UNIT-1

What is social web?

- The social web encompasses how websites and software are designed and developed in order to support and foster social interaction.

- Social web refers to web services, structures and interfaces that support social interactions among humans.

What is Social network Analysis

- A set of relational methods for systematically understanding, identifying, mapping, and measuring the relationships and flow among actors.
- The Actors can be:
  - ✓ people,
  - ✓ groups,
  - ✓ organizations,
  - ✓ computers,
  - ✓ URLs, etc.

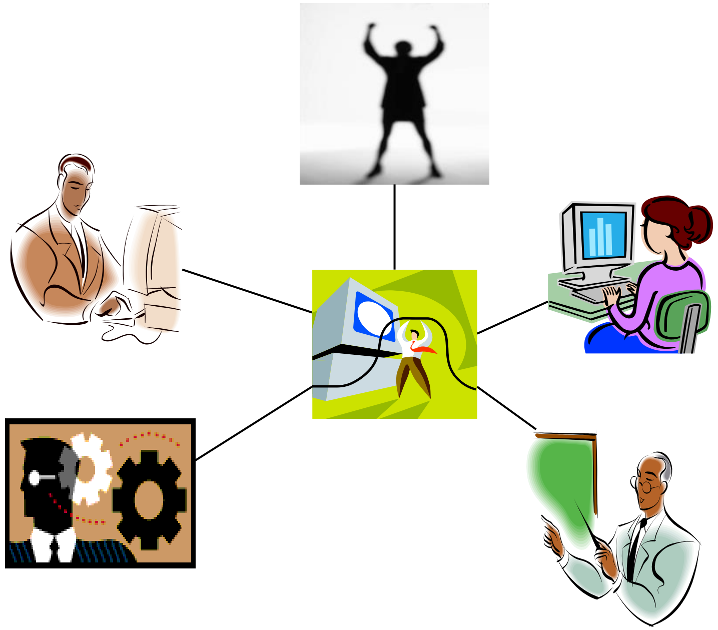
- SNA provides both a visual and a mathematical analysis of human relationships.

What is a Social Network ?

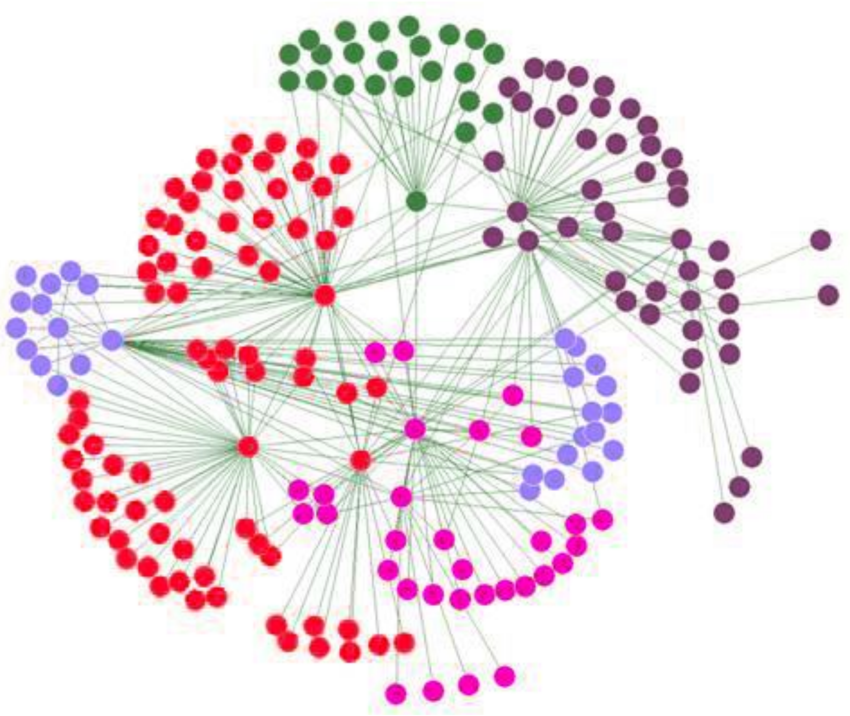
A social structure made up of individuals (or organizations) called "nodes", which are tied (connected) by one or more specific types of interdependency, such as friendship, common interest

-Nodes

- People
- Organizations
- Both (multi-modal network)



- **SNA** views social relationships in terms of network theory consisting of *nodes* and *ties* (also called *edges*, *links* or *connections*).

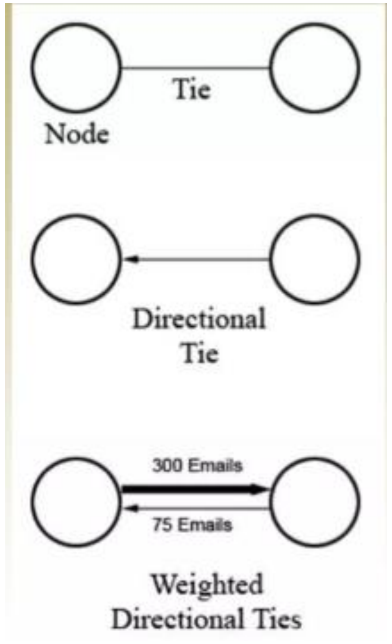


Basic concepts of networks (graphs)

- A **node or vertex** is an individual unit (actor) in the graph or system.

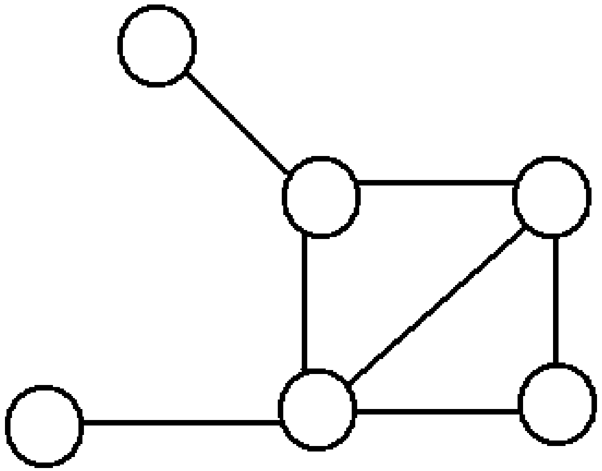
- A **graph or system or network** is a set of units that may be (but are not necessarily) connected to each other.

- Relations (lines, edges, ties) between pair of actors.
  - Directed or undirected
  - Binary or valued

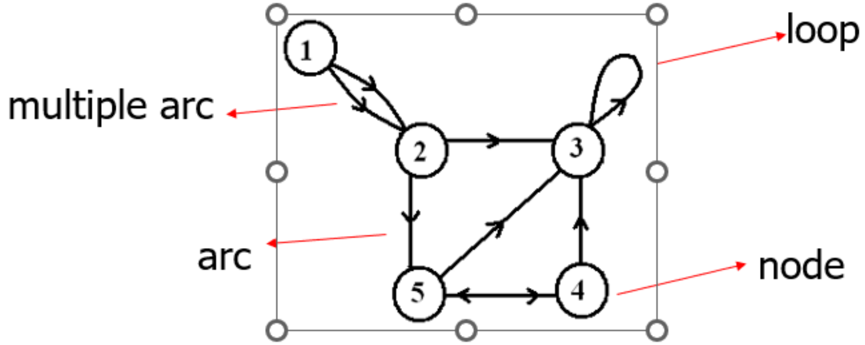


- Two actors are connected by social relationship.
  - ✓Kinship [ex: Son of]
  - ✓Role based [ex: Friend of, teacher of]
  - ✓Affective [ex: likes]
  - ✓Interactions [ex: gives advice, talks to]
  - ✓Affiliations [ex: belongs to same organization]

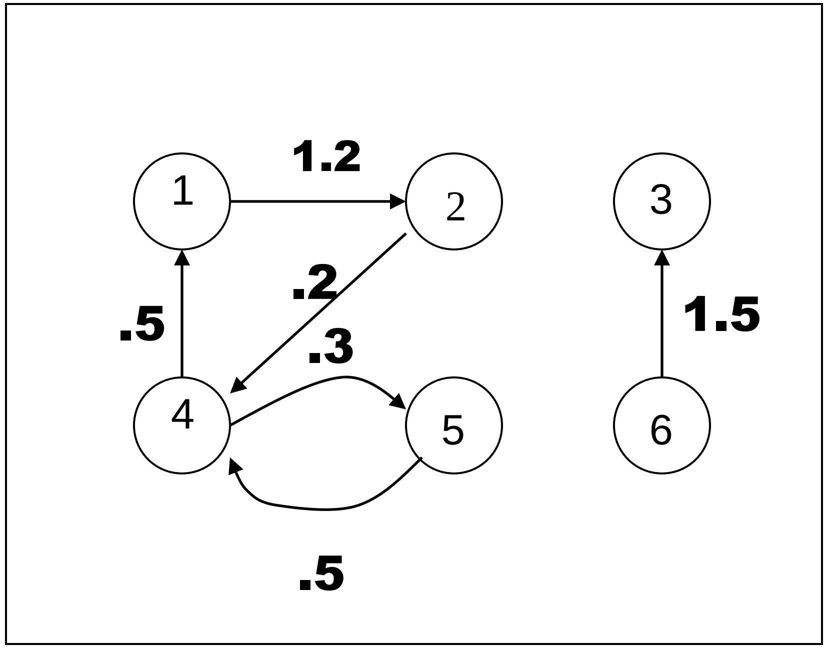
- ***Simple graphs*** are graphs without multiple edges or self-loops.

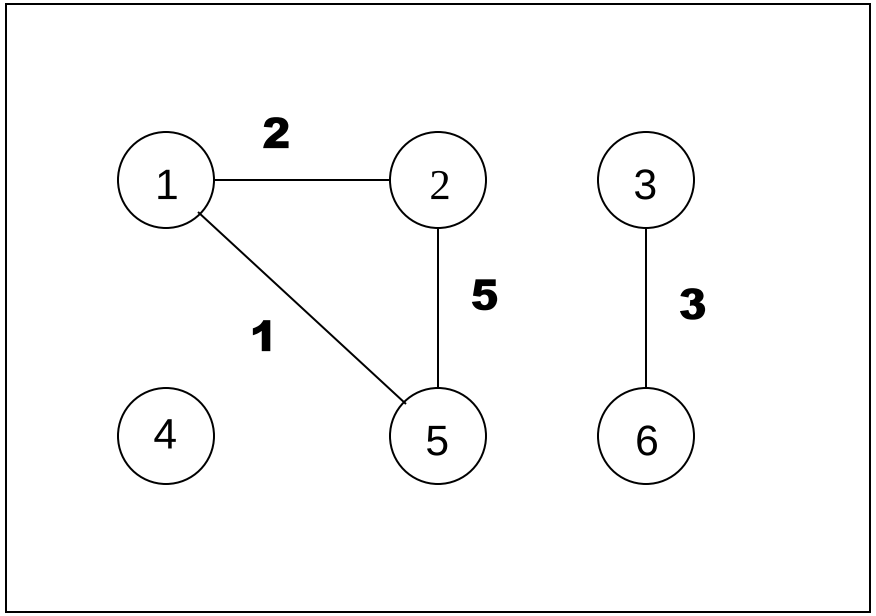


- Directed Graph (digraph)
  - Edges have directions
    - An edge is an *ordered* pair of nodes

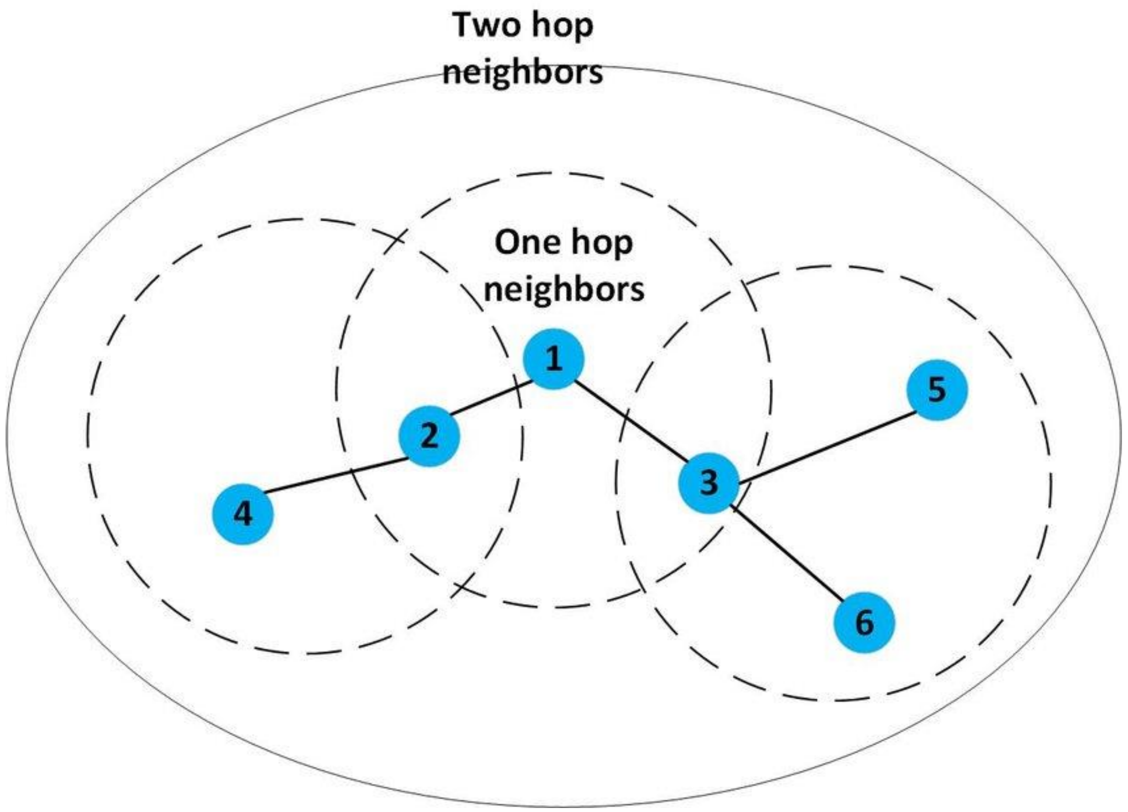


- **Weighted graphs:**
- Is a graph for which each edge has an associated ***weight***, usually given by a ***weight function*** *w: E* $\rightarrow$ **R**.





- A **neighborhood** N for a vertex or node is the set of its immediately connected nodes.
  - **Neighbor:** A vertex u is a neighbor of (or equivalently adjacent to) a vertex v in a graph G = (V, E) if there is an edge {u, v} ∈ E. For a directed graph a vertex u is an in-neighbor of a vertex v if (u, v) ∈ E and an out-neighbor if (v, u) ∈ E. We also say two edges or arcs are neighbors if they share a vertex.



- **Incident: A**n edge is incident on a vertex if the vertex is one of its endpoints. Similarly a vertex is incident on an edge if it is one of the endpoints of the edge.

- **Degree**: The degree k_i of a vertex or node is the number of other nodes in its neighborhood.

- **Paths:** A path in a graph is a sequence of adjacent vertices.

- **Reachability and connectivity:** A vertex v is reachable from a vertex u in G if there is a path starting at v and ending at u in G.

- **Cycles:** In a directed graph a cycle is a path that starts and ends at the same vertex. A cycle can have length one (i.e. a self loop).

  A simple cycle is a cycle that has no repeated vertices other than the start and end vertices being the same.

- **Trees and forests:** An undirected graph with no cycles is a forest and if it is connected it is called a tree.
- A **rooted tree** is a tree with one vertex designated as the root. For a directed graph the edges are typically all directed toward the root or away from the root.
- **Directed acyclic graphs**. A directed graph with no cycles is a directed acyclic graph (DAG).
- Distance.

Representation of Graphs

- **Adjacency matrix:**

An $n \times n$ matrix of binary values in which location (i, j) is 1 if $(i, j) \in E$ and 0 otherwise.

Adjacency list:

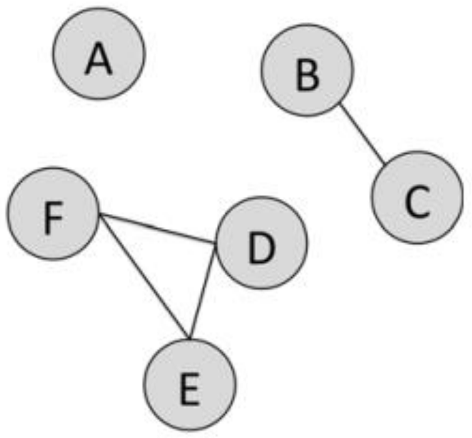
An array A of length n where each entry A[i] contains a pointer to a linked list of all the out-neighbors of vertex i. In an undirected graph with edge {u, v} the edge will appear in the adjacency list for both u and v.

Adjacency array:

Adjacency array keeps the neighbors of all vertices, one after another, in an array adj; and separately, keeps an array of indices that tell us where in the adj array to look for the neighbors of each vertex

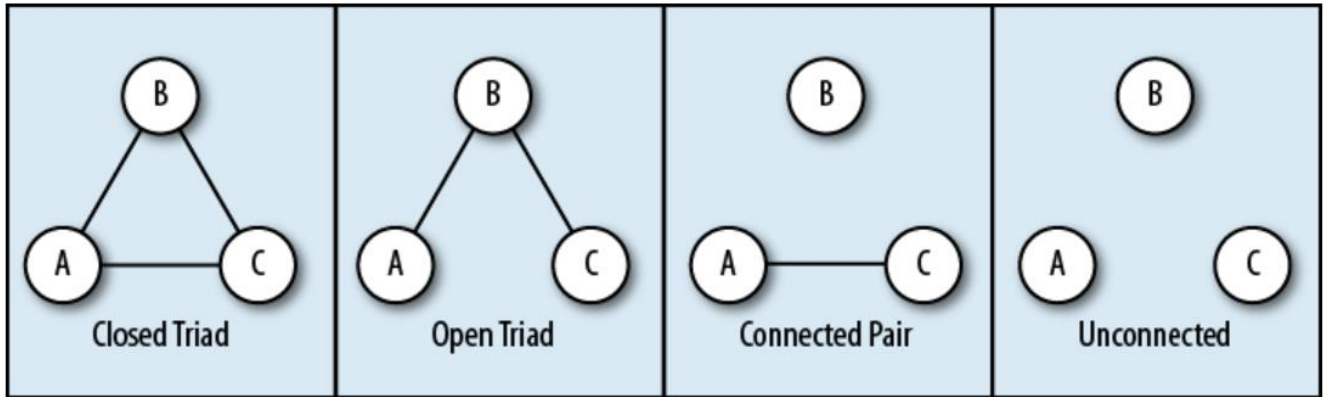
Basic network structures and properties

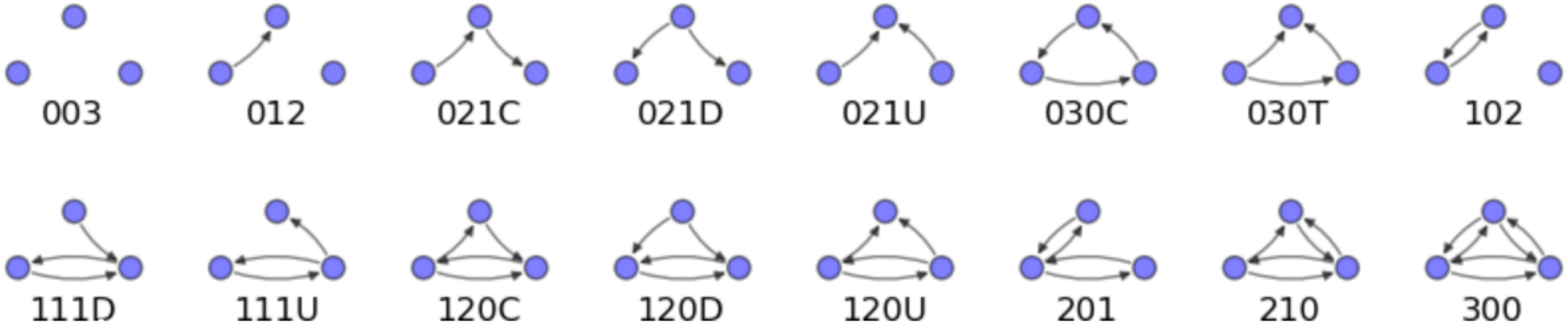
- **Subnetworks**
  - ✓Subset of the nodes and edges in a graph.
  - ✓Is a subset of the nodes of a network, and all of the edges linking these nodes
  - ✓Components are portions of the network that are disconnected from each other.
  - ✓Different types:
    - ➢ Singletons
    - ➢ Dyad
    - ➢ triad



A social network with a singleton, dyad, and triad

- Different types of triads.



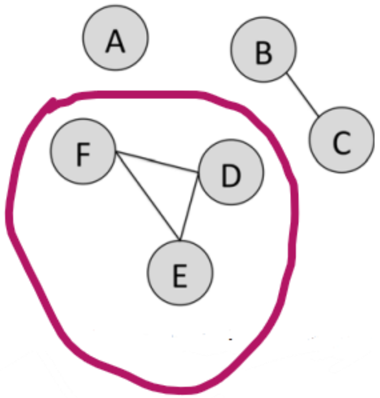


Triad census in directed networks

Fig. credit : https://www.oreilly.com/library/view/social-network-analysis/9781449311377/ch04.html

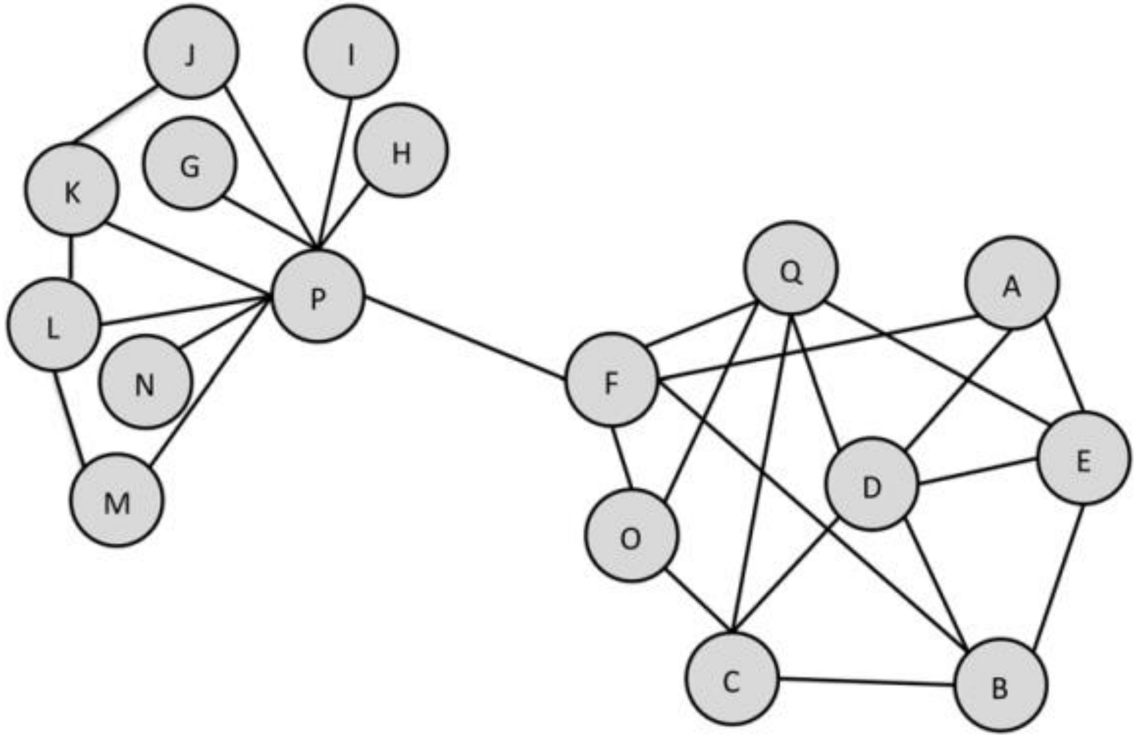
Basic network structures and properties

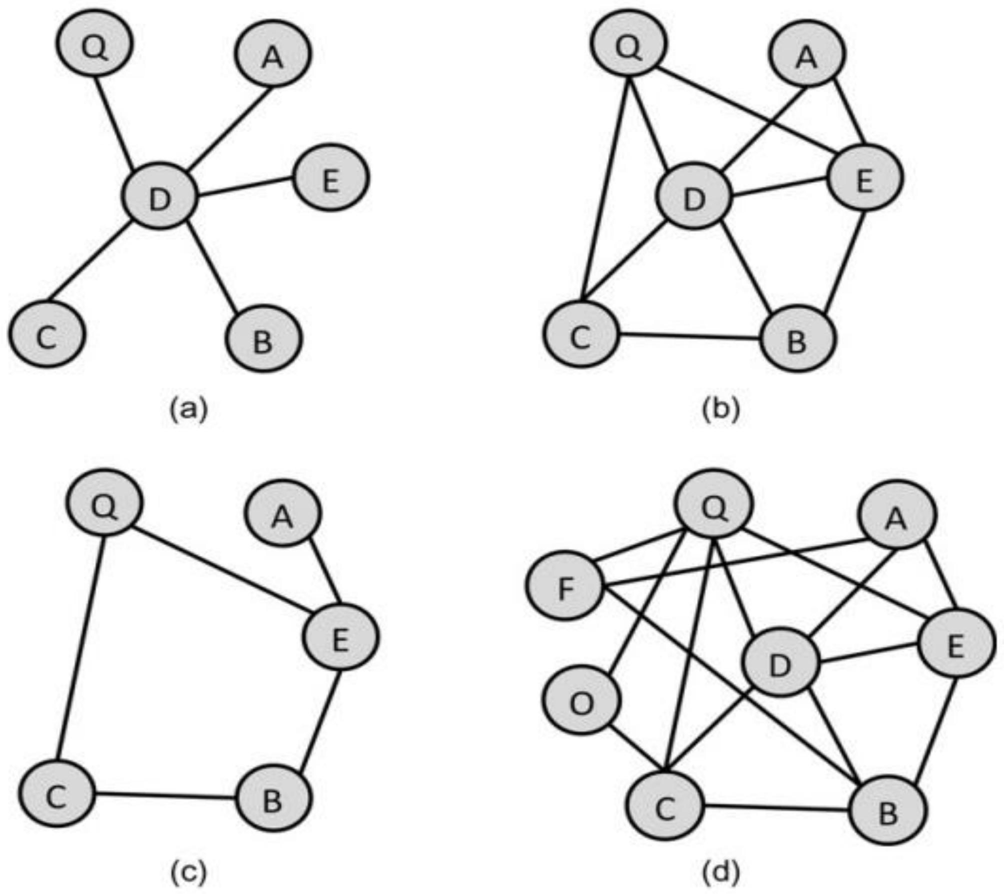
- **Cliques**
  - Groups of nodes of any size have properties that are interesting.
  - When all nodes in a group are connected to one another, it is called as **clique.**
  - A clique is defined as a **maximal complete subgraph** of a given graph—
  - Ex: a group of people where everybody is connected directly to everyone else.



- **Clusters**
  - A collection of nodes on the basis of similarity and dissimilarity between them.

- **Egocentric networks**
- Ego-centric networks (or shortened to “ego” networks) are a particular type of network which specifically maps the connections of and from the perspective of a single person (an “ego”).
- This is a network we pull out by selecting a node and all of its connections.
    - 1-degree egocentric network
    - 1.5-degree egocentric network
    - 1.5 egocentric network excluding center
    - 2-degree egocentric network


J
I
G
H
K
P
L
N
M
Q
A
F
E
D
O
C
B



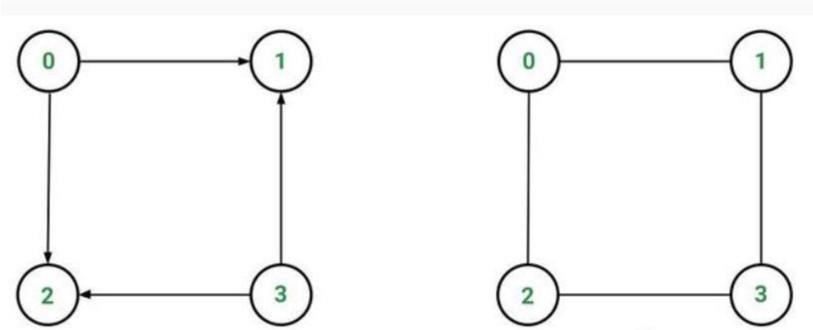
(a) The 1-degree egocentric network of D, (b) the 1.5-degree egocentric network of D, (c) the 1.5 egocentric network of D with D excluded, and (d) the 2-degree egocentric network of D.

Paths and connectedness

- **Path**
  - A path is a series of nodes that can be traversed following edges between them.
  - Length of a path= Number of edges in the path
  - **Shortest paths** is important measure in network analysis and are sometimes called **geodesic distances.**

Connectedness

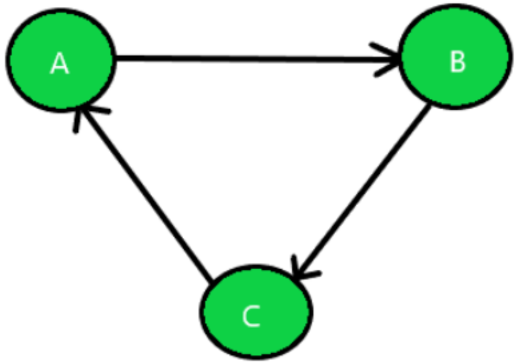
- Two nodes in a graph are called connected if there is a path between them in the network.
- An entire graph is called connected if all pairs of nodes are connected.
- ***Strongly Connected:*** *A graph is said to be* ***strongly connected*** *if every pair of vertices(u, v) in the graph contains a path between each other.*
- If a path cannot be found between all pairs of nodes using the direction of the edges, but paths can be found if the directed edges are treated as undirected, then the graph is called **weakly connected.**
- If a graph is not connected, it may have subgraphs that are connected. These are called **connected components**.



G

G’

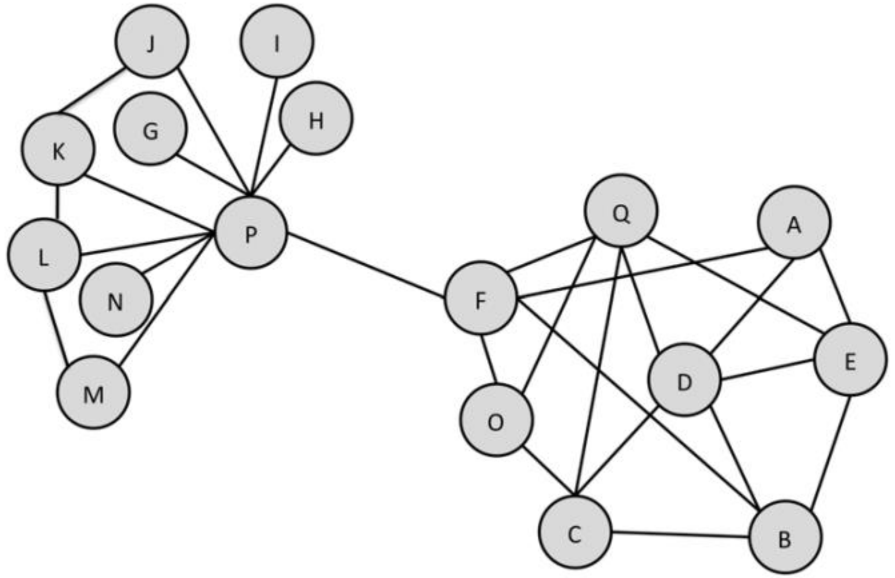
Graph G is weakly connected.



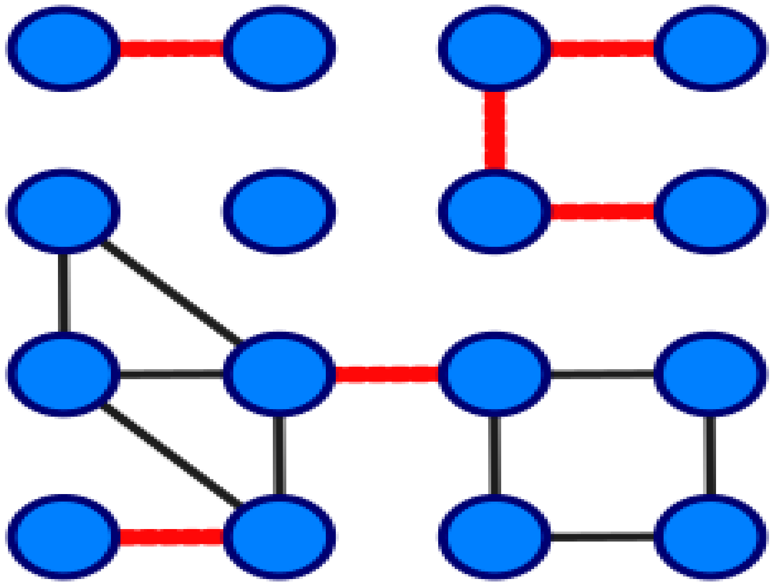
Strongly connected Graph

Bridges and Hubs

- A **bridge** is an edge in graph that, if removed, will increase the number of connected components in a graph.
- A hub is the most connected nodes in the network.



Edge PF is a bridge
Node P is the hub



A graph with 16 vertices and six bridges (highlighted in red)

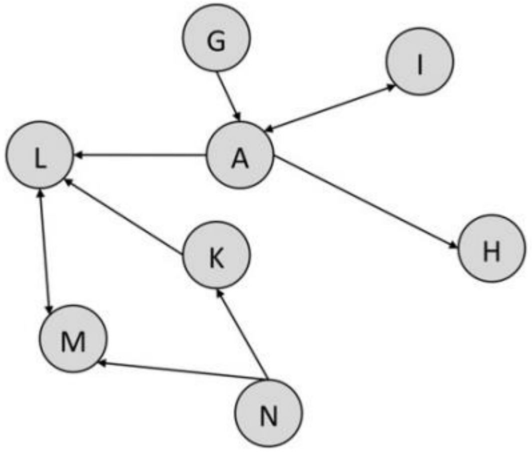
Fig Credit: https://en.wikipedia.org/wiki/Bridge_(graph_theory)

Network Structure and Measures

- Tool for understanding connected data structures (such as graphs, or networks.)

Describing Nodes and Edges:

- Measures which describe how nodes are connected each other and network as a whole:
  - **Degree of a node.**
    - ✓ Undirected graph number of edges connected to it.
    - ✓ Directed graph-→ indegree, outdegree.

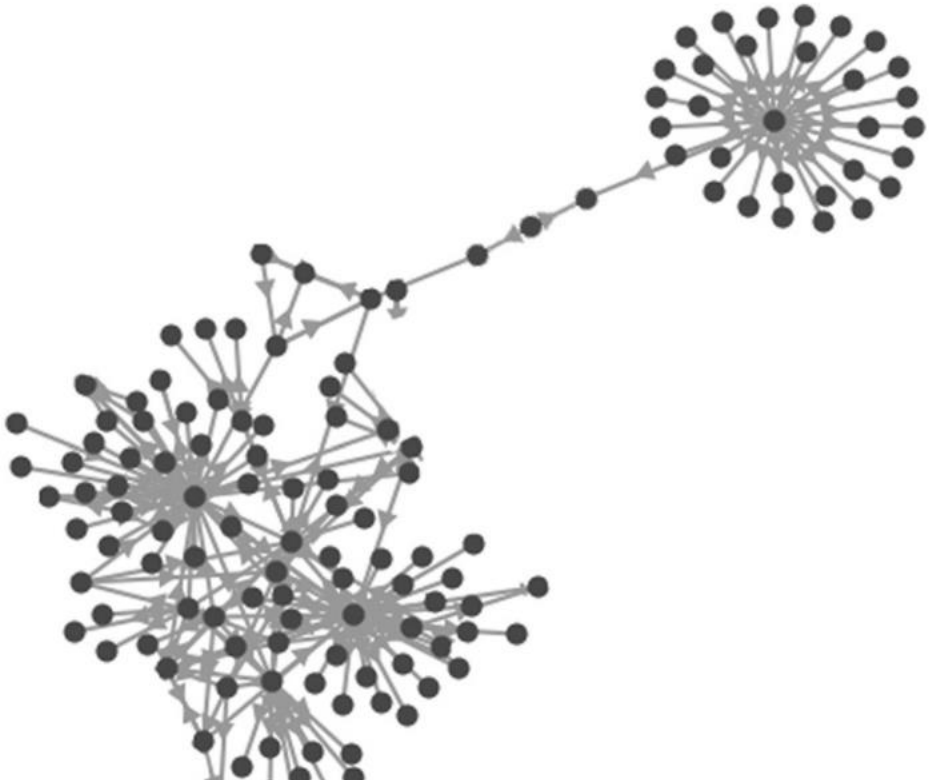


Sample undirected graph: Find indegree and outdegree of each node

- **Centrality**
  - ✓ Centrality of a node is an estimate of its importance in the network.
  - ✓ “central” measure may vary depending on the context.
  - ✓ The interpretation of the centrality measures is left to a human analyst.
  - ✓ Different ways to measure centrality of a node.
    - Degree centrality
    - Closeness centrality
    - Betweenness centrality
    - Eigenvector centrality.

➢Degree Centrality

- ✓Degree centrality is the *number of edges connected to a given node*.
- ✓Ex: In a social network, this might mean the number of friends an individual has.
- ✓**When to use it:** For finding very connected individuals, popular individuals, individuals who are likely to hold most information or individuals who can quickly connect with the wider network.



- **Closeness centrality**
  - ✓ Indicates how close a node is to all other nodes in the network.
  - ✓ It is calculated as the average of the shortest path length from the node to every other node in the network.
  - ✓ Lower values indicate more central nodes. [If reciprocals are taken then higher values indicate more central nodes]
  - ✓ **When to use it:** For finding the individuals who are best placed to influence the entire network most quickly.

Table 3.1 The Shortest Path Lengths from D to each Other Node in the Network

Node	Shortest Path from D
A	3 (D C B A)
B	2
C	1
E	1
F	2
G	2
H	1

$(3 + 2 + 1 + 1 + 2 + 2 + 1) \div 7 = 12 \div 7 = \mathbf{1.71}.$

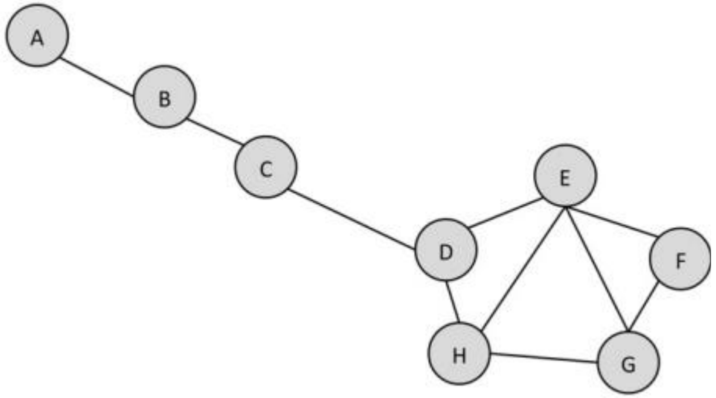
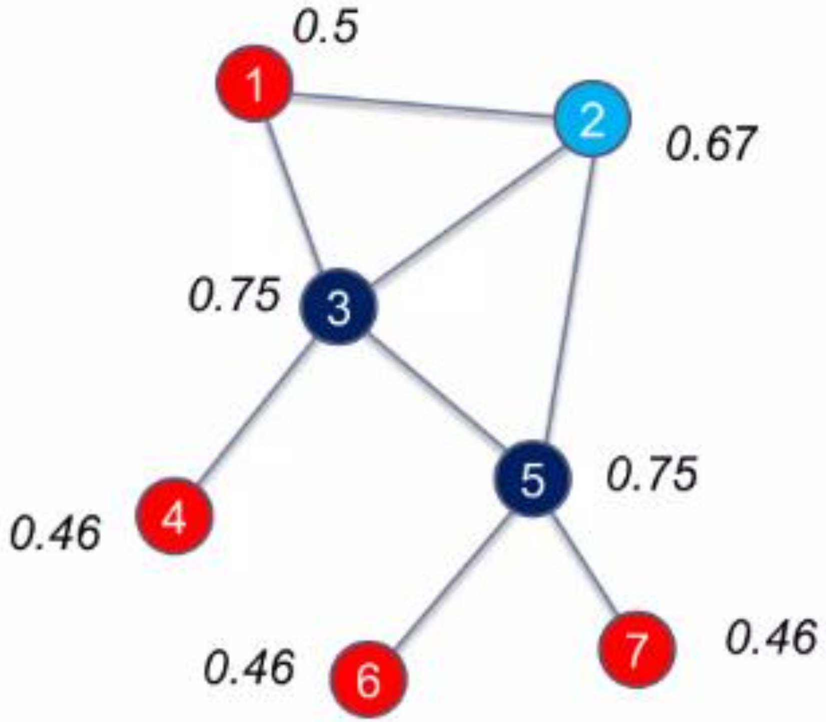


Table 3.2 The Shortest Path Length from node A to Every Other Node in the Network

Node	Shortest Path from A
B	1
C	2
D	3
E	4
F	5
G	5
H	4

$(1 + 2 + 3 + 4 + 5 + 5 + 4) \div 7 = 24 \div 7 = \mathbf{3.43}.$



0.5
1
2
0.67
0.75
3
5
0.75
0.46
4
0.46
6
7
0.46

➢ Betweenness centrality

- ✓ Betweenness centrality measures how important a node is to the shortest paths through the network.
- ✓ Defined as *the number of shortest paths that go through a given node*.
- ✓ Nodes with high betweenness is influential in a network.
- ✓ They capture the most amount of information flowing through the network because the information tends to flow through them.
- ✓ **When to use it:** For finding the individuals who influence the flow around a system.

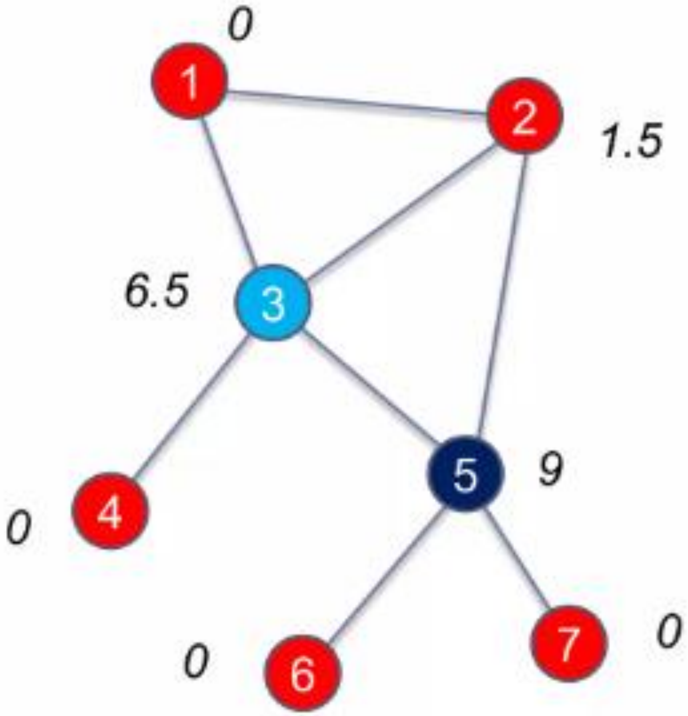
✓ To compute betweenness of a node N

- Select a pair of nodes and find all the shortest paths between those nodes.
- Then compute the fraction of those shortest paths that include node N.
- Repeat this process for every pair of nodes in the network.
- Add up the fractions computed, and this is the betweenness centrality for node N.

The betweenness centrality of a node v is given by the expression:

$$g(v) = \sum_{s \neq v \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}}$$

where σ_{st} is the total number of shortest paths from node s to node t and $\sigma_{st}(v)$ is the number of those paths that pass through v (not where v is an end point).[2]


0
1
2
1.5
6.5
3
5
9
0
4
0
6
7
0

➢ Eigenvector centrality

- ✓ Eigenvector centrality measures a node's importance while giving consideration to the importance of its neighbors.
- ✓ **When to use it:** Eigen Centrality is a good 'all-round' centrality score, handy for understanding human social networks,
- ✓ Ex:a node with 300 relatively unpopular friends on Facebook would have lower eigenvector centrality than someone with 300 very popular friends.
- ✓ It is determined by performing a matrix calculation to determine what is called the principal eigenvector using the adjacency matrix.

Describing networks

- A number of measures can be used to describe the structure of a network as a whole.

➢ **Degree distribution**

  ✓ Degree is used to describe individual nodes.

  ✓ To get an idea of the degree for all the nodes in the network, **degree distribution** is used.

  ✓ This shows how many nodes have each possible degree.

  - **Steps:**
    - Calculate the degree for each node in the network.
    - Count how many nodes have each degree. This is totaled for each degree, including those for which there are no nodes with that count.
    - The most common way to show a degree distribution is in a bar graph. The x-axis has the degrees in ascending order, and the Y-axis indicates how many nodes have a given-degree

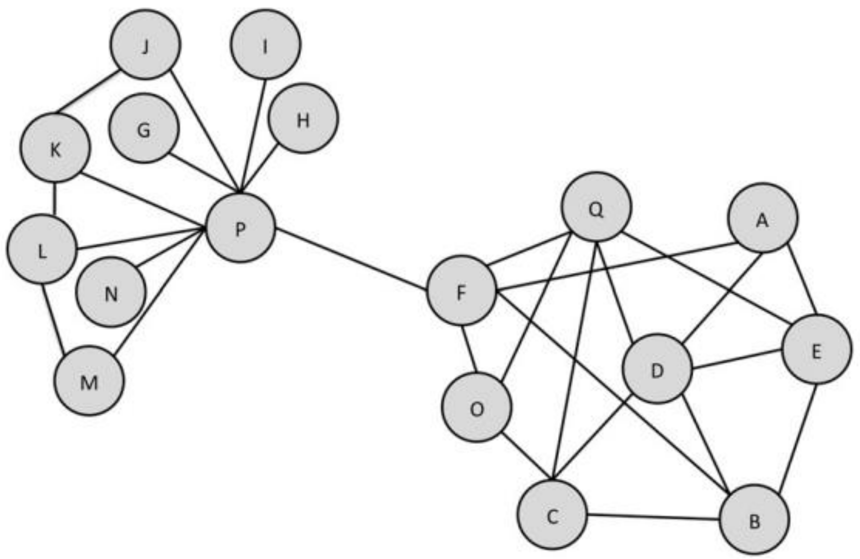
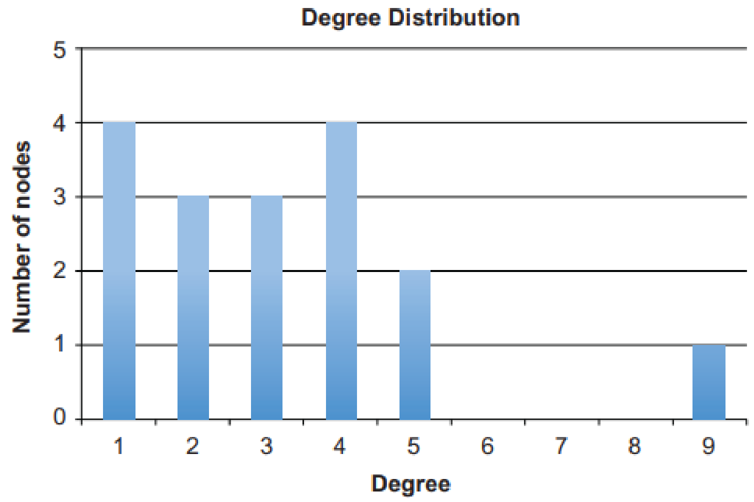


Table 3.3 Degrees for each Node Shown in

Node	Degree
A	3
B	4
C	4
D	5
E	4
F	4
G	1
H	1
I	1
J	2
K	3
L	3
M	2
N	1
O	2
P	9
Q	5

Table 3.4 The Degree Distribution for the Network The First Column Shows the Degree, and the Second Column Shows How Many Nodes have that Degree

Degree	Number of Nodes
1	4
2	3
3	3
4	4
5	2
6	0
7	0
8	0
9	1

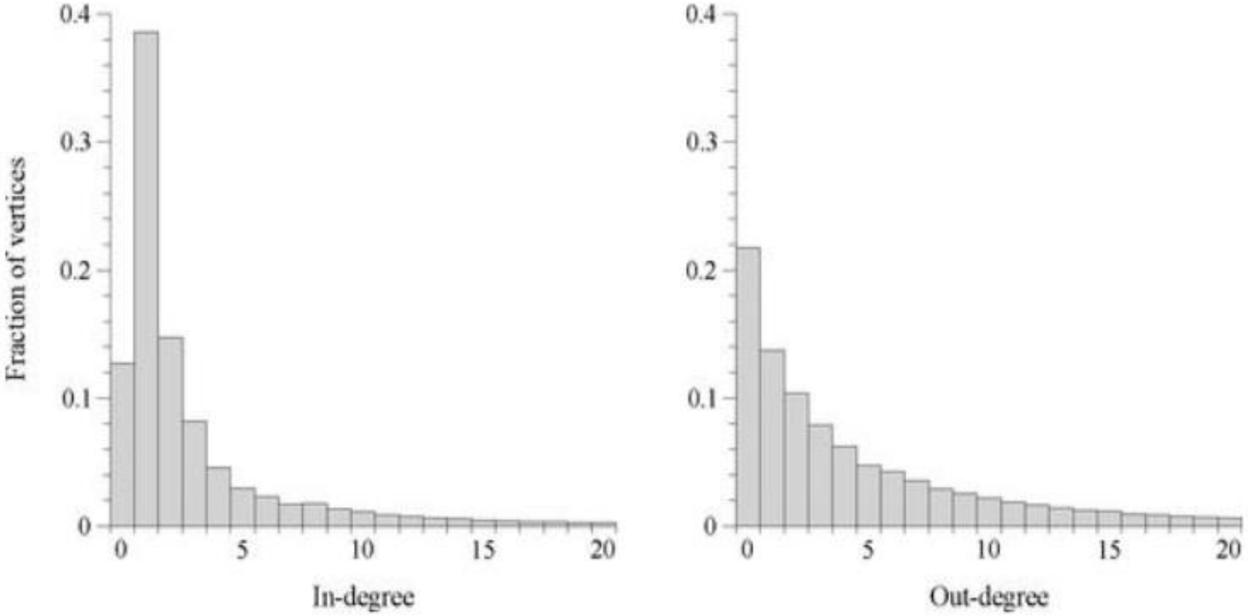


The degree distribution for the graph

The degree distribution $P(k)$ of a network is then defined to be the fraction of nodes in the network with degree k. Thus if there are n nodes in total in a network and n_k of them have degree k, .

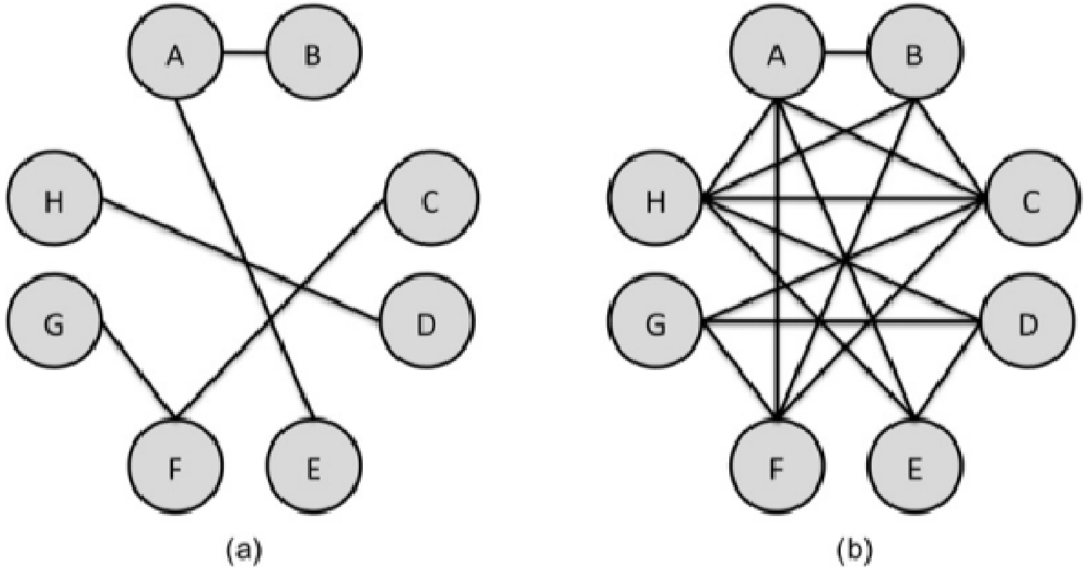
$$P(k) = \frac{n_k}{n}$$

For directed networks we have both in- and out-degree distributions



- **Density**
  - ✓Density refers to the "connections" between participants.
  - ✓Density describes how connected a network is.
  - ✓*More formally, it is a statistic comparing the number of edges that exist in a network to the number of edges that could possibly exist.*
  - ✓Density is defined as the number of connections a participant has, divided by the total possible connections a participant could have.
  - ✓For example, if there are 20 people participating, each person could potentially connect to 19 other people.
  - ✓Formula to calculate

density: number of edges ÷ number of possible edge



Network (a) on the left has fewer edges than network (b) on the right. Since they both have the same number of nodes and thus the same number of possible edges, network (b) is more dense.

- For Directed networks, the number of possible edges in a graph with n nodes is: **PC= n × (n- 1)**

- For undirected networks **PC= [n × (n- 1)] /2**

Network Density:

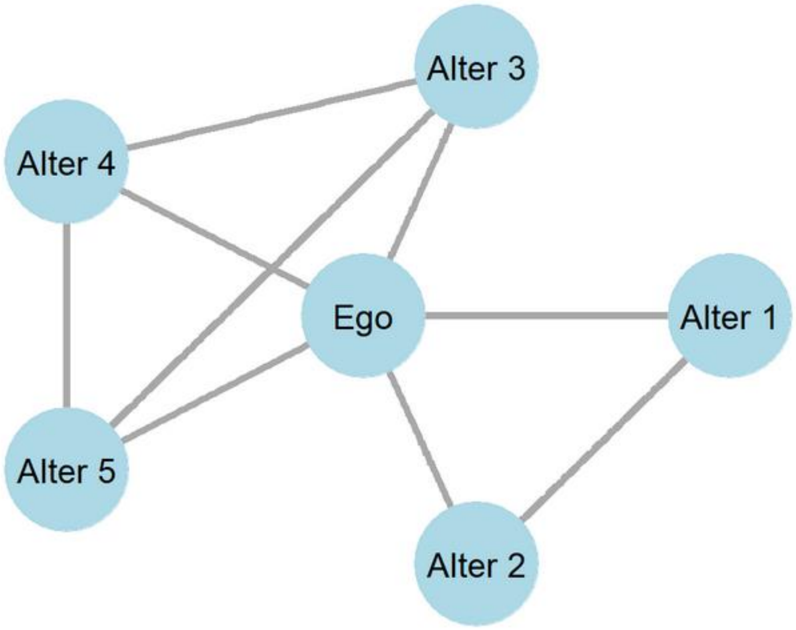
$$\frac{\text{Actual Connections}}{\text{Potential Connections}}$$

- Density is always between 0 and 1, where 0 is the lowest possible density and 1 is the highest.

➢Density in egocentric networks

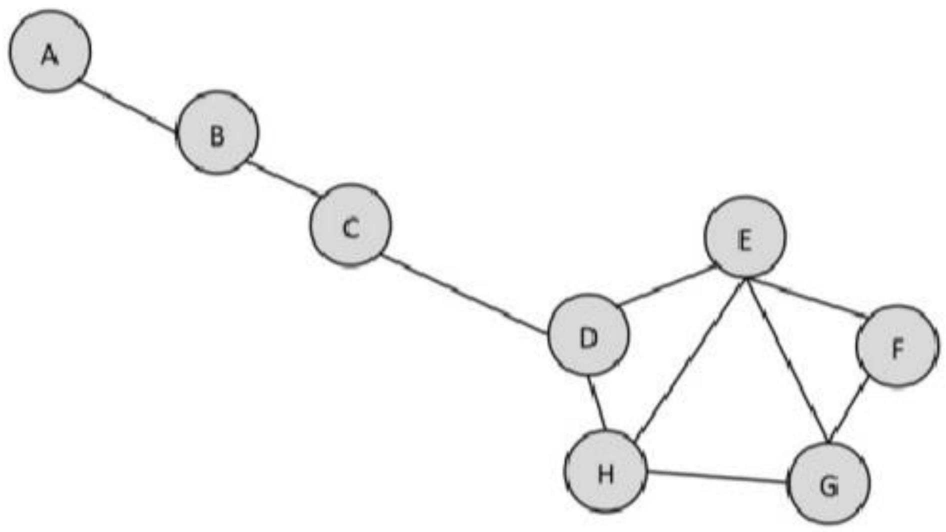
- ✓Density is more commonly used to compare subnetworks especially egocentric networks.
- ✓Computing the density of each node’s egocentric network gives us a way to compare nodes.
- ✓Some will have dense egocentric networks, which means a lot of their friends know one another. Others will have sparse egocentric networks, and where their connections often do not know one another.
- ✓The density of an egocentric network is sometimes referred to as the **local clustering coefficient**
- ✓To compute the density of an egocentric network, 1.5-diameter network is used
- ✓Which considers the node’s connections and all the connections between those nodes.
- ✓For this calculation, the ego-node will be excluded from its egocentric network because the density of interest is that of the connections between the node’s friends

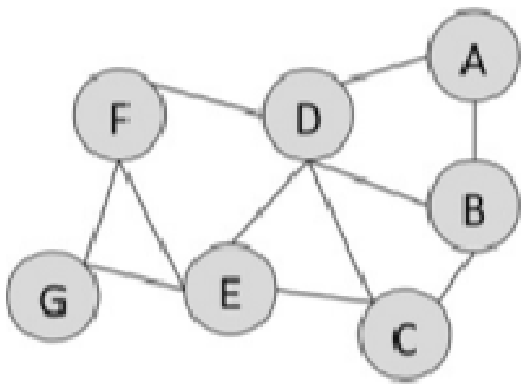
- Find density of egocentric node



➢ Connectivity

- ✓ Connectivity, also known as **cohesion**, measures how the edges in the network are distributed.
- ✓ It is a count of the minimum number of **nodes** that would have to be removed before the graph becomes disconnected; that is, there is no longer a path from each node to every other node.





➢ Centralization

- ✓ **Centrality** is an important way to understand the role of a node in the network and to compare nodes.
- ✓ Centralization uses the distribution of a centrality measure to understand the network as a whole.
- ✓ Any one of the centrality measures must be used to decide Centralization
- ✓ If one node has extremely high centrality while most other nodes have low centrality, the centralization of the graph is high.
- ✓ If centrality is more evenly distributed, then the centralization of the network is low.
- ✓ Centralization measures the extent to which the ties of a given network are concentrated on a single actor or group of actors.
- ✓ Centralization is computed by looking at the sum of the differences in centrality between the most central node and every other node in the network, and dividing this by the maximum possible difference in centrality that could exist in the graph.

- Let C(n) be the centrality of node n, (using whatever centrality measure).
- Say n is the most central node.
- Find the difference in centrality between n and every other node in the network, and add those up.
- If there are N nodes in the network, the formula for this is:

$$\sum_{i=1}^{N} C(n^*) - C(n_i)$$

- Divide this by the sum of the maximum possible differences between n and every other node. [Maximum possible centrality will change depending on which centrality measure we are using].
- Denoted as $\max \sum_{i=1}^{N} C(n^*) - C(n_i)$
- Centralization= $\dfrac{\sum_{i=1}^{N} C(n^*) - C(n_i)}{\max \sum_{i=1}^{N} C(n^*) - C(n_i)}$