Social Networks Measures

- **Single-node Measures**: Based on some properties of specific nodes
- **Graph-based measures**: Based on the graph-structure of the network

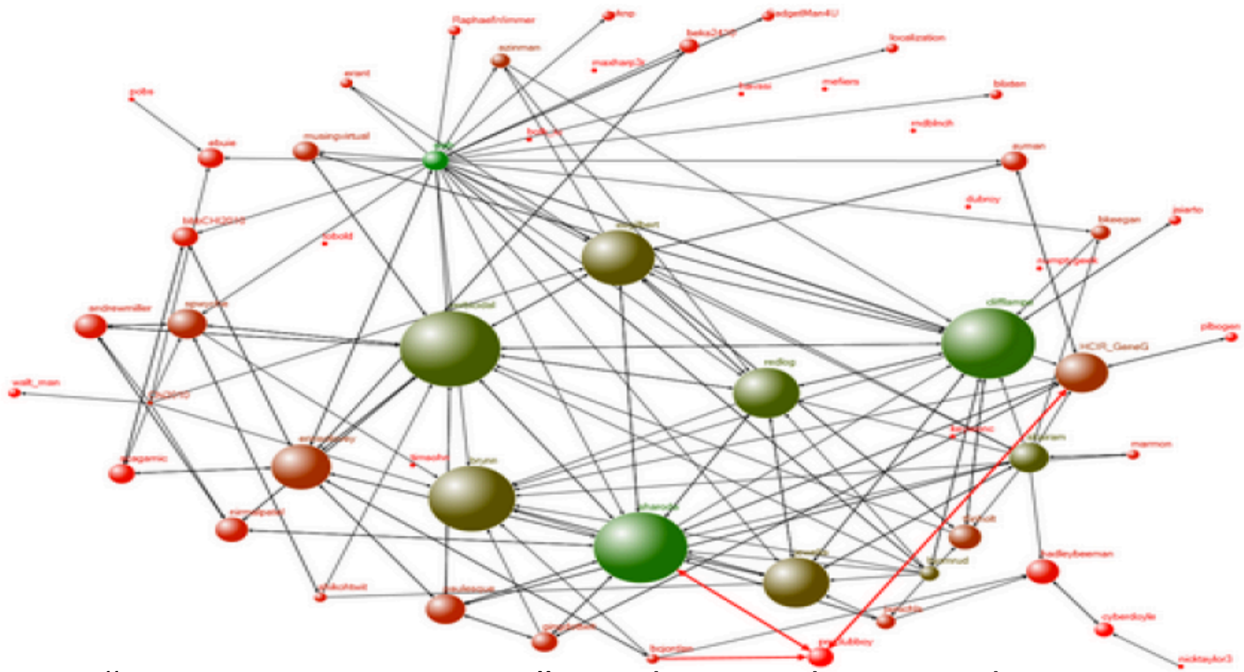
Graph-based measures of social influence

- Previously surveyed measures of influence, such as buzz, applause etc. are based on surface metrics (e.g. number of retweets, etc): graph-based measures go more in-depth.
- Objective here: model the social network as a graph
- Use graph-based methods/algorithms to identify “relevant players” in the network
  - Relevant players = more influential, according to some criterion
- Use graph-based methods to identify communities (community detection)
- Use graph-based methods to analyze the “spread” of information

Graph-based measures of social influence

- **Use graph-based methods/algorithms to identify "relevant players" in the network**
  - Relevant players = more influential, according to some criterion
- Use graph-based methods to identify global network properties and communities (community detection)
- Use graph-based methods to analyze the "spread" of information

Modeling a Social Network as a graph



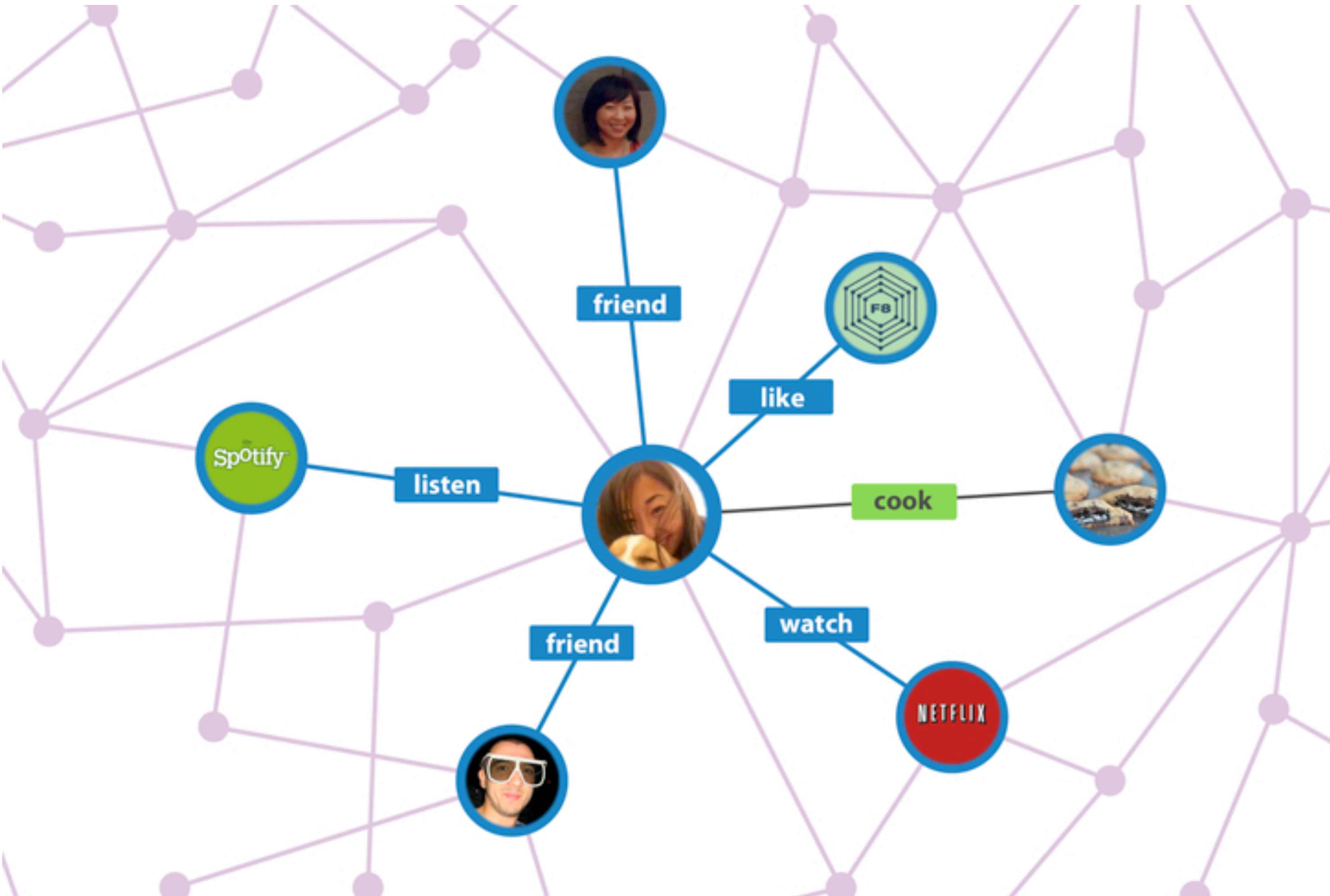
NODE= “actor, vertices, points” i.e. the social entity who participates in a certain network

EDGE= “connection, edges, arcs, lines, ties” is defined by some type of relationship between these actors (e.g. friendship, reply/re-tweet, partnership between connected companies..)

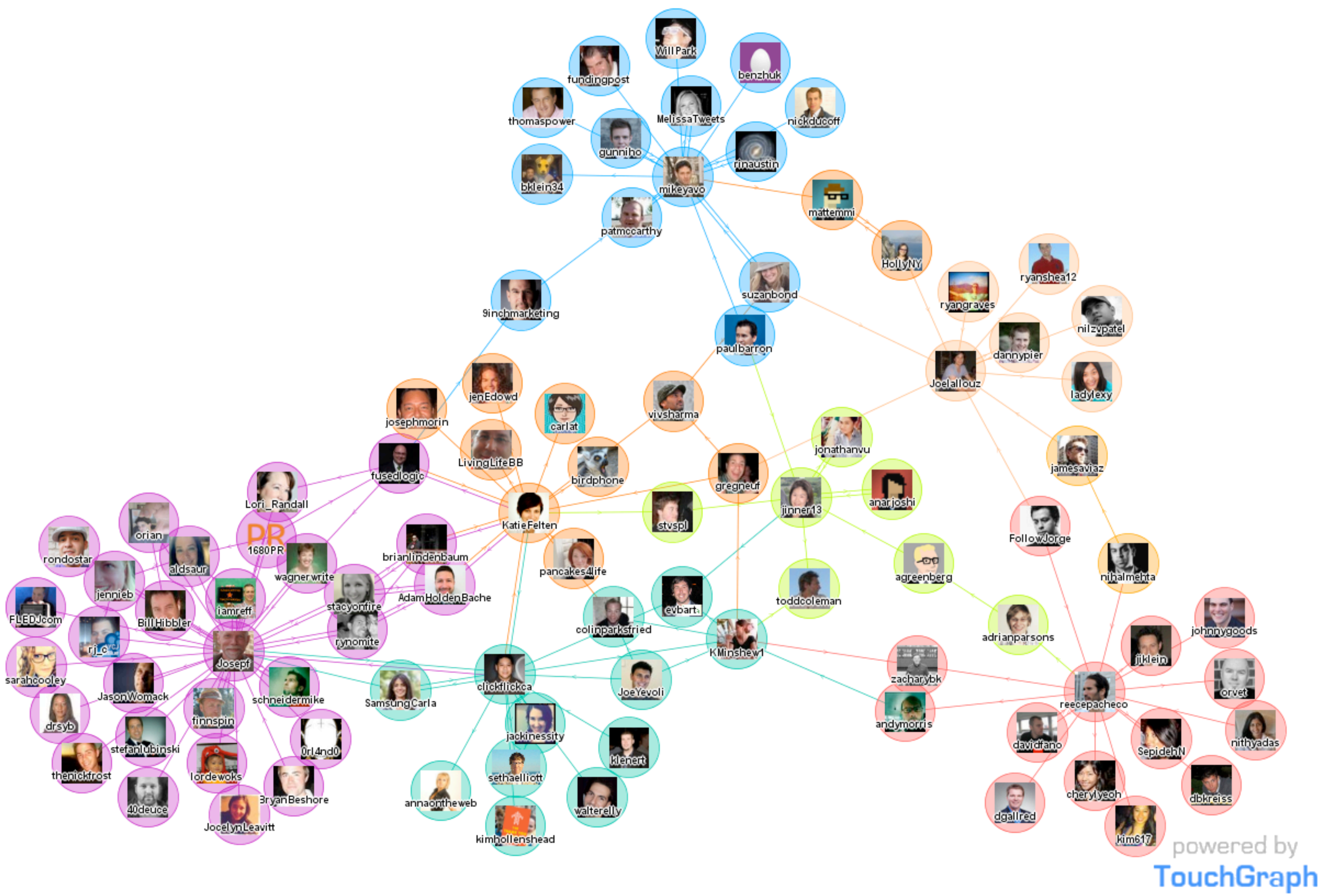
SN = graph

- A network can then be represented as a graph data structure
- We can apply a variety of measures and analysis to the graph representing a given SN
- Edges in a SN can be **directed or undirected** (e.g. friendship, co-authorship are usually undirected, emails are directed)

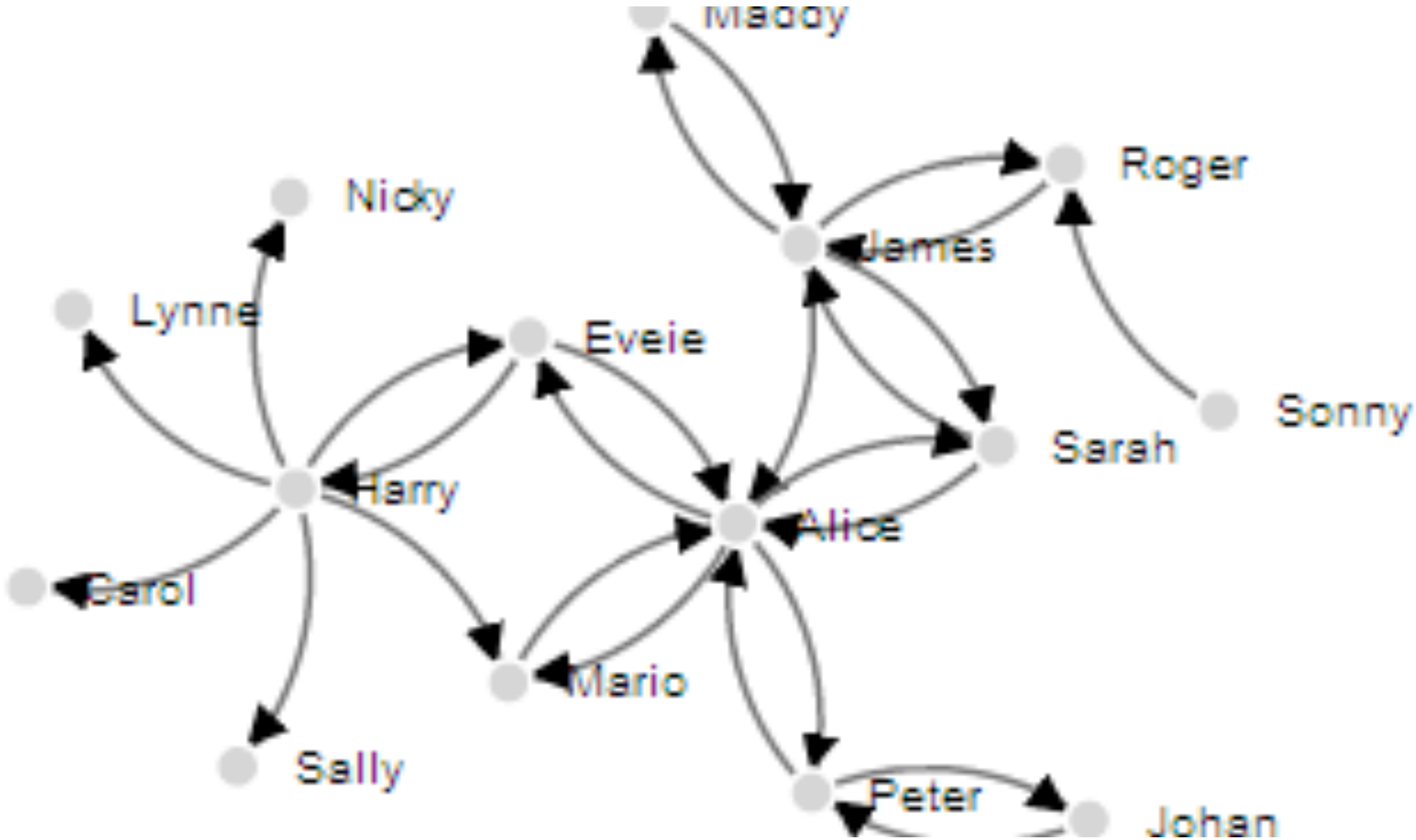
What is the meaning of edges?



Facebook in undirected (friendship is mutual)



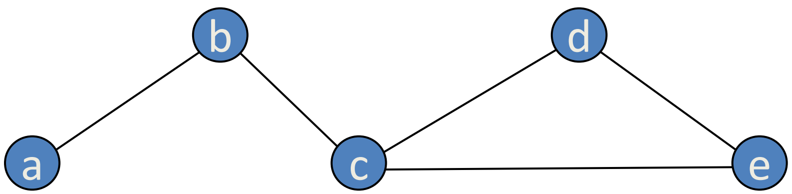
Twitter is a directed graph (friendship is not necessarily bidirectional)



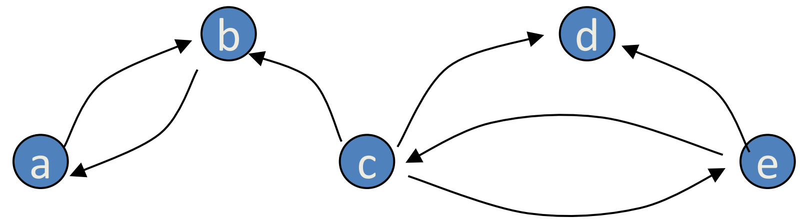
Social Network as a graph

In general, a relation can be:

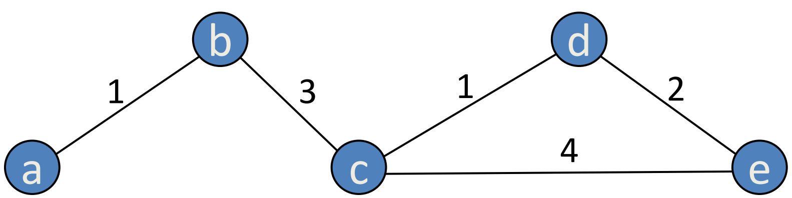
- Binary or Valued
- Directed or Undirected



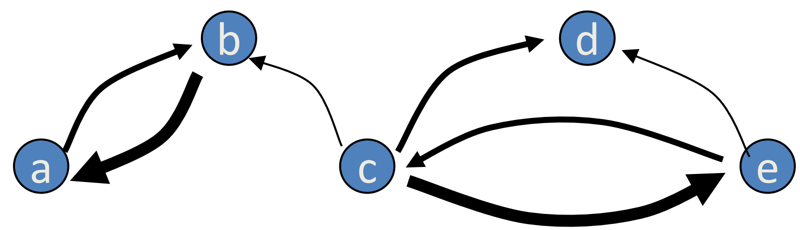
Undirected, binary



Directed, binary

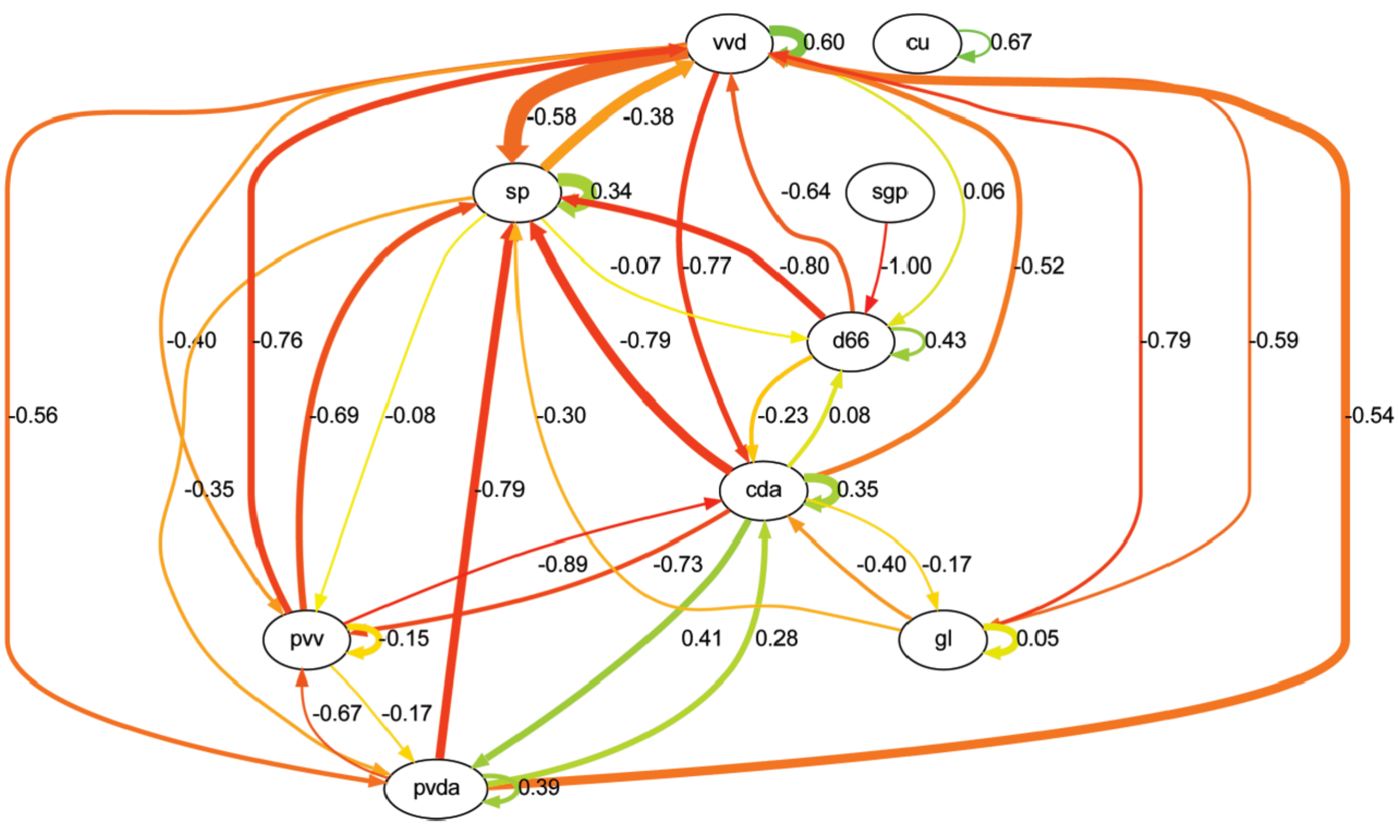


Undirected, Valued



Directed, Valued

Example of directed, valued: Sentiment relations among parties during a political campaign.
Color: positive (green) negative (red).
Intensity (thikness of edges): related to number of mutual references



Graph-based measures of social influence: key players

Key players

- Using graph theory, we can identify **key players** in a social network
- Key players are nodes (or actors, or vertexes) with some measurable **connectivity property**
- Two important concepts in a network are the ideas of **centrality** and **prestige** of an actor.
- Centrality more suited for undirected, prestige for directed

Measuring Networks: Centrality

Centrality refers to (one dimension of) *location,* identifying *where* an actor resides in a network. Mostly used for **undirected** networks.

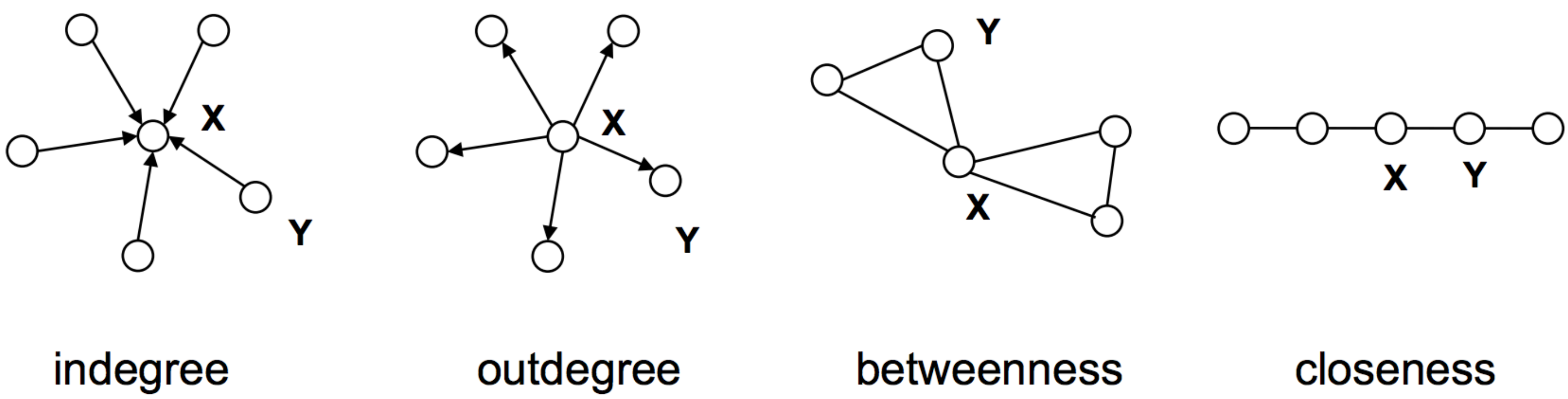
- For example, we can compare actors at the edge of the network to actors at the center.

- In general, this is a way to formalize intuitive notions about the distinction between *insiders and outsiders*.

Measuring Networks: **Centrality**

Conceptually, centrality is fairly straight forward: we want to identify which nodes are in the 'center' of the network. Who is important based on network position.

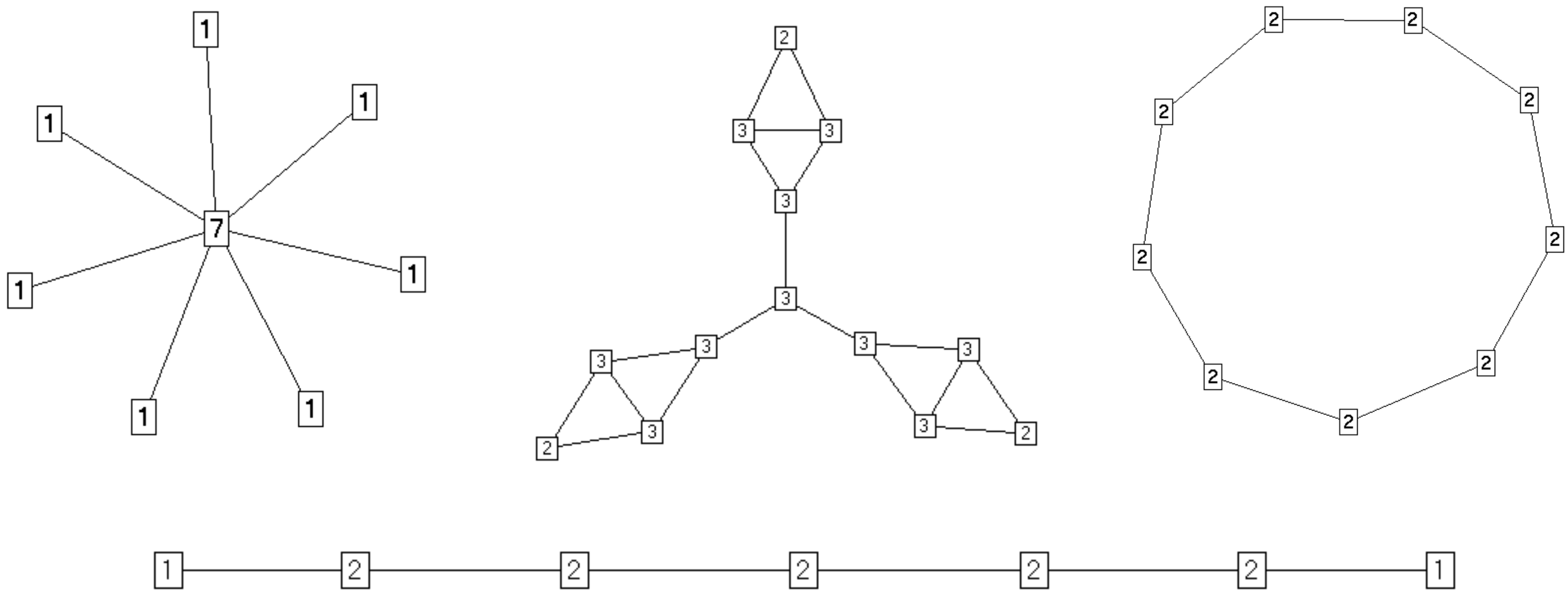
Several types of centrality measures:



Measuring Networks: **Centrality**

1.Centrality Degree

The most intuitive notion of centrality focuses on ***degree***. Degree is the number of ties, and the actor with the most ties is the most important:

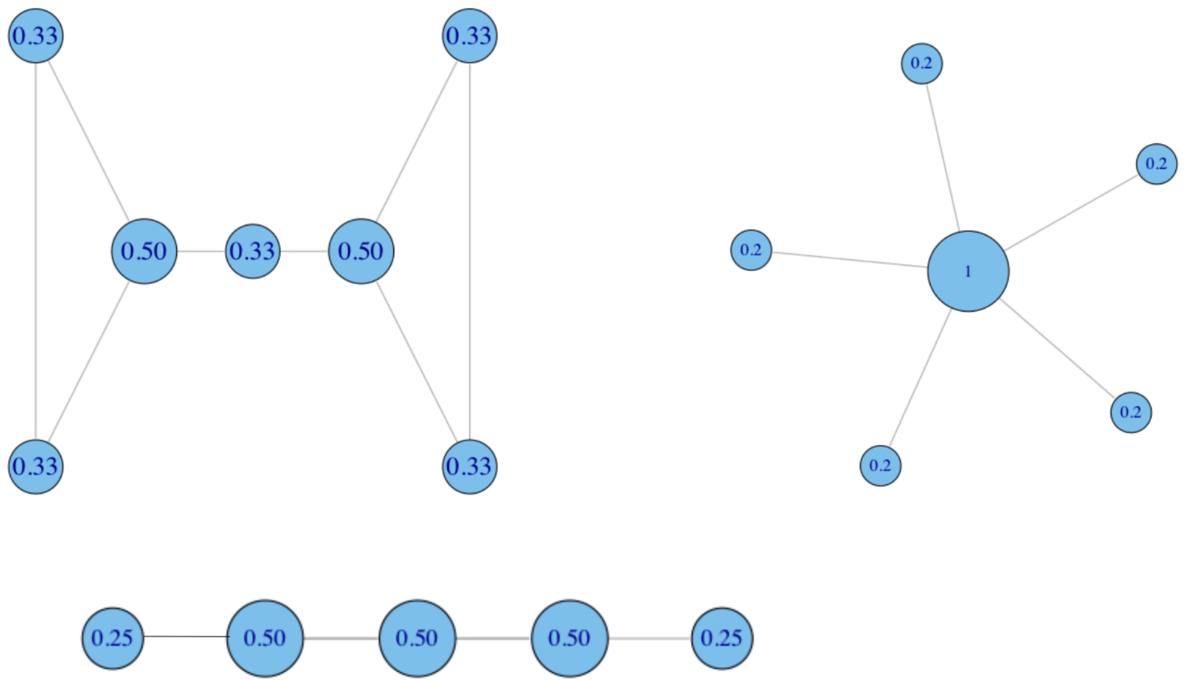


$$C_D = d(n_i) = X_{i+} = \sum_j X_{ij}$$

Measuring Networks: **Centrality**

2.Normalized Centrality Degree

Divide by the maximum, e.g. the number of nodes N:
$C'_D(n)=C_D(n)/(N-1)$



Measuring Networks: Closeness **Centrality**

A second measure of centrality is **closeness centrality**. An actor is considered important if he/she is relatively close to all other actors.

Closeness is based on the inverse of the distance of each actor to every other actor in the network.

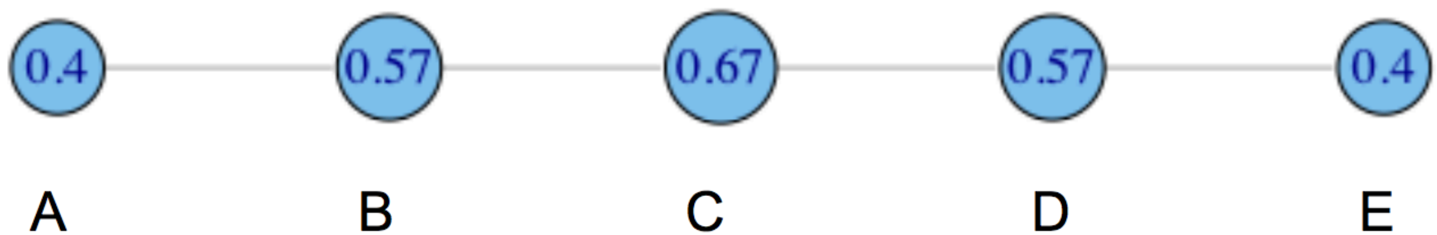
Closeness Centrality:

$$C_c(n_i) = \left[\sum_{j=1}^{g} d(n_i, n_j) \right]^{-1}$$

Normalized Closeness Centrality (*g* is is the maximum, e.g., the number of nodes in the network)

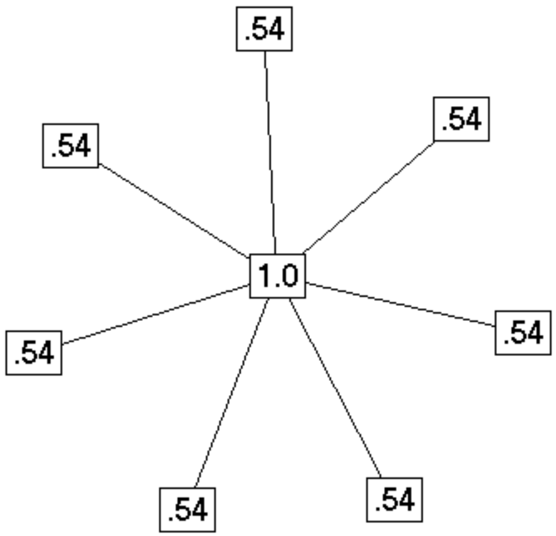
$$C'_C(n) = \frac{C_C(n)}{g-1}$$

Closeness centrality simple example



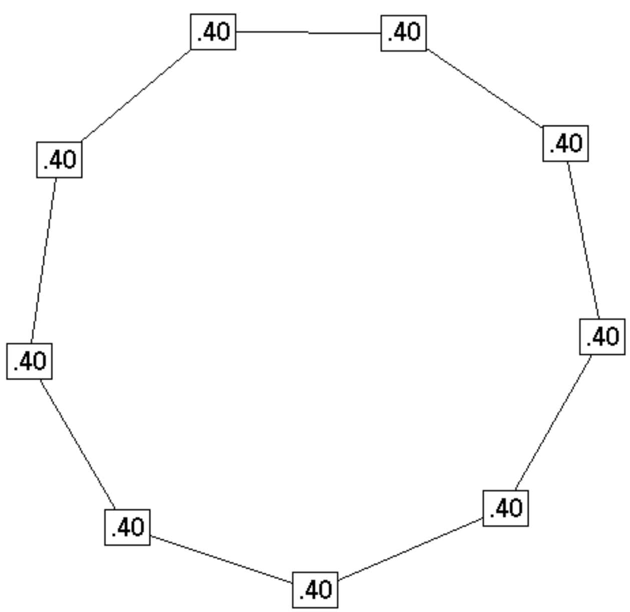
$$C_c^{'}(A) = \left[\frac{\sum_{j=1}^{N} d(A,j)}{N-1} \right]^{-1} = \left[\frac{1+2+3+4}{4} \right]^{-1} = \left[\frac{10}{4} \right]^{-1} = 0.4$$

Measuring Networks: examples of closeness **Centrality**



Distance	Closeness	normalized
0 1 1 1 1 1 1 1	.143	1.00
1 0 2 2 2 2 2 2	.077	.538
1 2 0 2 2 2 2 2	.077	.538
1 2 2 0 2 2 2 2	.077	.538
1 2 2 2 0 2 2 2	.077	.538
1 2 2 2 2 0 2 2	.077	.538
1 2 2 2 2 2 0 2	.077	.538
1 2 2 2 2 2 2 0	.077	.538

$$C_c(n_i) = \left[\sum_{j=1}^{g} d(n_i, n_j)\right]^{-1}$$



Distance	Closeness	normalized
0 1 2 3 4 4 3 2 1	.050	.400
1 0 1 2 3 4 4 3 2	.050	.400
2 1 0 1 2 3 4 4 3	.050	.400
3 2 1 0 1 2 3 4 4	.050	.400
4 3 2 1 0 1 2 3 4	.050	.400
4 4 3 2 1 0 1 2 3	.050	.400
3 4 4 3 2 1 0 1 2	.050	.400
2 3 4 4 3 2 1 0 1	.050	.400
1 2 3 4 4 3 2 1 0	.050	.400

Measuring Networks: ex. Closeness **Centrality**

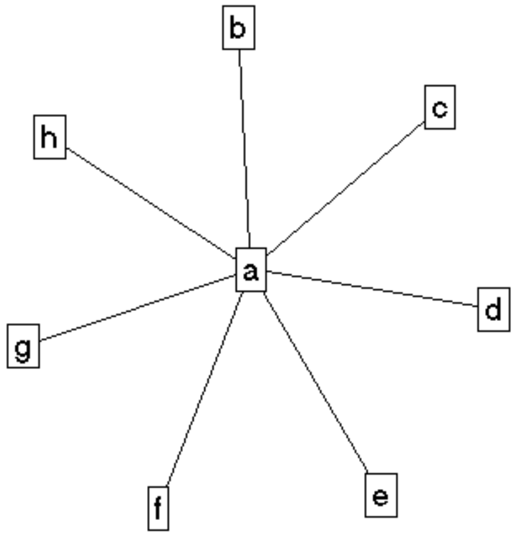
Distance	Closeness	normalized
0 1 2 3 4 5 6	.048	.286
1 0 1 2 3 4 5	.063	.375
2 1 0 1 2 3 4	.077	.462
3 2 1 0 1 2 3	.083	.500
4 3 2 1 0 1 2	.077	.462
5 4 3 2 1 0 1	.063	.375
6 5 4 3 2 1 0	.048	.286

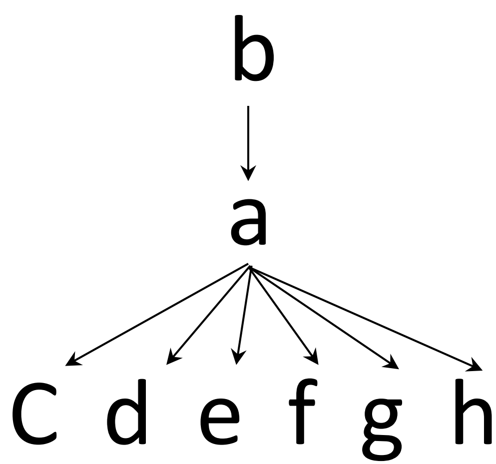
$$C_c(n_i) = \left[\sum_{j=1}^{g} d(n_i, n_j) \right]^{-1}$$

Measuring Networks: **Betweenness Centrality**

Model based on communication flow: A person who lies on communication paths can control communication flow, and is thus important.

Betweenness centrality counts the number of geodesic paths between *i* and *k* **that actor *j* resides on**. Geodesics are defined as the shortest path between points



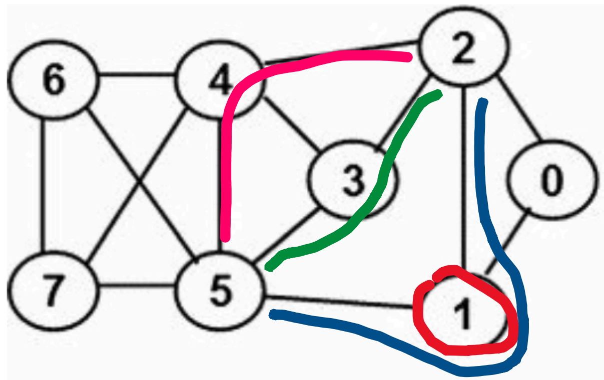


Measuring Networks: **Betweenness Centrality**

$$C_B(n_i) = \sum_{j<k} g_{jk}(n_i) / g_{jk}$$

Where g_{jk} = the number of geodesics (shortest) connecting *jk*, and g_{jk} (ni)= the number of such paths that node *i* is on (**count also in the start-end nodes of the path**).

Can also compute **edge betweenness** in the very same way




```
betweeness for node 0
Betweenness of 0 : 0.0
************************************
betweeness for node 1
Pair (5,0) --->1 / 1
Pair (6,0) --->1 / 2
Pair (7,0) --->1 / 2
Pair (5,2) --->1 / 3
Pair (0,5) --->1 / 1
Pair (2,5) --->1 / 3
Pair (0,6) --->1 / 2
Pair (0,7) --->1 / 2
Betweenness of 1 : 4.666666666666667
************************************
betweeness for node 2
Pair (3,0) --->1 / 1
Pair (4,0) --->1 / 1
Pair (6,0) --->1 / 2
Pair (7,0) --->1 / 2
Pair (3,1) --->1 / 2
Pair (4,1) --->1 / 2
Pair (0,3) --->1 / 1
Pair (1,3) --->1 / 2
Pair (0,4) --->1 / 1
Pair (1,4) --->1 / 2
Pair (0,6) --->1 / 2
Pair (0,7) --->1 / 2
Betweenness of 2 : 8.0
************************************
betweeness for node 3
Pair (5,2) --->1 / 3
Pair (2,5) --->1 / 3
Betweenness of 3 : 0.6666666666666666
************************************
betweeness for node 4
Pair (6,0) --->1 / 2
Pair (7,0) --->1 / 2
Pair (5,2) --->1 / 3
Pair (6,2) --->1 / 1
Pair (7,2) --->1 / 1
Pair (6,3) --->1 / 2
Pair (7,3) --->1 / 2
Pair (2,5) --->1 / 3
Pair (0,6) --->1 / 2
Pair (2,6) --->1 / 1
Pair (3,6) --->1 / 2
Pair (0,7) --->1 / 2
Pair (2,7) --->1 / 1
Pair (3,7) --->1 / 2
Betweenness of 4 : 8.666666666666666
************************************
betweeness for node 5
Pair (6,0) --->1 / 2
Pair (7,0) --->1 / 2
Pair (3,1) --->1 / 2
Pair (4,1) --->1 / 2
Pair (6,1) --->1 / 1
Pair (7,1) --->1 / 1
Pair (1,3) --->1 / 2
Pair (6,3) --->1 / 2
Pair (7,3) --->1 / 2
Pair (1,4) --->1 / 2
Pair (0,6) --->1 / 2
Pair (1,6) --->1 / 1
Pair (3,6) --->1 / 2
Pair (0,7) --->1 / 2
Pair (1,7) --->1 / 1
Pair (3,7) --->1 / 2
Betweenness of 5 : 10.0
************************************
betweeness for node 6
Betweenness of 6 : 0.0
************************************
betweeness for node 7
Betweenness of 7 : 0.0
************************************
```

ID	Betweenness	ID	Betweenness
0	0.0	4	8.67
1	4.67	5	10.0
2	8.0	6	0.0
3	0.67	7	0.0

Method (to avoid computing shortest paths for all nodes /edges)

BFS breadth first search

- For each node A:
  1. BFS starting at A
  2. Count the number of shortest paths from A to each other node
  3. Based on this number, determine the amount of flow from A to all other nodes

Formal definition of betweenness

- Directed graph $G = < V, E >$
- $\sigma(s, t)$: number of shortest paths between nodes s and t
- $\sigma(s, t|v)$: number of shortest paths between nodes s and t that pass through v.
- $C_B(v)$, the betweenness centrality of v:

$$C_B(v) = \sum_{s,t \in V} \frac{\sigma(s, t|v)}{\sigma(s, t)}$$

- If $s = t$, then $\sigma(s, t) = 1$
- If $v \in (s, t)$ then $\sigma(s, t|v) = 0$

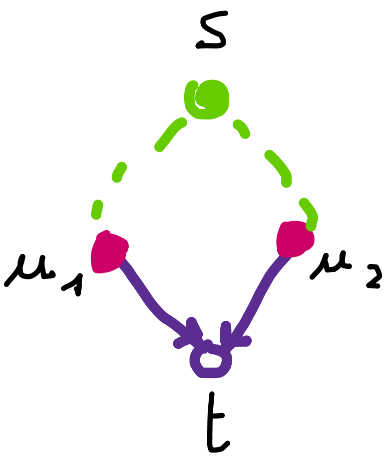
https://www.cl.cam.ac.uk/teaching/1617/MLRD/slides/slides13.pdf

1) Recursive calculation of shortest paths

- $\sigma(s, t)$ can be calculated recursively:

$$\sigma(s,t) = \sum_{u \in Pred(t)} \sigma(s,u)$$

  - $\text{Pred}(t) = \{u : (u,t) \in E, d(s,t) = d(s,u) + 1\}$ predecessors of t on shortest path from s
  - $d(s, u)$: Distance between nodes s and u
- This can be done by running Breadth First search with each node as source s once, for total complexity of $O(V(V + E))$.



Example if there are 3 shortest paths between s and u1 and 2 between s and u2, then, there will be 5 shortest paths between s and t

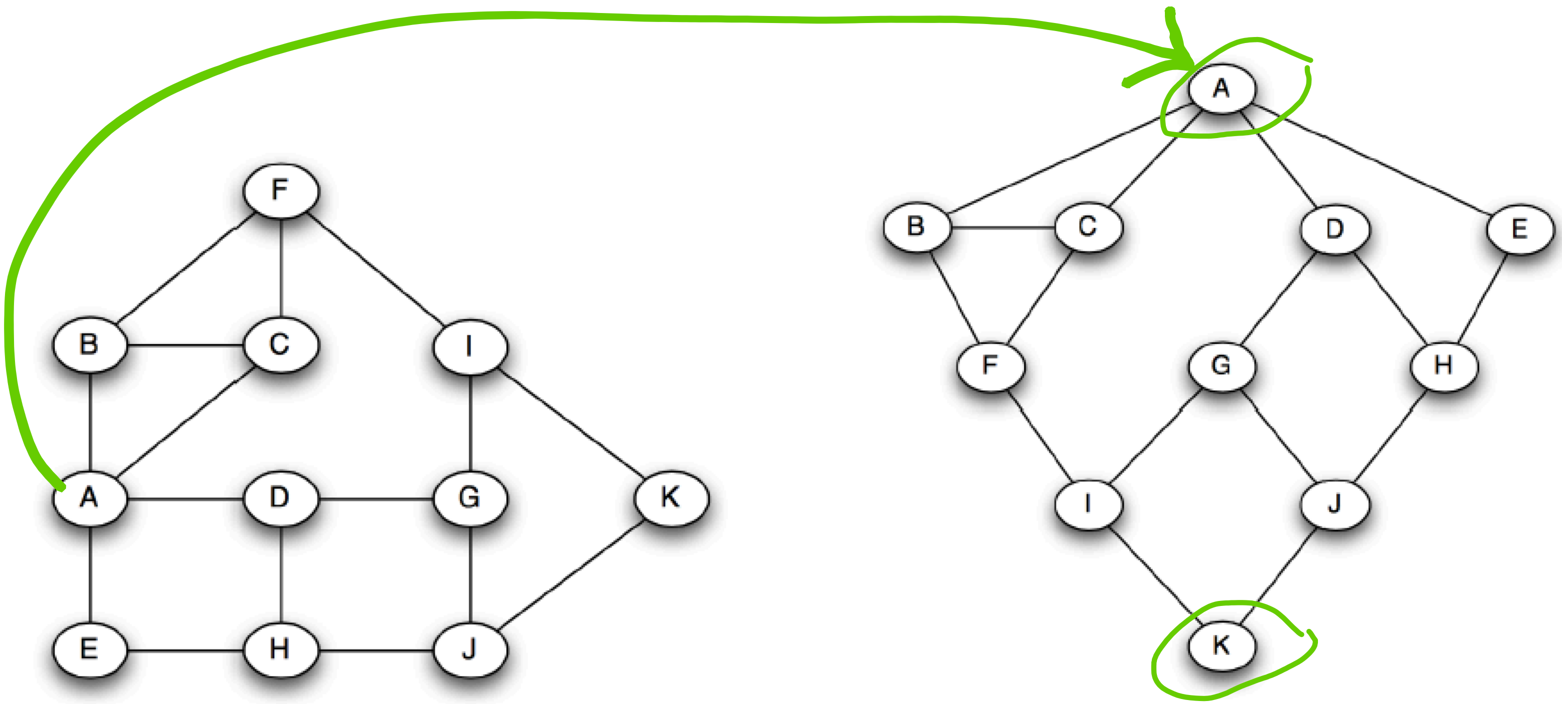
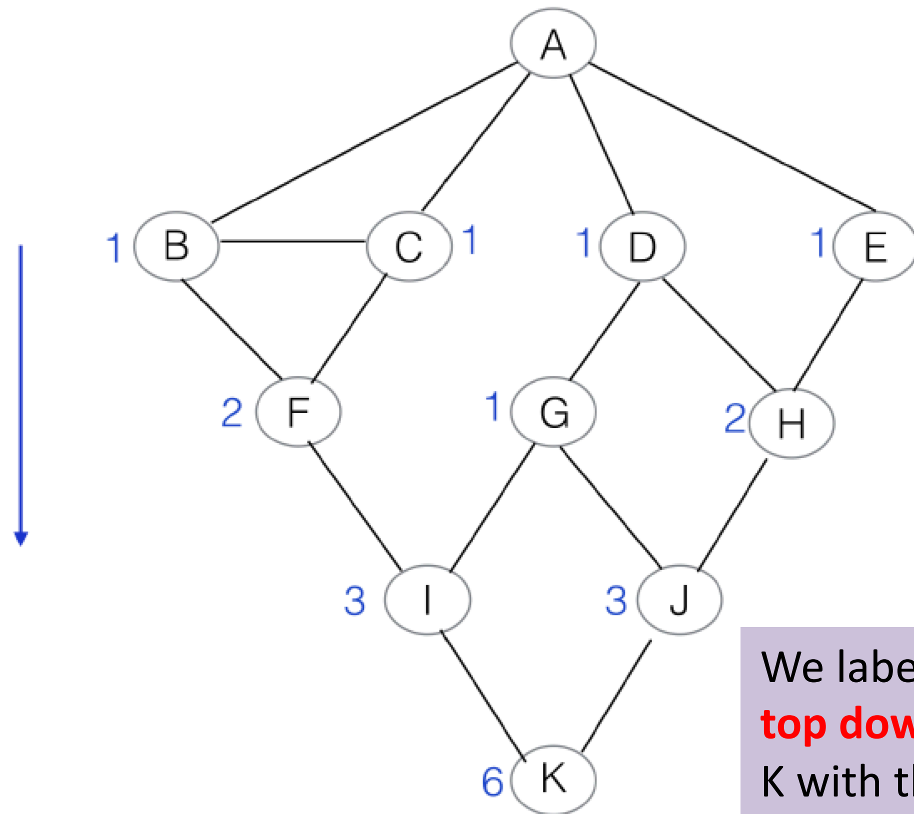


Figure 3-18 from Easley and Kleinberg (2010)

How many shortest paths between A and K??



We label each node Ni
top down from A towards
K with the number of shortest
paths from A

$$\sigma(s,t) = \sum_{u \in Pred(t)} \sigma(s,u)$$

2) Recursive calculation of flow

$$\delta(s,t|v) = \frac{\sigma(s,t|v)}{\sigma(s,t)}$$

Fraction of shortes paths between s and t that v lies on

$$\delta(s|v) = \sum_{t \in V} \delta(s,t|v)$$

Betweenness of *v* w.r.t. paths starting from *s*

Then Brandes (2001) shows:

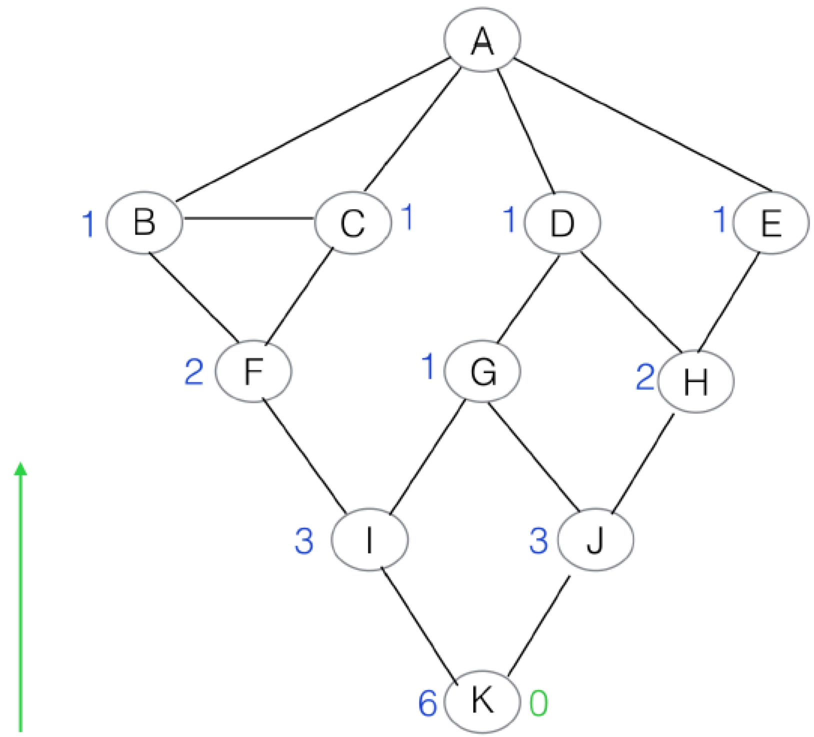
$$\delta(s|v) = \sum_{\substack{(v,w) \in E \\ w:\ d(s,w) = d(s,v)+1}} \frac{\sigma(s,v)}{\sigma(s,w)}.(1 + \delta(s|w))$$

w is a node «below» node *v*

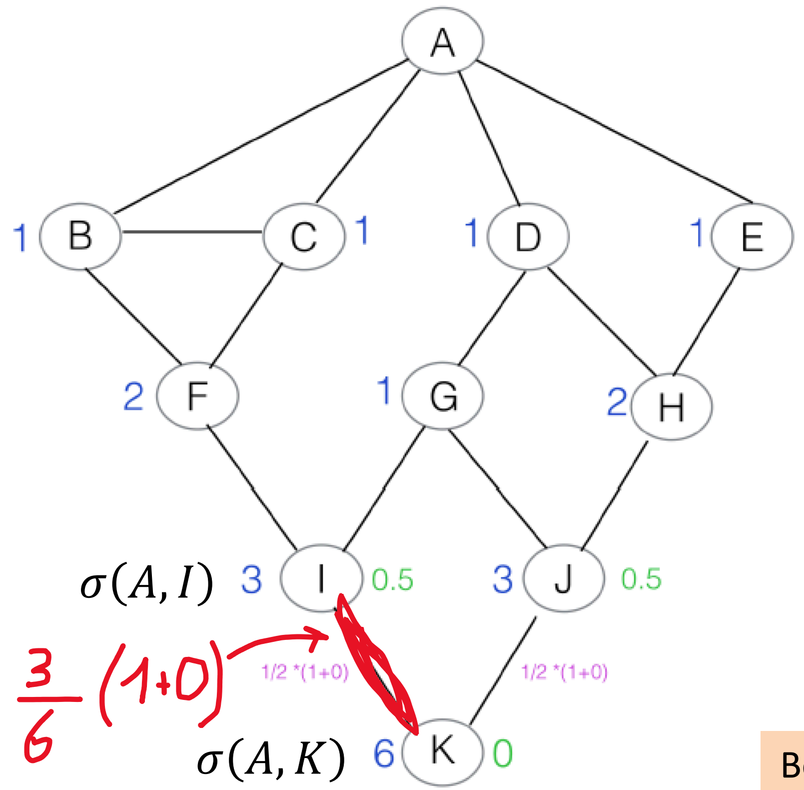
And:

$$C_B(v) = \sum_{s \in V} \delta(s|v)$$

$\delta(s|v)$ can also be iteratively calculated bottom up!



$$\delta(s|v) = \sum_{\substack{(v,w)\in E\\ w:\ d(s,w)=d(s,v)+1}} \sigma(s,v)/\sigma(s,w).(1+\delta(s|w))$$

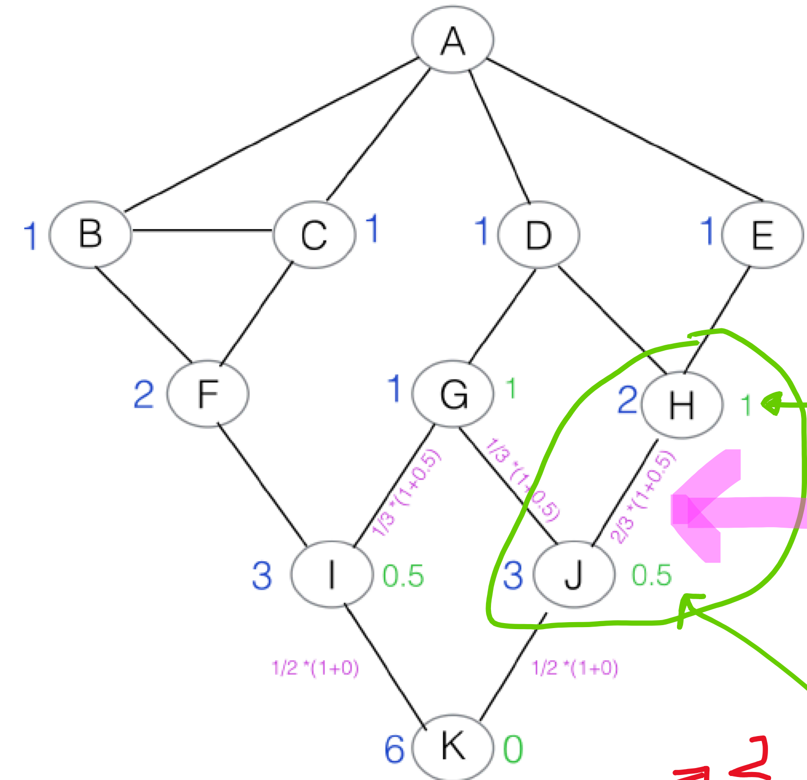


$\delta(A|K) = 0$

$\frac{3}{6}(1+0)$

$$\delta(s|v) = \sum_{\substack{(v,w) \in E \\ w:\ d(s,w)=d(s,v)+1}} \sigma(s,v)/\sigma(s,w) \cdot (1 + \delta(s|w))$$

Betweenness of *w* w.r.t. paths starting from *s*



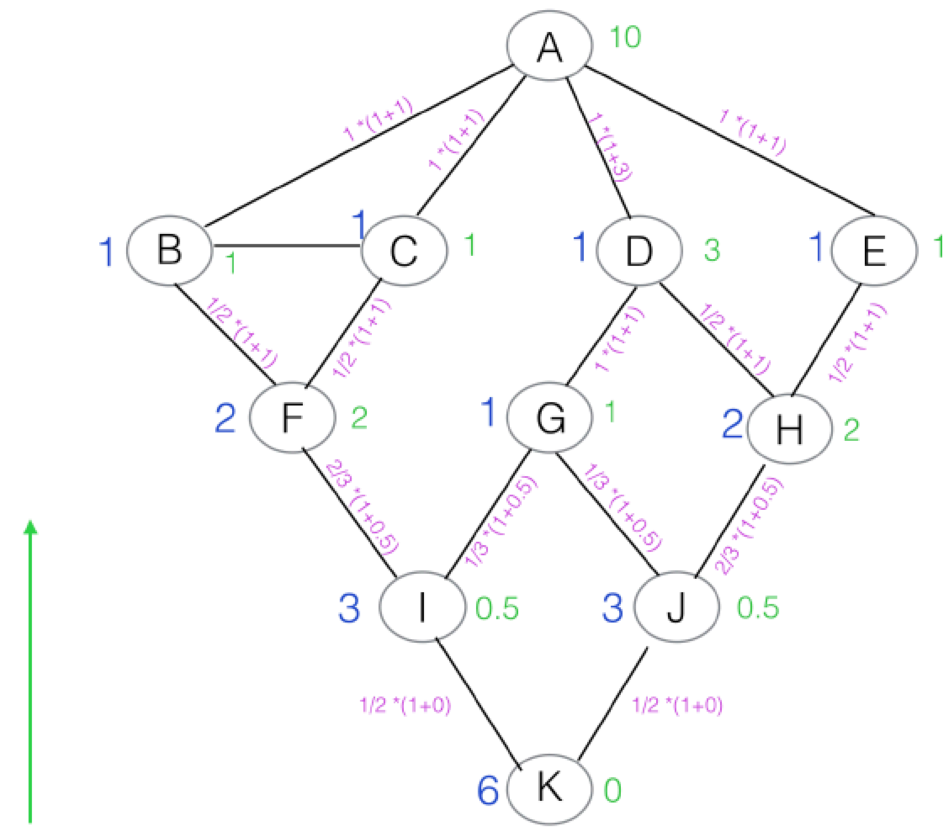
s = A
w = J
v = H

$$\frac{2}{3}\left(1+\frac{1}{2}\right) = 1$$

$\delta(A,H)$

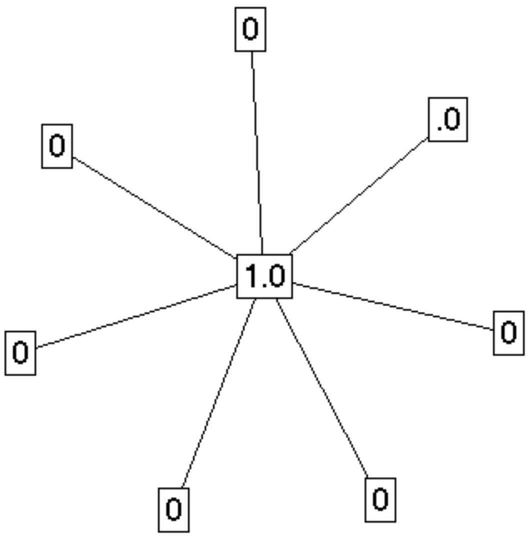
$\frac{2}{3}$

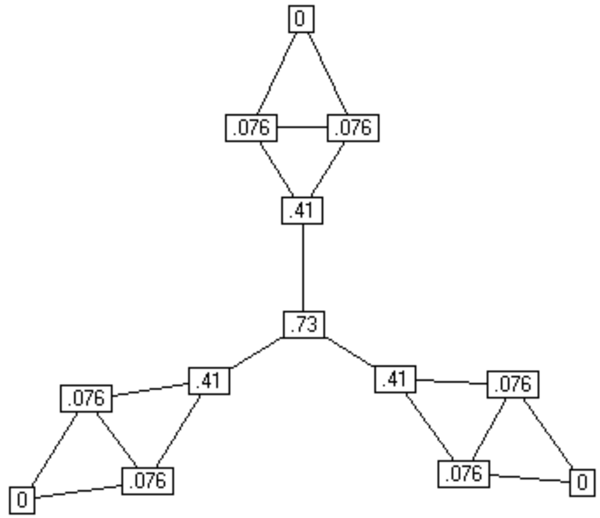
$$\delta(s|v) = \sum_{\substack{(v,w)\in E \\ w:\ d(s,w)=d(s,v)+1}} \sigma(s,v)/\sigma(s,w)(1+\delta(s|w))$$

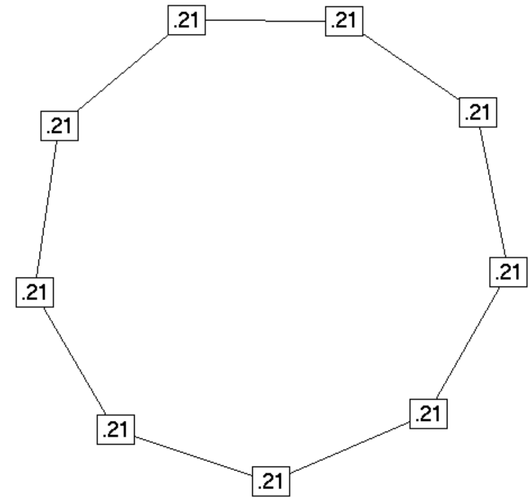


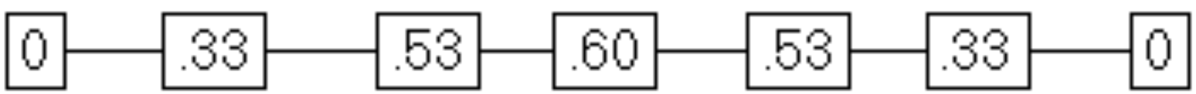
$$\delta(s|v) = \sum_{\substack{(v,w)\in E \\ w:\ d(s,w)=d(s,v)+1}} \sigma(s,v)/\sigma(s,w).(1+\delta(s|w))$$

Other examples (node betweenness)





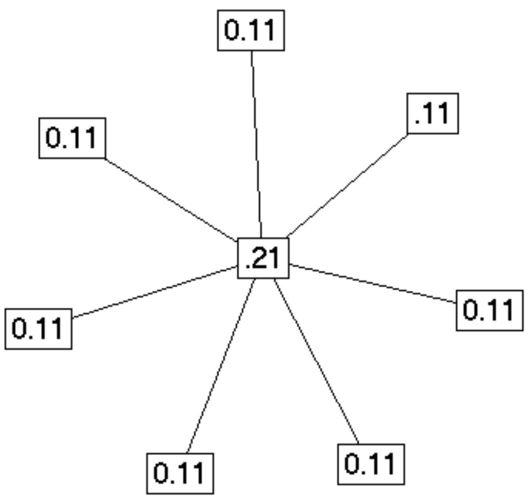


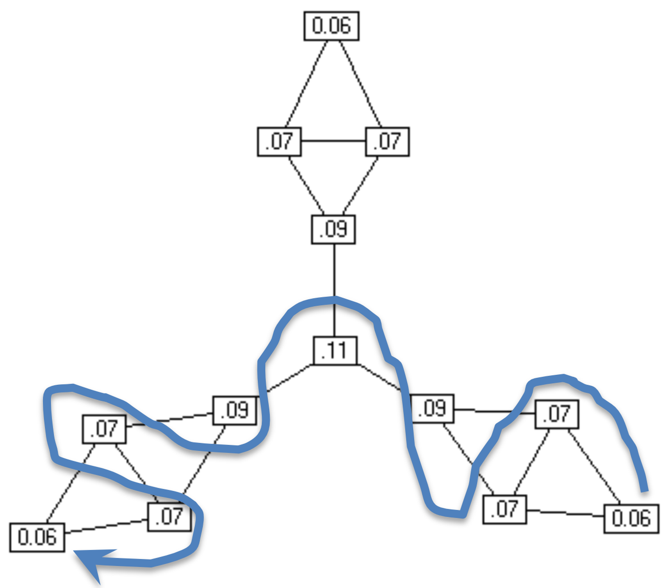


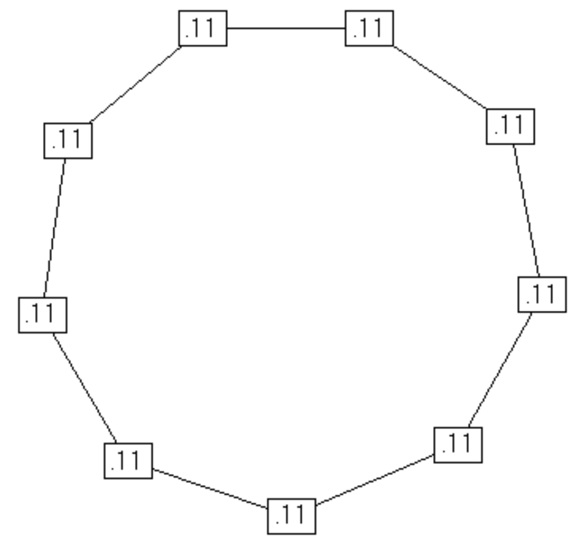
$$C_B(n_i) = \sum_{j<k} g_{jk}(n_i) / g_{jk}$$

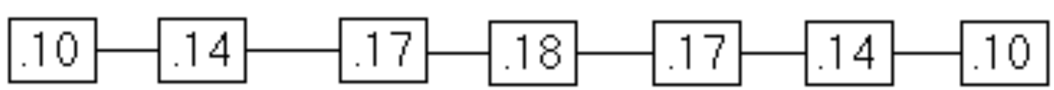
Measuring Networks: **Information Centrality**

It is quite likely that information can flow through paths *other* than the geodesic. The Information Centrality score uses **all paths** in the network, and weights them based on their length.









Computationall very demanding for large graphs!!

Measuring Networks: **Prestige**

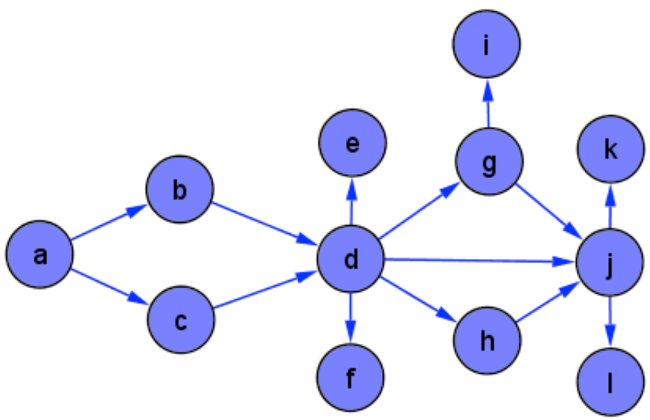
- The term prestige is used for **directed networks** since for this measure the direction is an important property of the relation.
- In this case we can define two different types of prestige:
  - one for outgoing arcs (measures of influence),
  - one for incoming arcs (measures of support).
- Examples:
  - An actor has high influence, if he/she gives hints to several other actors (e.g. in Yahoo! Answers, or if he/she has many followers).
  - An actor has high support, if a lot of people vote for him/her (many “likes”, many friends)
  - Very similar to the concept of hubs and authorities in HITS algorithm

Measures of prestige in directed networks

- Influence and support
- Influence domain
- Hubs and authorities
- Brockers

Measuring prestige: influence and support

- **Influence and support**: According to the direction/meaning of a relation, in and outdegree represent support or influence. (e.g., likes, votes for,. . .).

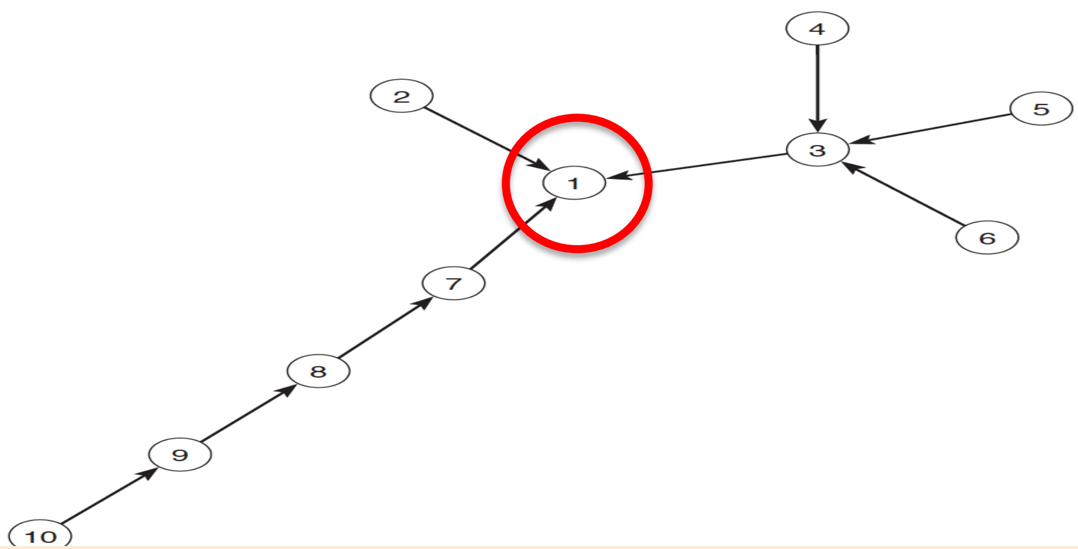


$$InDegree(x) = \#\ \ incoming\ \ edges(x)$$

$$InDegree^{N}(x) = \frac{\#\ \ incoming\ \ edges(x)}{\max\limits_{y \in network}(InDegree^{N}(y))}$$

Measuring prestige: influence domain

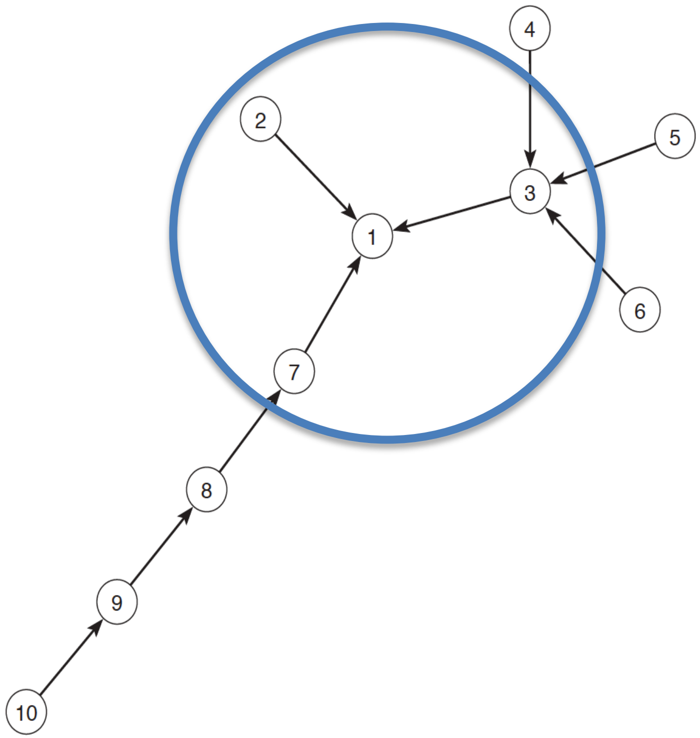
- **Influence domain**: The influence domain of an actor (node) in a directed network is the number (or proportion) of all other nodes *which are connected by a path to this node*.



All other actors are in influence domain of actor 1: Prest(1)=10/10=1.

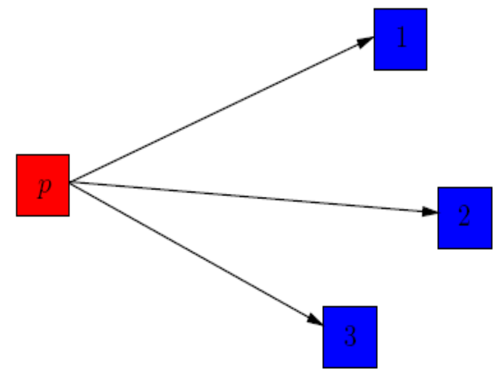
Limits of Influence domain

- Influence domain has an important limitation: *all the nodes contribute equally to influence*.
- Choices by actors 2, 3, and 7 are more important to person 1 than **indirect** choices by 4, 5, 6, and 8. Individuals 9 and 10 contribute even less to the prestige of 1.

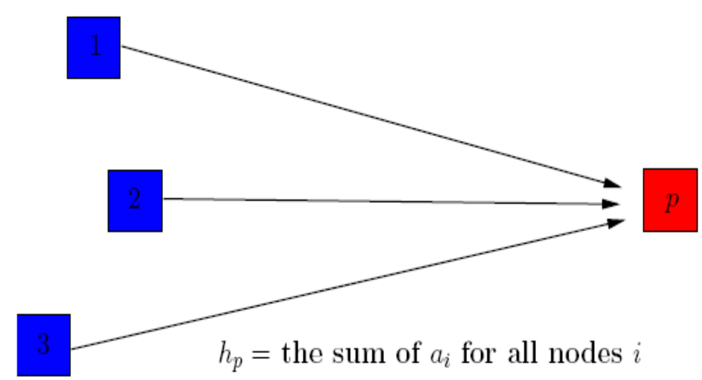


Measuring prestige: Hubs and Authorities, Page Rank

- Hubness is a good measure of influence
- Authority is a good measure of support
- **Kleinberg’s algorithm** (HITS) to compute authority and hubness degree of nodes, same as for link analysis
- **Page Rank** is a good measure of support
- HITS, Page Rank: see previous lessons

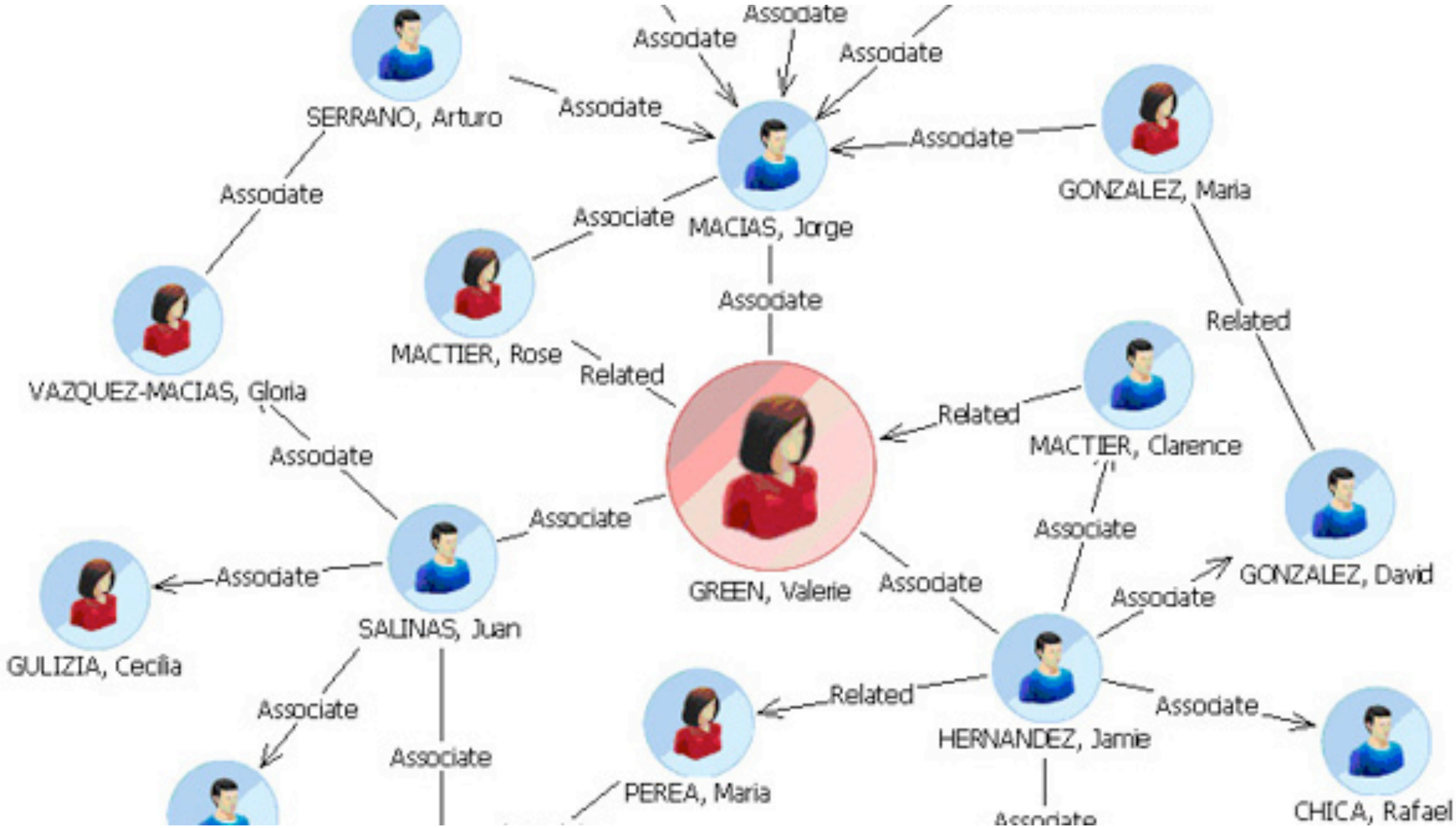


a_p = the sum of h_i for all nodes i pointing to p



h_p = the sum of a_i for all nodes i pointed to by p

Example



If Mrs. Green is the boss, employees referring directly to her are more important

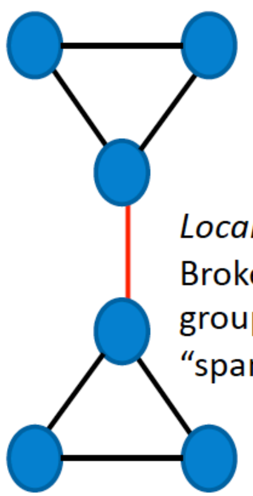
High-level scheme

- Hubs and authorities can be computed in sub-communities, i.e. on parts of a large social network graph, or on the entire graph
- Initial step (create a sub-graph):
  1. Extract from the graph a <u>base set</u> of users that *could* be good hubs or authorities (e.g. with many incoming or outgoing links).
  2. From these, identify a small set of top hub and authority users;

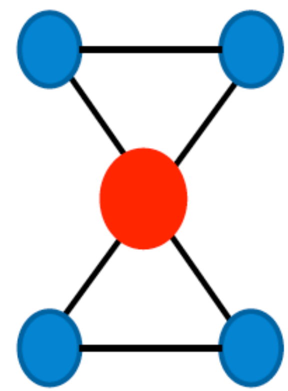
  →using the iterative HITS algorithm.

Measuring prestige: Brockers (bridges)

- Network brokerage: Links between different groups/communites (very similar to betweenness)



Local bridging ties:
Brokerage of disconnected groups:
"spanning structural holes"

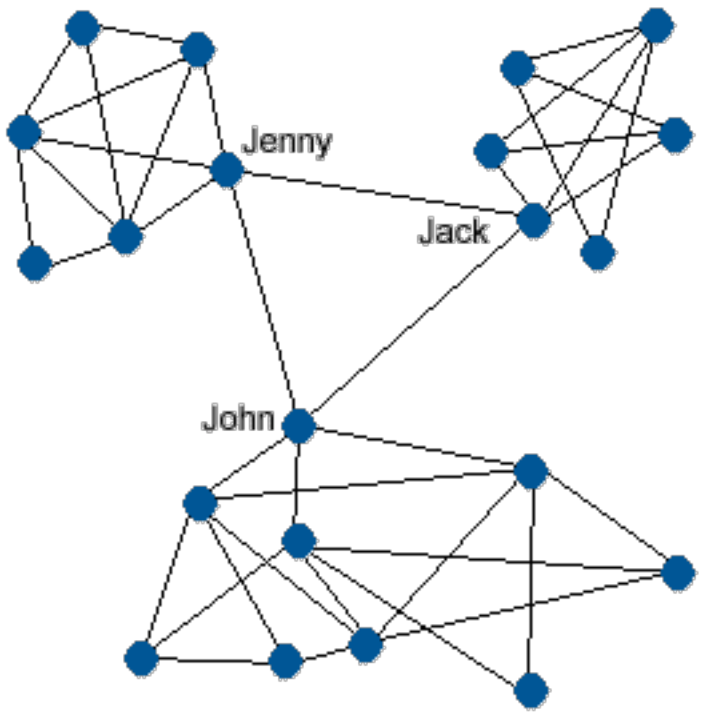


Local cut points:
Brokerage through overlapping group membership

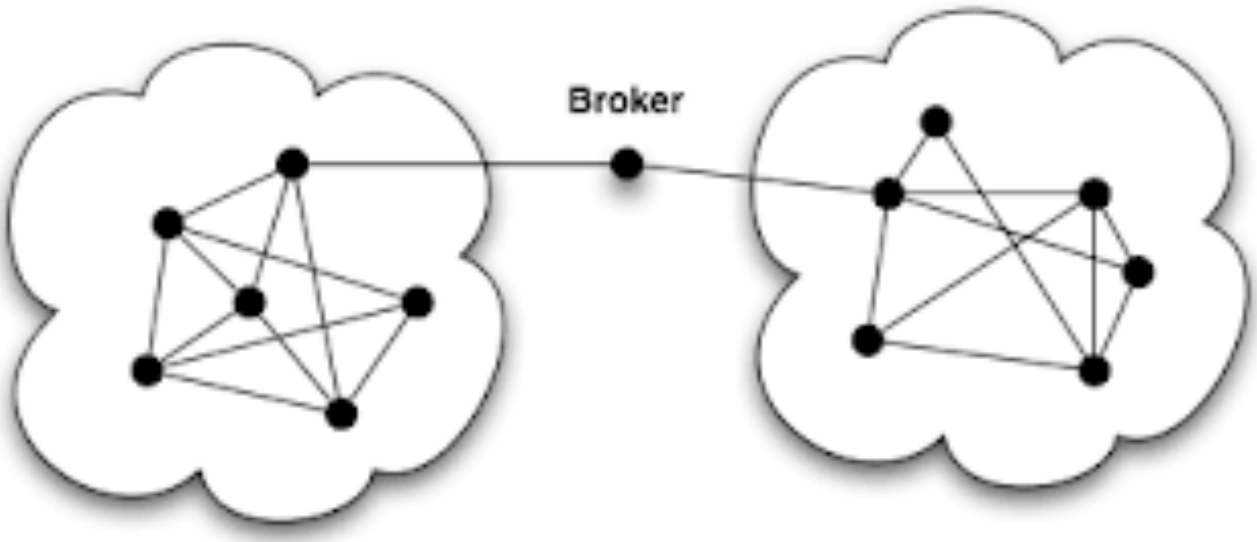
Measuring prestige: Brockers

Finding Brockers

- Brockers are "**intermediaries**", people that create relationships between communities
- As for graph representation, a brocker is a node that, if removed from the graph, **reduces graph connectivity.** For example, it causes the creation of disconnected components (*Jenny*, *Jack* and *John* in the graph)
- Brockers are also called key separators



Example of key separator



Algorithms to identify brockers are all based on some measure of the **graph connectivity**.

Algorithm for KPP_NEG (Keblady 2010)

- Let C_G be a measure of graph connectivity (e.g *reachability,* see later) for a graph G; V is the set of actors in G(nodes, vertexes)
- Algorithm KPP-neg (greedy algorithm)

Compute proposed measure of entire graph, C_G

$\forall\ v_i \in V$, remove v_i from the graph

Compute $C_{G-\{v_i\}}$ for the graph $G - \{v_i\}$.

Rank the nodes based on $|C_G$- $C_{G-\{v_i\}}|$ difference. Larger difference ranks higher.

Top ranked nodes are considered as key separators.

KPP-neg (2)

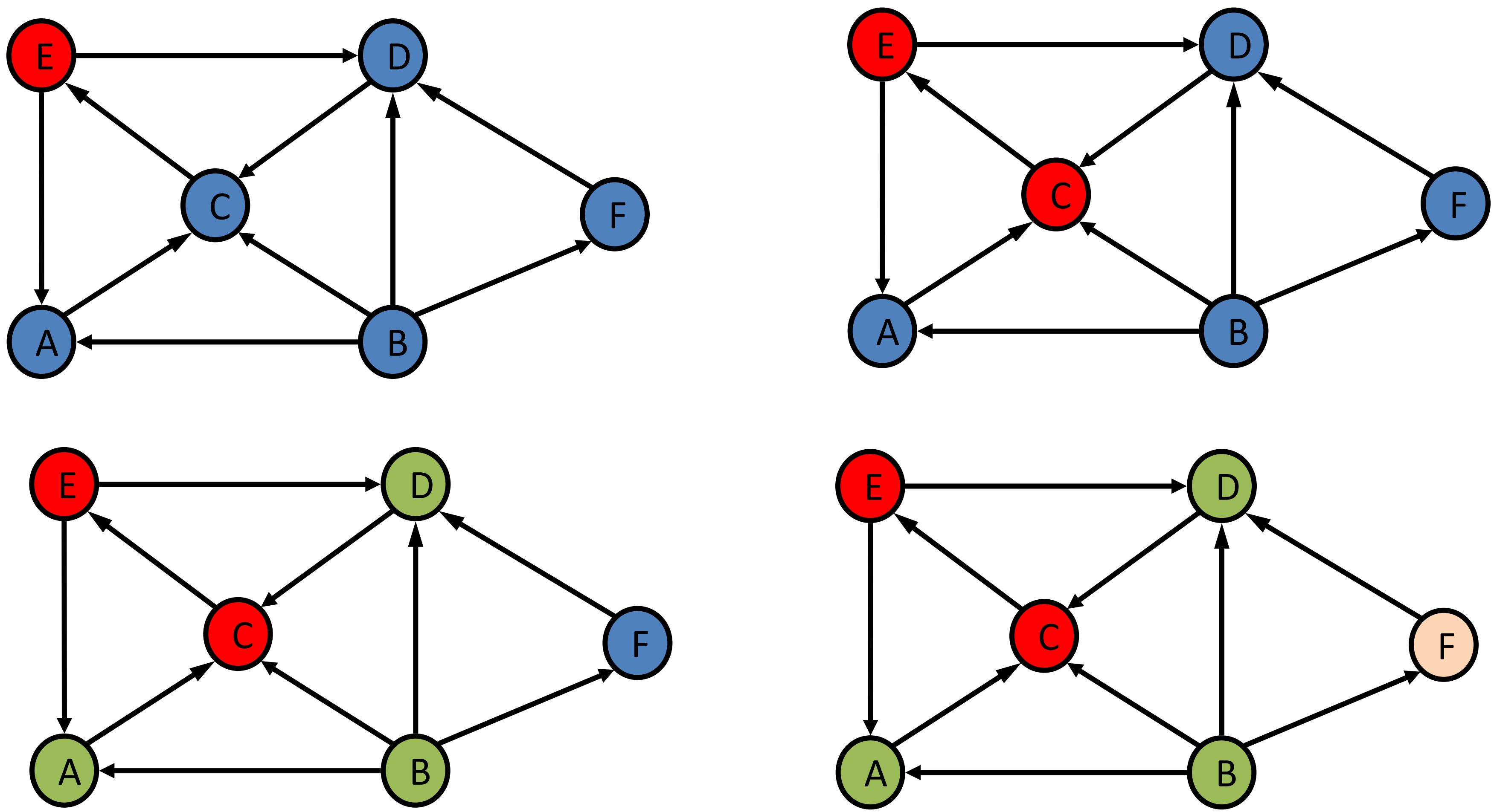
Every node reach itself

- A measure of connectivity: *reachability*

Pseudocode 1: *Reach*(v_i) – number of nodes reachable from v_i
Go to Source vertex v_i and mark it as *visited* and add to the set $Reach(v_i)$
For each adjacent vertex, A , of v_i,
If A is not already visited,
Add adjacent vertex A to the set $Reach(v_i)$ and mark A as *visited*
Call $Reach(A)$

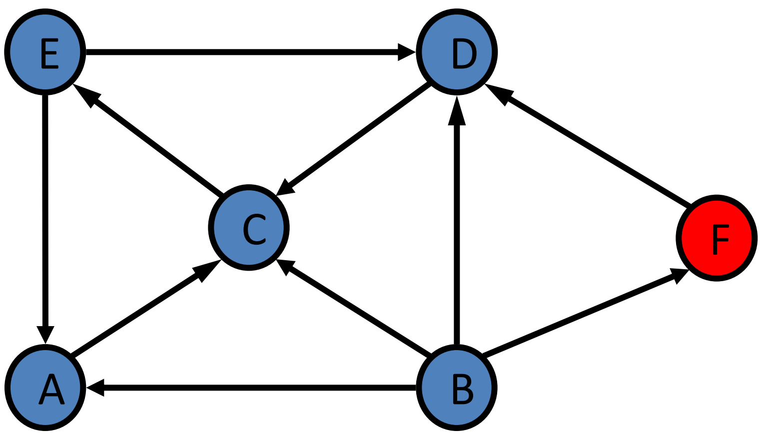
$$C_G = \sum_{i=1}^{n} Reach(v_i)$$

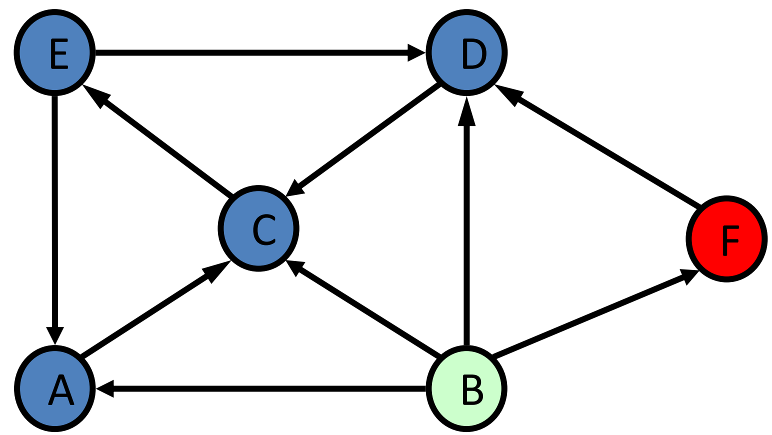
Example



R(E)= E,C,A,B,D,F

Example (2)





R(F)=F,B

NOTE: node reachability is a more accurate measure than previously seen “REACH”

Graph-based measures of social influence

1. Use graph-based methods/algorithms to identify “relevant players” in the network

   Relevant players = more influential, according to some criterion

2. **Use graph-based methods to identify <u>global</u> network properties** and communities (community detection)
3. Use graph-based methods to analyze the “spread” of information

Global Network Analysis

- Global properties of the network
- Community detection
- Spread of influence

Network Centrality

If we want to measure the degree to which the graph **as a whole** is centralized, we look at the **dispersion of centrality**:

Simple!: **variance of the individual centrality scores.**

$$S_D^2 = \left[\sum_{i=1}^{g} (C_D(n_i) - \overline{C}_d)^2 \right] / g$$

Or, using Freeman's general formula for centralization:

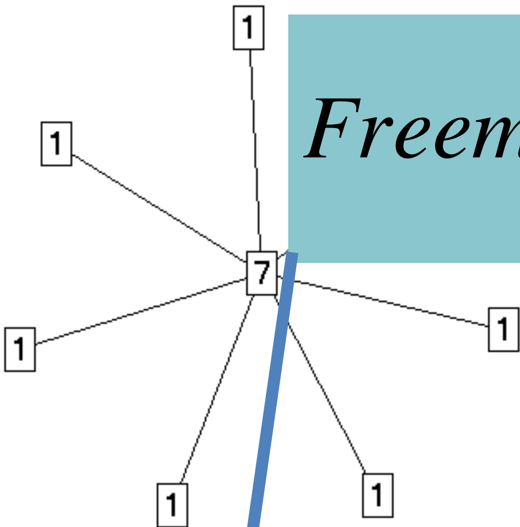
$$C_D = \frac{\sum_{i=1}^{g} \left[C_D(n^*) - C_D(n_i) \right]}{[(g-1)(g-2)]}$$

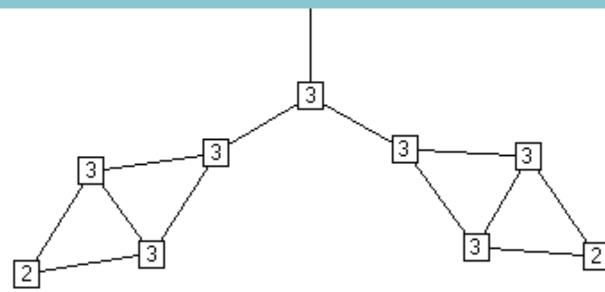
$C_D(n^*)$ is the maximum obtained value , therefore we are measuring the dispersion around that value

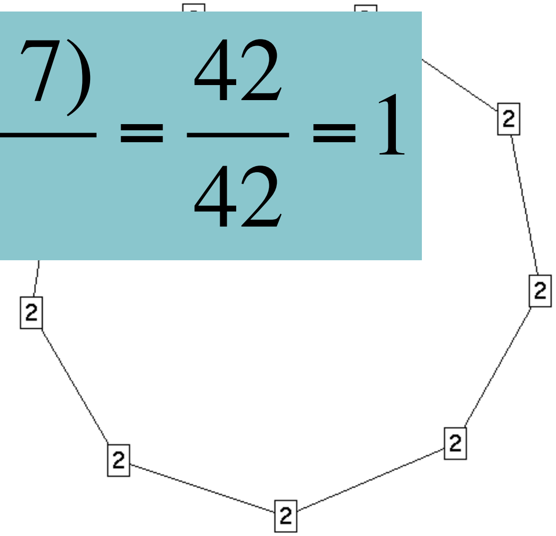
Network Centrality

Degree Centralization Scores

$$Freeman: \frac{(7-1)\times 7+(7-7)}{(8-1)(8-2)} = \frac{42}{42} = 1$$



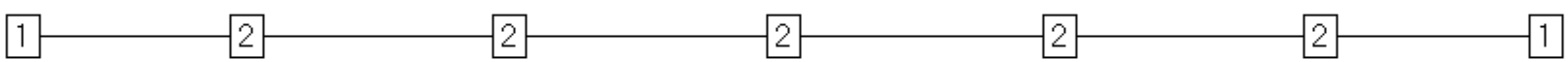




Freeman: 1.0
Variance: 3.9

Freeman: .02
Variance: .17

Freeman: 0.0
Variance: 0.0



Freeman: .07
Variance: .20

Global Network Analysis

- Global properties of the network
- **Community detection**
- Spread of influence

Community detection

- Community: It is formed by individuals such that those within a group interact with each other **more frequently than with those outside the group**
  - a.k.a. group, cluster, cohesive subgroup, module in different contexts
- Community detection: discovering groups in a network where individuals' group memberships are not explicitly given
- (next lesson)