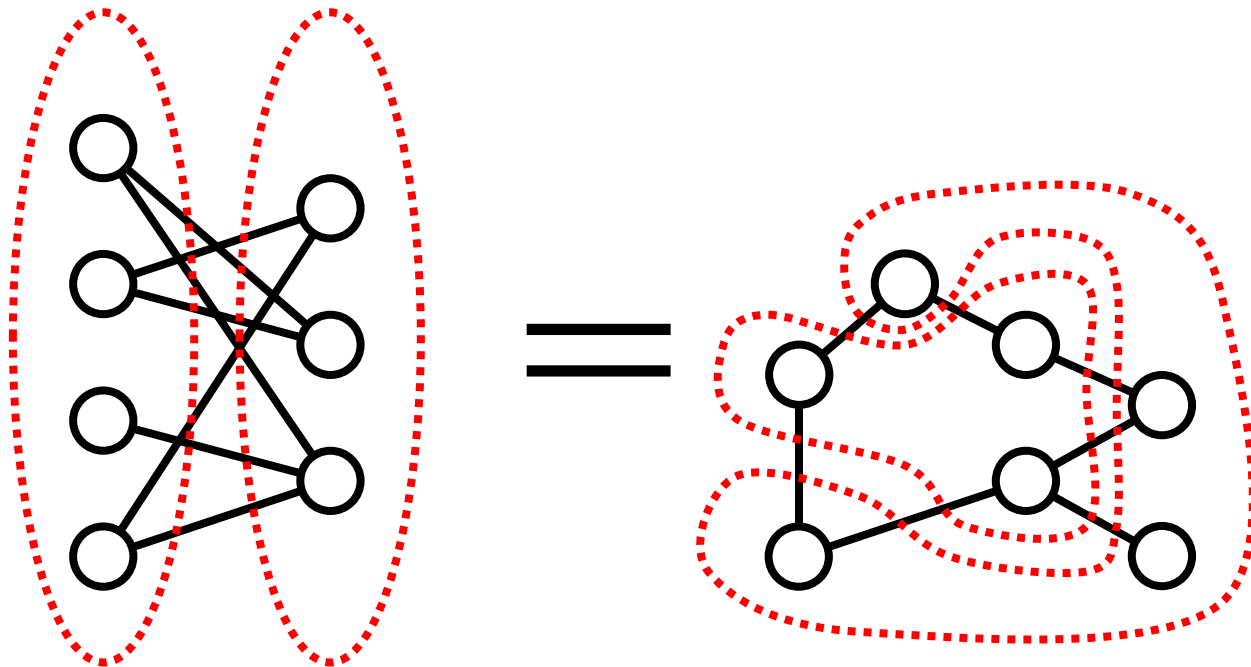
- What value can  $k$  take when a  $k$ -regular graph has 5 nodes?
- What value can  $k$  take when a  $k$ -regular graph has 6 nodes?

(Assume there is no multiple links or loops)

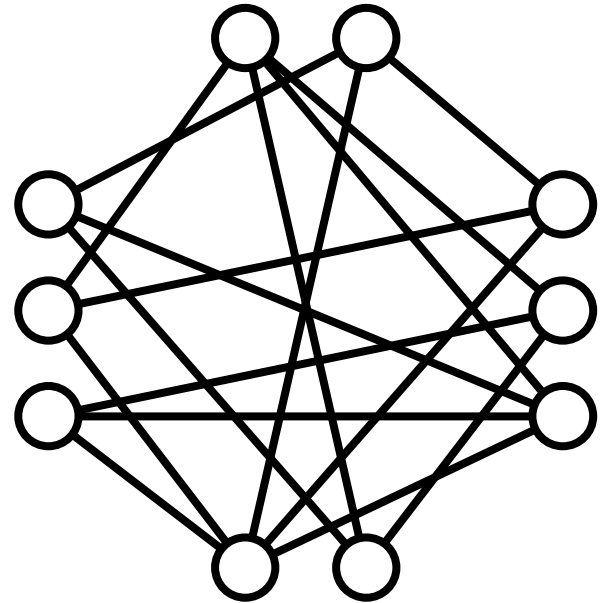
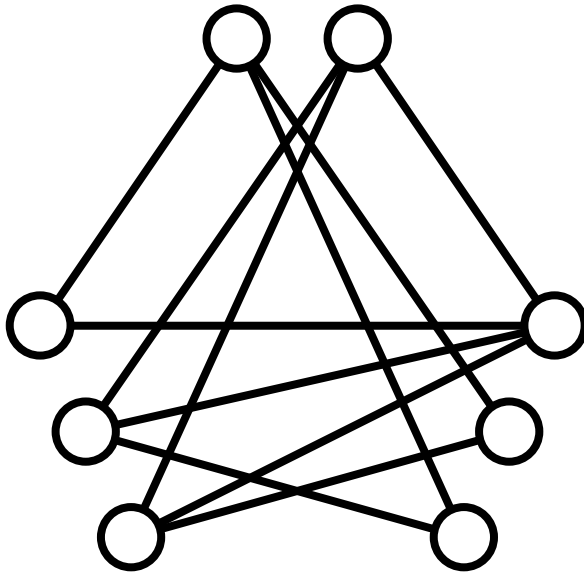
# Bipartite graph

---

- A graph whose nodes can be divided into two subsets so that no link connects nodes within the same subset



# n-partite graph

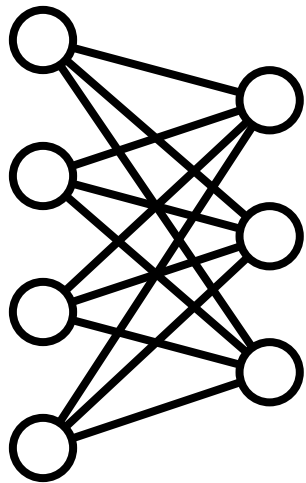




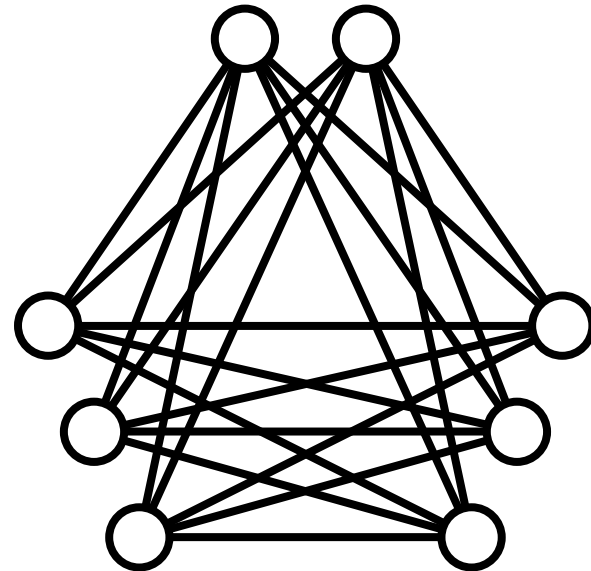
# Complete n-partite graph

---

- An n-partite graph in which each node is connected to all the other nodes that belong to different subsets



$K_{4,3}$

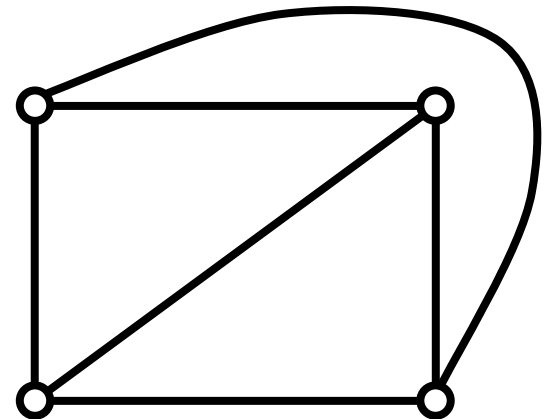
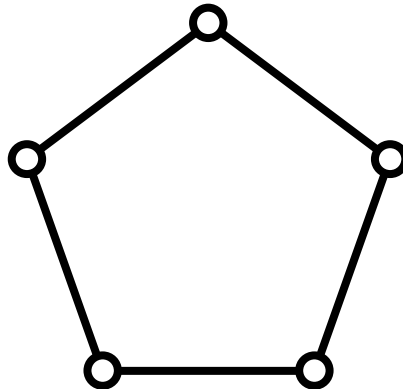
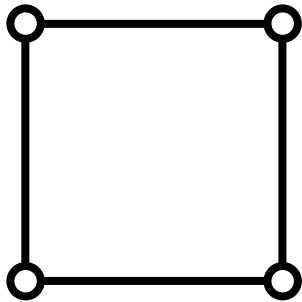


- $K_{m,n}$  represents a complete bipartite graph

# Planar graph

---

- A graph that can be graphically drawn in a two-dimensional plane with no link crossing



# Exercise

---

- Check if  $K_4$ ,  $K_5$ ,  $K_{2,2}$ ,  $K_{2,3}$ , and  $K_{3,3}$  are planar or not

# Planar graph and Euler's formula

---

- Euler's formula for any planar graph:

$$|V| - |E| + R = s + 1$$

- R: # of regions
- s: # of connected components
- For a **simple planar graph** (no multiple edges, loops, or edge crossings), each region must be encircled by at least **three** edges, i.e.

$$|E| \geq 3R / 2$$

# Exercise

---

- By using the previous two formulae, show that  $K_5$  cannot be a planar graph

# Exercise

---

- Prove Euler's formula
  - Hint: use mathematical induction for the number of edges  $|E|$

# A (rough) proof

---

- With  $|E|=0$ , the formula is true ( $|V|-0+1 = |V|+1$ )
- Assume that the formula is true with  $|E|=p$ ; When a new edge is added to the graph ( $|E| \rightarrow p+1$ ):
  - If the added edge divides a region into two, it must be part of a cycle and thus included in a connected graph (i.e. # of connected components does not change):  
 $|V|-(p+1)+R+1 = s+1$  (the formula still holds)
  - If the added edge does not divide any region, it must be a bridge between two originally disconnected components (i.e. # of connected components decreases by 1):  
 $|V|-(p+1)+R = (s-1)+1$  (the formula still holds)

# Dual graph

---

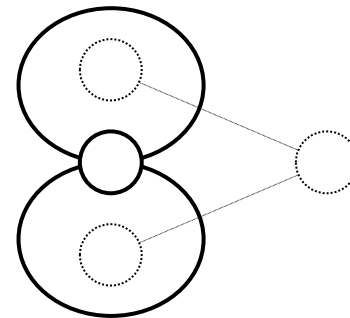
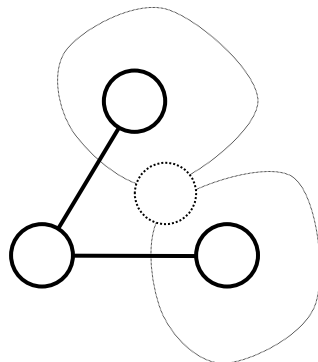
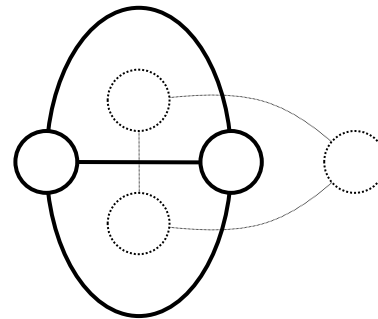
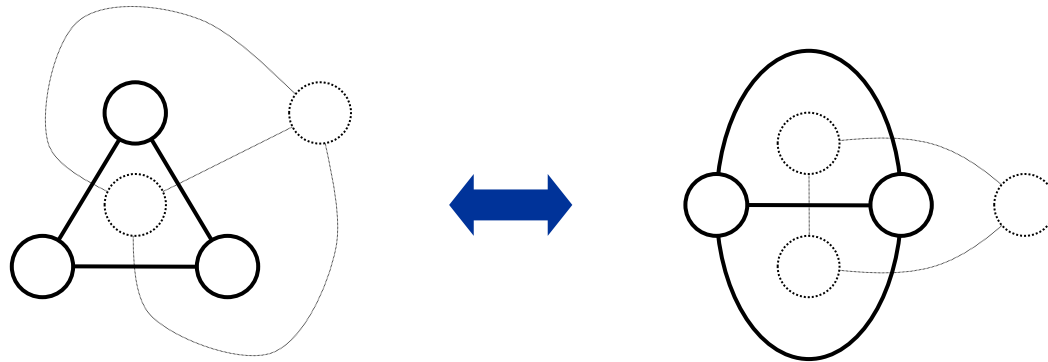
- For any planar graph, one can obtain a new planar graph (**dual graph**) by
  - assigning a node to each separate region (including background), and
  - connect nodes if the corresponding regions are adjacent across one edge

(A dual of a dual will be the original planar graph)



# Examples of dual graphs

---



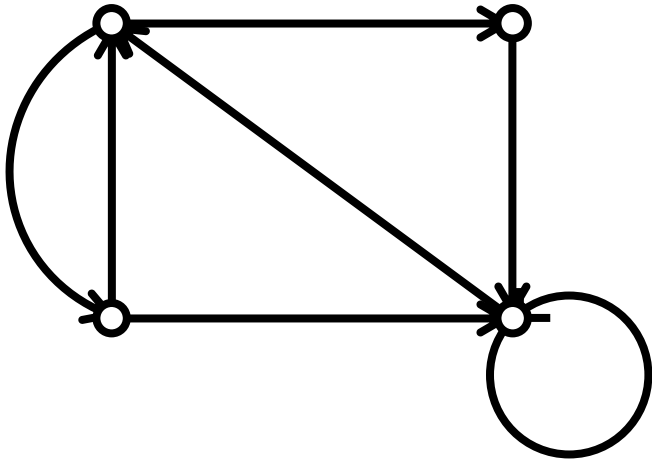
# Exercise

---

- Represent the numbers of nodes and edges of a dual graph of  $G$  using the numbers of nodes and edges of  $G$   
(Assume that  $G$  is a connected graph)
- By using the result above, obtain all the complete graphs whose dual graphs are identical to themselves

# Directed graph

---

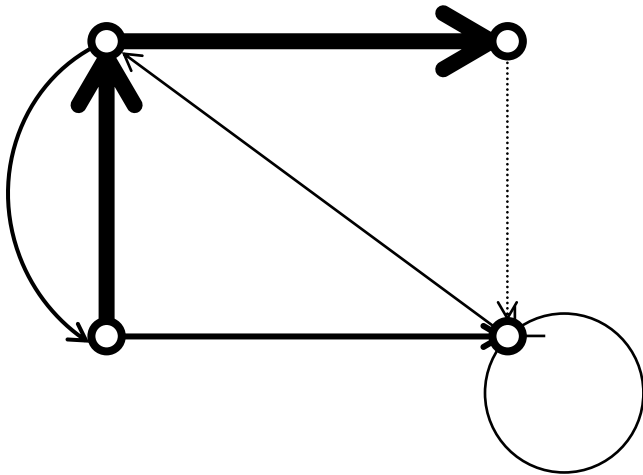


- Each link is directed
- Direction represents either order of relationship or accessibility between nodes

E.g. genealogy

# Weighted directed graph

---



- Most general version of graphs
- Both weight and direction is assigned to each link

E.g. traffic network

# Levels of connectivity

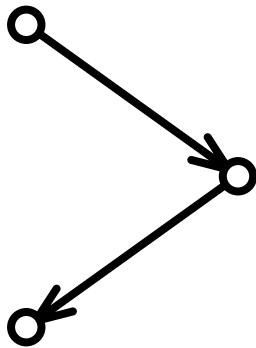
---

- For every pair of nodes in a directed graph:
  - if there is a semi-path (path ignoring directions) between them, the graph is **weakly connected**
  - if there is a path from one to the other, the graph is **unilaterally connected**
  - if there are paths from either direction between them, the graph is **strongly connected**

# Exercise

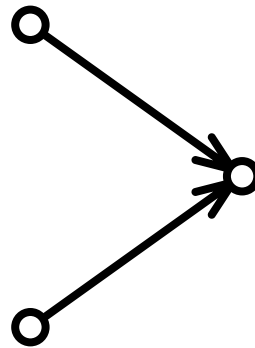
---

- Determine the connectivity of the graphs below



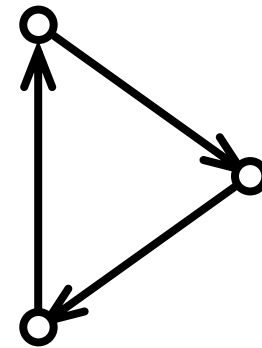
**Unilateral**

There is always  
some causal  
relationship  
between any pair



**Weak**

Some pairs are  
completely  
independent  
from each other



**Strong**

There is always  
bidirectional  
influences  
between any pair

# General structure of large directed networks

---

