Social and Information Networks

CSCC46H, Fall 2022
Lecture 2

Prof. Ashton Anderson
ashton@cs.toronto.edu

Logistics

Tutorials on Tuesdays and Thursdays, starting next week (I'm sorry for yesterday, TUT01 students! 🤦)

TUT0003 time change from Fridays 9–10am to Thursdays 3–4pm (please let me know if this causes any issues)

My office hours are Weds 4:30–5:30pm

Please answer the polls and introduce yourself in Discord #general (thanks to those who have already!)

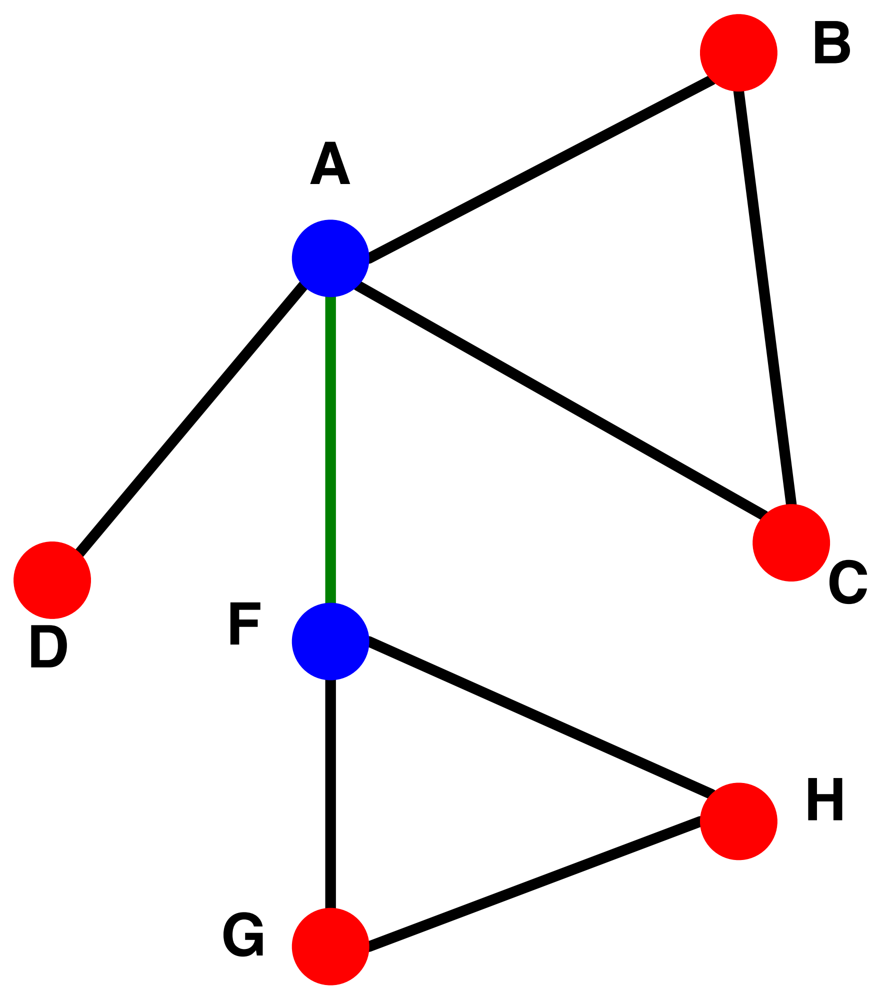
A1 out next week

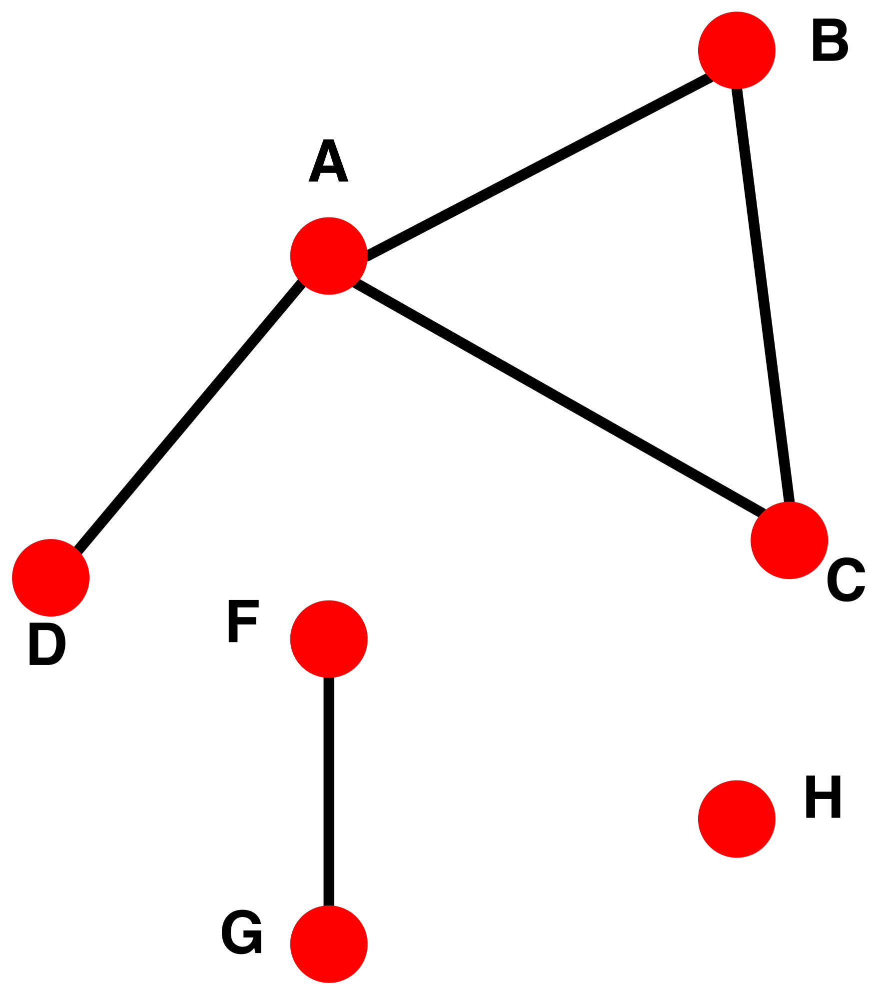
Today

1) Graph structure of the Web
2) Building up our network vocabulary
3) Measuring networks; basic properties
4) Random graph model: G_{np}

Connectivity of Graphs

- **Connected component (undirected):**
  - Any two vertices can be joined by a path
  - No superset with the same property
- A disconnected graph is made up of two or more connected components



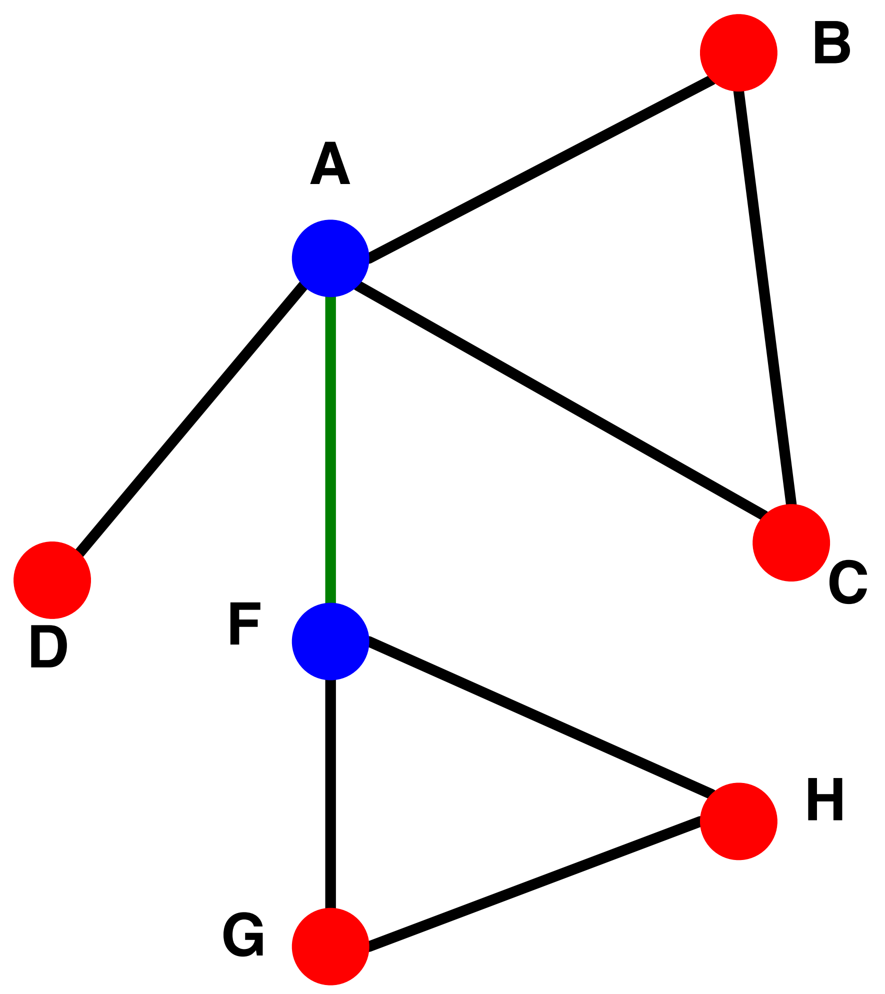


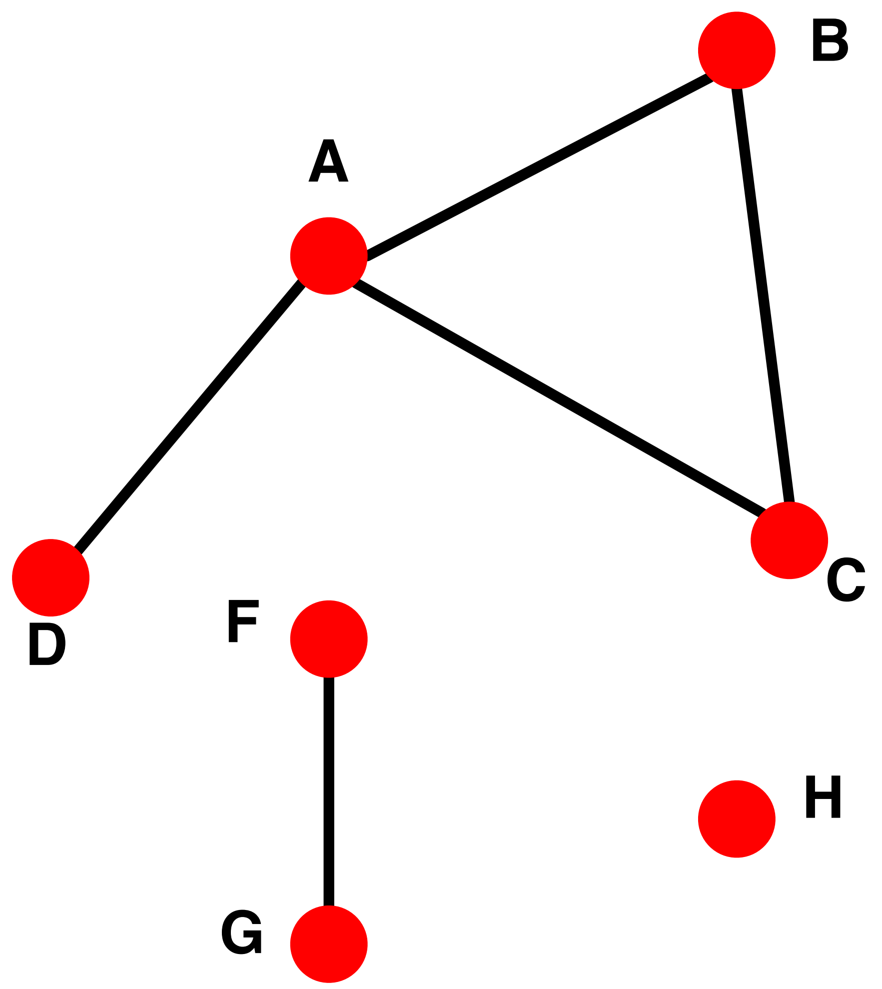
Largest Component:
Giant Component

Isolated node (node H)

Connectivity of Graphs

- **Connected component (undirected):**
  - Any two vertices can be joined by a path
  - No superset with the same property
- A disconnected graph is made up of two or more connected components





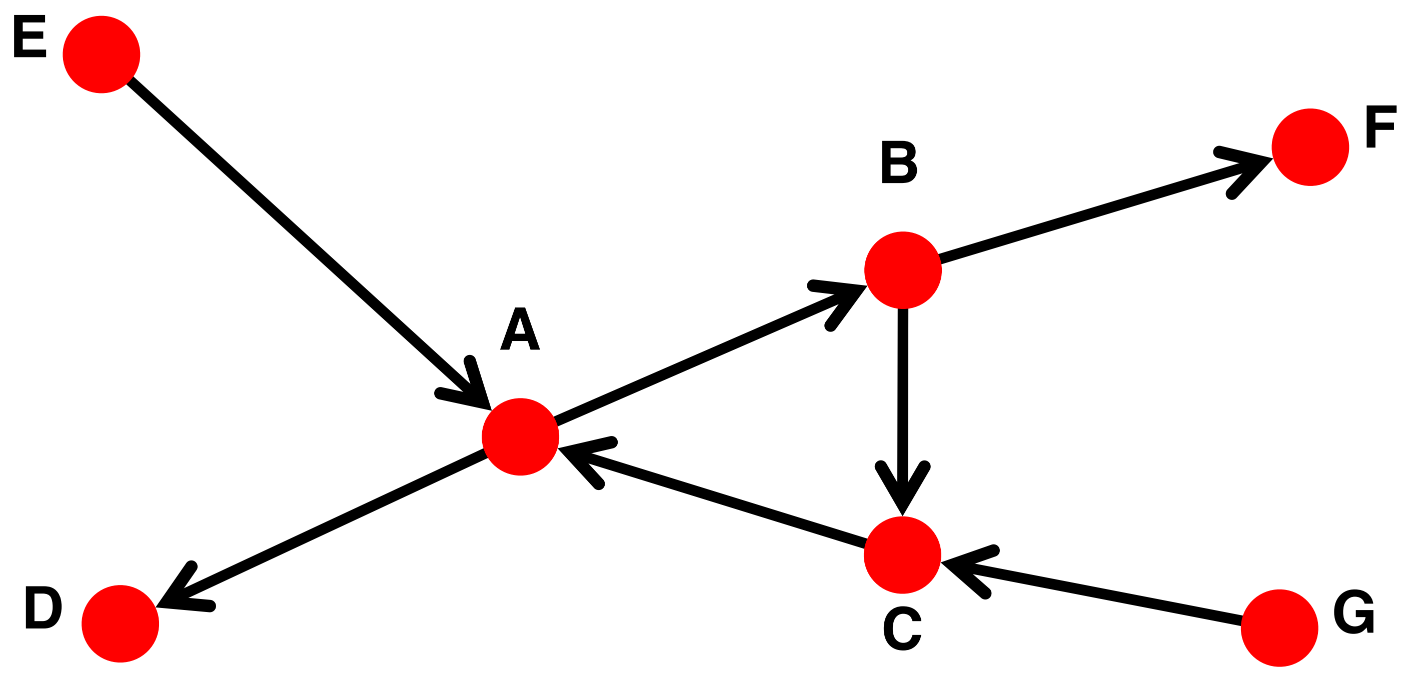
Largest Component:
Giant Component

Isolated node (node H)

Bridge edge: If we erase it, the graph becomes disconnected.

Connectivity of Directed Graphs

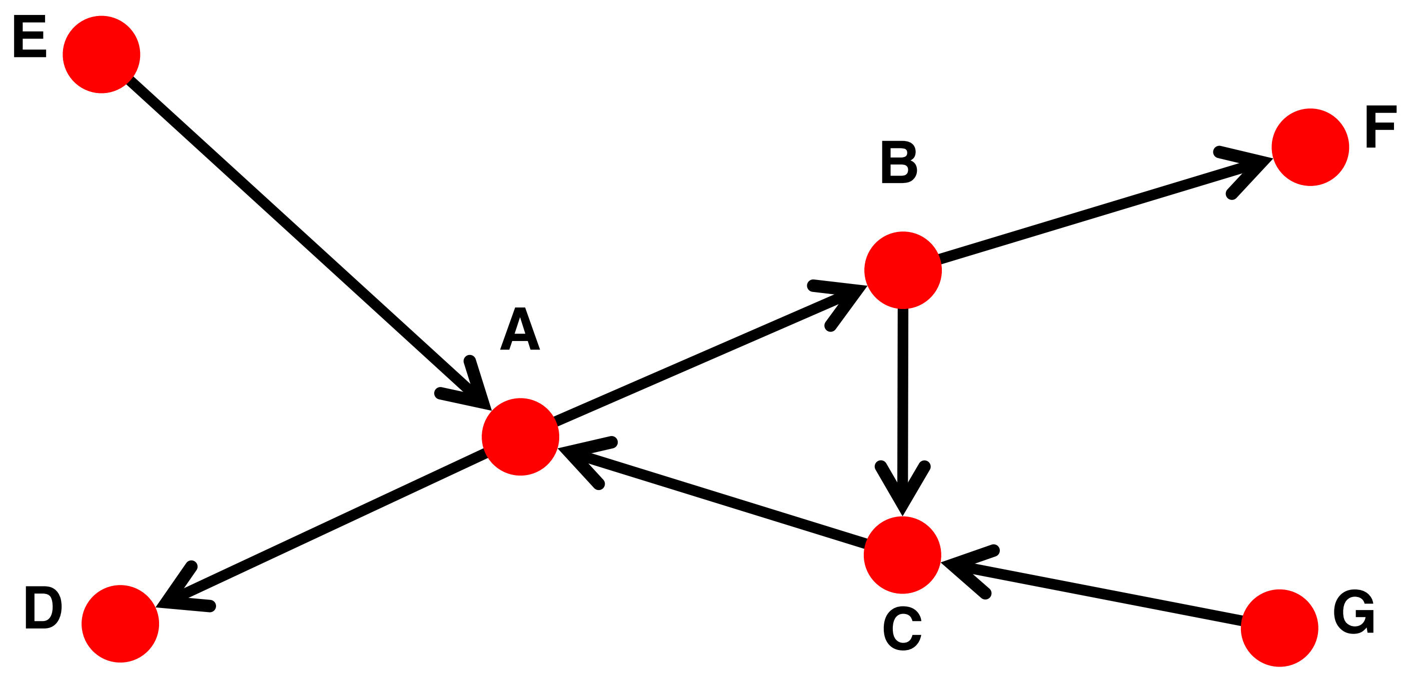
- **Strongly connected directed graph**
  - has a path from each node to every other node and vice versa (e.g., A-B path and B-A path)
- **Weakly connected directed graph**
  - is connected if we disregard the edge directions



Is this graph weakly connected?
Strongly connected?

Connectivity of Directed Graphs

- **Strongly connected directed graph**
  - has a path from each node to every other node and vice versa (e.g., A-B path and B-A path)
- **Weakly connected directed graph**
  - is connected if we disregard the edge directions



It is weakly connected but not strongly connected (e.g., there is no way to get from F to G by following the edge directions)

What is the large-scale structure of the Web?

The Structure of the Web

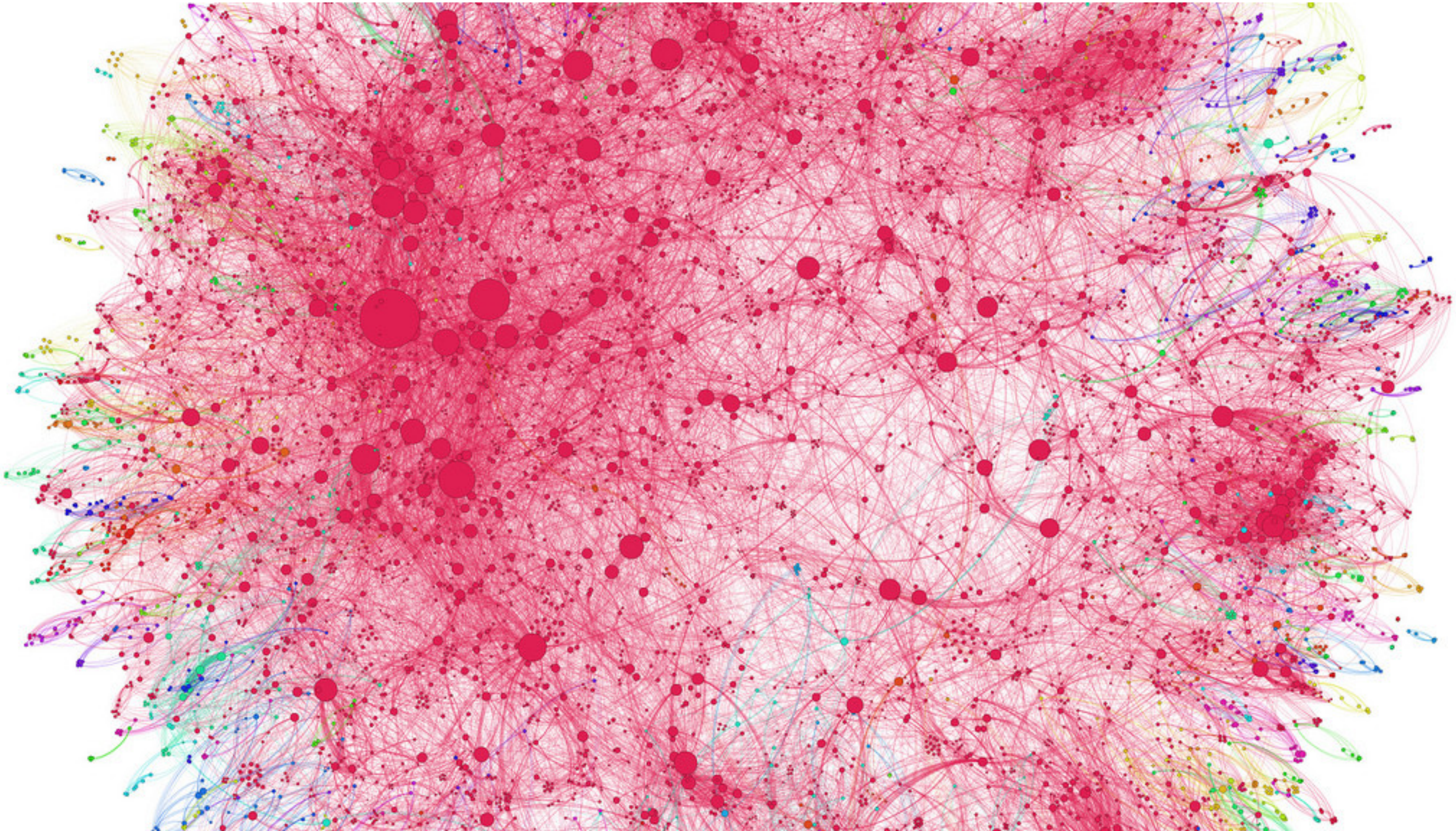
- Q: What does the Web "look like"?

The Structure of the Web

- Q: What does the Web "look like"?

The Structure of the Web

A network!



Web as a Graph

Here is what we will do next:

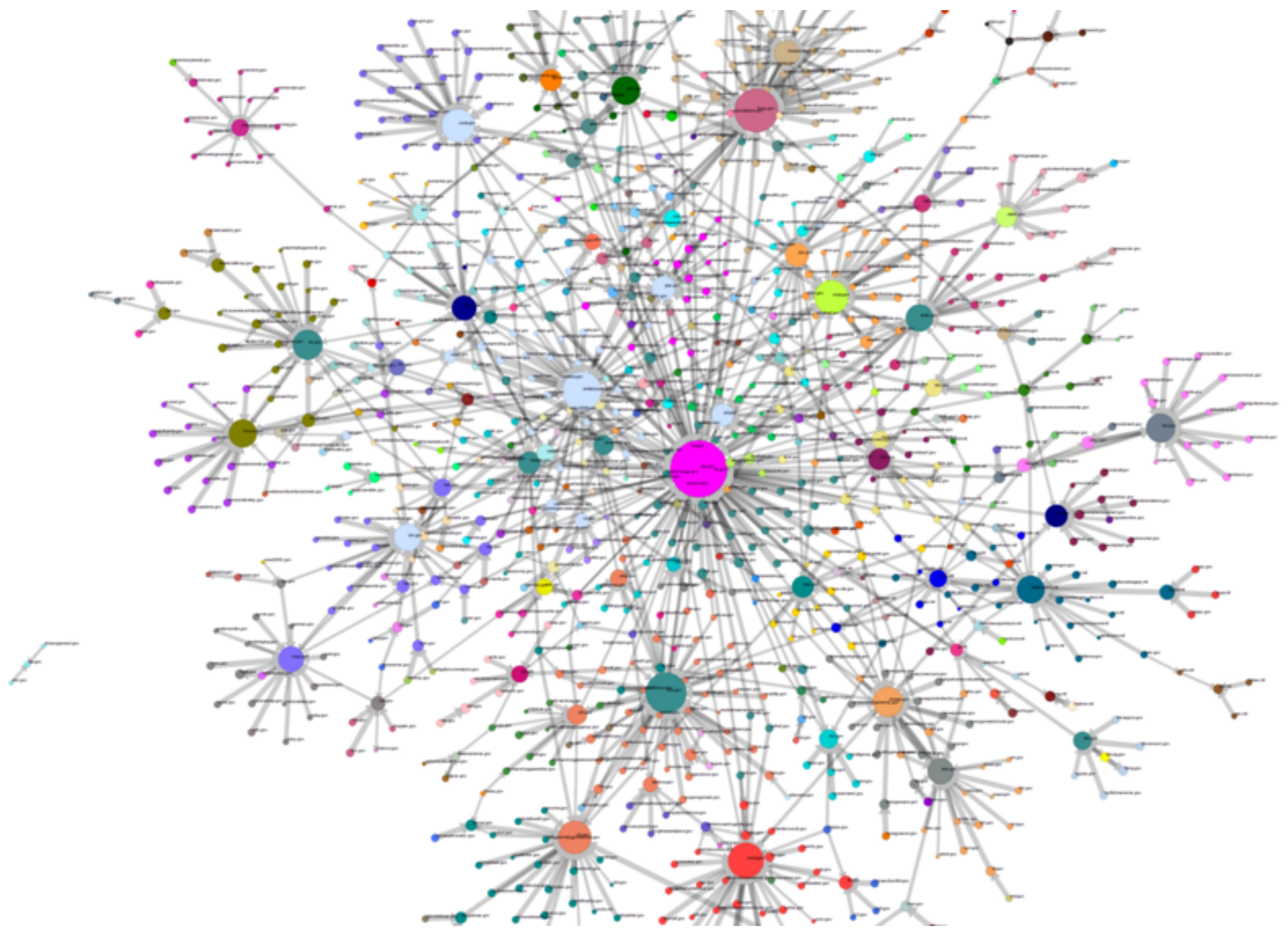
- We will take a real system (i.e., the Web)
- We will represent the Web as a graph
- We will use language of graph theory to reason about the structure of the graph
- Do a computational experiment on the Web graph
- **Learn something about the structure of the Web!**

Web as a Graph

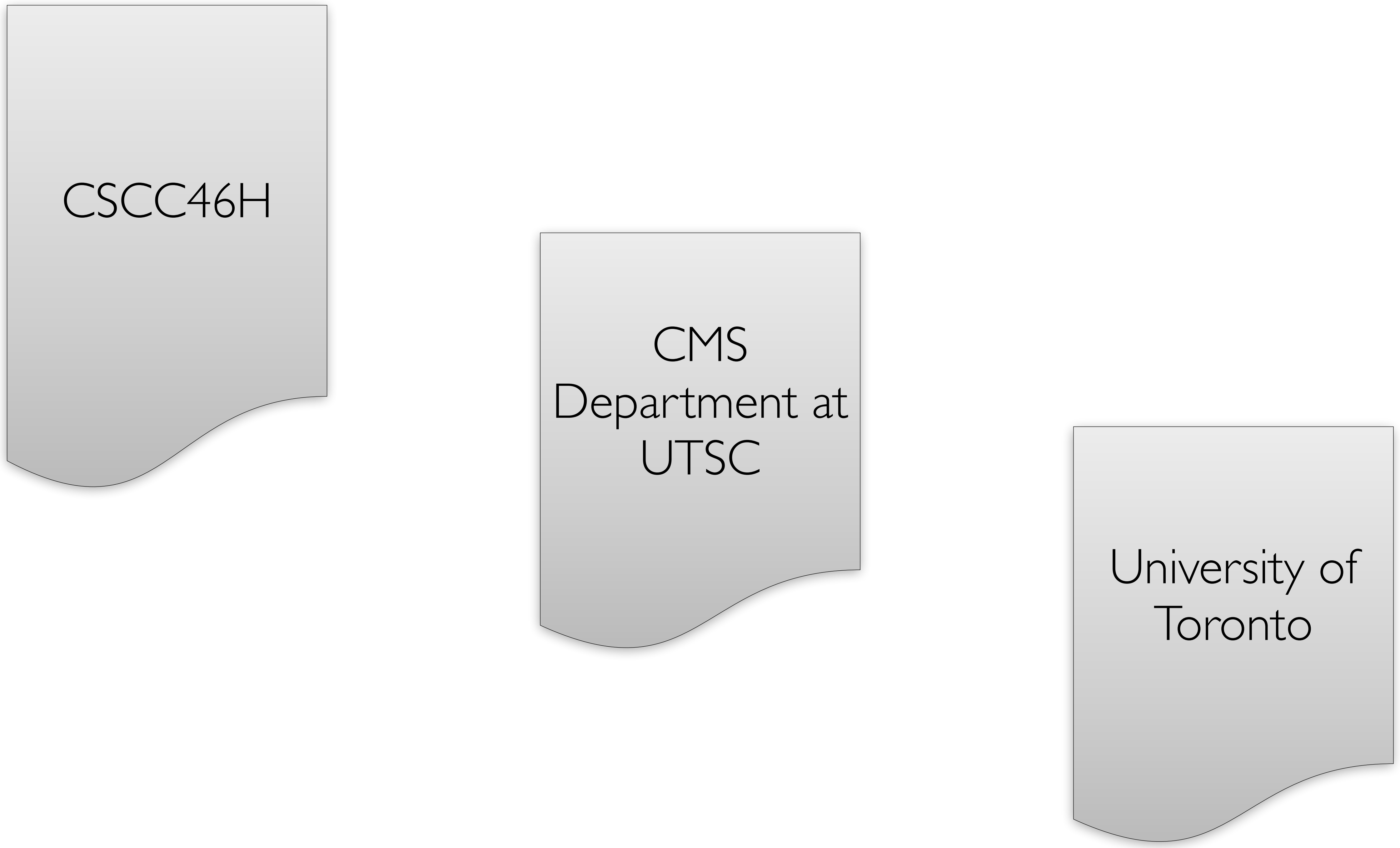
Q: What does the Web "look like" at a global level?

- **Web as a graph:**
  - Nodes = web pages
  - Edges = hyperlinks

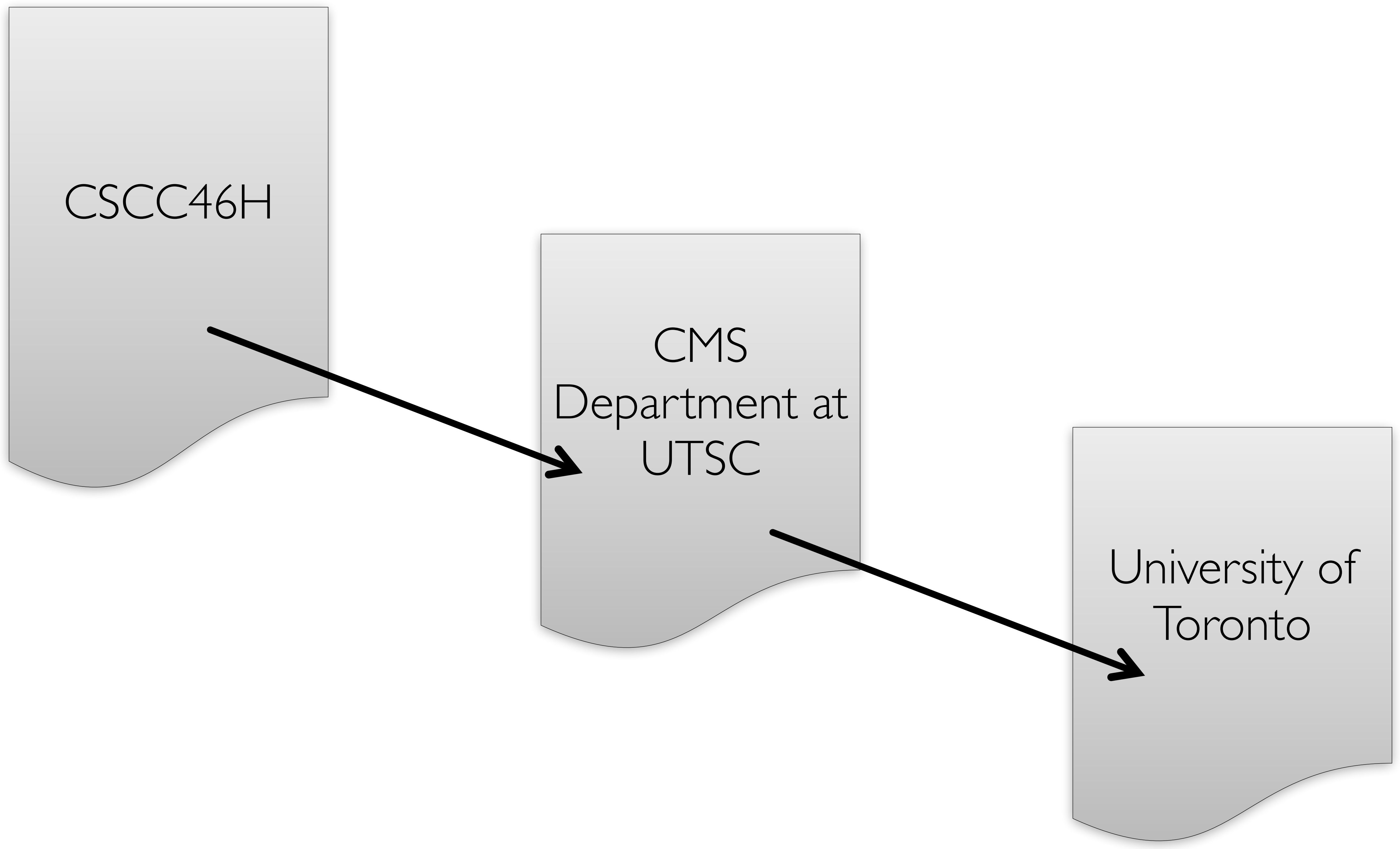
  - **Side issue:** What is a node?
    - Dynamic pages created on the fly
    - "dark matter" – inaccessible database generated pages



The Web as a Graph

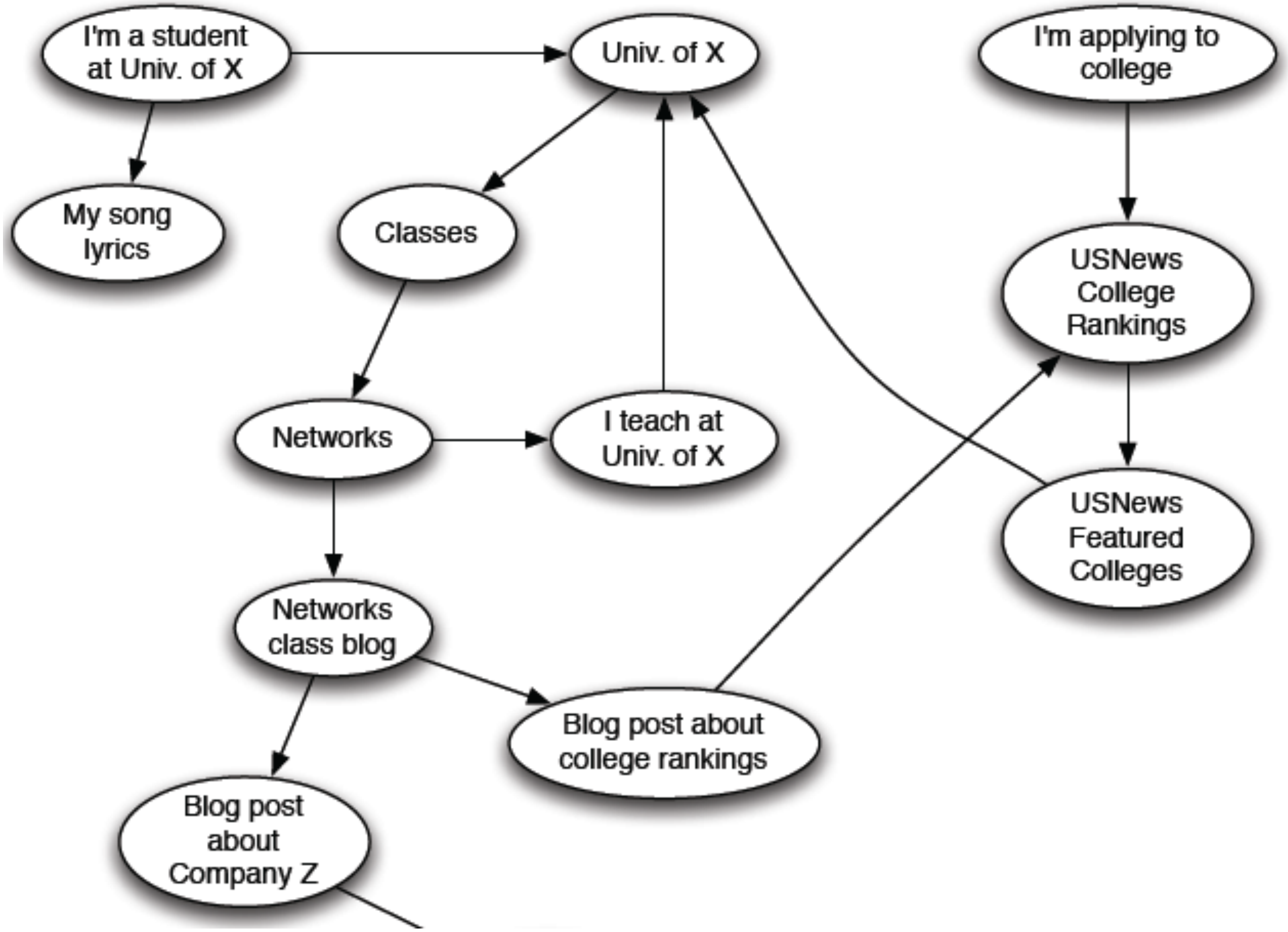


The Web as a Graph

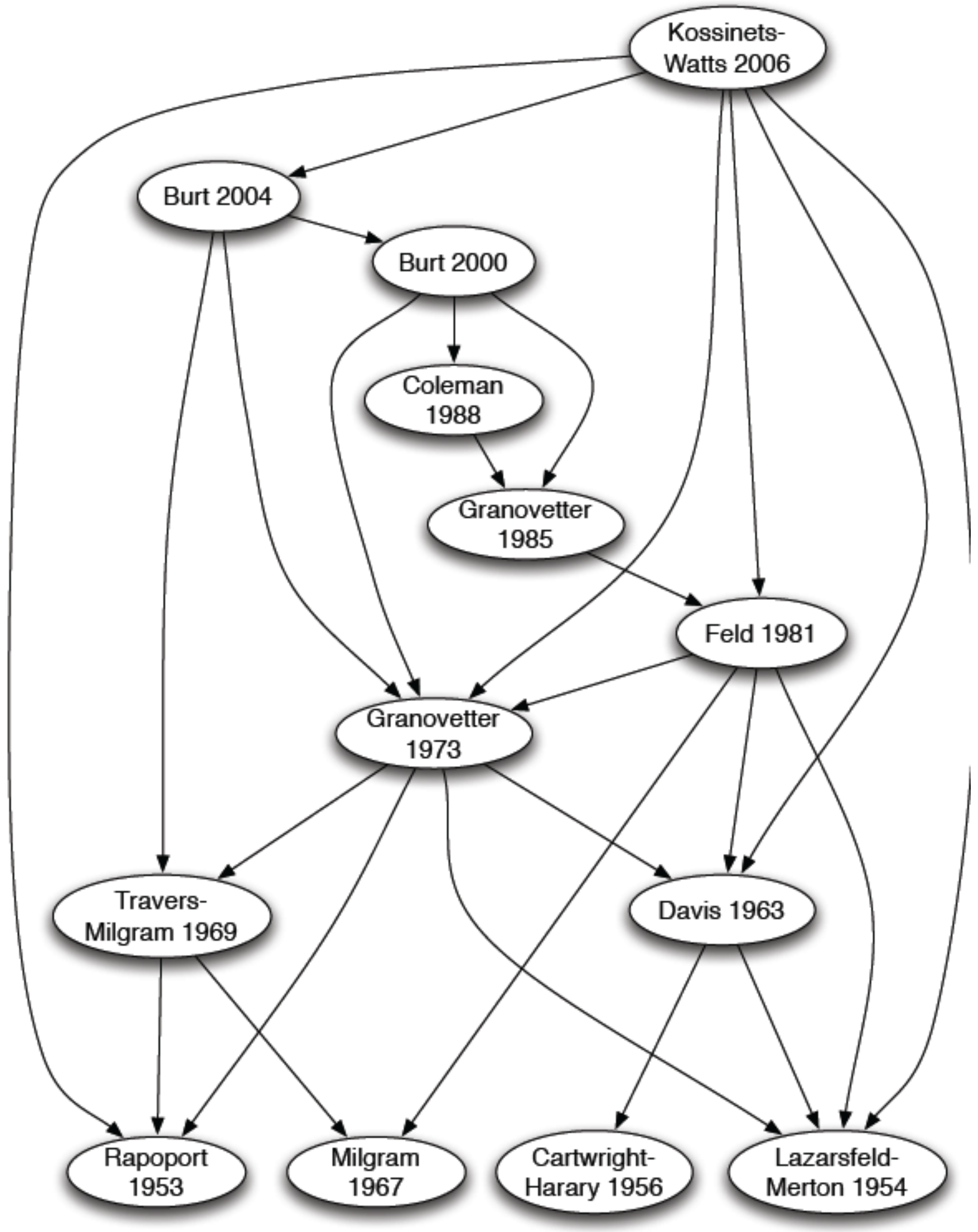


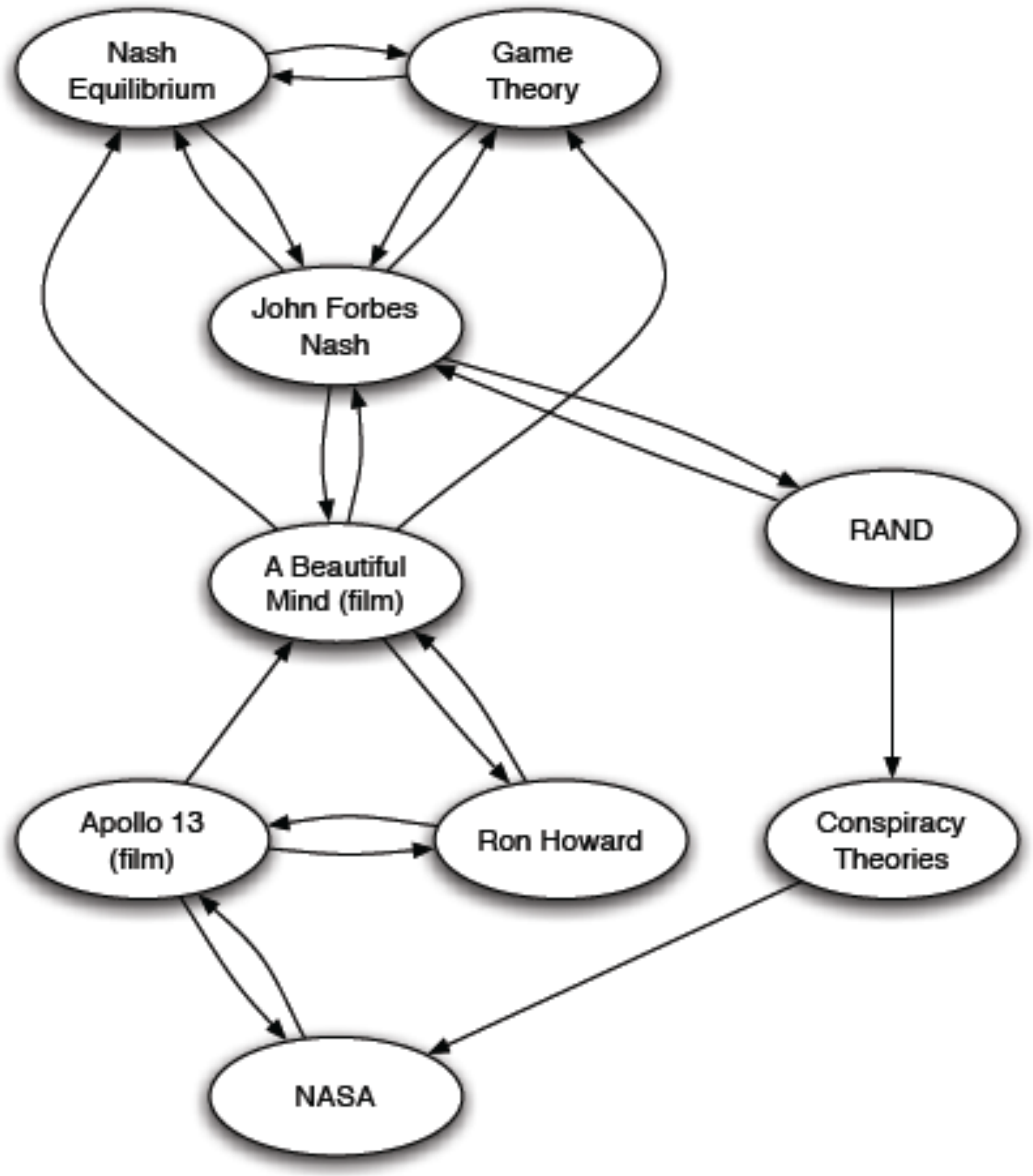
- In early days of the Web links were **navigational**
- Today many links are **transactional**

The Web as a Directed Graph



Other Information Networks

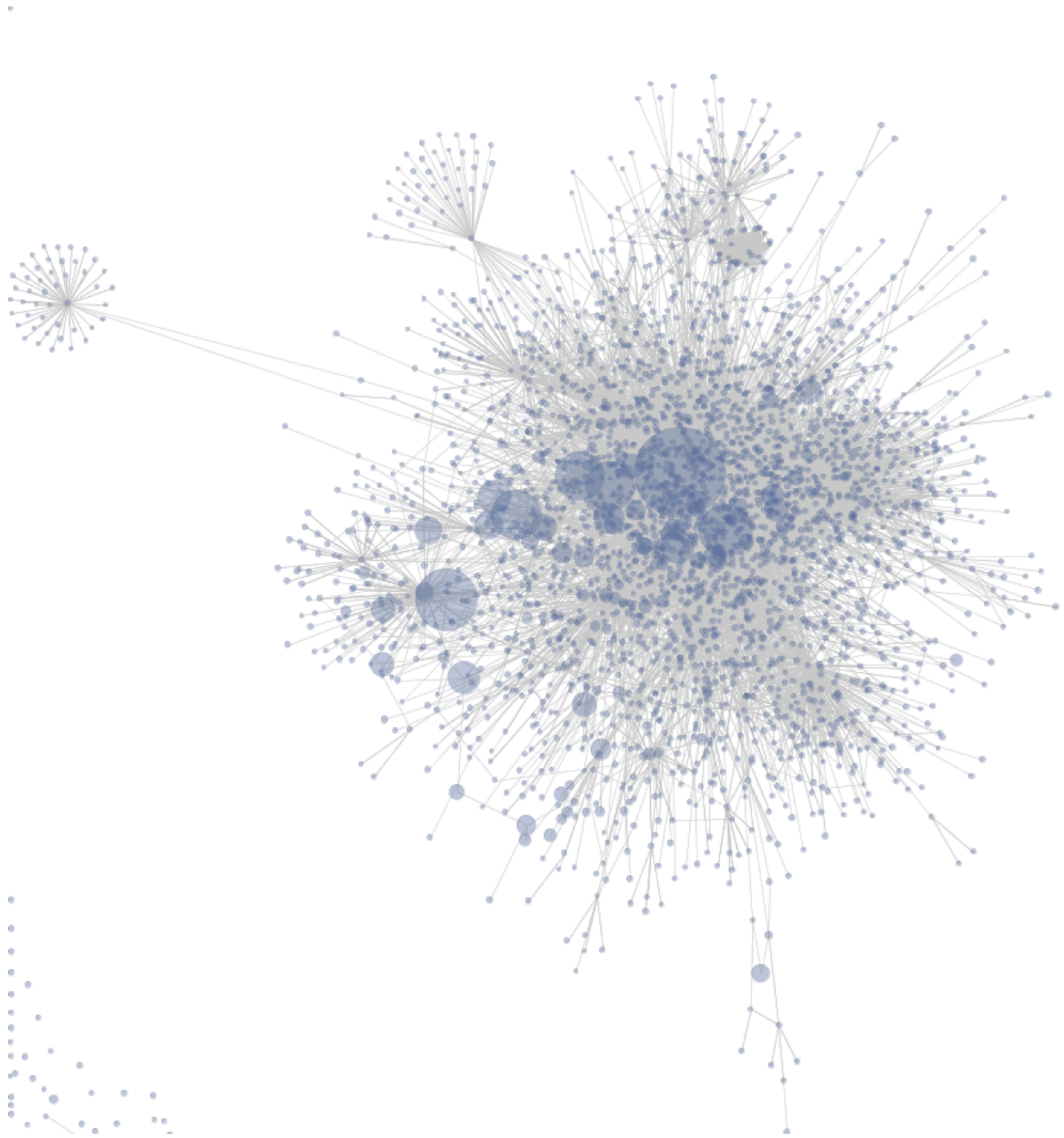




Citations

References in an encyclopedia

Other Information Networks



References between pages in a part of Wikipedia

What Does the Web Look Like?

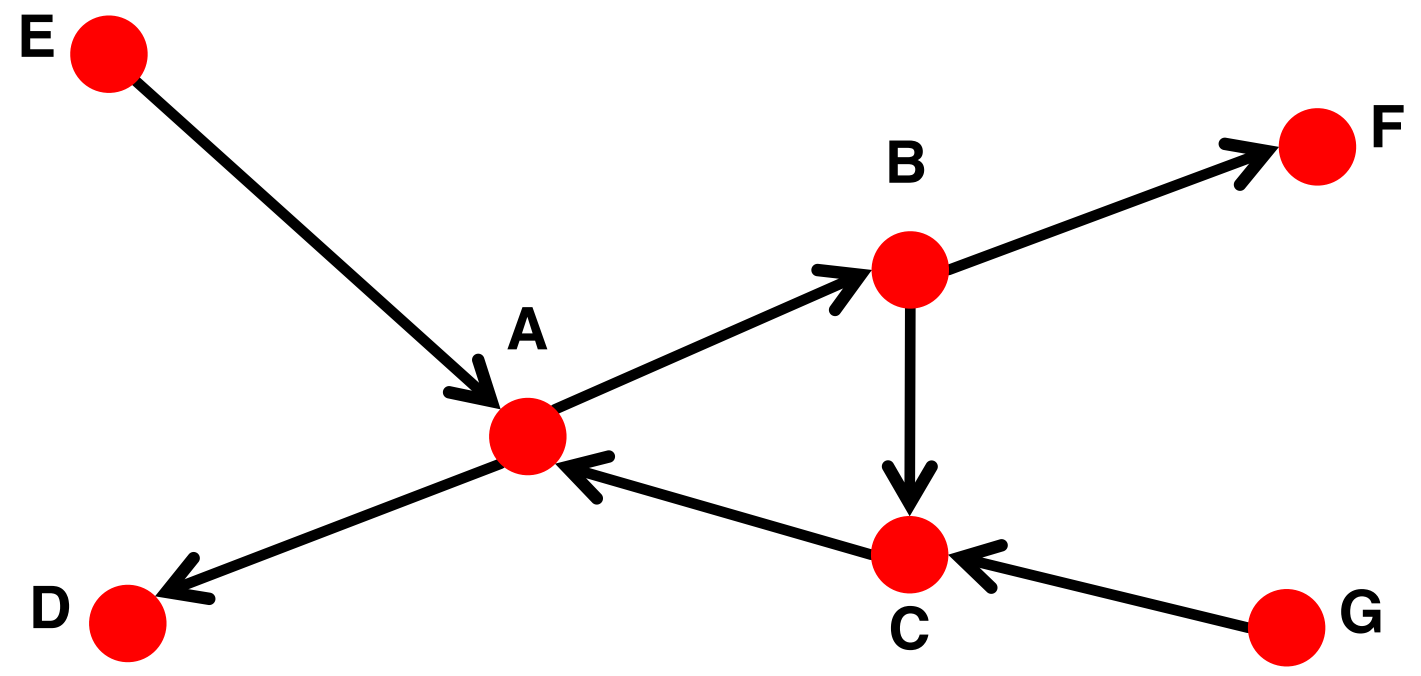
- How is the Web linked?
- What is the “map” of the Web?

What Does the Web Look Like?

- **How is the Web linked?**
- **What is the “map” of the Web?**

Web as a directed graph [Broder et al. 2000]:

- Given node ***v***, what can ***v*** reach?
- What other nodes can reach ***v***?



$$In(v) = \{w \mid w \text{ can reach } v\}$$
$$Out(v) = \{w \mid v \text{ can reach } w\}$$

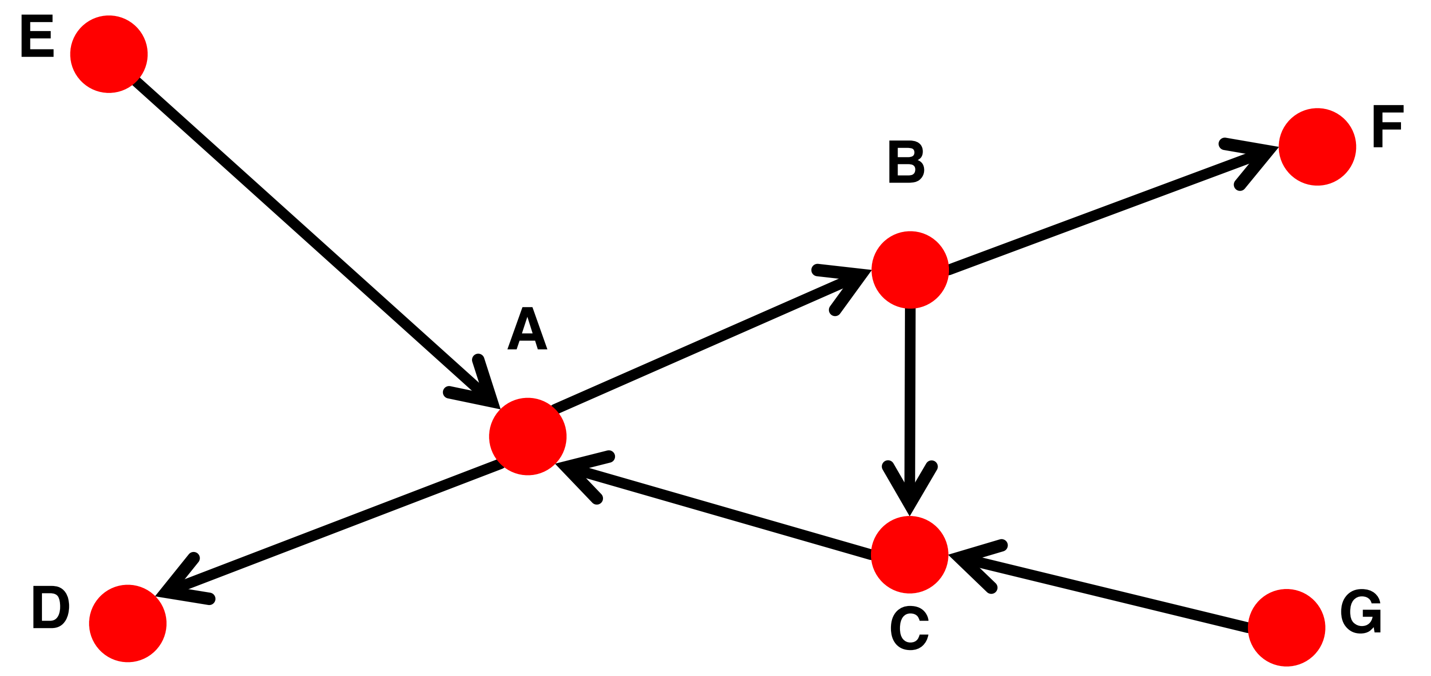
For example:
In(A) = {?}
Out(A)={?}

What Does the Web Look Like?

- **How is the Web linked?**
- **What is the "map" of the Web?**

Web as a <u>directed graph</u> [Broder et al. 2000]:

- Given node ***v***, what can ***v*** reach?
- What other nodes can reach ***v***?



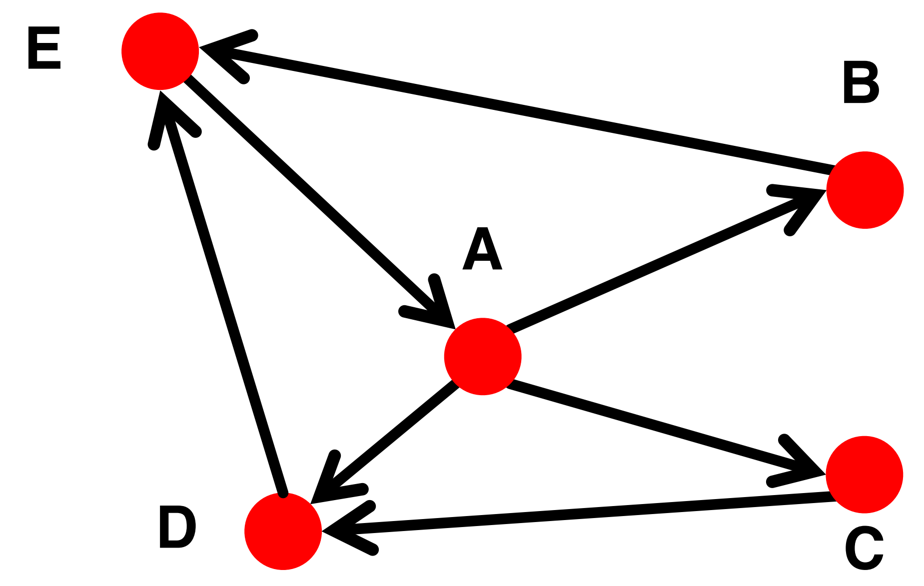
$$In(v) = \{w \mid w \text{ can reach } v\}$$
$$Out(v) = \{w \mid v \text{ can reach } w\}$$

For example:
In(A) = {A,B,C,E,G}
Out(A)={A,B,C,D,F}

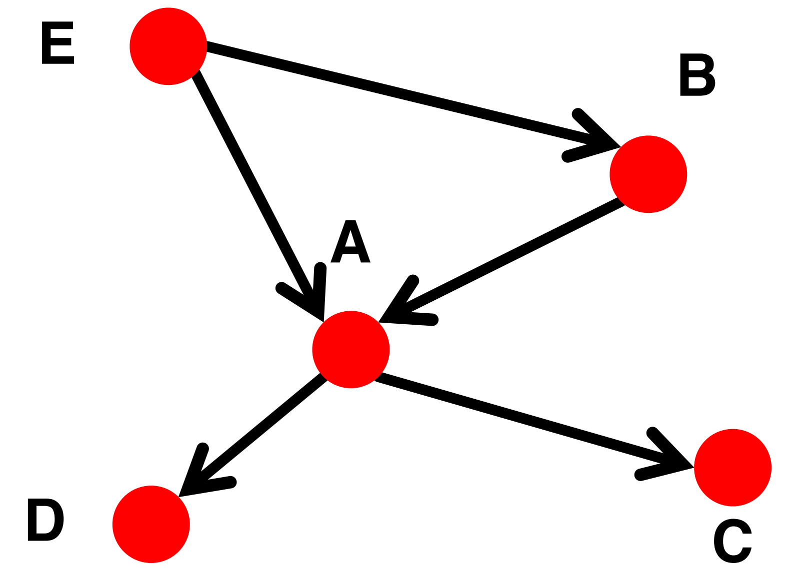
Directed Graphs

- **Two types of directed graphs:**
  - **Strongly connected graph:**
    - Any node can reach any node via a directed path

      In(A)=Out(A)={A,B,C,D,E}



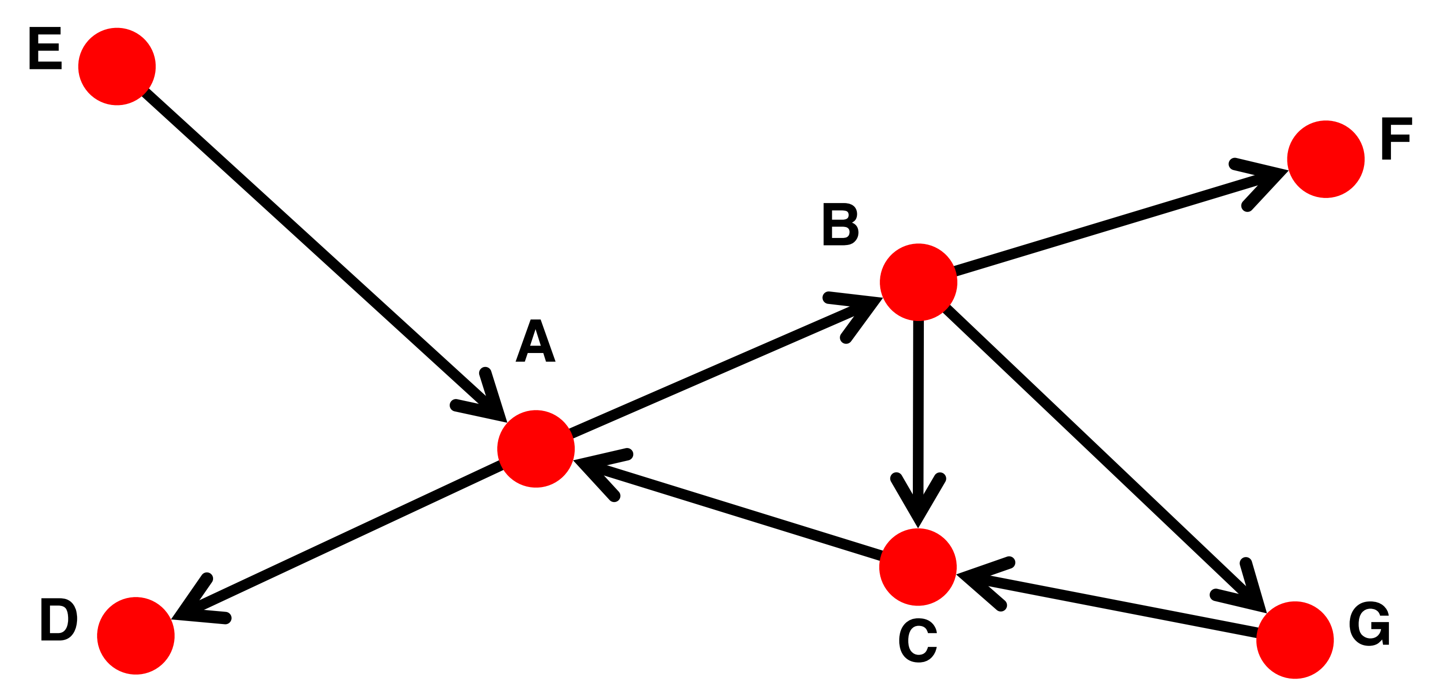
  - **DAG – Directed Acyclic Graph:**
    - Has no cycles: if ***u*** can reach ***v***, then ***v*** can not reach ***u***



Any directed graph can be expressed in terms of these two types!

Strongly Connected Component

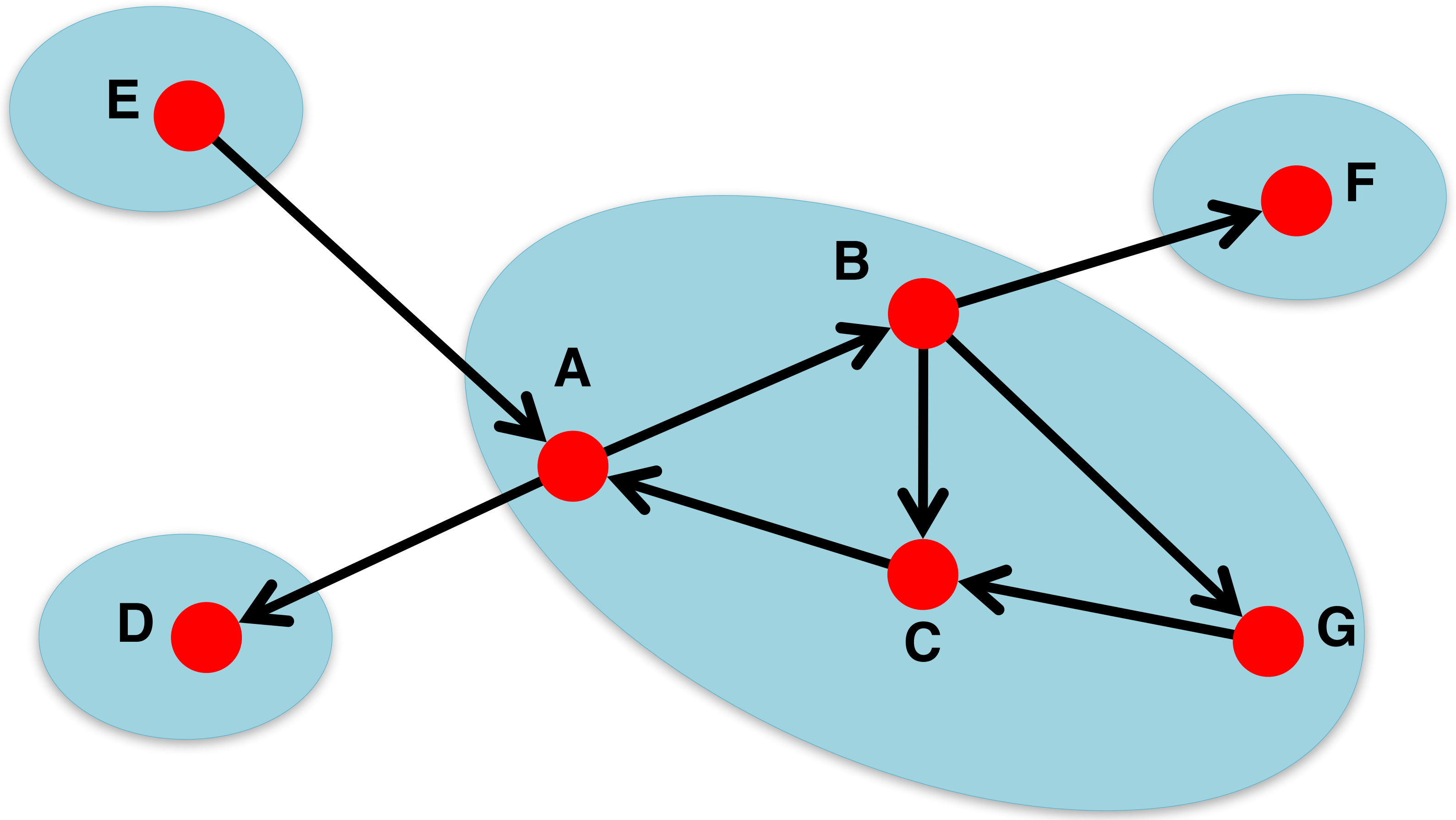
- **Strongly connected component (SCC)** is a set of nodes ***S*** so that:
  - Every pair of nodes in ***S*** can reach each other
  - There is no larger set containing ***S*** with this property



What are the strongly connected components of this graph?

Strongly Connected Component

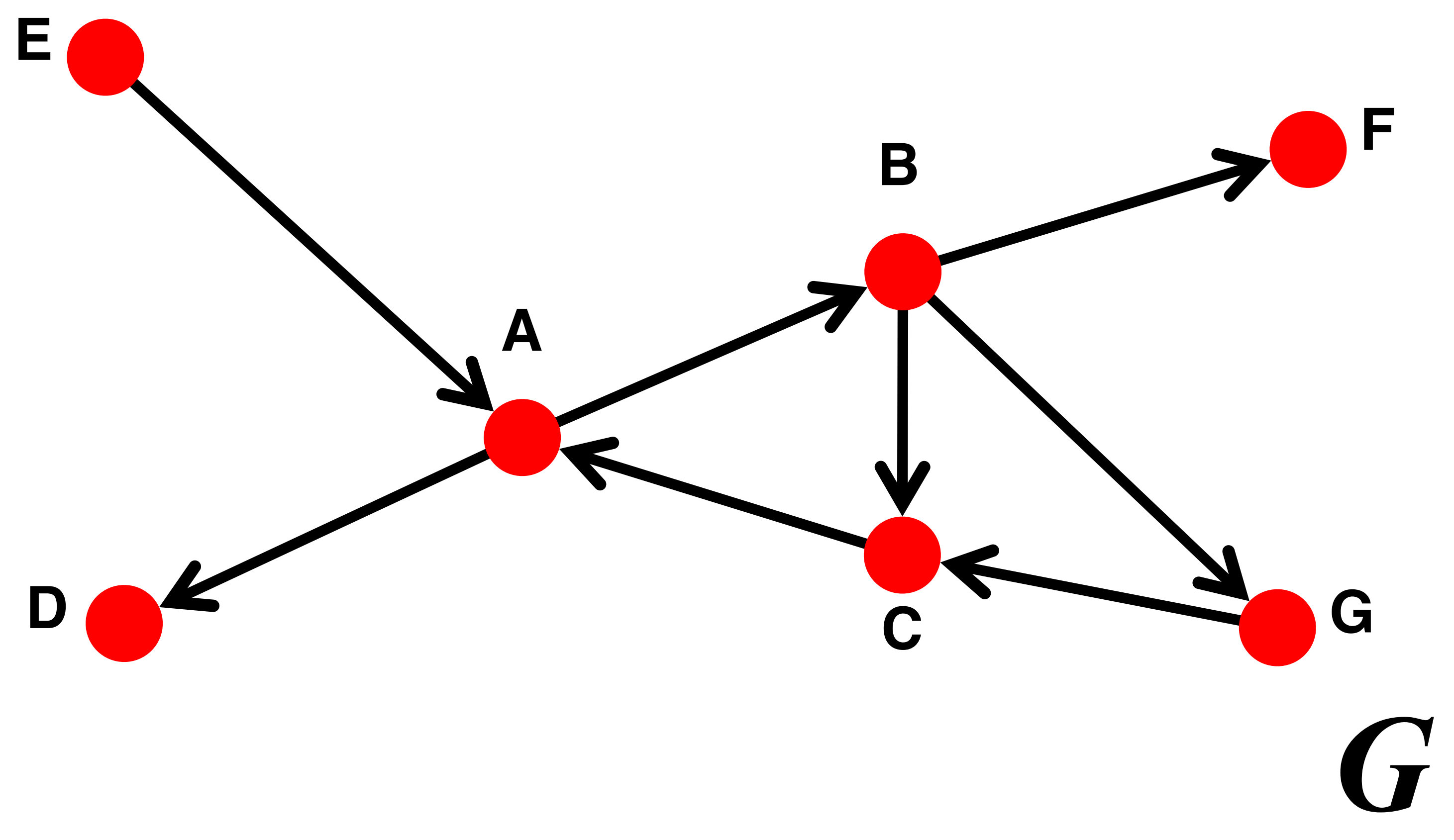
- **Strongly connected component (SCC)** is a set of nodes ***S*** so that:
  - Every pair of nodes in ***S*** can reach each other
  - There is no larger set containing ***S*** with this property



Strongly connected components of the graph: {A,B,C,G}, {D}, {E}, {F}

Strongly Connected Component

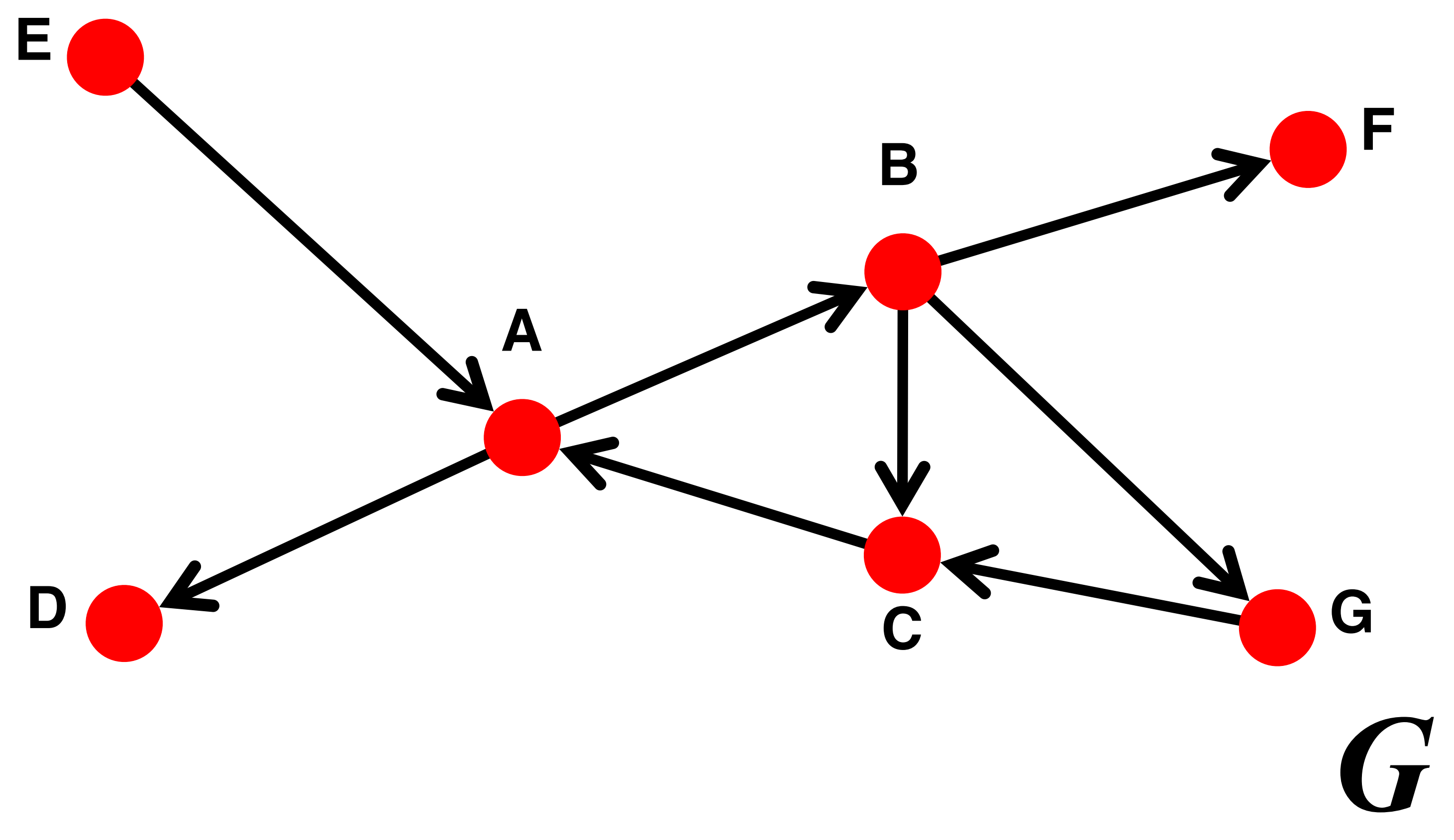
- **Fact:** **Every directed graph is a DAG on its SCCs**
  - **(1)** SCCs partitions the nodes of ***G***
    - That is, each node is in exactly one SCC
  - **(2)** If we build a graph ***G'*** whose nodes are SCCs, and with an edge between nodes of ***G'*** if there is an edge between corresponding SCCs in ***G***, then ***G'*** is a DAG



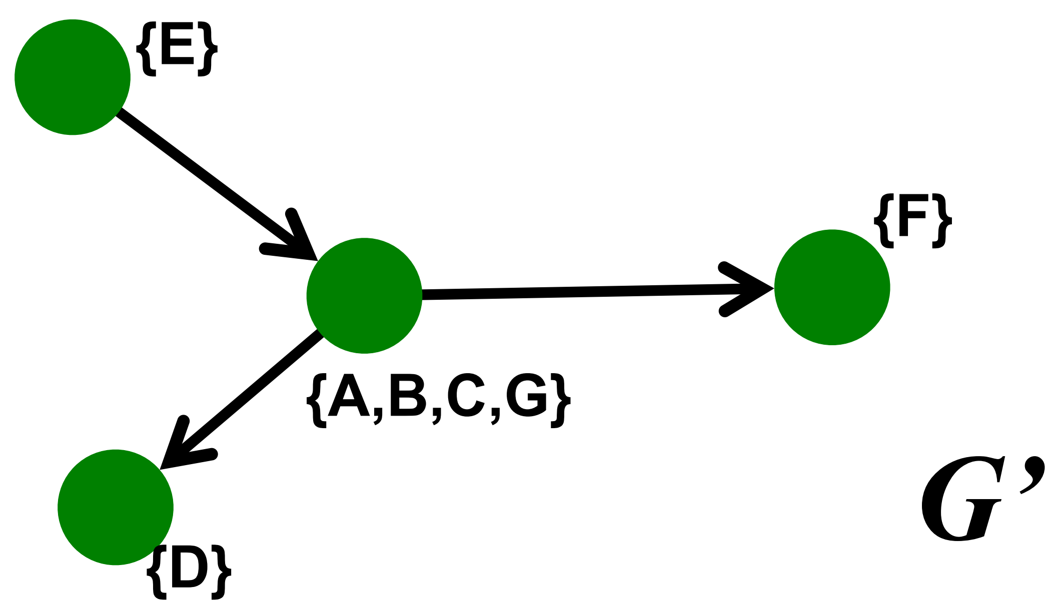
G'?

Strongly Connected Component

- **Fact:** **Every directed graph is a DAG on its SCCs**
  - **(1)** SCCs partitions the nodes of ***G***
    - That is, each node is in exactly one SCC
  - **(2)** If we build a graph ***G'*** whose nodes are SCCs, and with an edge between nodes of ***G'*** if there is an edge between corresponding SCCs in ***G***, then ***G'*** is a DAG

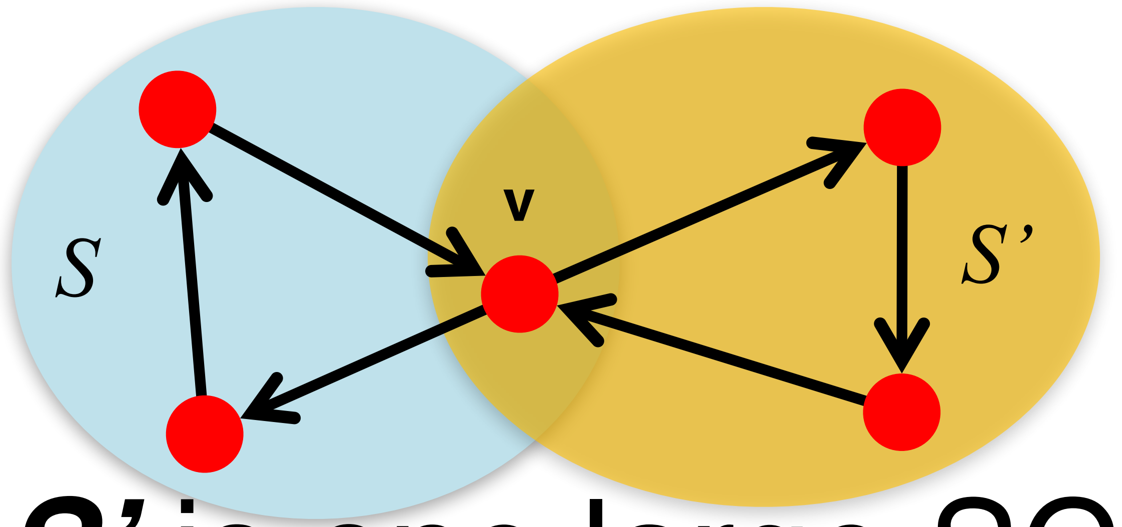


(1) Strongly connected components of graph G: {A,B,C,G}, {D}, {E}, {F}
(2) G' is a DAG:



Proof of (1)

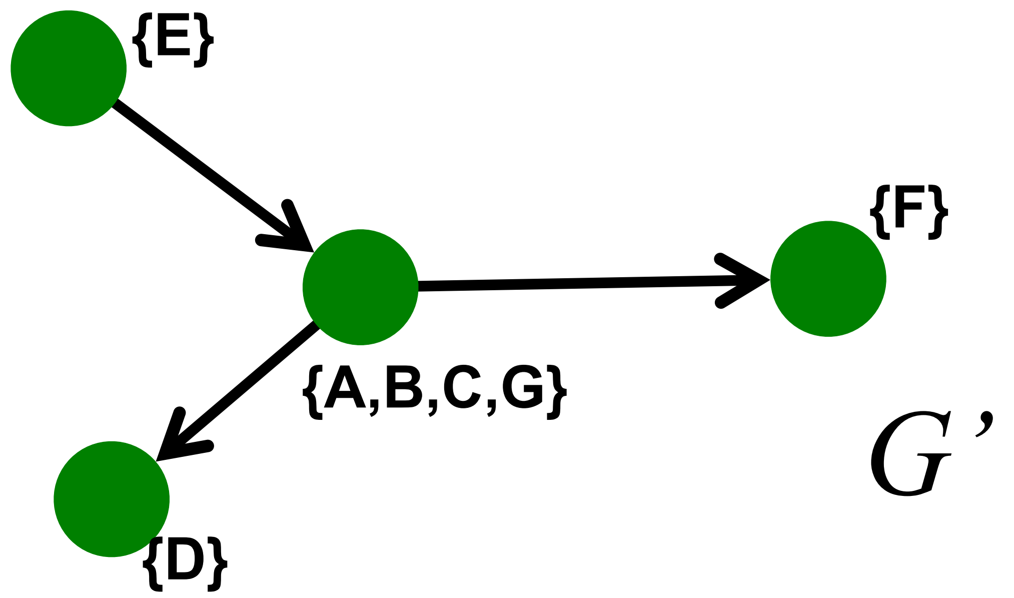
- **Claim: SCCs partitions nodes of G.**
  - This means: Each node is member of exactly 1 SCC
- Proof by contradiction:
  - Suppose there exists a node ***v*** which is a member of two SCCs ***S*** and ***S'***

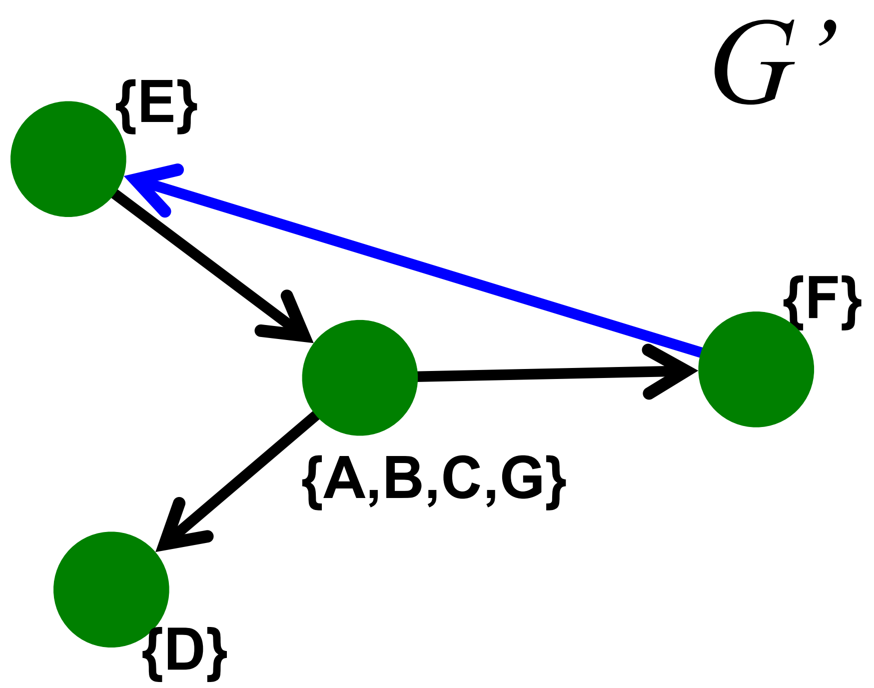


  - But then ***S*** ∪ ***S'*** is one large SCC!
    - Contradiction!

Proof of (2)

- **Claim: *G'* (graph of SCCs) is a DAG.**
  - This means: ***G'*** has no cycles
- Proof by contradiction:
  - Assume ***G'*** is not a DAG
  - Then ***G'*** has a directed cycle
  - Now all nodes on the cycle are mutually reachable, and all are part of the same SCC
  - But then ***G'*** is not a graph of connections between SCCs (SCCs are defined as maximal sets)
    - Contradiction!



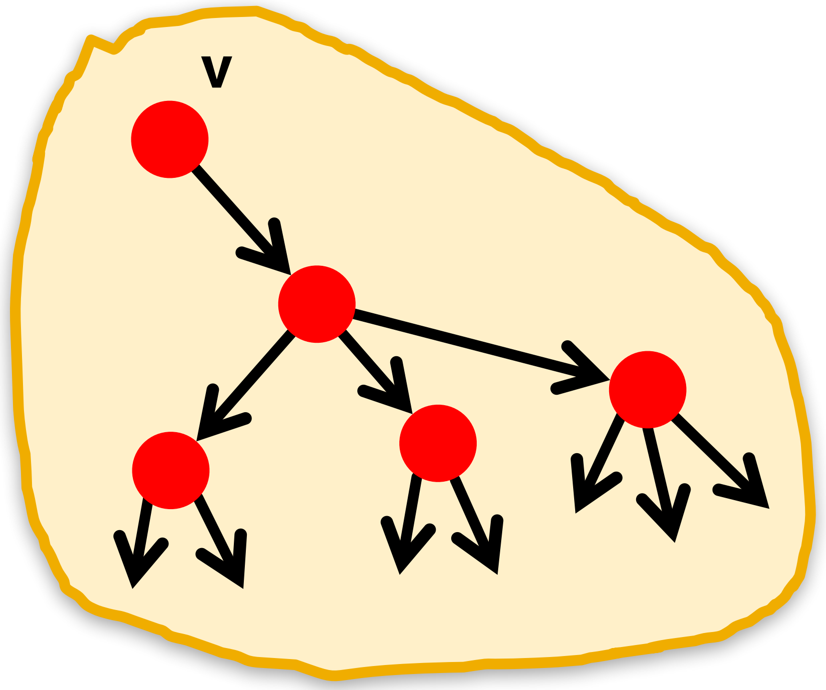


Now {A,B,C,G,E,F} is a SCC!

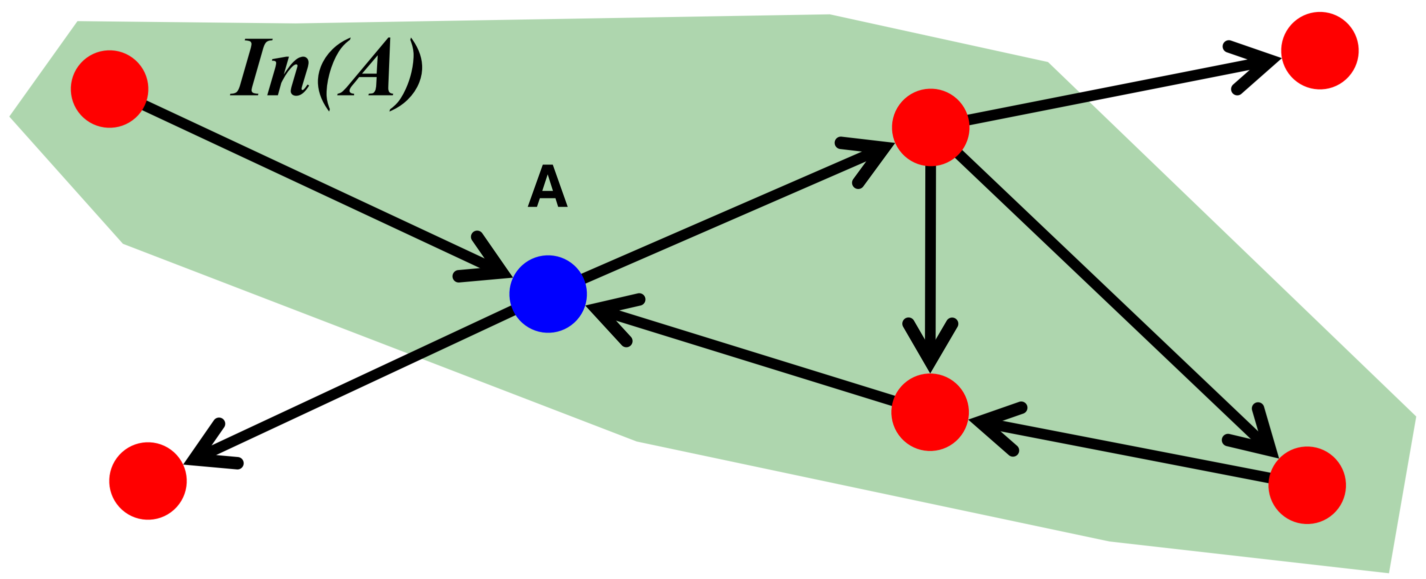
Graph Structure of the Web

- **<u>Goal:</u> Take a large snapshot of the Web and try to understand how its SCCs "fit together" as a DAG**
- **Computational issue:**
  - Want to find a SCC containing node ***v***?
  - **Observation:**
    - ***Out(v)*** … nodes that can be reached from *v*
    - **SCC containing *v* is:** $Out(v) \cap In(v)$

      $= Out(v,G) \cap Out(v,\overline{G})$, where $\overline{G}$ is G with all edge directions flipped

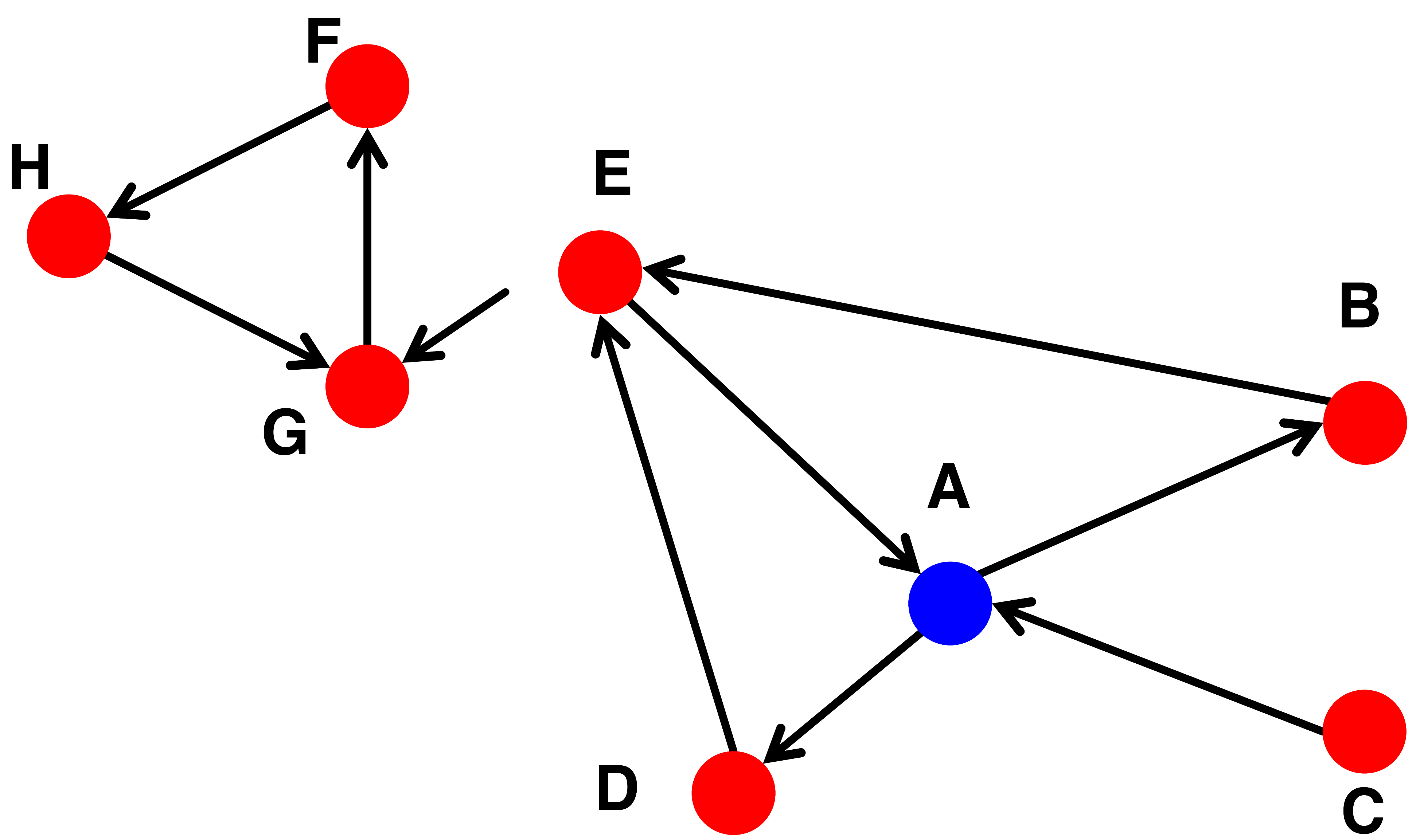


Out(v)



Out(A) ∩ In(A) = SCC

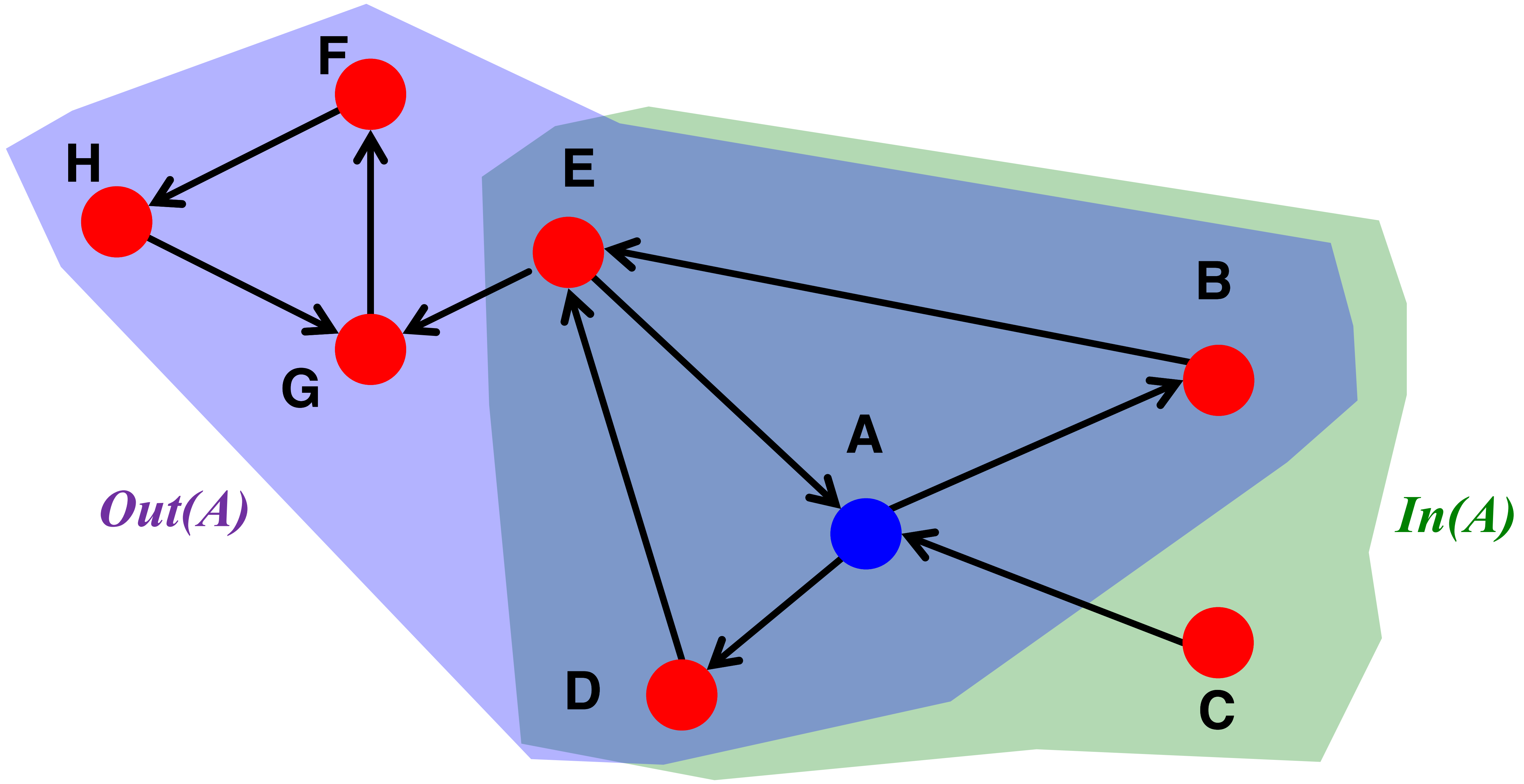
Example:



- Out(A) = {?}
- In(A) = {?}

Out(A) ∩ In(A) = SCC

Example:



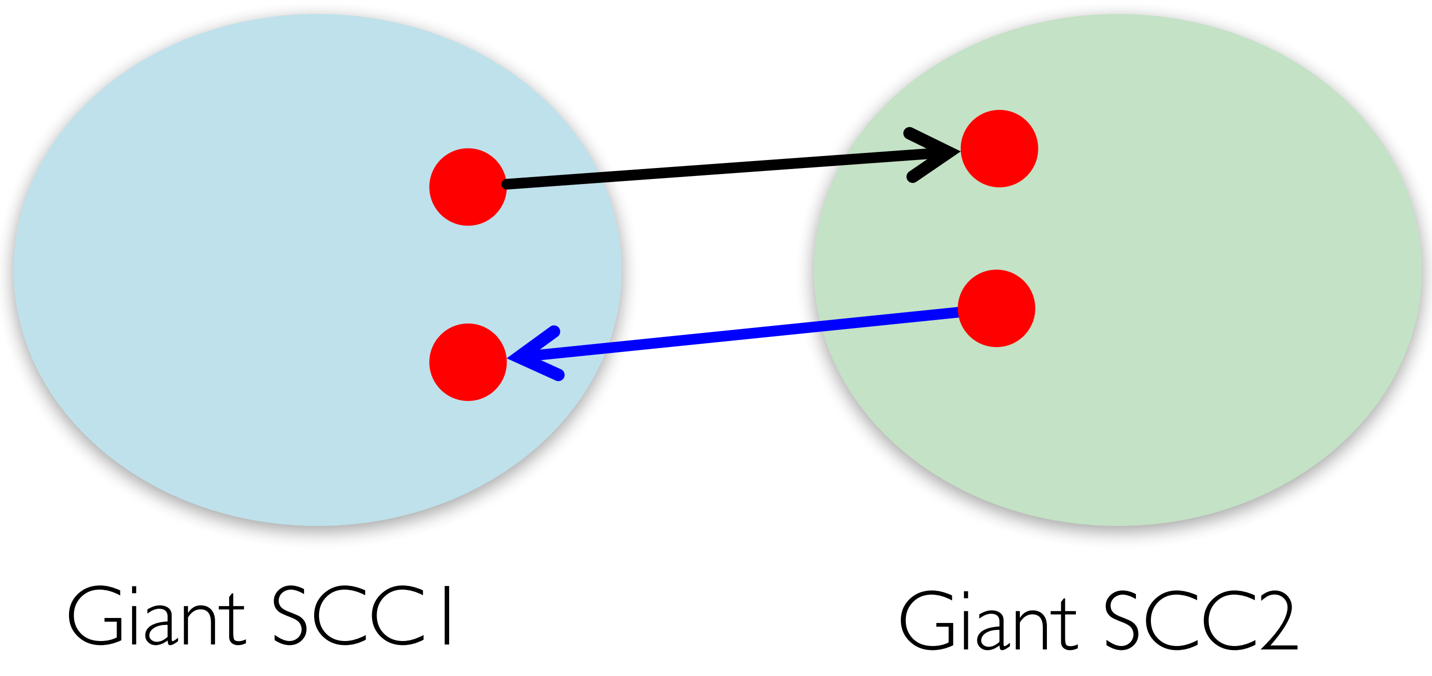
- Out(A) = {A,B,D,E,F,G,H}
- In(A) = {A,B,C,D,E}
- Therefore, SCC(A) = {A,B,D,E}

Graph Structure of the Web

- **How many “big” SCCs?**

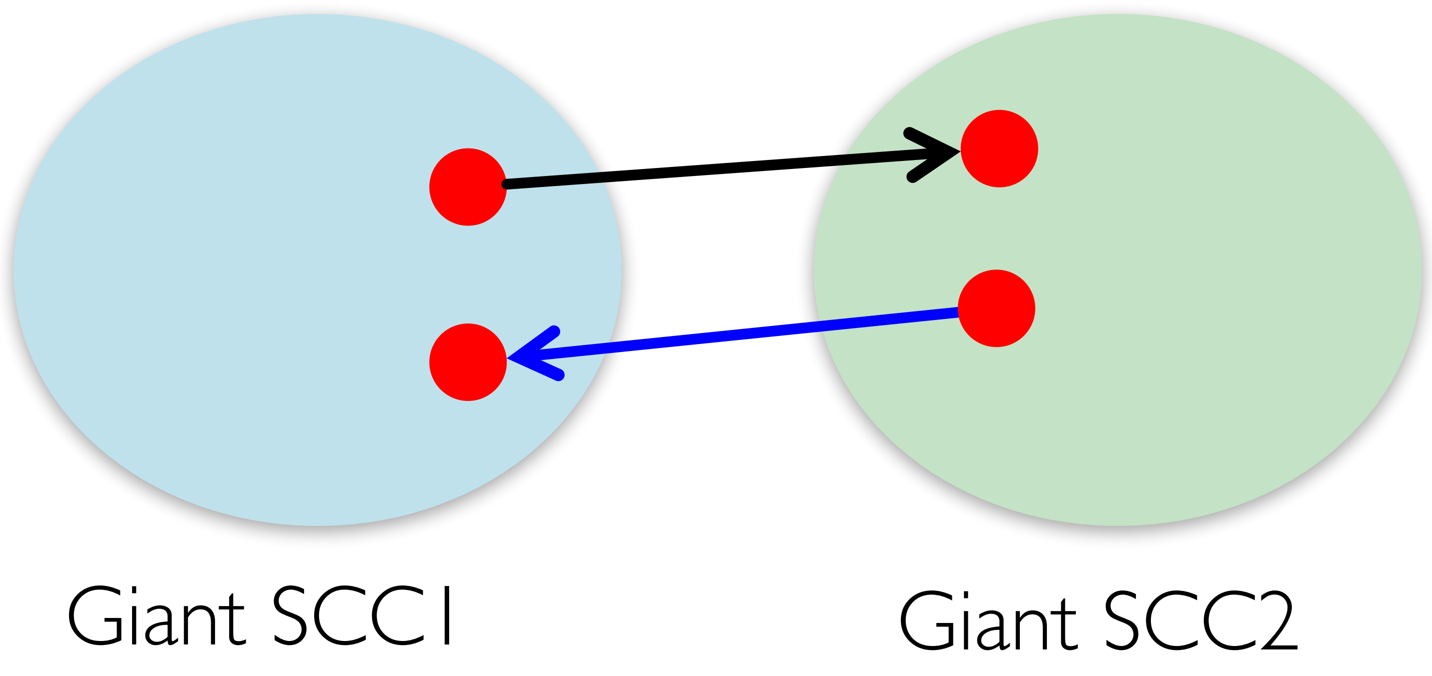
Graph Structure of the Web

- **How many "big" SCCs?**



Graph Structure of the Web

- **There is a single giant SCC**
  - That is, there won't be two SCCs
- **Heuristic argument:**
  - It just takes 1 page from one SCC to link to the other SCC
  - If the 2 SCCs have millions of pages the likelihood of this not happening is very very small



Structure of the Web

- **Broder et al., 2000:**
  - Altavista crawl from October 1999
    - 203 million URLS
    - 1.5 billion links
  - Computer: Server with 12GB of memory
- **Undirected version of the Web graph:**
  - 91% nodes in the largest weakly connected component
  - Are hubs making the web graph connected?
    - Even if they deleted links to pages with in-degree >10 WCC was still ≈50% of the graph

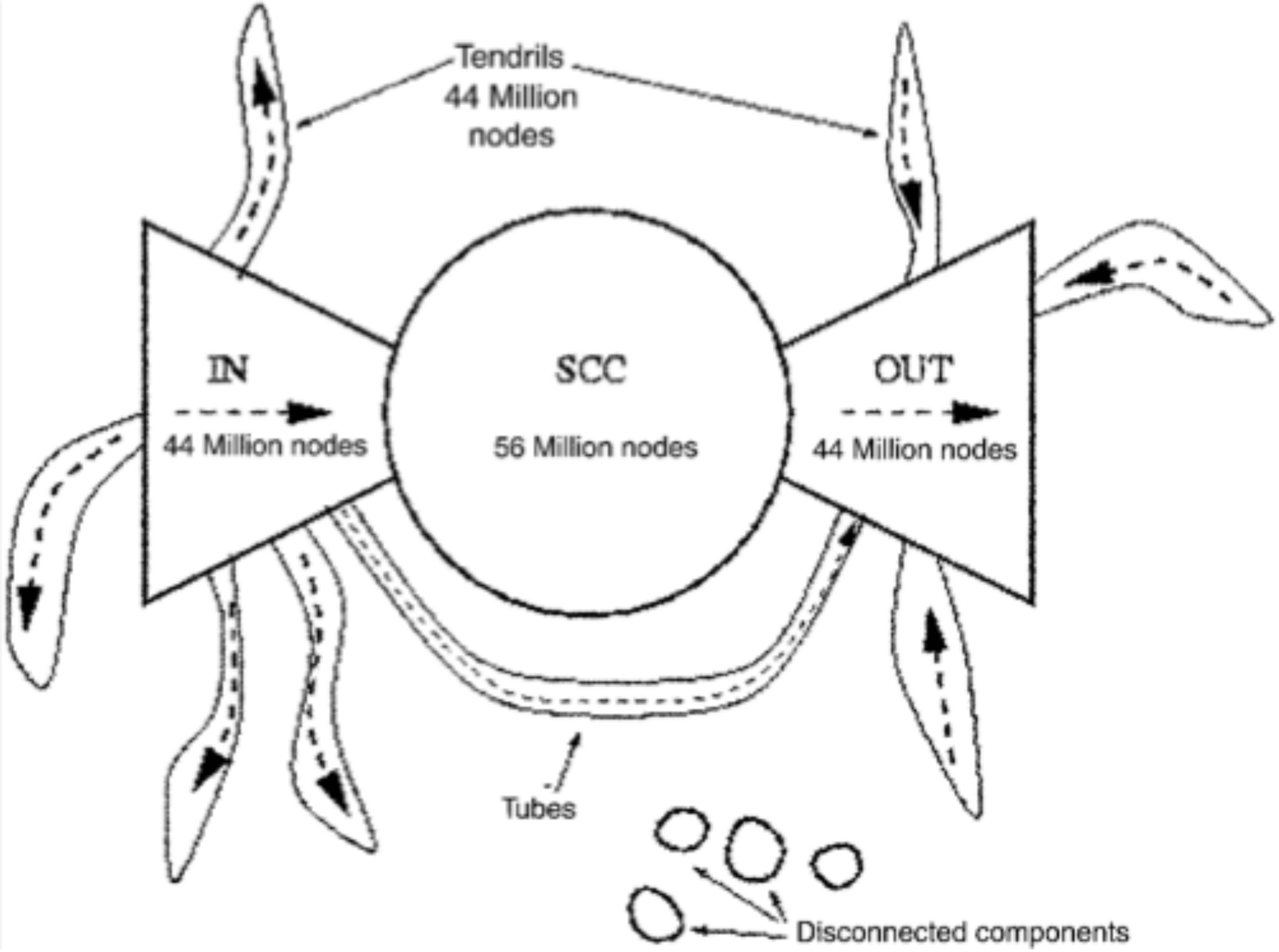
Structure of the Web

- Directed version of the Web graph:
  - **Largest SCC:** 28% of the nodes (56 million)
  - Taking a random node v
    - **Out(v) ≈ 50%** (100 million)
    - **In(v) ≈ 50%** (100 million)

- What does this tell us about the conceptual picture of the Web graph?

Bow-tie Structure of the Web

203 million pages, 1.5 billion links [Broder et al. 2000]



What did we do?

- **Here is what we've already done**
  - We took a real system (the Web)
  - We represented the Web as a graph
  - We used the language of graph theory to reason about the structure of the graph
  - We did a computational experiment on the Web graph
  - **Learned something about the structure of the Web!**

What did We Learn/Not Learn ?

- **What did we learn:**
  - Some conceptual organization of the Web (i.e., the bowtie)
- **What did we not learn:**
  - **Treats all pages as equal**
    - Google's homepage == my homepage
  - **What are the most important pages**
    - How many pages have *k* in-links as a function of *k*?

      The degree distribution: $\sim k^{-2}$
    - Link analysis ranking -- as done by search engines (PageRank)
  - **Internal structure inside giant SCC**
    - Clusters, implicit communities?
  - **How far apart are nodes in the giant SCC:**
    - Distance = # of edges in shortest path
    - Avg = 16 [Broder et al.]

Recap

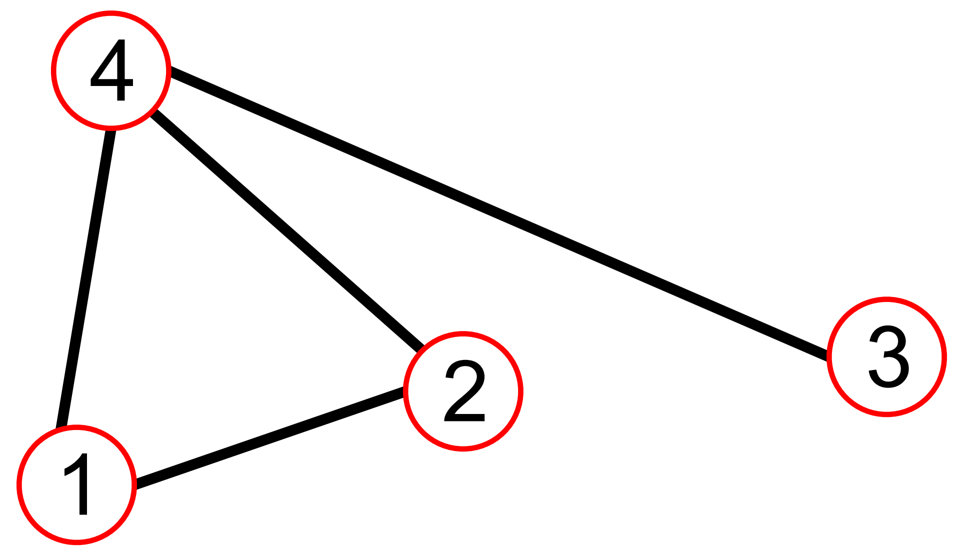
- **Network analysis is the language of connectedness**
  - Represent real-world networks from many different domains as graphs, use graph theory and algorithms to reason about them
  - Social networks, information networks, knowledge networks, biological networks, etc.
- **Network analysis fundamentals**
  - Nodes, edges, paths, cycles, un/directed, connected components (weak and strong)
  - Choices of representation
  - Every directed graph is a DAG on its SCCs
- **Structure of the Web**
  - Looks like a bow-tie: big giant component, IN & OUT components, tendrils, disconnected components

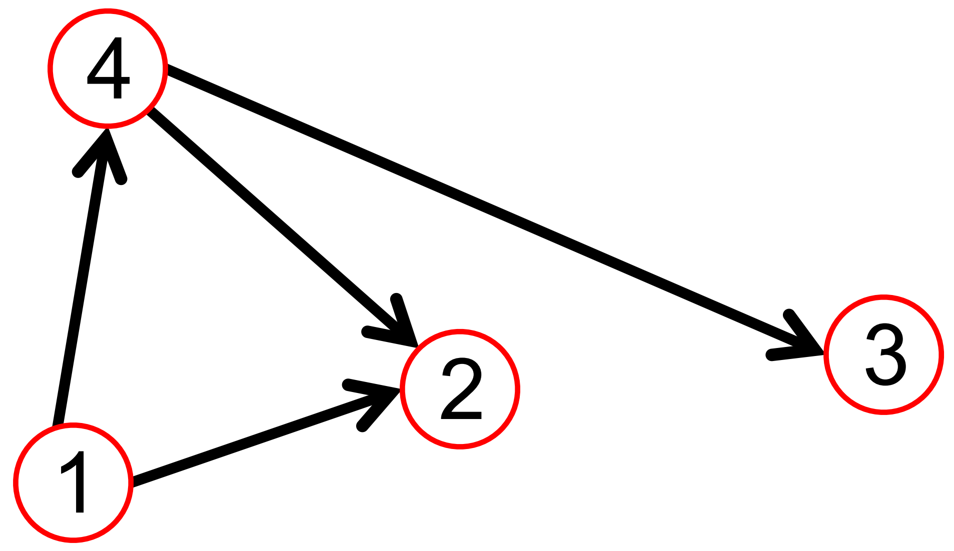
Network Representations

How do we represent graphs as mathematical objects?

What are our choices when we're translating real-world networks into a graph representation?

Edge List

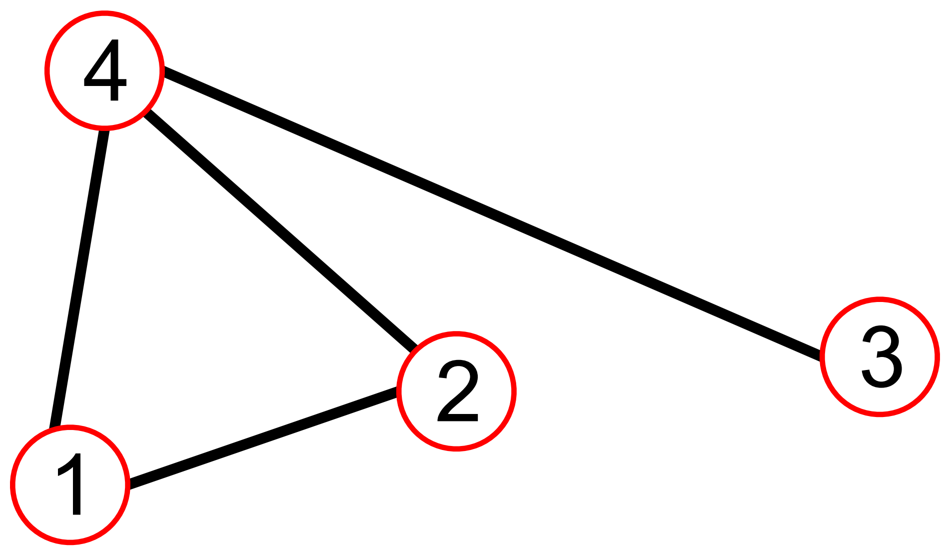


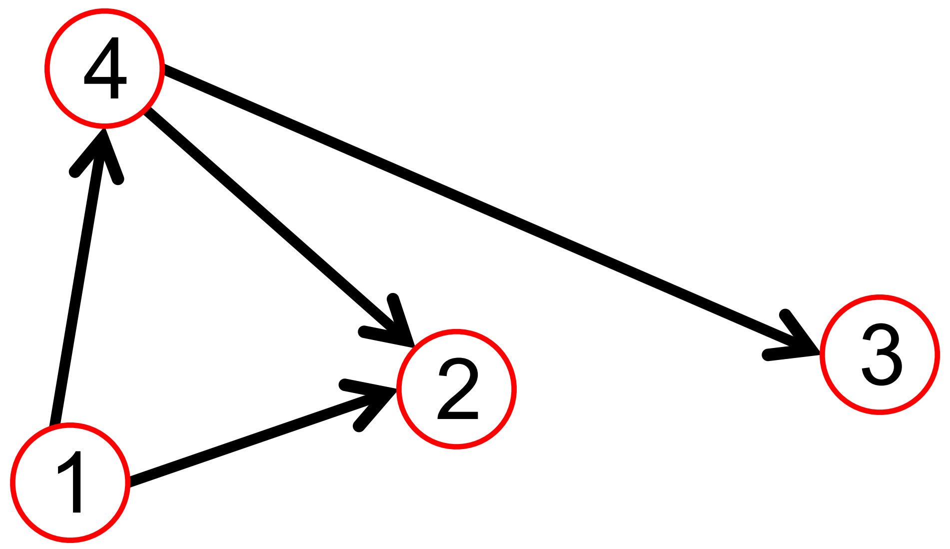


[(1,2),
(1,4),
(2,4),
(3,4)]

[(1,2),
(1,4),
(4,2),
(4,3)]

Adjacency List



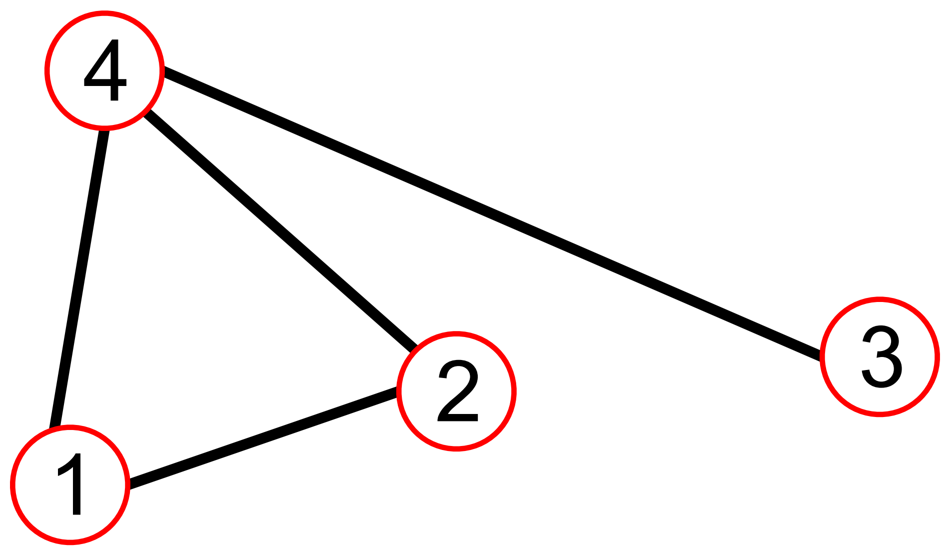


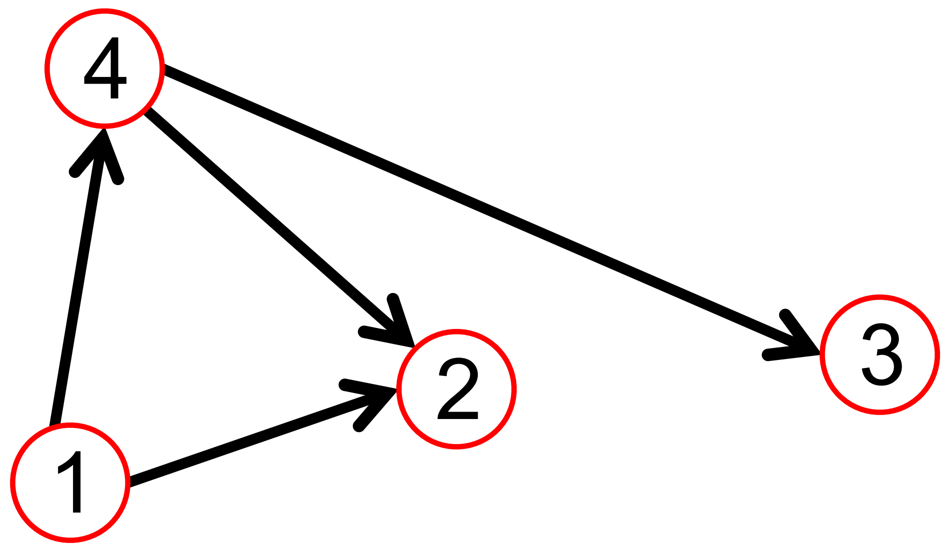
{1: [2,4],
 2: [1,4],
 3: [4],
 4: [1,2,3]}

{1: [2,4],
 4: [2,3]}

Total length of lists?

Adjacency Matrix





$A_{ij} = 1$ if there is a link from node *i* to node *j*

$A_{ij} = 0$ otherwise

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

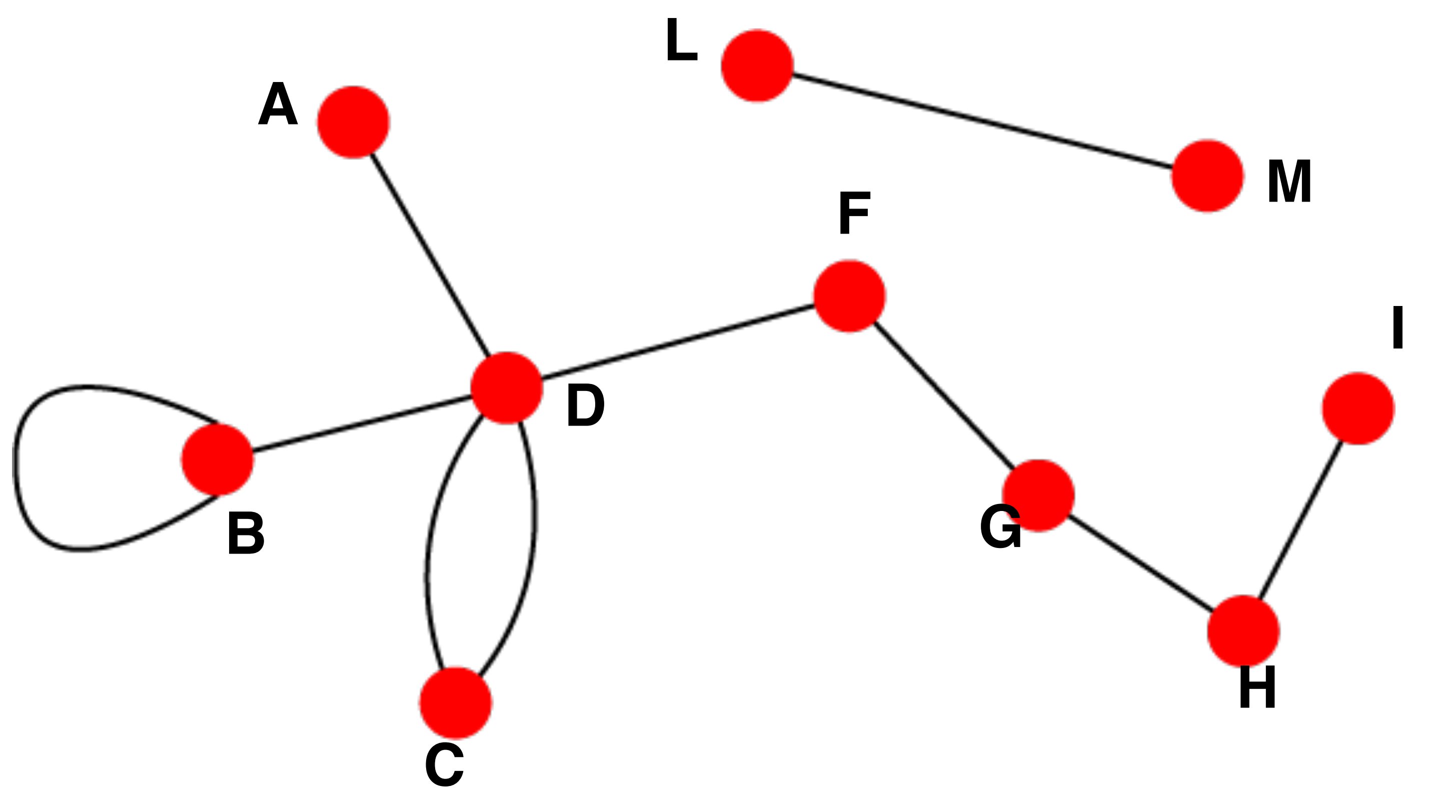
$$A = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{pmatrix}$$

Note that for a directed graph (right) the matrix is not symmetric.

Undirected vs. Directed Networks

Undirected graphs

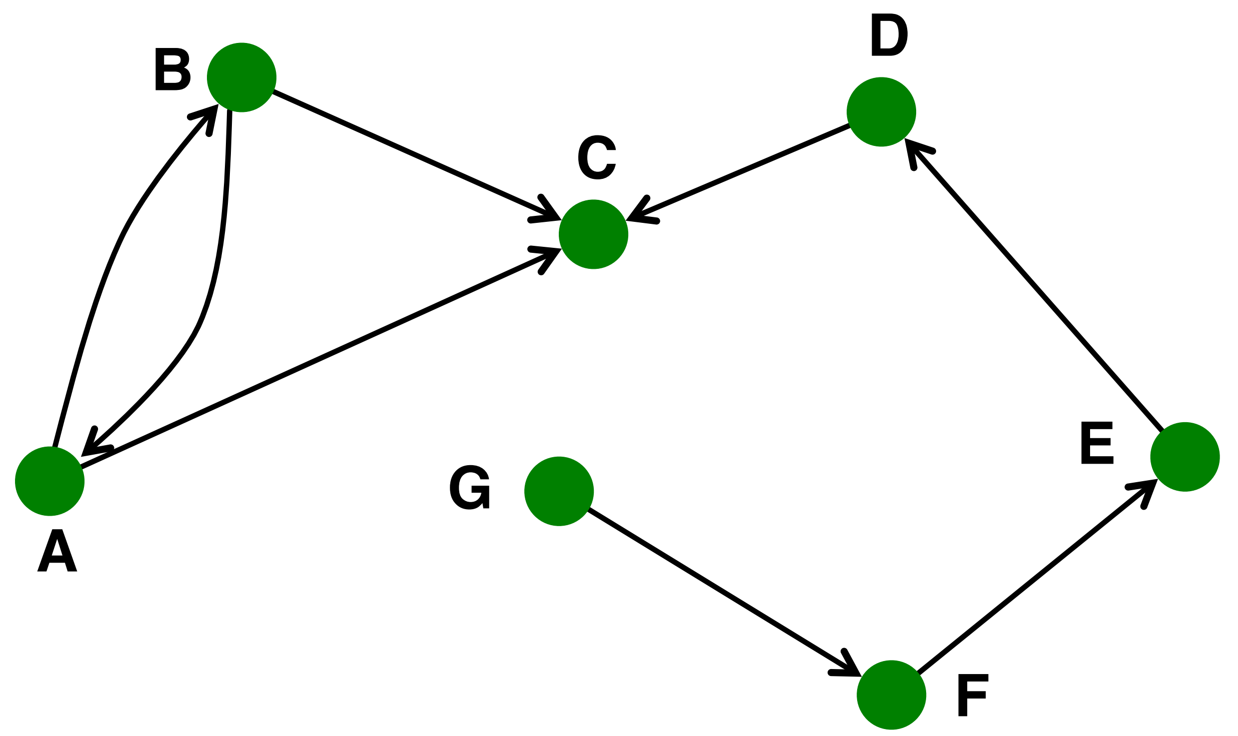
- **Links:** undirected (symmetrical, reciprocal relations)



- Undirected links:
  - Collaborations
  - Friendship on Facebook

Directed graphs

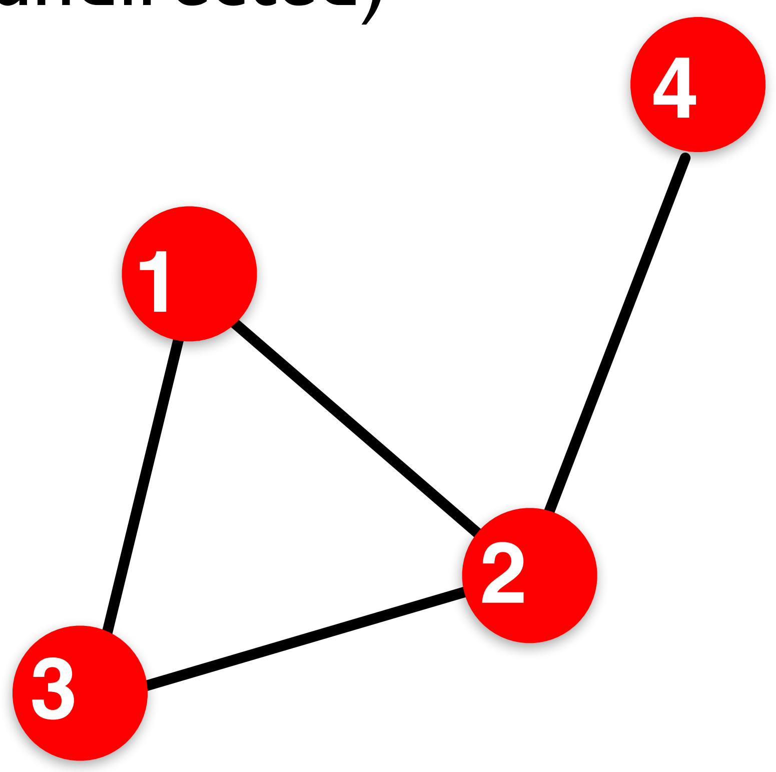
- **Links:** directed (asymmetrical relations)



- Directed links:
  - Phone calls
  - Following on Twitter

More Types of Graphs:

Unweighted
(undirected)



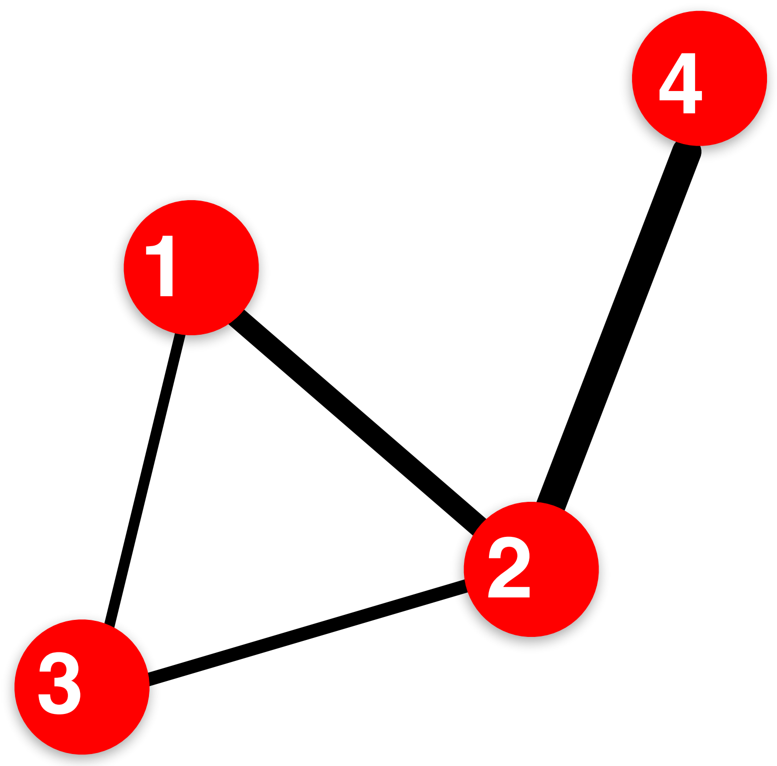
$$A_{ij} = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

$$A_{ii} = 0 \qquad A_{ij} = A_{ji}$$

$$E = \frac{1}{2}\sum_{i,j=1}^{N} A_{ij} \qquad \bar{k} = \frac{2E}{N}$$

Examples: Friendship, Hyperlink

Weighted
(undirected)



$$A_{ij} = \begin{pmatrix} 0 & 2 & 0.5 & 0 \\ 2 & 0 & 1 & 4 \\ 0.5 & 1 & 0 & 0 \\ 0 & 4 & 0 & 0 \end{pmatrix}$$

$$A_{ii} = 0 \qquad A_{ij} = A_{ji}$$

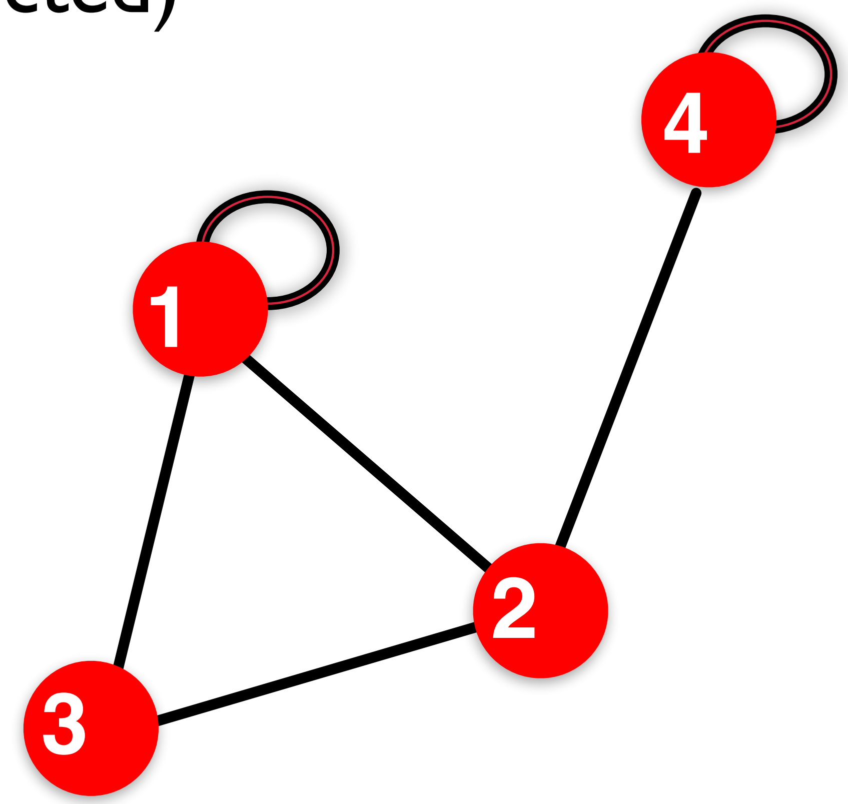
$$E = \frac{1}{2}\sum_{i,j=1}^{N} nonzero(A_{ij}) \qquad \bar{k} = \frac{2E}{N}$$

Examples: Collaboration, Internet, Roads

More Types of Graphs:

Graphs with self-edges

(undirected)



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

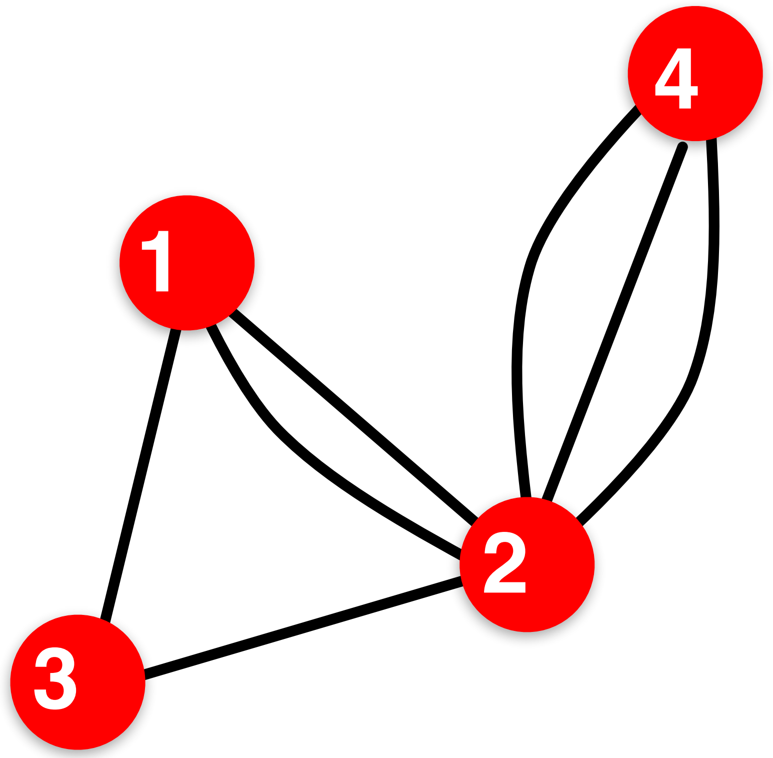
$A_{ii} \neq 0$ $\quad A_{ij} = A_{ji}$

$$E = \frac{1}{2} \sum_{i,j=1, i \neq j}^{N} A_{ij} + \sum_{i=1}^{N} A_{ii}$$

Examples: Proteins, Hyperlinks

Multigraph

(undirected)



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

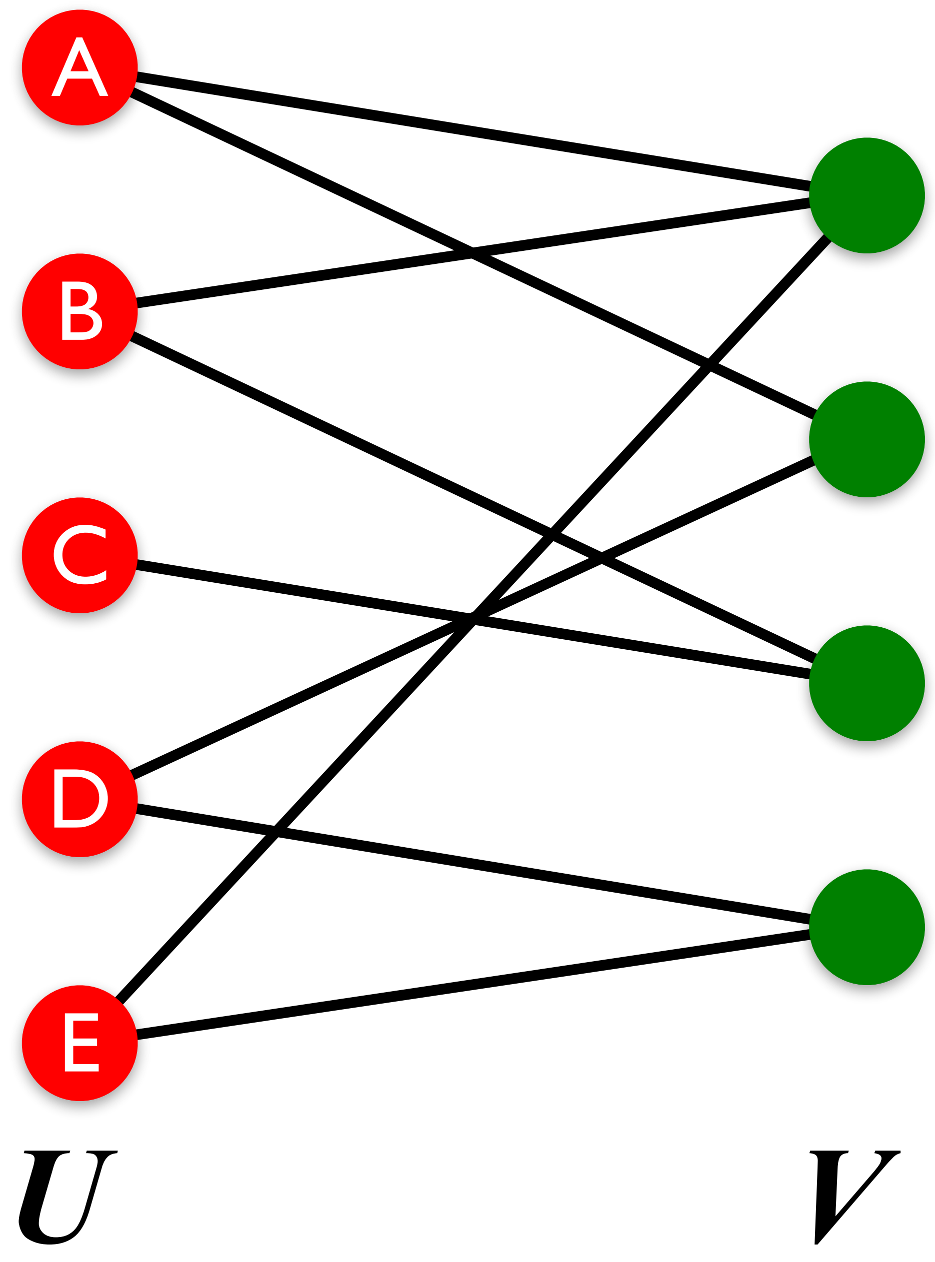
$A_{ii} = 0$ $\quad A_{ij} = A_{ji}$

$$E = \frac{1}{2} \sum_{i,j=1}^{N} nonzero(A_{ij}) \qquad \bar{k} = \frac{2E}{N}$$

Examples: Communication, Collaboration

Bipartite Graph

Bipartite graph is a graph whose nodes can be divided into two disjoint sets U and V such that every link connects a node in U to one in V; that is, U and V are **independent sets**

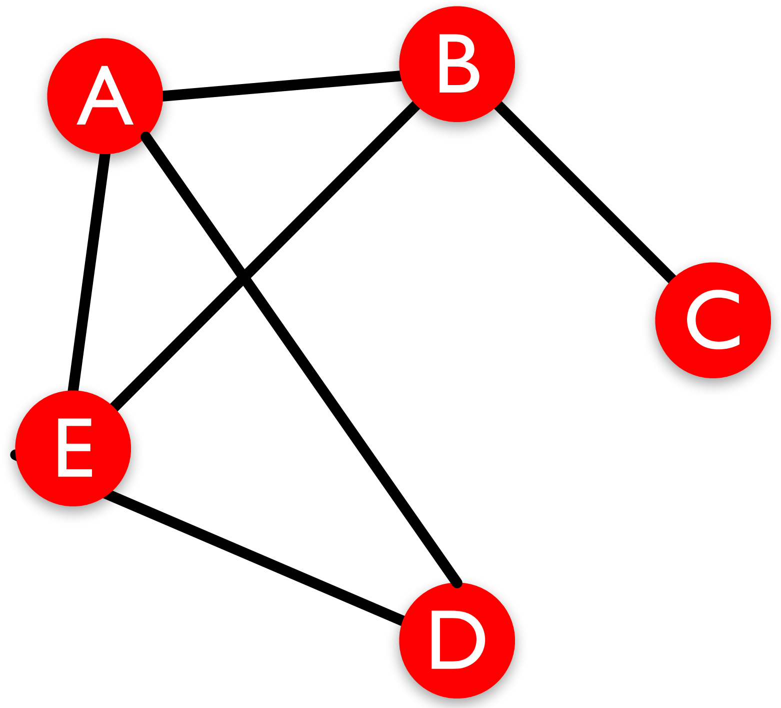


Examples:

- Authors-to-papers (they authored)
- Actors-to-Movies (they appeared in)
- Users-to-Movies (they rated)

"Folded" networks:

- Author collaboration networks
- Movie co-rating networks



Folded version of the graph above

Networks are Sparse Graphs

Most real-world networks are sparse

$E \ll E_{max}$ (or $\bar{k} \ll N-1$)

WWW (Stanford-Berkeley):	N=319,717	$\langle k \rangle$ =9.65
Social networks (LinkedIn):	N=6,946,668	$\langle k \rangle$ =8.87
Communication (MSN IM):	N=242,720,596	$\langle k \rangle$ =11.1
Coauthorships (DBLP):	N=317,080	$\langle k \rangle$ =6.62
Internet (AS-Skitter):	N=1,719,037	$\langle k \rangle$ =14.91
Roads (California):	N=1,957,027	$\langle k \rangle$ =2.82
Proteins (S. Cerevisiae):	N=1,870	$\langle k \rangle$ =2.39

(Source: *Leskovec et al., Internet Mathematics, 2009*)

Consequence: Adjacency matrix is filled with zeros!

(Density of the matrix (E/N^2): WWW=1.51×10^{-5}, MSN IM = 2.27×10^{-8})

Network Representations

WWW ➢

Facebook friendships ➢

Citation networks ➢

Collaboration networks ➢

Mobile phone calls ➢

Protein Interactions ➢

Network Representations

WWW ➢ 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

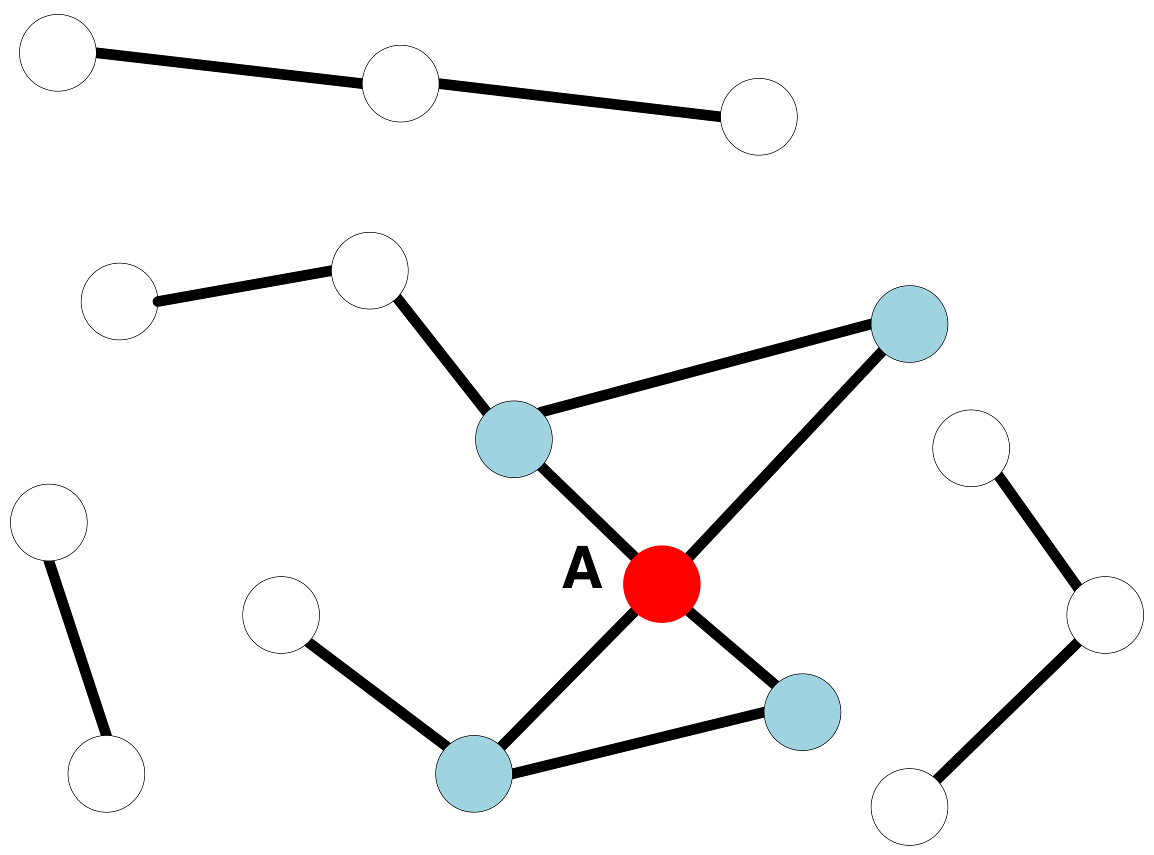
Network Properties:
How to Characterize/Measure a Network?

How do we measure properties in the graph representation of a network?

Focus on connectivity and distance

Connectivity: Node Degrees

Undirected

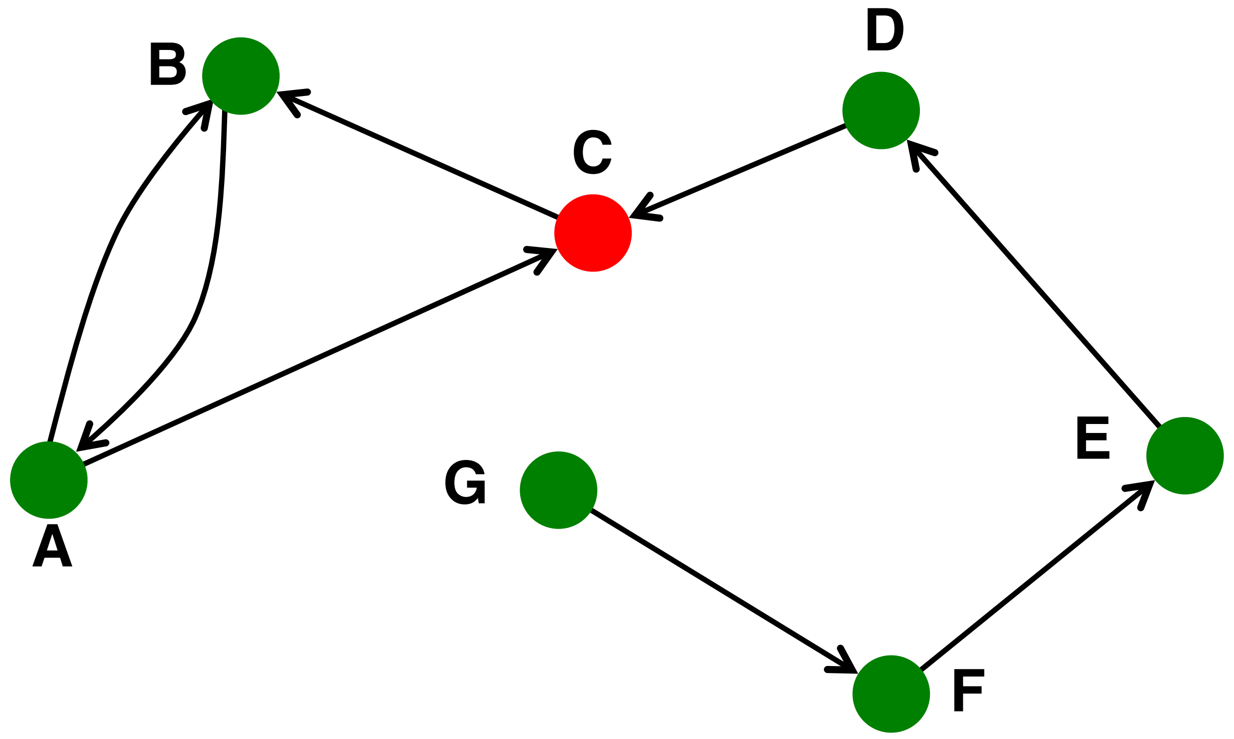


Node degree, k_i: the number of edges adjacent to node *i*

e.g. $k_A = 4$

Avg. degree: $\bar{k} = \langle k \rangle = \frac{1}{N}\sum_{i=1}^{N} k_i = \frac{2E}{N}$

Directed



Source: Node with $k^{in} = 0$
Sink: Node with $k^{out} = 0$

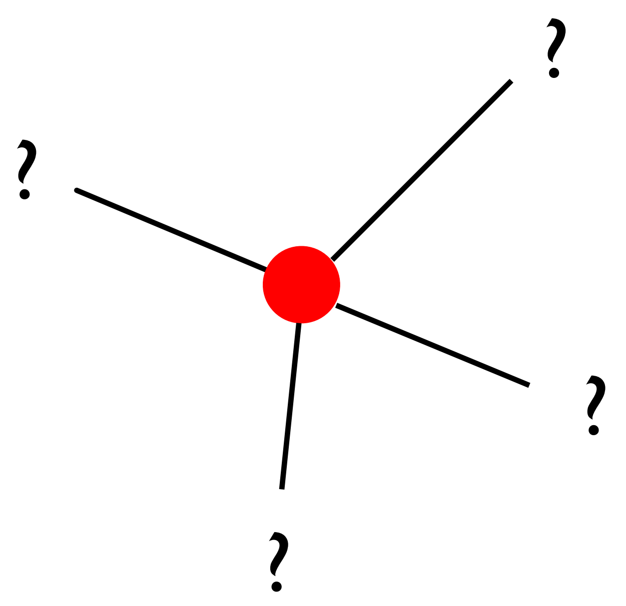
In directed networks we define an in-degree and out-degree.
The (total) degree of a node is the sum of in- and out-degrees.

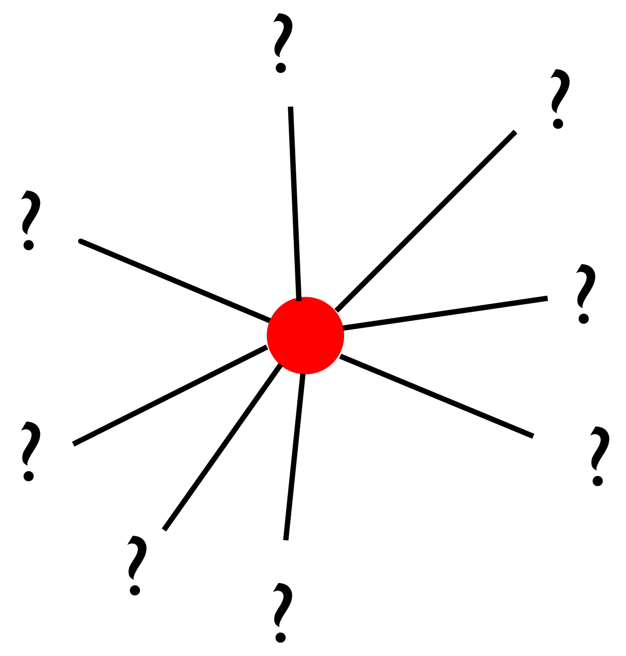
$k_C^{in} = 2 \quad k_C^{out} = 1 \quad k_C = 3$

$$\overline{k^{in}} = \overline{k^{out}}$$

Connectivity: How Connected Are Nodes?

How many neighbours do nodes tend to have in your graph?





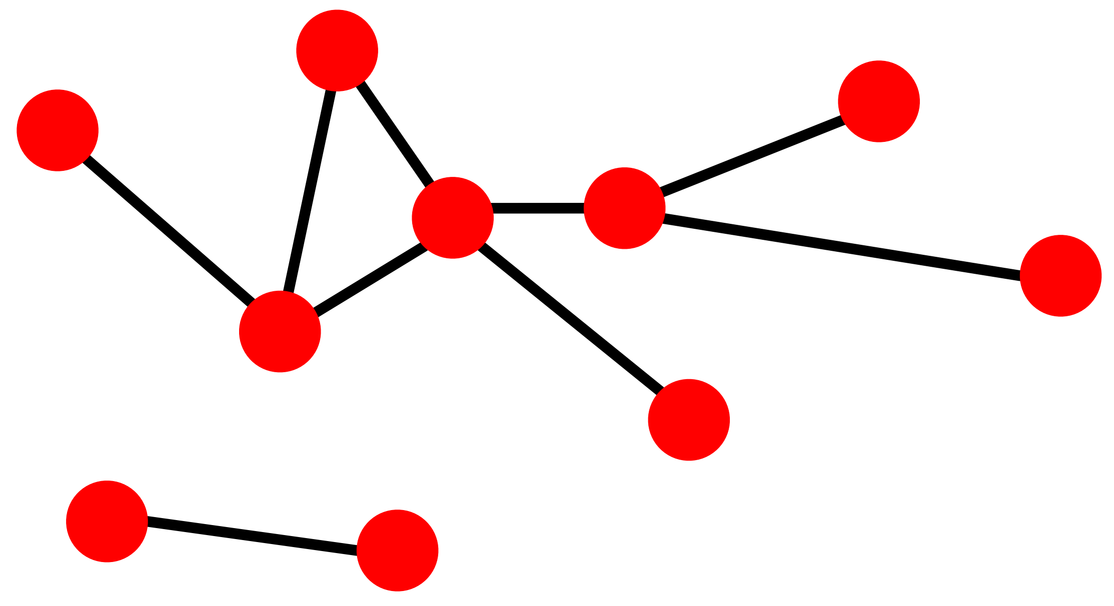
Connectivity: Degree Distribution

Degree distribution $P(k)$: Probability that a randomly chosen node has degree k

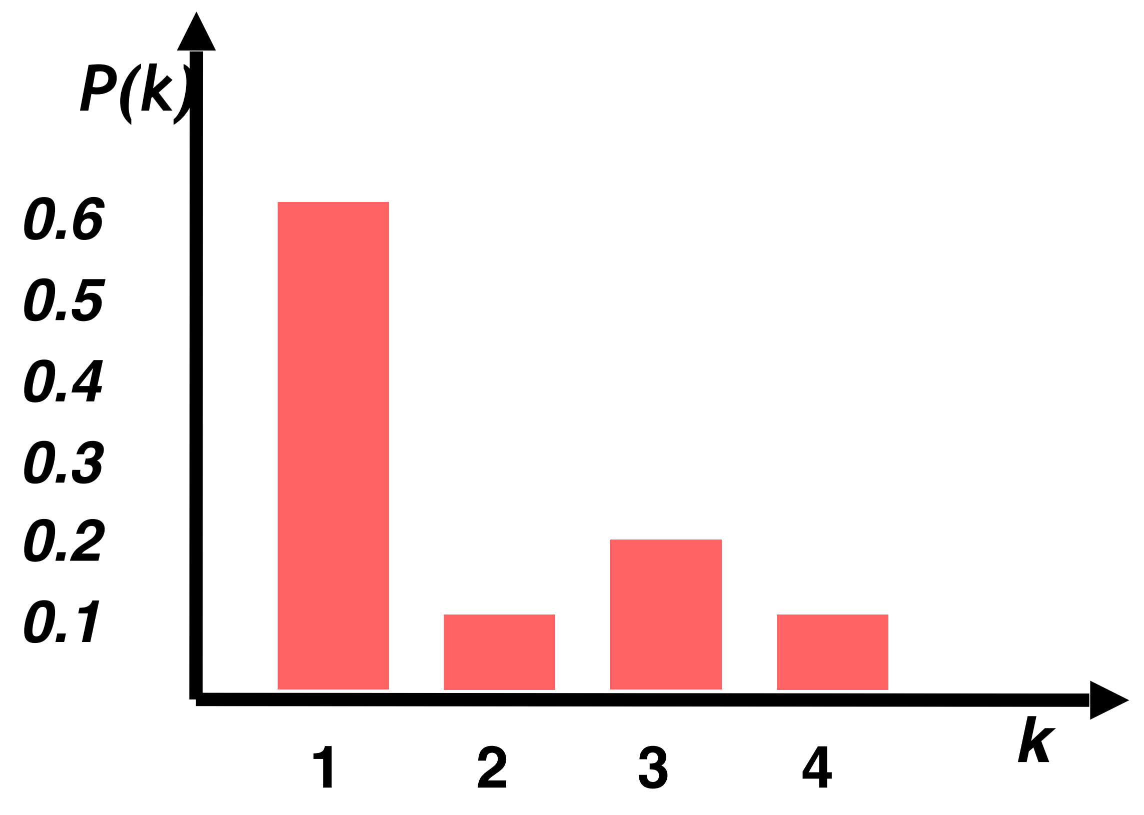
N_k = # nodes with degree k

Normalized histogram:

$P(k) = N_k / N$ → plot



→



Connectivity: Local Clustering

Are the nodes "clustered" in the graph? Do nodes with common neighbours tend to know each other?

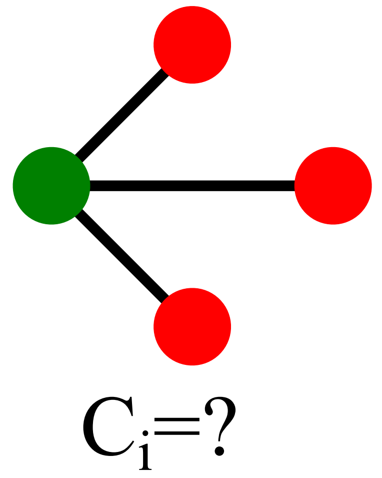
Connectivity: Clustering Coefficient

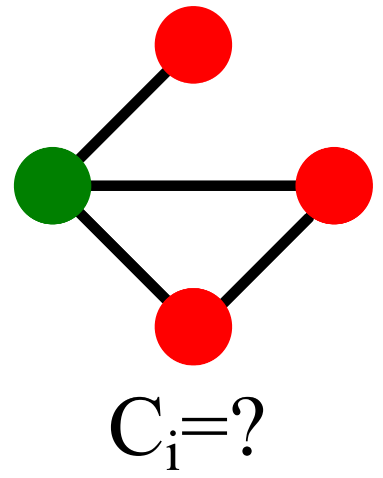
What's the probability that a random pair of your friends are connected?

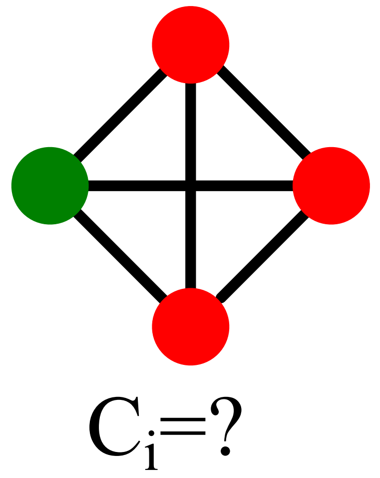
$$C_i \in [0, 1]$$

$$C_i = \frac{e_i}{\binom{k_i}{2}} = \frac{e_i}{k_i(k_i - 1)/2} = \frac{2e_i}{k_i(k_i - 1)}$$

where e_i is the number of edges between the neighbours of node i and k_i is the degree of node i







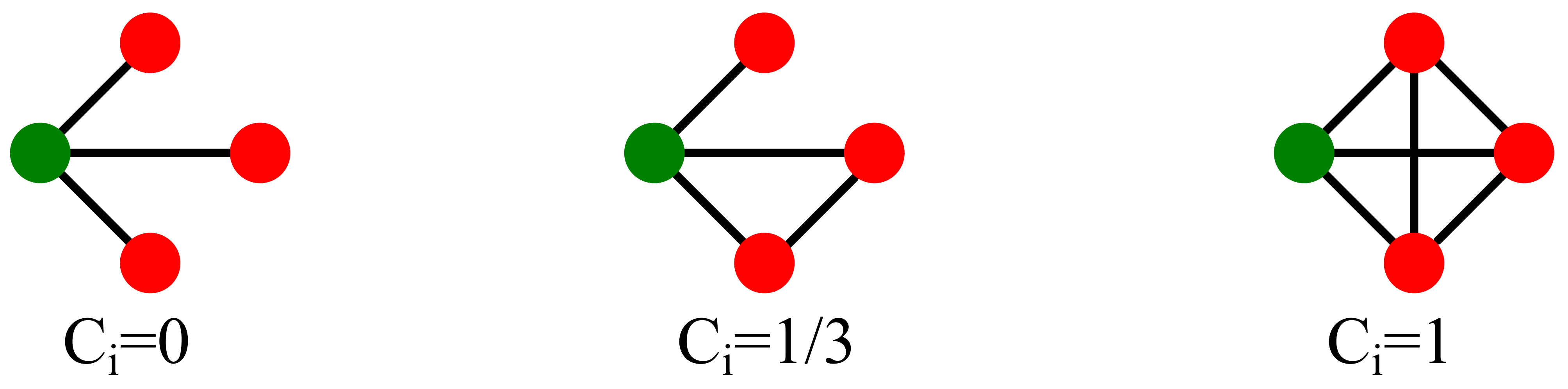
Connectivity: Clustering Coefficient

What's the probability that a random pair of your friends are connected?

$$C_i \in [0, 1]$$

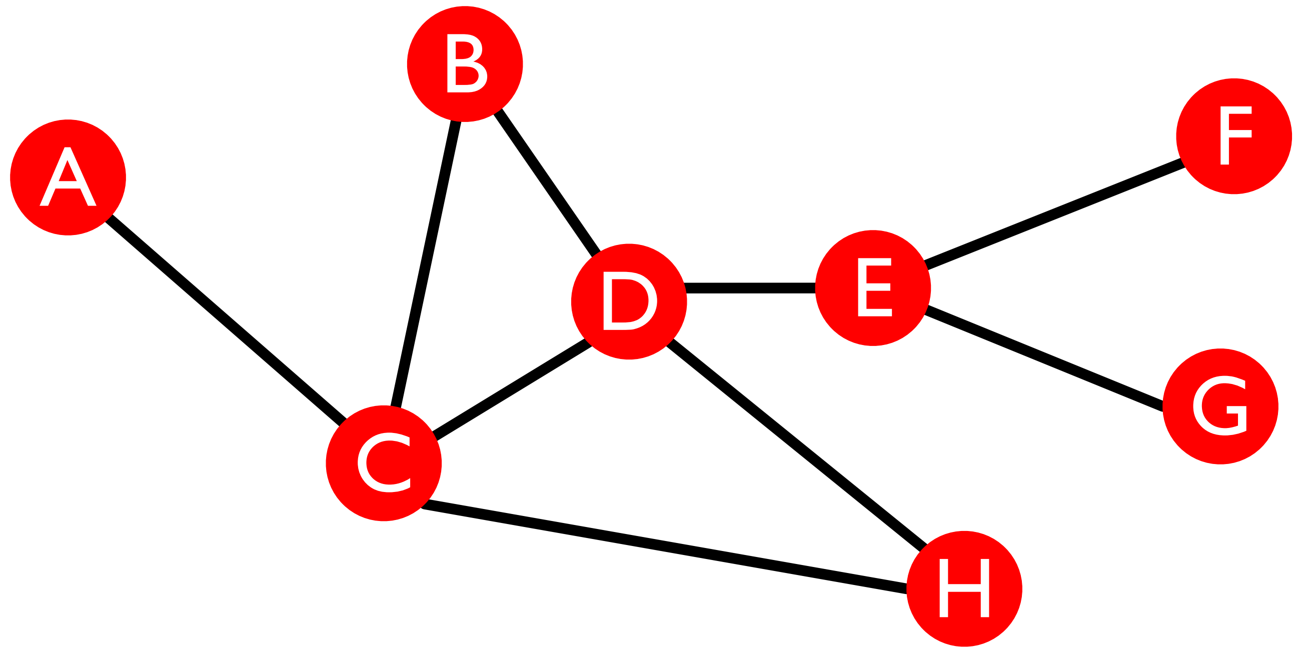
$$C_i = \frac{e_i}{\binom{k_i}{2}} = \frac{e_i}{k_i(k_i-1)/2} = \frac{2e_i}{k_i(k_i-1)}$$

where e_i is the number of edges between the neighbors of node I and k_i is the degree of node I



Average clustering coefficient: $$C = \frac{1}{N}\sum_{i}^{N} C_i$$

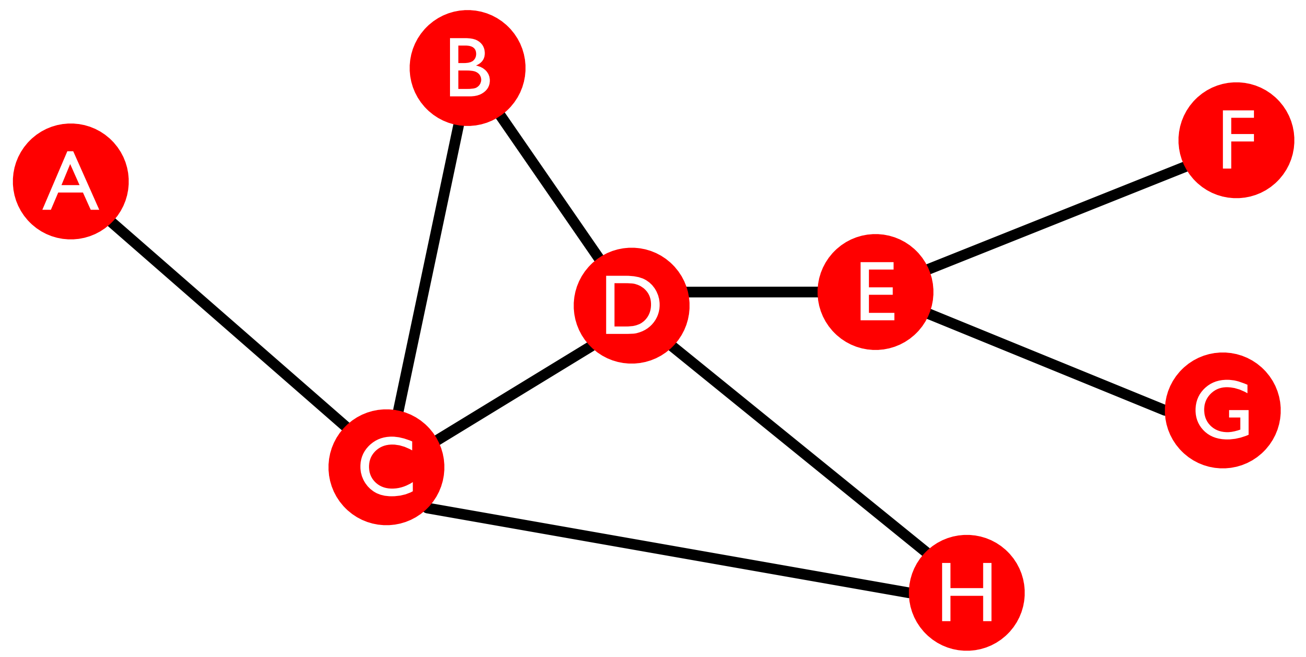
Connectivity: Clustering Coefficient



k_B=?, e_B=?, C_B=? = ?

k_D=?, e_D=?, C_D=? = ?

Connectivity: Clustering Coefficient



$k_B=2, \; e_B=1, \; C_B=2/2 = 1$

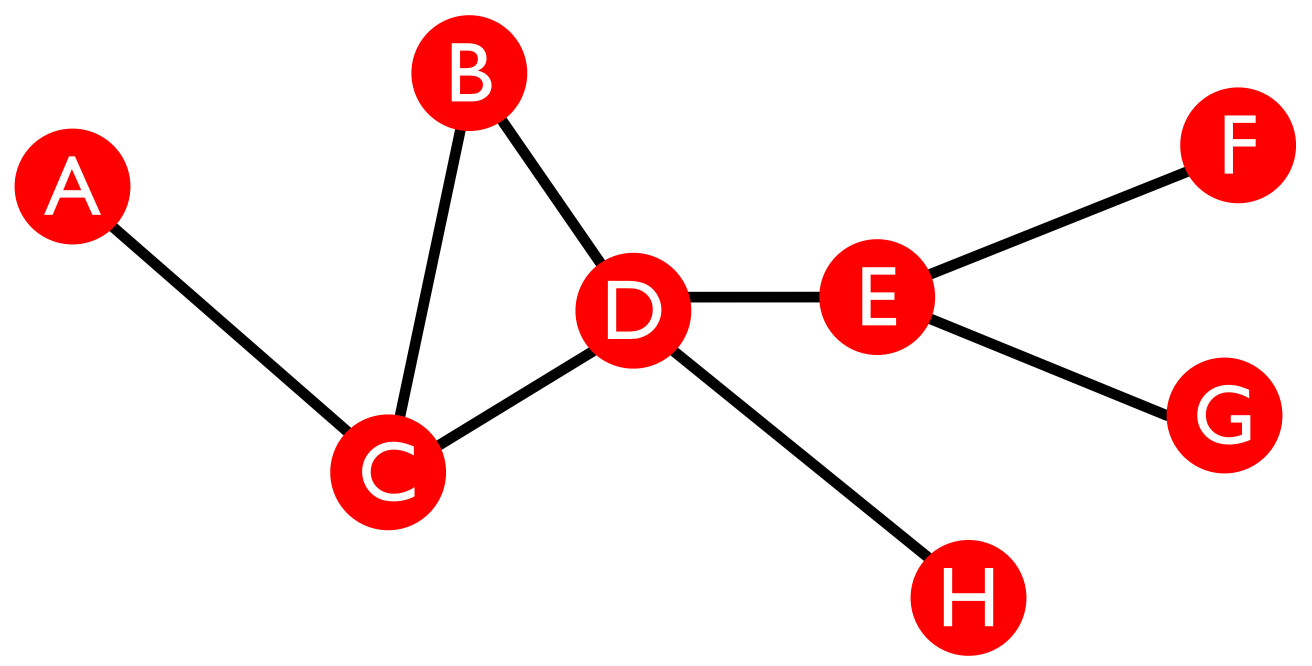
$k_D=4, \; e_D=2, \; C_D=(2*2)/(4*3) = 4/12 = 1/3$

Distance: Paths in a Graph

- A ***path*** is a sequence of nodes in which each node is linked to the next one

$$P_n = \{i_0, i_1, i_2, \ldots, i_n\} \qquad P_n = \{(i_0, i_1), (i_1, i_2), (i_2, i_3), \ldots, (i_{n-1}, i_n)\}$$

- Path can intersect itself and pass through the same edge multiple times
  - E.g.: ACBDCDEG
  - In a directed graph a path can only follow the direction of the “arrow”



Distance: Number of Paths

Number of paths between nodes *u* and *v*:

Length $h=1$: If there is a link between u and v, $A_{uv}=1$ else $A_{uv}=0$

Length $h=2$: If there is a path of length two between *u* and *v* then $A_{uk}A_{kv}=1$ else $A_{uk}A_{kv}=0$

$$H_{uv}^{(2)} = \sum_{k=1}^{N} A_{uk}A_{kv} = [A^2]_{uv}$$

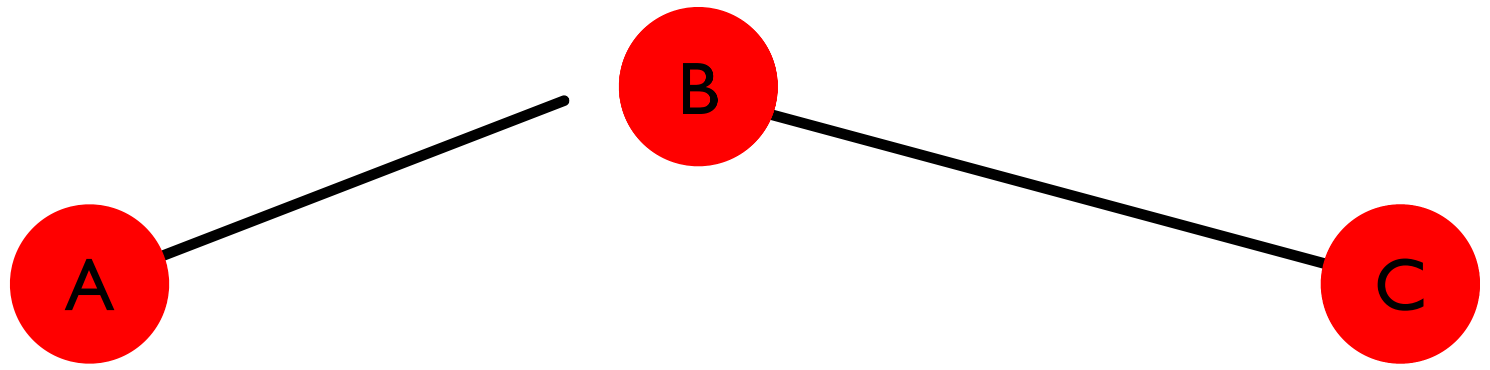
Length h: If there is a path of length *h* between *u* and *v* then $A_{uk} \ldots A_{kv}=1$ else $A_{uk} \ldots A_{kv}=0$

So, the no. of paths of length *h* between *u* and *v* is

$$H_{uv}^{(h)} = [A^h]_{uv}$$

(holds for both directed and undirected graphs)

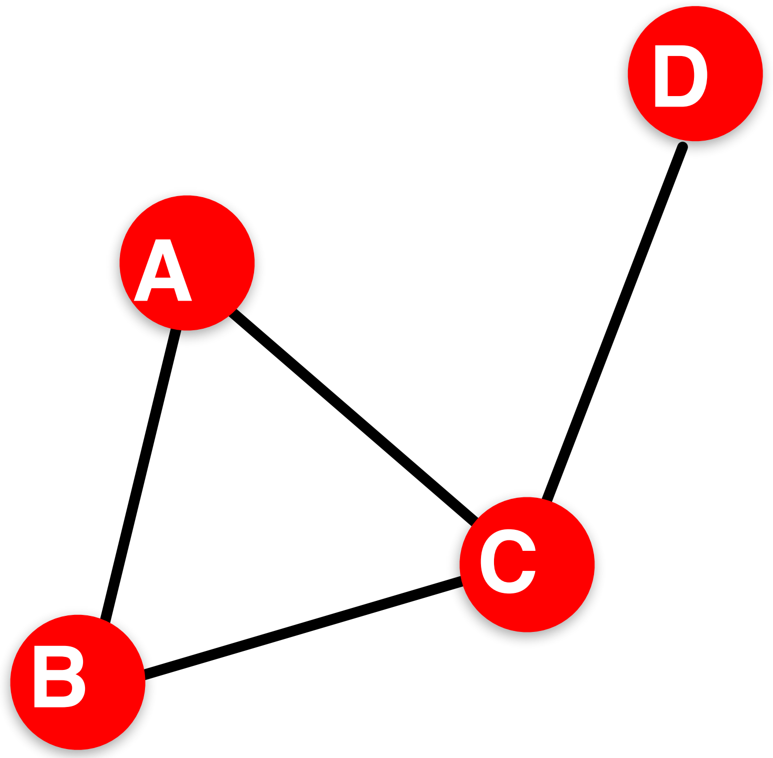
Distance: Number of Paths



$$H^{(1)} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

$$H^{(2)} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 2 & 0 \\ 1 & 0 & 1 \end{pmatrix}$$

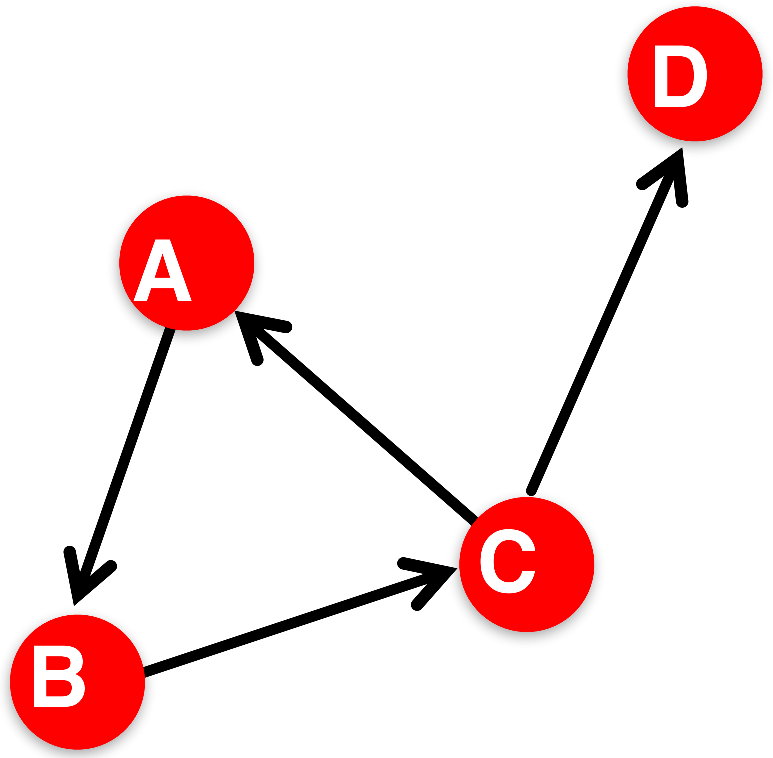
Distance: definition



$h_{B,D} = 2$

Distance (shortest path, geodesic) between a pair of nodes is defined as the number of edges along the shortest path connecting the nodes

*If the two nodes are disconnected, the distance is usually defined as infinite



$h_{B,C} = 1$, $h_{C,B} = 2$

In **directed graphs** paths need to follow the direction of the arrows

Consequence: Distance is **not symmetric**: $h_{A,C} \neq h_{C,A}$

Distance: Graph-level measures

- **Diameter:** the maximum (shortest path) distance between any pair of nodes in a graph

- **Average path length** for a connected graph (component) or a strongly connected (component of a) directed graph

$$\bar{h} = \frac{1}{2E_{max}} \sum_{i,j \neq i} h_{ij}$$

where $\boldsymbol{h_{ij}}$ is the distance from node $\boldsymbol{i}$ to node $\boldsymbol{j}$,
And Emax is the maximum number of edges (=n*(n-1)/2)

  - Many times we compute the average only over the connected pairs of nodes (that is, we ignore “infinite” length paths)

Key Network Properties

Degree distribution: $P(k)$

Clustering coefficients: C

Path lengths: L

Diameter: D