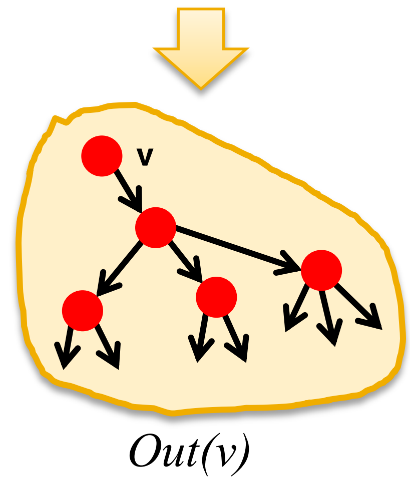
Link Analysis: PageRank

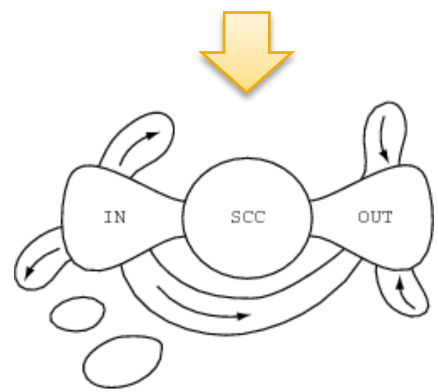
CS224W: Analysis of Networks
Jure Leskovec, Stanford University
http://cs224w.stanford.edu

Web as a Graph

Structure of the Web

- **Today we will talk about how does the Web graph look like:**
  - **1)** We will take a real system: **the Web**
  - **2)** We will represent it as a **directed graph**
  - **3)** We will use the language of graph theory
    - **Strongly Connected Components**
  - **4)** We will design a **computational experiment:**
    - Find In- and Out-components of a given node v
  - **5) We will learn something about the structure of the Web: BOWTIE!**

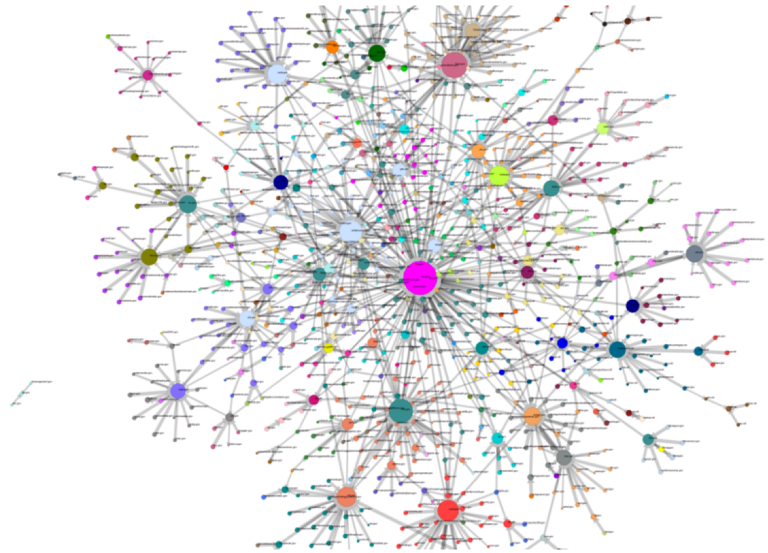




The Web as a Graph

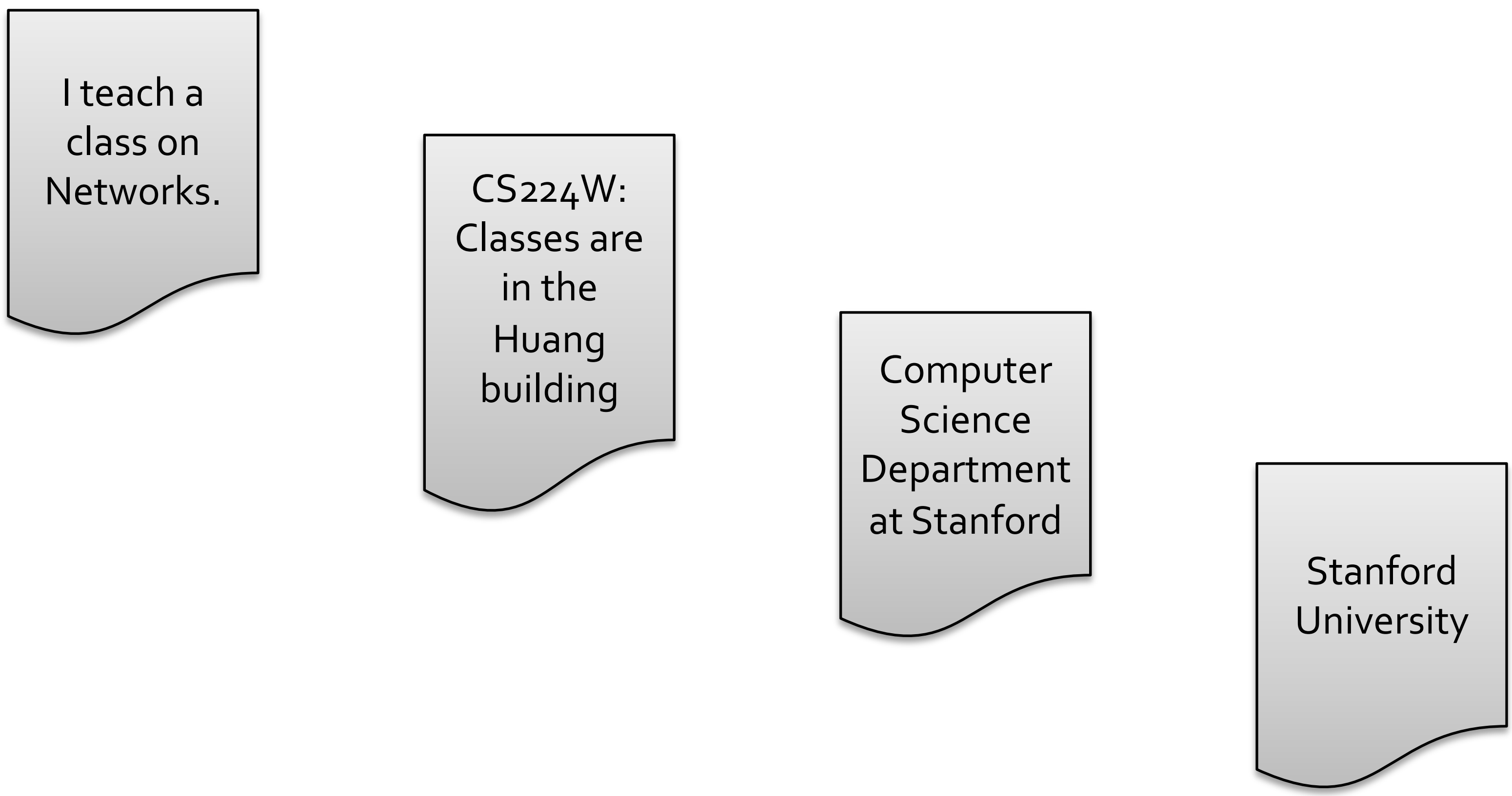
Q: What does the Web "look like" at a global level?

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

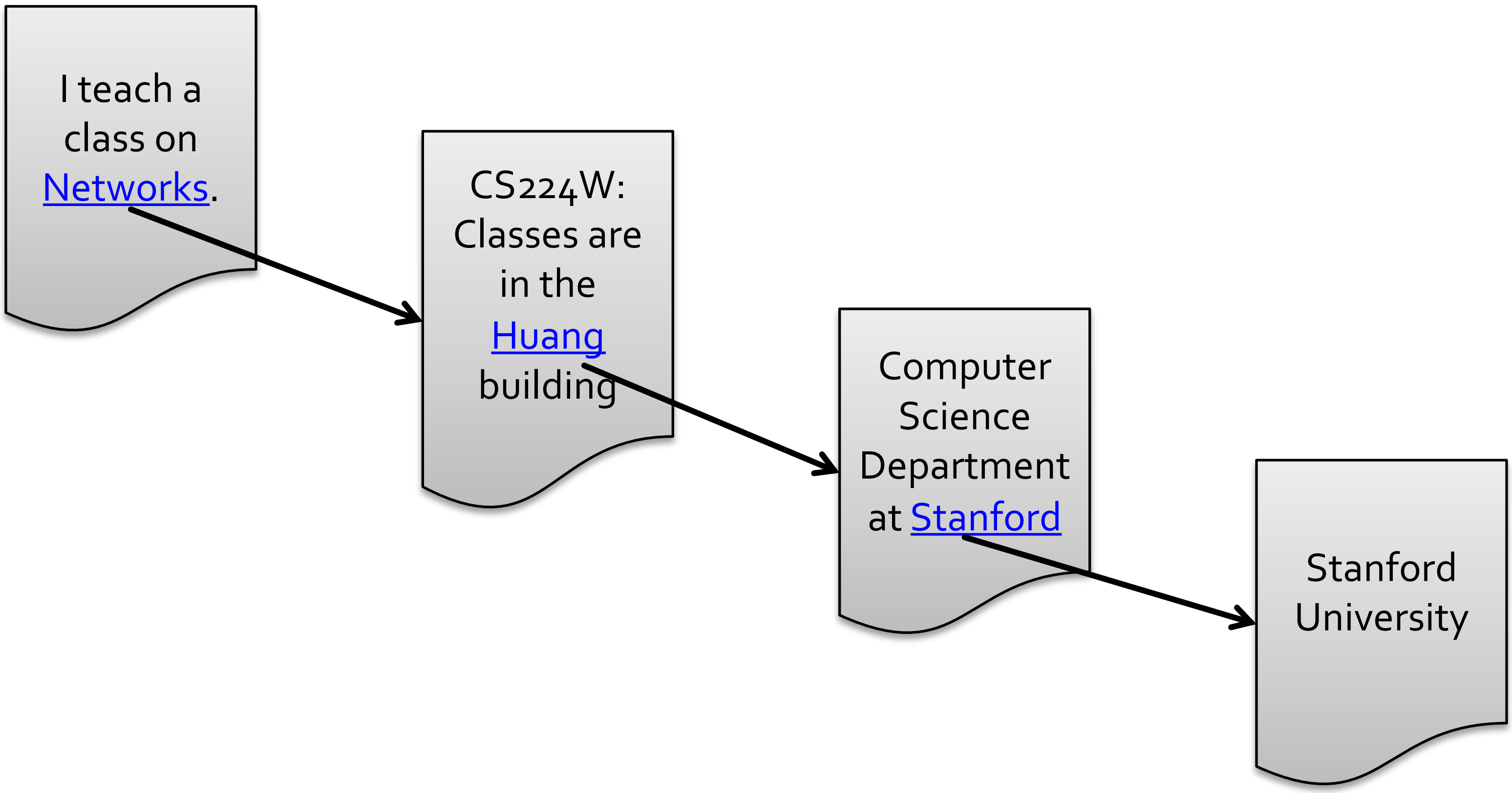


  - **Side issue:** What is a node?
    - Dynamic pages created on the fly
    - "dark matter" – inaccessible database generated pages

The Web as a Graph

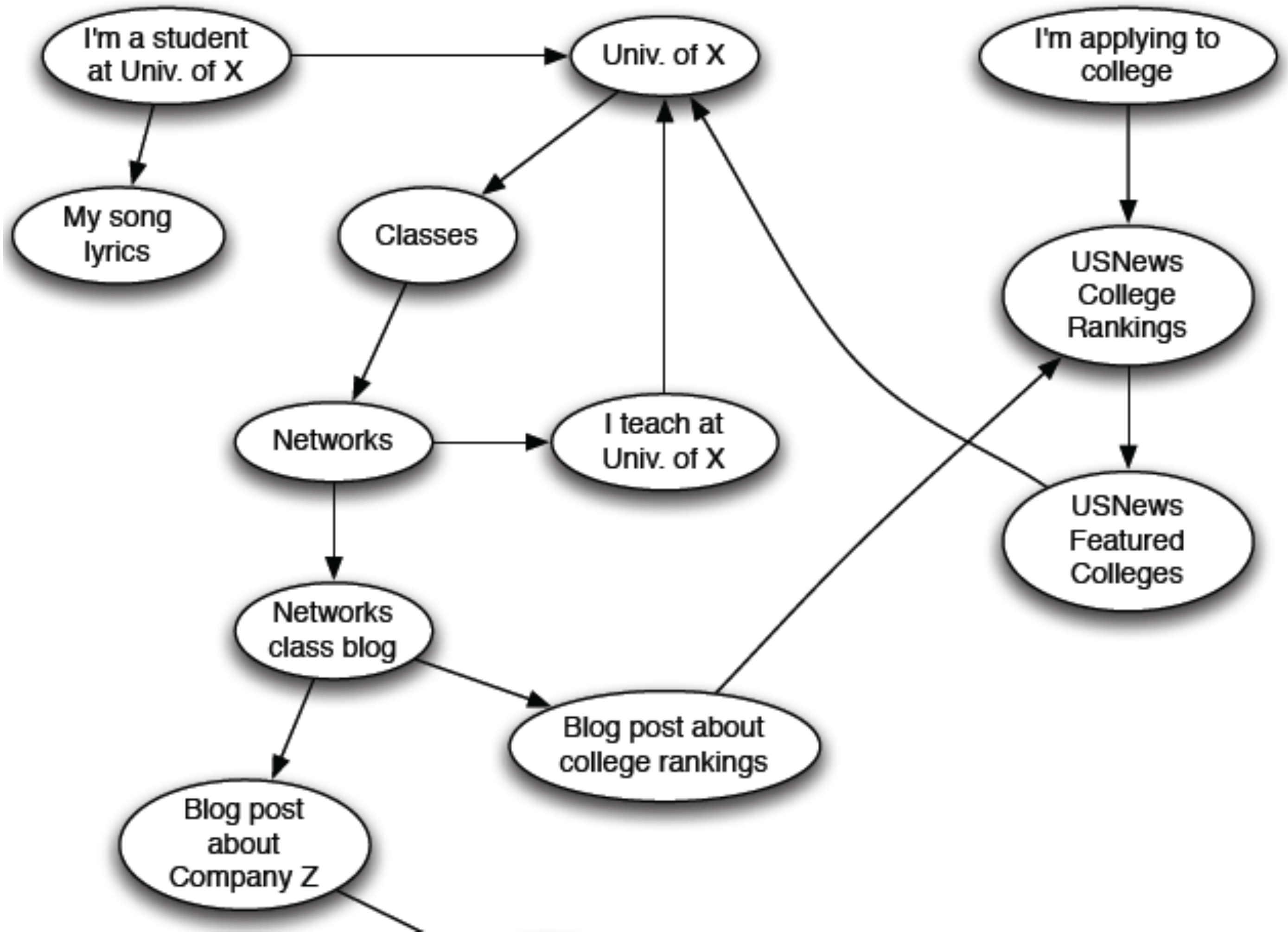


The Web as a Graph

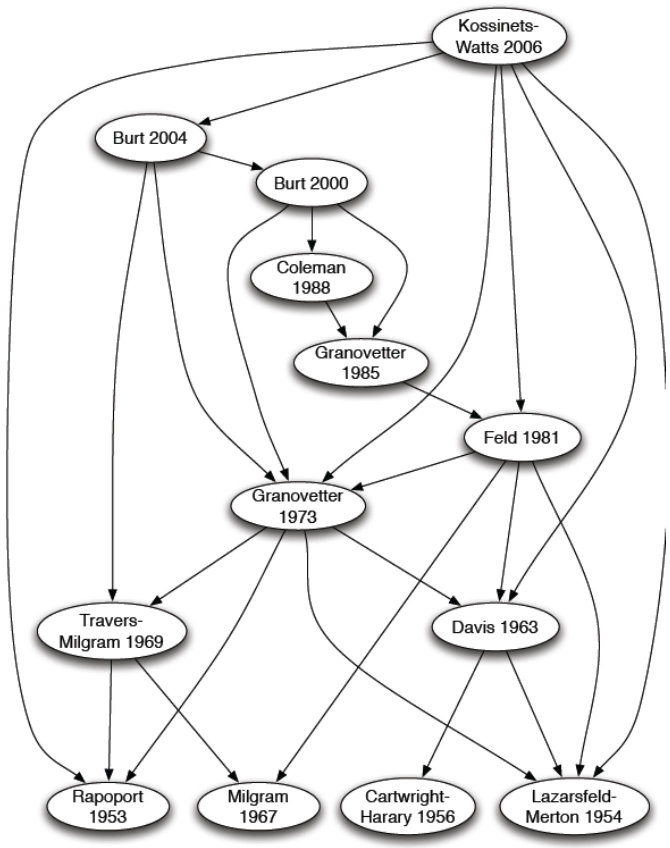


- In early days of the Web links were **navigational**
- Today many links are **transactional** (used **not** to navigate from page to page, but to post, comment, like, buy, …)

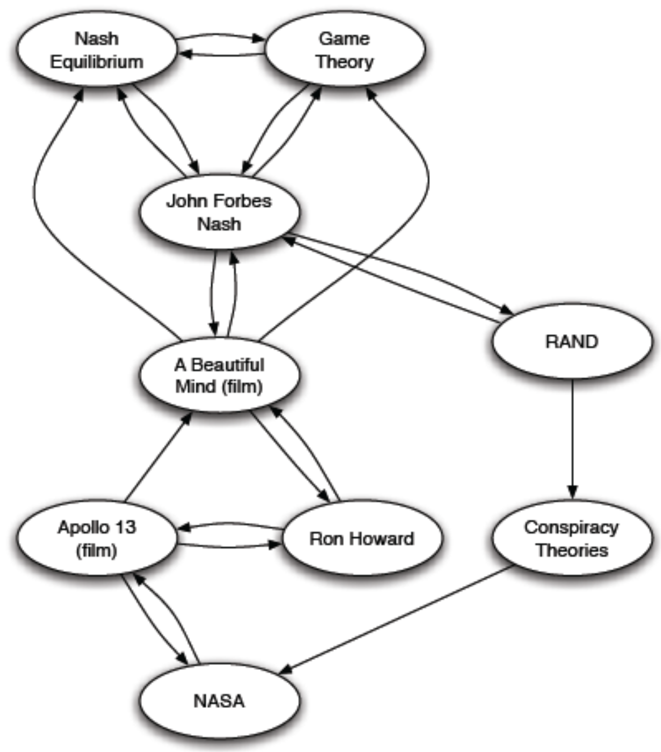
The Web as a Directed Graph



Other Information Networks



Citations



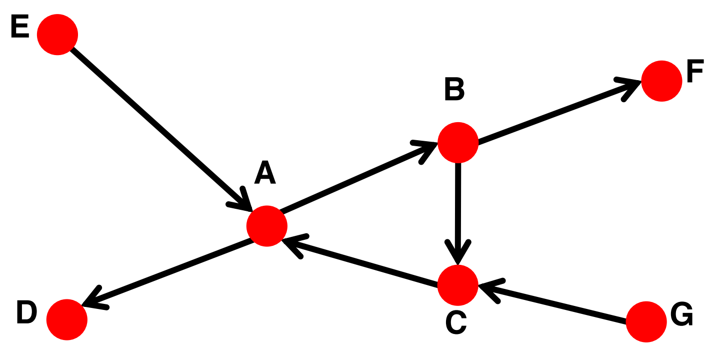
References in an Encyclopedia

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?



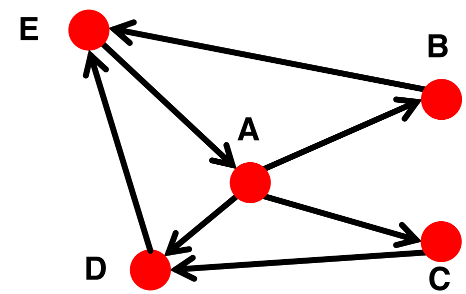
$$In(v) = \{w \mid w \text{ can reach } v\}$$
$$Out(v) = \{w \mid v \text{ can reach } w\}$$

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

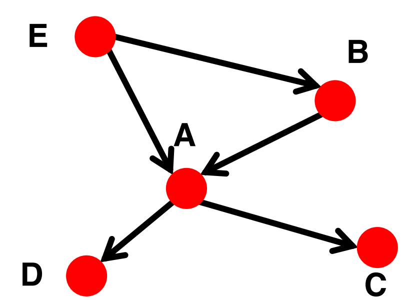
Reasoning about Directed Graphs

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

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



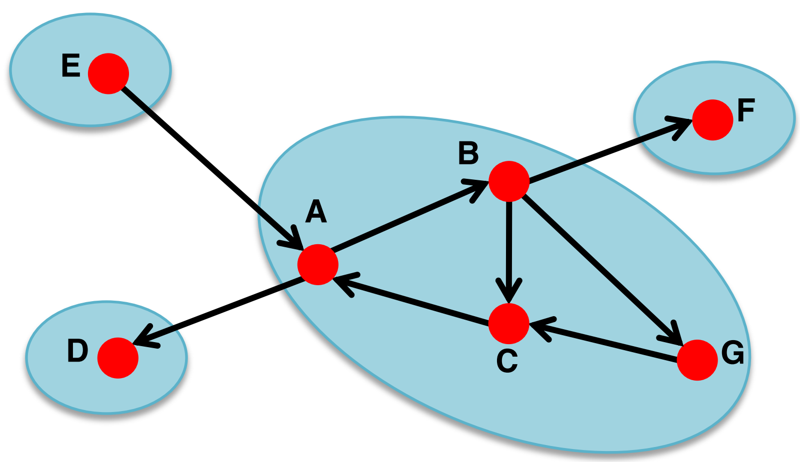
  - **Directed Acyclic Graph (DAG):**
    - Has no cycles: if u can reach v, then v cannot reach u



- **Any directed graph (the Web) can be expressed in terms of these two types!**
  - Is the Web a big strongly connected graph or a DAG?

Strongly Connected Component

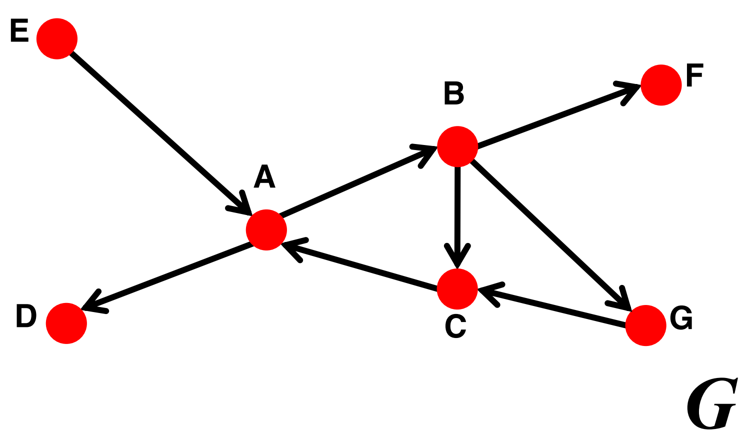
- **A Strongly Connected Component (SCC)** is a set of nodes S so that:
  - Every pair of nodes in S can reach each other
  - There is no larger set containing S with this property



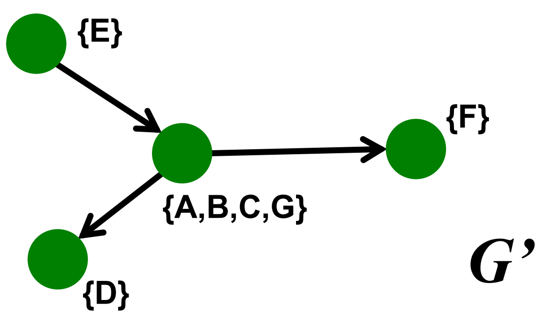
Strongly connected components of the graph: {A,B,C,G}, {D}, {E}, {F}

Strongly Connected Component

- **Fact:** **Every directed graph is a DAG on its SCCs**
  - **(1)** SCCs partition 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:



Structure of the Web

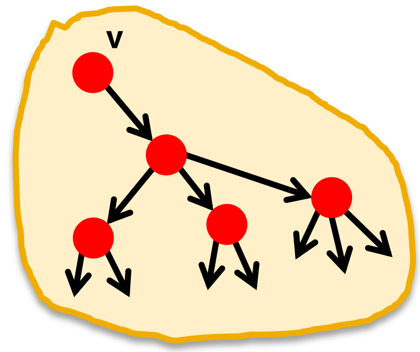
- **Broder et al.: Altavista web crawl (Oct '99)**
  - Web crawl is based on a large set of starting points accumulated over time from various sources, including voluntary submissions.
  - 203 million URLS and 1.5 billion links

Goal: Take a large snapshot of the Web and try to understand how its SCCs "fit together" as a DAG

Tomkins, Broder, and Kumar

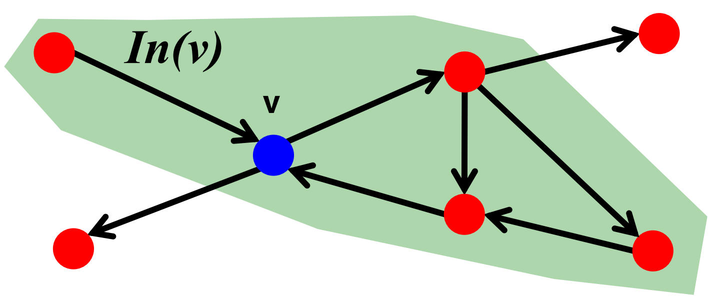
Graph Structure of the Web

- **Computational issue:**
  - Want to find a SCC containing node v?



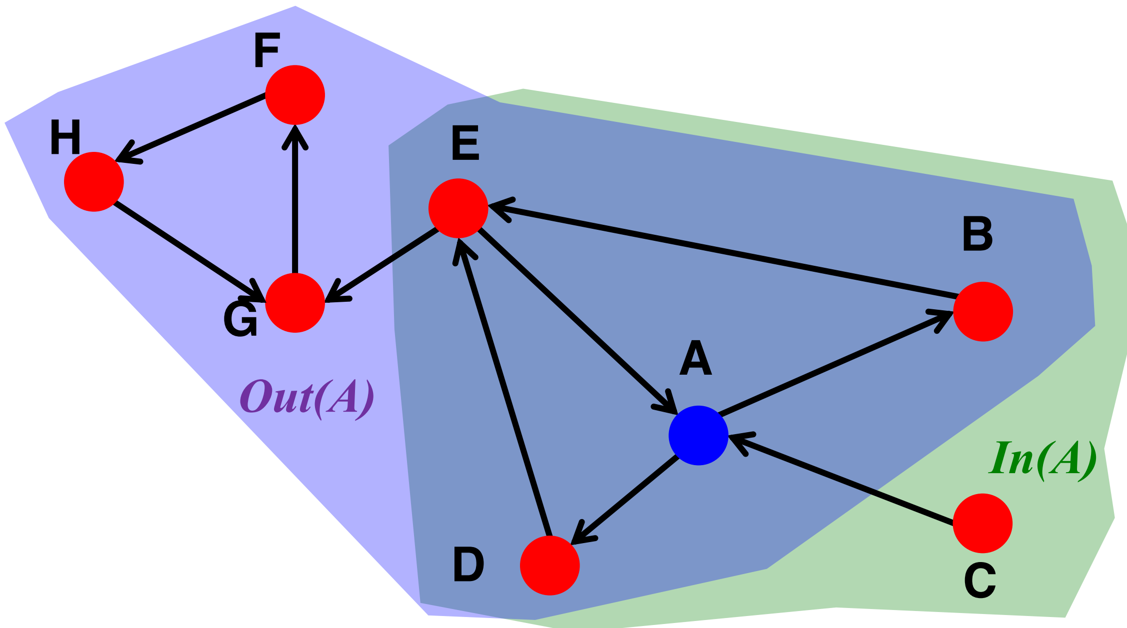
Out(v)

- **Observation:**
  - ***Out(v)*** … nodes that can be reached from v (w/ BFS)
  - **SCC containing v is:** $Out(v) \cap In(v)$
    $= Out(v,G) \cap Out(v,G')$, where G' is G with all edge directions flipped



Out(A) ∩ In(A) = SCC

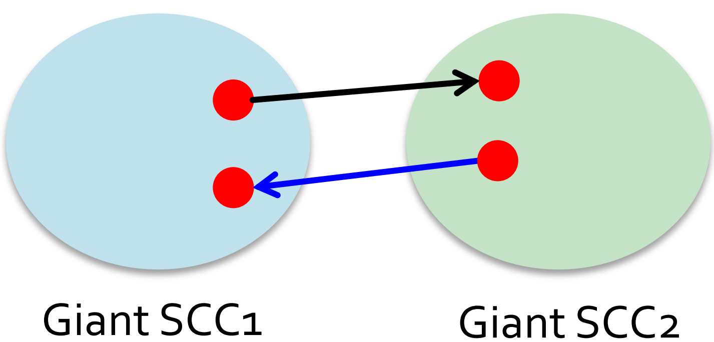
- **Example:**



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

Graph Structure of the Web

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

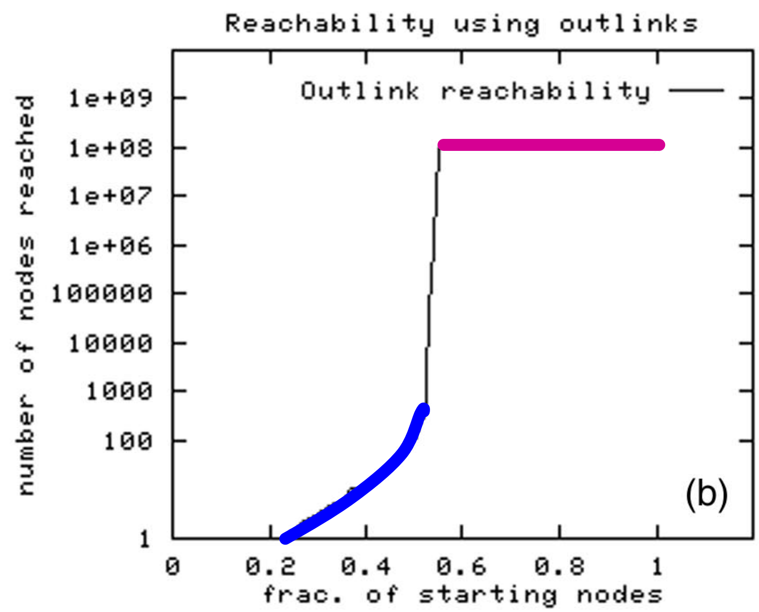


Structure of the Web

- **Directed version of the Web graph:**
  - Altavista crawl from October 1999
    - 203 million URLs, 1.5 billion links

Computation:

- Compute $IN(v)$ and OUT(v) by starting at random nodes.
- **Observation:** The BFS either visits **many nodes** or **very few**

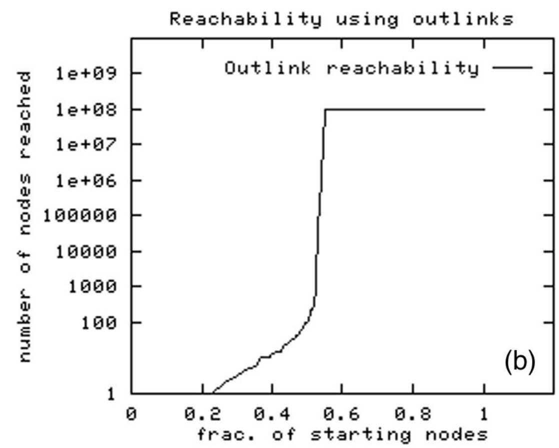


x-axis: rank
y-axis: number of reached nodes

Structure of the Web

Result: Based on IN and OUT of a random node *v*:

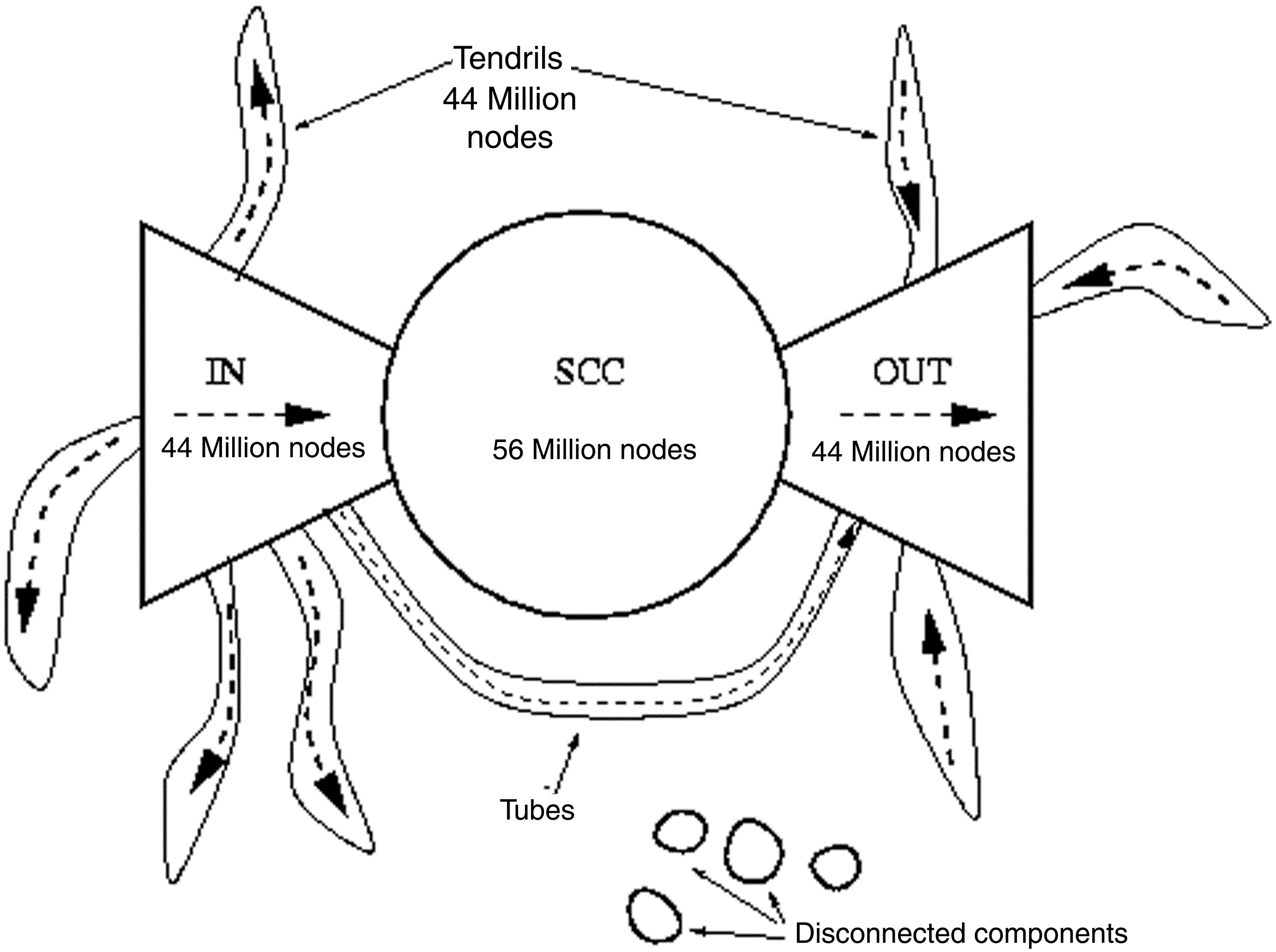
- **Out(*v*)** ≈ 100 million (**50%** nodes)
- **In(*v*)** ≈ 100 million (**50%** nodes)
- **Largest SCC:** 56 million (**28%** nodes)



x-axis: rank
y-axis: number of reached nodes

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

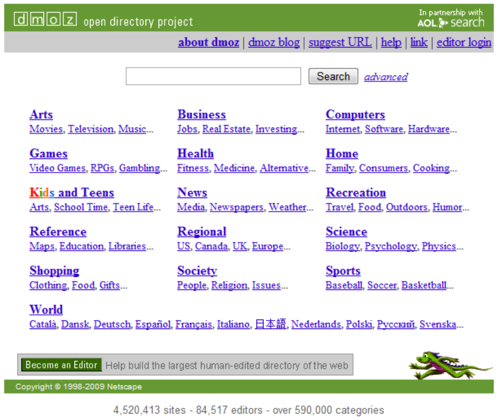
Bowtie Structure of the Web



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

How to Organize the Web
PageRank
(aka the Google Algorithm)

How to Organize the Web?



- **How to organize the Web?**
- **First try:** Human curated **Web directories**
  - Yahoo, DMOZ, LookSmart
- **Second try: Web Search**
  - **Information Retrieval** attempts to find relevant docs in a small and trusted set
    - Newspaper articles, Patents, etc.
  - **But:** Web is **huge**, full of untrusted documents, random things, web spam, etc.
  - **So we need a good way to rank webpages!**

Web Search: 2 Challenges

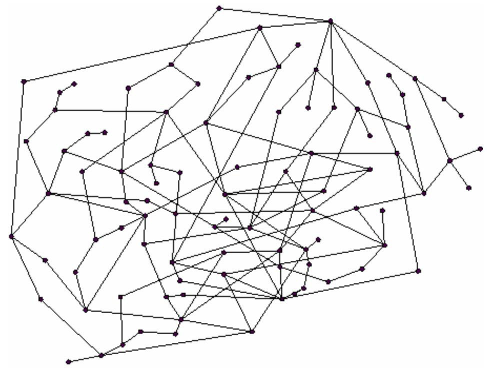
2 challenges of web search:

- **(1) Web contains many sources of information**
  Who to "trust"?
  - **Insight:** Trustworthy pages may point to each other!

- **(2) What is the "best" answer to query "newspaper"?**
  - No single right answer
  - **Insight:** Pages that actually know about newspapers might all be pointing to many newspapers

Ranking Nodes on the Graph

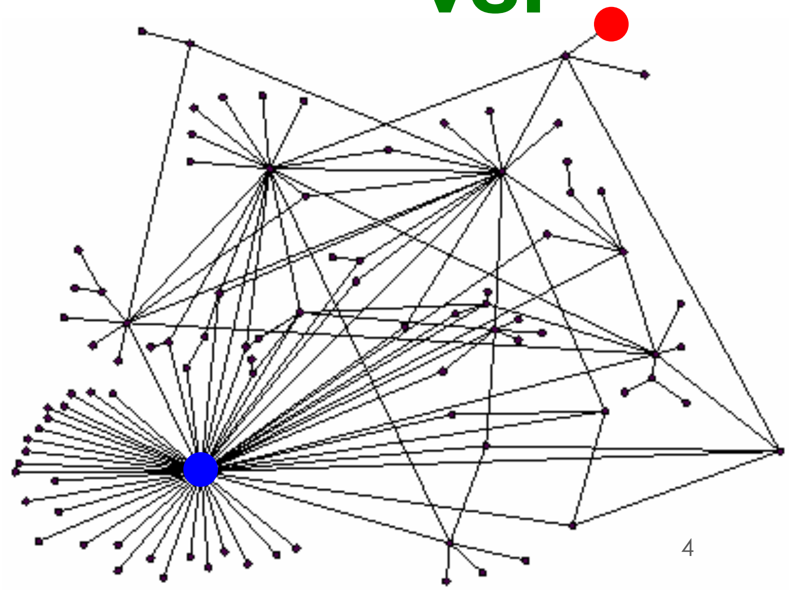
- **All web pages are not equally "important"**
  www.joe-schmoe.com vs. www.stanford.edu



- **We already know:**
  There is large diversity in the web-graph node connectivity.

vs.

- **So, let's rank the pages using the web graph link structure!**



Link Analysis Algorithms

- **We will cover the following Link Analysis approaches** to computing **importance** of nodes in a graph:
  - Hubs and Authorities (HITS)
  - Page Rank
  - ~~Topic-Specific (Personalized) Page Rank~~ <- another time

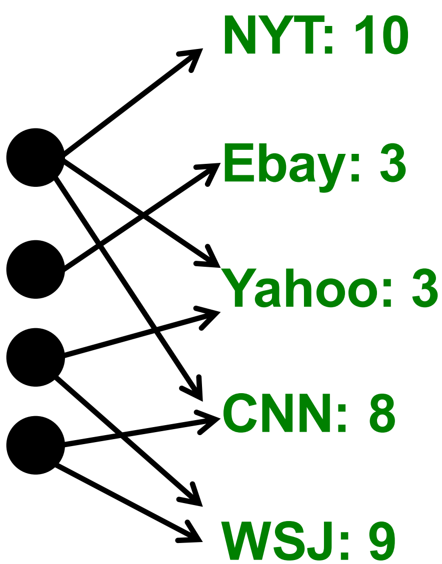
Sidenote: Various notions of **node centrality: Node u**

- **Degree centrality** = degree of u
- **Betweenness centrality** = #shortest paths passing through u
- **Closeness centrality** = avg. length of shortest paths from u to all other nodes of the network
- **Eigenvector centrality** = like PageRank

Hubs and Authorities

Link Analysis

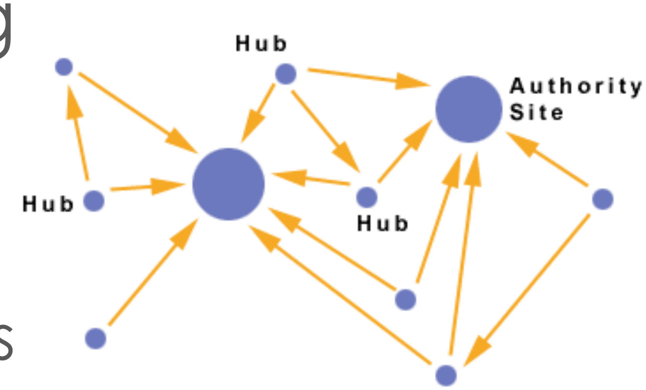
- **Goal** (back to the newspaper example):
  - Don't just find newspapers. Find "experts" – pages that link in a coordinated way to good newspapers
- **Idea: Links as votes**
  - **Page is more important if it has more links**
    - In-coming links? Out-going links?
- **Hubs and Authorities**
  Each page has **2** scores:
  - **Quality as an expert (hub):**
    - Total sum of votes of pages pointed to
  - **Quality as an content (authority):**
    - Total sum of votes of experts
  - **Principle of repeated improvement**



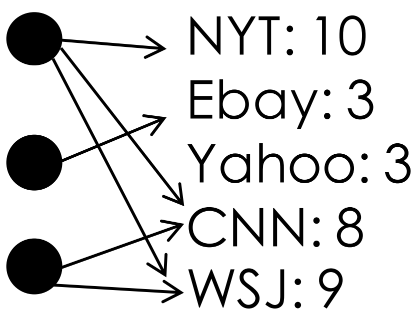
Hubs and Authorities

Interesting pages fall into two classes:

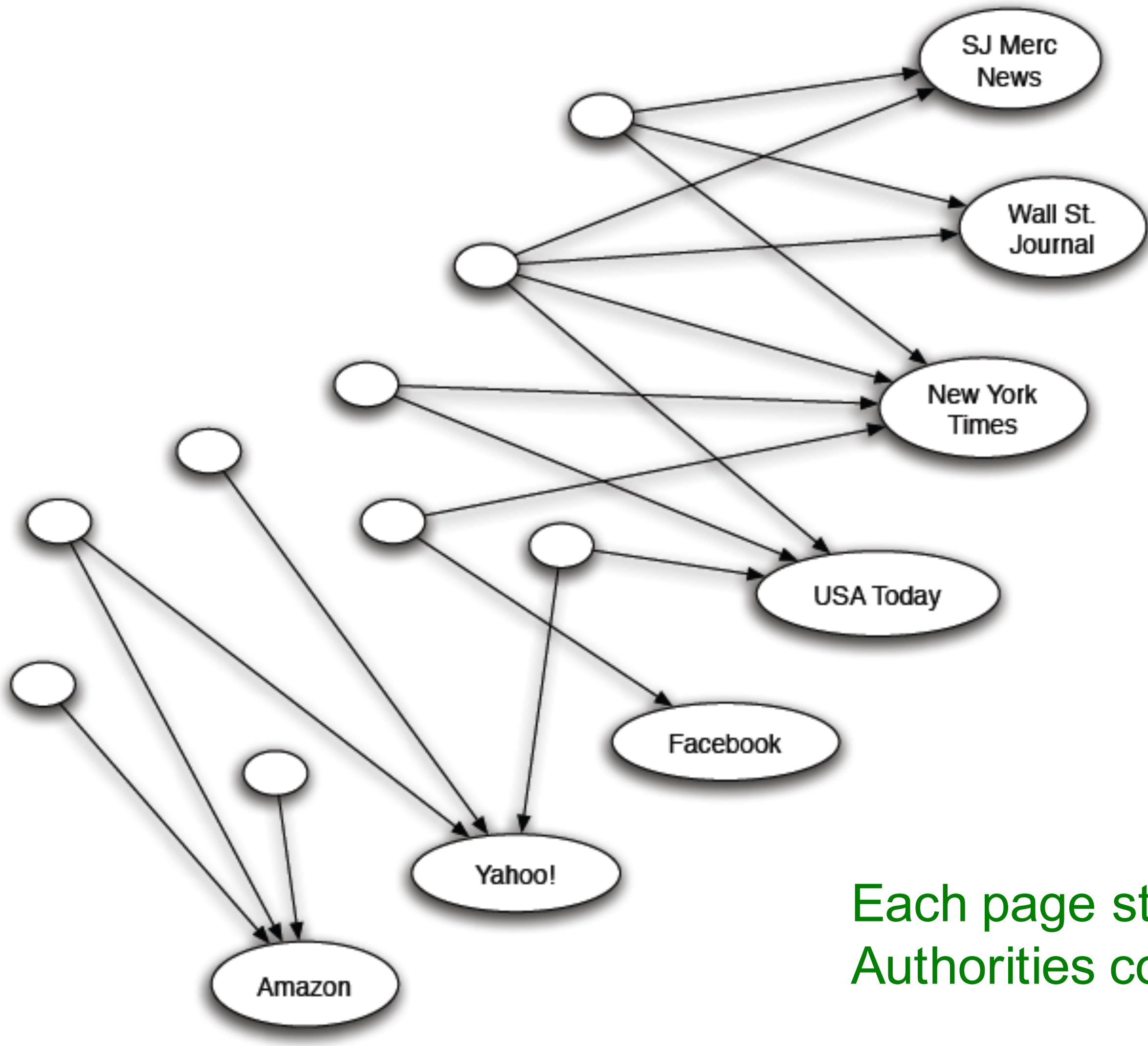
1. **Authorities** are pages containing useful information
   - Newspaper home pages
   - Course home pages
   - Home pages of auto manufacturers



2. **Hubs** are pages that link to authorities
   - List of newspapers
   - Course bulletin
   - List of U.S. auto manufacturers



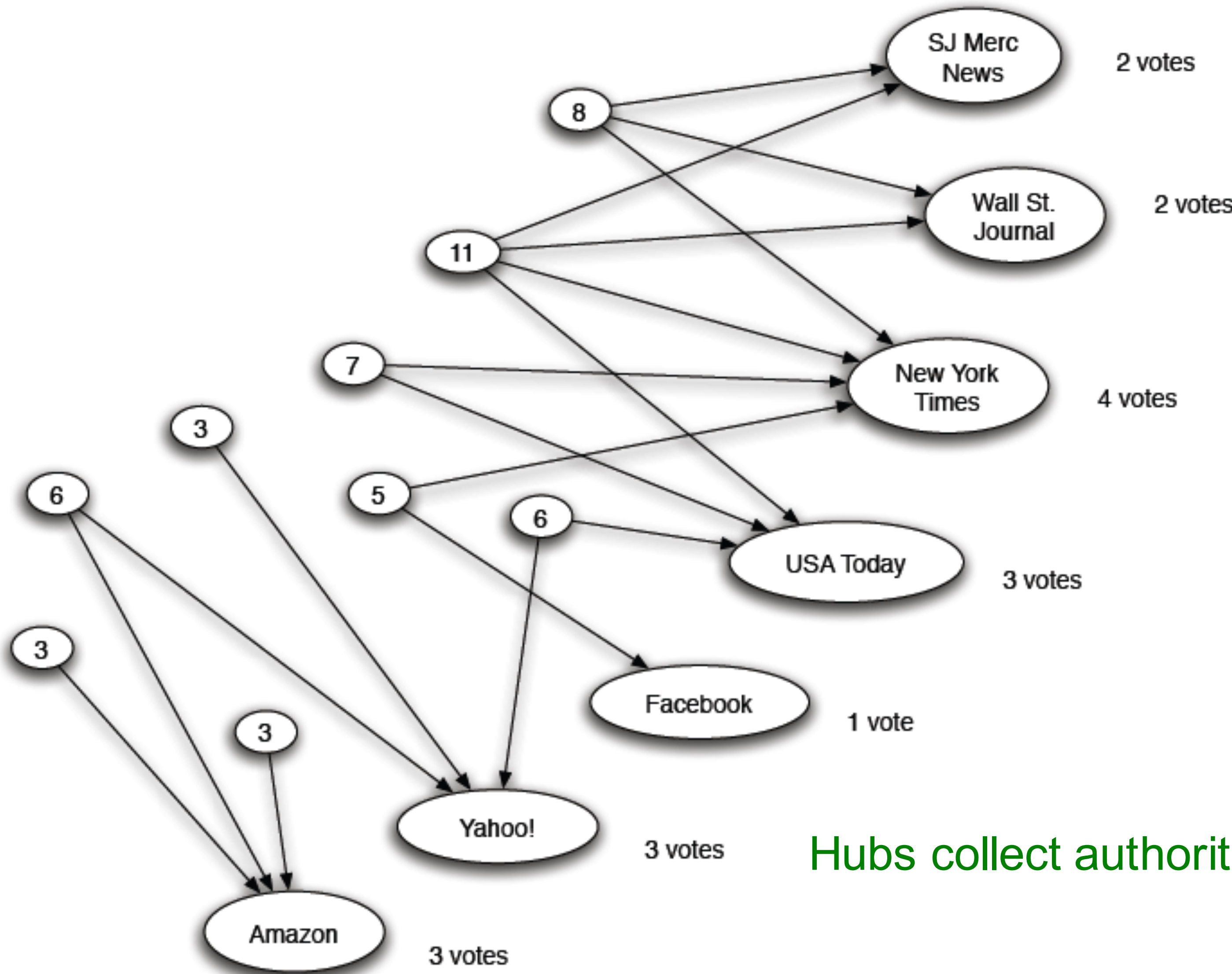
Counting in-links: Authority



Each page starts with **hub score 1**
Authorities collect their votes

(Note this is idealized example. In reality graph is not bipartite and each page has both a hub and the authority score)

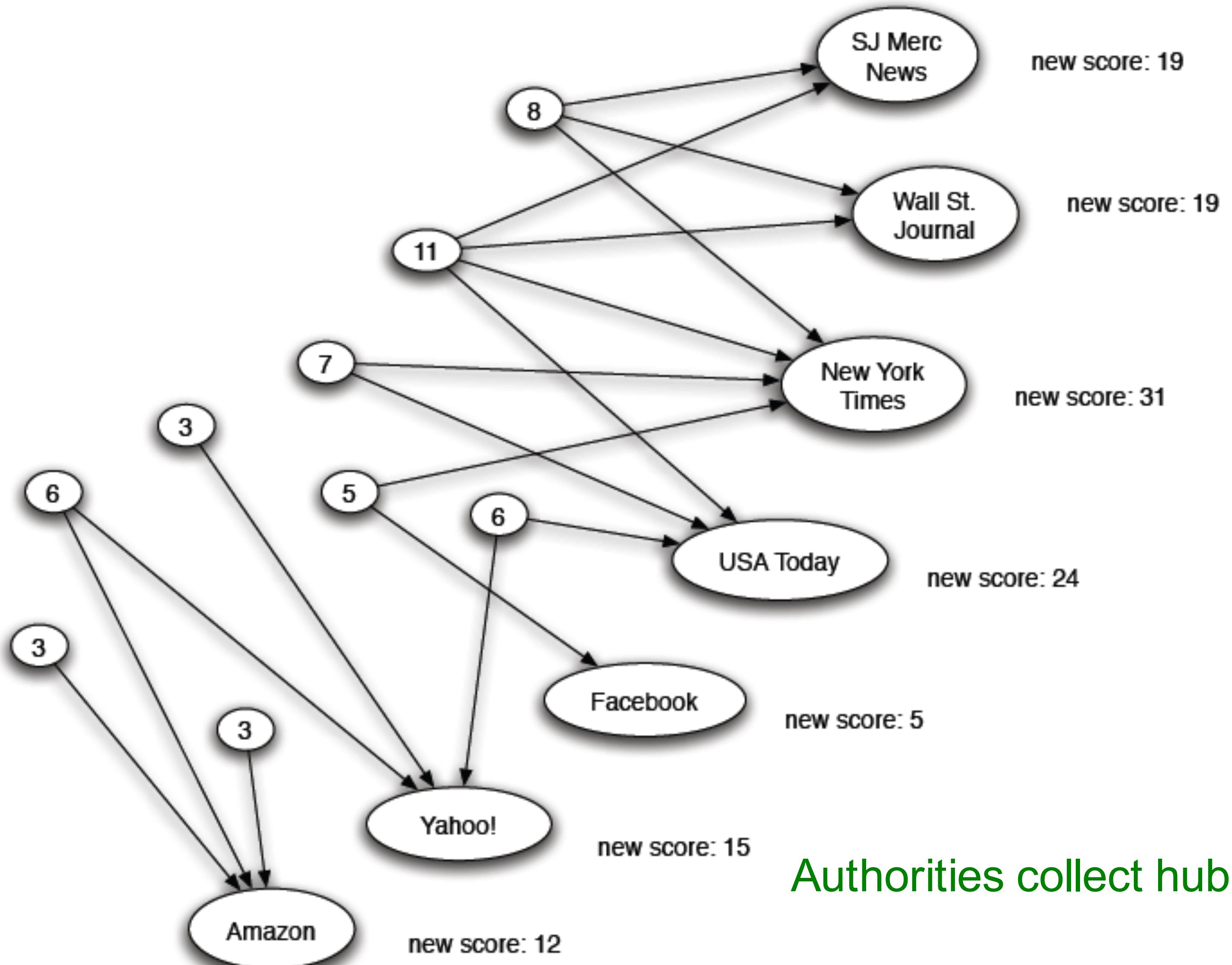
Expert Quality: Hub



Hubs collect authority scores

(Note this is idealized example. In reality graph is not bipartite and each page has both a hub and authority score)

Reweighting



Authorities collect hub scores

(Note this is idealized example. In reality graph is not bipartite and each page has both a hub and authority score)

Mutually Recursive Definition

- **A good hub links to many good authorities**
- **A good authority is linked from many good hubs**
  - Note a self-reinforcing recursive definition
- **Model using two scores for each node:**
  - **Hub** score and **Authority** score
  - Represented as vectors $\boldsymbol{h}$ and $\boldsymbol{a}$, where the i-th element is the hub/authority score of the i-th node

Hubs and Authorities

- **Each page i has 2 scores:**
  - Authority score: $\boldsymbol{a_i}$
  - Hub score: $\boldsymbol{h_i}$

Convergence criteria:

$$\sum_i \left(h_i^{(t)} - h_i^{(t+1)}\right)^2 < \varepsilon$$

$$\sum_i \left(a_i^{(t)} - a_i^{(t+1)}\right)^2 < \varepsilon$$

HITS algorithm:

- Initialize: $a_j^{(0)} = 1/\sqrt{\mathrm{n}},\ \mathrm{h}_\mathrm{j}^{(0)} = 1/\sqrt{\mathrm{n}}$
- Then keep iterating until **convergence**:
  - $\forall \boldsymbol{i}$: Authority: $a_i^{(t+1)} = \sum_{j \to i} h_j^{(t)}$
  - $\forall \boldsymbol{i}$: Hub: $h_i^{(t+1)} = \sum_{i \to j} a_j^{(t)}$
  - $\forall \boldsymbol{i}$: Normalize:

$$\sum_i \left(a_i^{(t+1)}\right)^2 = 1,\ \sum_j \left(h_j^{(t+1)}\right)^2 = 1$$

[K]

Details!

Hubs and Authorities

- **Hits in the vector notation:**
  - Vector $\boldsymbol{a} = (\boldsymbol{a_1} \dots, \boldsymbol{a_n})$, $\boldsymbol{h} = (\boldsymbol{h_1} \dots, \boldsymbol{h_n})$
  - Adjacency matrix $\boldsymbol{A}$ (n x n): $\boldsymbol{A_{ij}} = \boldsymbol{1}$ if $\boldsymbol{i} \rightarrow \boldsymbol{j}$
- **Can rewrite** $\boldsymbol{h_i} = \sum_{i \rightarrow j} \boldsymbol{a_j}$ **as** $\boldsymbol{h_i} = \sum_j \boldsymbol{A_{ij}} \cdot \boldsymbol{a_j}$
- **So:** $\boldsymbol{h} = \boldsymbol{A} \cdot \boldsymbol{a}$ **And similarly:** $\boldsymbol{a} = \boldsymbol{A^T} \cdot \boldsymbol{h}$
- **Repeat until convergence:**
  - $h^{(t+1)} = A \cdot a^{(t)}$
  - $a^{(t+1)} = A^T \cdot h^{(t)}$
  - Normalize $a^{(t+1)}$ and $h^{(t+1)}$

Details!

Hubs and Authorities

- What is $a = A^T \cdot h$?
- Then: $a = \underbrace{A^T \cdot \underbrace{(A \cdot a)}_{\text{new } h}}_{\text{new } a}$
- a is updated (in 2 steps):

$$a = A^T (A\, a) = (A^T A)\, a$$

- h is updated (in 2 steps)

$$h = A\, (A^T h) = (A\, A^T)\, h$$

- Thus, in $2k$ steps:

$$a = (A^T \cdot A)^k \cdot a$$

$$h = (A \cdot A^T)^k \cdot h$$

Repeated matrix powering

Hubs and Authorities

- **<u>Definition:</u> Eigenvectors & Eigenvalues**
- Let $\boldsymbol{R} \cdot \boldsymbol{x} = \boldsymbol{\lambda} \cdot \boldsymbol{x}$
  for some scalar $\boldsymbol{\lambda}$, vector $\boldsymbol{x}$, matrix $\boldsymbol{R}$
  - Then $\boldsymbol{x}$ **is an eigenvector**, and $\boldsymbol{\lambda}$ is its **eigenvalue**
- **The <u>steady state</u> (HITS has converged):**
  - $\boldsymbol{A}^{\boldsymbol{T}} \cdot \boldsymbol{A} \cdot \boldsymbol{a} = \boldsymbol{c}' \cdot \boldsymbol{a}$
  - $\boldsymbol{A} \cdot \boldsymbol{A}^{\boldsymbol{T}} \cdot \boldsymbol{h} = \boldsymbol{c}'' \cdot \boldsymbol{h}$

Note constants *c',c"* don't matter as we normalize them out every step of HITS

- **So**, **authority** $\boldsymbol{a}$ is eigenvector of $\boldsymbol{A}^{\boldsymbol{T}}\boldsymbol{A}$ (associated with the largest eigenvalue)
  Similarly: **hub** $\boldsymbol{h}$ is eigenvector of $\boldsymbol{A}\boldsymbol{A}^{\boldsymbol{T}}$

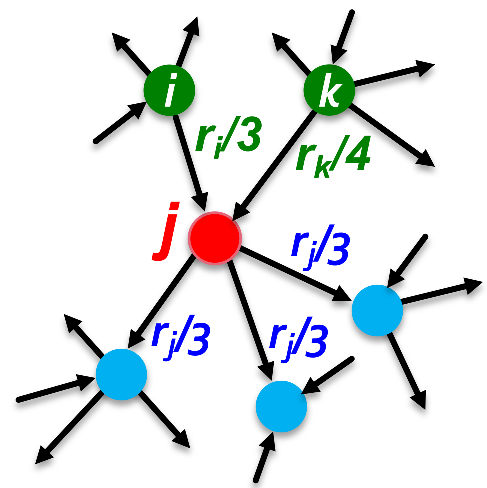
PageRank

Links as Votes

- **Still the same idea: Links as votes**
  - **Page is more important if it has more links**
    - In-coming links? Out-going links?
- **Think of in-links as votes:**
  - www.stanford.edu has 23,400 in-links
  - www.joe-schmoe.com has 1 in-link
- **Are all in-links equal?**
  - Links from important pages count more
  - Recursive question!

PageRank: The "Flow" Model

- **A "vote" from an important page is worth more:**
  - Each link's vote is proportional to the **importance** of its source page
  - If page i with importance r_i has d_i out-links, each link gets r_i / d_i votes
  - Page j's own importance r_j is the sum of the votes on its in-links



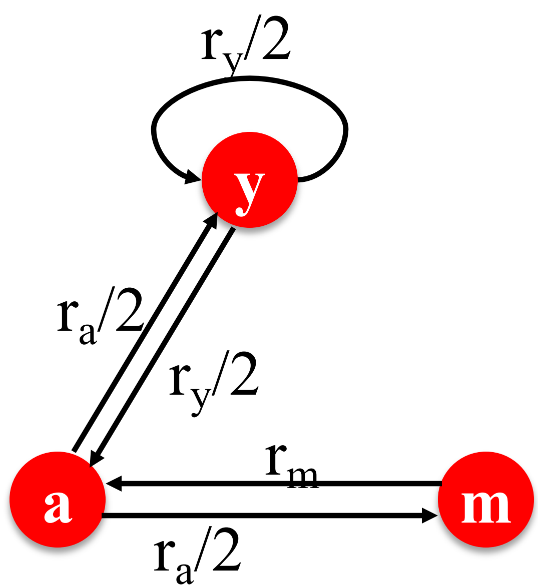
$$r_j = r_i/3 + r_k/4$$

PageRank: The "Flow" Model

- **A page is important if it is pointed to by other important pages**
- **Define a "rank" r_j for node j**

$$r_j = \sum_{i \to j} \frac{r_i}{d_i}$$

d_i ... **out-degree of node i**



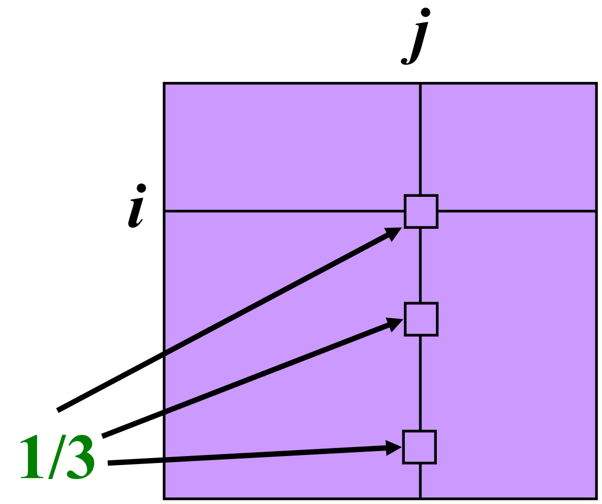
"Flow" equations:

$$\mathbf{r_y} = \mathbf{r_y}/2 + \mathbf{r_a}/2$$
$$\mathbf{r_a} = \mathbf{r_y}/2 + \mathbf{r_m}$$
$$\mathbf{r_m} = \mathbf{r_a}/2$$

You might wonder: Let's just use Gaussian elimination to solve this system of linear equations. Bad idea (G is **too** large!)

PageRank: Matrix Formulation

- **Stochastic adjacency matrix M**
  - Let page j have d_j out-links
  - If $j \rightarrow i$, then $M_{ij} = \frac{1}{d_j}$
    - M is a **column stochastic matrix**
      - **Columns** sum to **1**



M

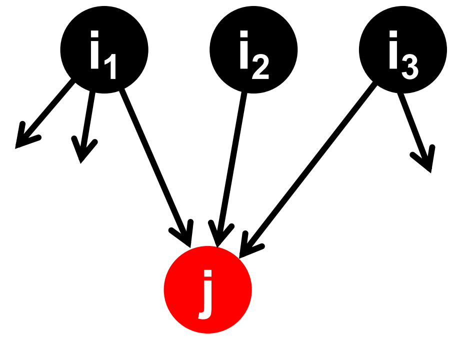
- **Rank vector r:** An entry per page
  - r_i is the importance score of page i
  - $\sum_i r_i = 1$
- **The flow equations can be written**

$$r = M \cdot r$$

$$r_j = \sum_{i \to j} \frac{r_i}{d_i}$$

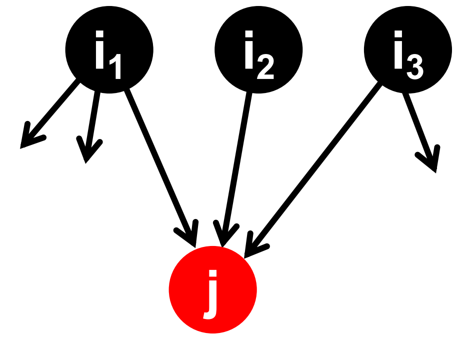
Random Walk Interpretation

- **Imagine a random web surfer:**
  - At any time $\boldsymbol{t}$, surfer is on some page i
  - At time $\boldsymbol{t+1}$, the surfer follows an out-link from $\boldsymbol{i}$ uniformly at random
  - Ends up on some page $\boldsymbol{j}$ linked from $\boldsymbol{i}$
  - Process repeats indefinitely
- **Let:**
  - $\boldsymbol{p(t)}$ … vector whose i^{th} coordinate is the prob. that the surfer is at page i at time t
  - So, $\boldsymbol{p(t)}$ is a probability distribution over pages



$$r_j = \sum_{i \to j} \frac{r_i}{\mathrm{d}_{\mathrm{out}}(\mathrm{i})}$$

The Stationary Distribution



$$p(t+1) = \mathrm{M} \cdot p(t)$$

- **Where is the surfer at time $t+1$?**
  - Follows a link uniformly at random

    $$\boldsymbol{p}(\boldsymbol{t}+\mathbf{1}) = \boldsymbol{M} \cdot \boldsymbol{p}(\boldsymbol{t})$$

- Suppose the random walk reaches a state

  $$\boldsymbol{p}(\boldsymbol{t}+\mathbf{1}) = \boldsymbol{M} \cdot \boldsymbol{p}(\boldsymbol{t}) = \boldsymbol{p}(\boldsymbol{t})$$

  then $\boldsymbol{p}(t)$ is **stationary distribution** of a random walk
- **Our original rank vector** $\boldsymbol{r}$ satisfies $\boldsymbol{r} = \boldsymbol{M} \cdot \boldsymbol{r}$
  - **So, $\boldsymbol{r}$ is a stationary distribution for the random walk**

PageRank: How to solve?

PageRank: How to solve?

Given a web graph with *n* nodes, where the nodes are pages and edges are hyperlinks

- Assign each node an initial page rank
- Repeat until convergence ($\Sigma_i \, |r_i^{(t+1)} - r_i^{(t)}| < \varepsilon$)
  - Calculate the page rank of each node

$$r_j^{(t+1)} = \sum_{i \to j} \frac{r_i^{(t)}}{\mathrm{d}_i}$$

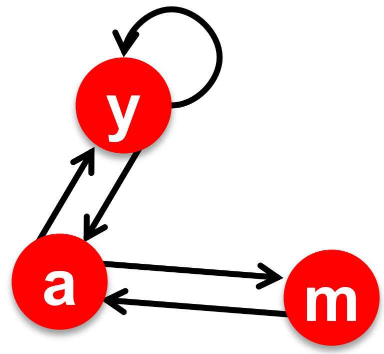
d_i **…. out-degree of node** i

PageRank: How to solve?

- **Power Iteration:**
  - Set $r_j \leftarrow 1/N$
  - **1:** $r'_j \leftarrow \sum_{i \rightarrow j} \frac{r_i}{d_i}$
  - **2:** $r \leftarrow r'$
  - If $|r - r'| > \varepsilon$: goto **1**
- **Example:**

$$\begin{bmatrix} r_y \\ r_a \\ r_m \end{bmatrix} = \begin{matrix} 1/3 \\ 1/3 \\ 1/3 \end{matrix}$$

Iteration 0, 1, 2, …

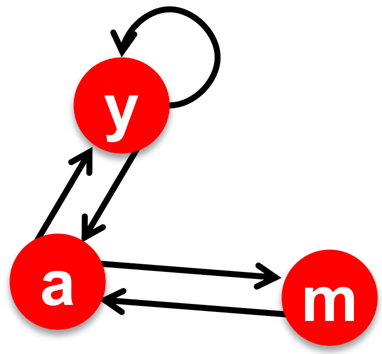


	y	a	m
y	½	½	0
a	½	0	1
m	0	½	0

$$\mathbf{r_y} = \mathbf{r_y}/2 + \mathbf{r_a}/2$$
$$\mathbf{r_a} = \mathbf{r_y}/2 + \mathbf{r_m}$$
$$\mathbf{r_m} = \mathbf{r_a}/2$$

PageRank: How to solve?

- **Power Iteration:**
  - Set $r_j \leftarrow 1/N$
  - **1:** $r'_j \leftarrow \sum_{i \to j} \frac{r_i}{d_i}$
  - **2:** $r \leftarrow r'$
  - If $|r - r'| > \varepsilon$: goto **1**
- **Example:**



	y	a	m
y	½	½	0
a	½	0	1
m	0	½	0

$$\mathbf{r_y} = \mathbf{r_y}/2 + \mathbf{r_a}/2$$
$$\mathbf{r_a} = \mathbf{r_y}/2 + \mathbf{r_m}$$
$$\mathbf{r_m} = \mathbf{r_a}/2$$

$$\begin{bmatrix} r_y \\ r_a \\ r_m \end{bmatrix} = \begin{array}{cccccc} 1/3 & 1/3 & 5/12 & 9/24 & & 6/15 \\ 1/3 & 3/6 & 1/3 & 11/24 & \ldots & 6/15 \\ 1/3 & 1/6 & 3/12 & 1/6 & & 3/15 \end{array}$$

Iteration 0, 1, 2, …

PageRank: Three Questions

$$r_j^{(t+1)} = \sum_{i \to j} \frac{r_i^{(t)}}{\mathrm{d_i}}$$

or equivalently

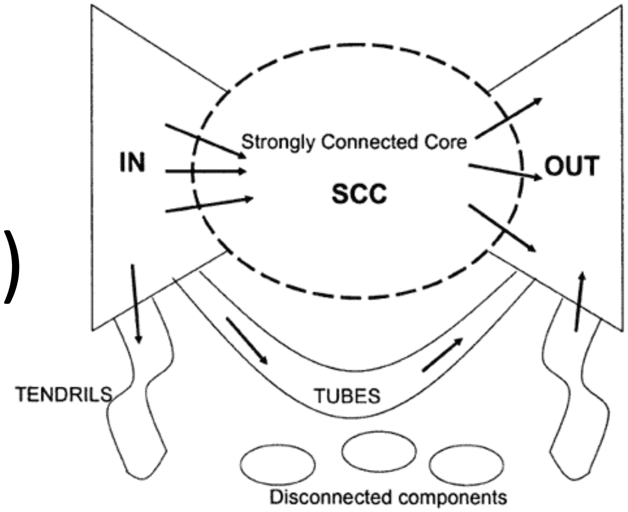
$$r = Mr$$

- **Does this converge?**
- **Does it converge to what we want?**
- **Are the results reasonable?**

RageRank: Problems

Two problems:

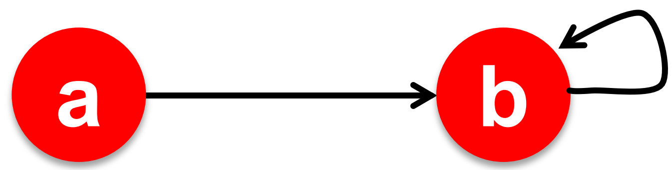
- **(1)** Some pages are **dead ends** (have no out-links)
  - Such pages cause importance to "leak out"



- **(2)** **Spider traps** (all out-links are within the group)
  - Eventually spider traps absorb all importance

Does this converge to what we want?

- **The "Spider trap" problem:**



$$r_j^{(t+1)} = \sum_{i \to j} \frac{r_i^{(t)}}{\mathrm{d_i}}$$

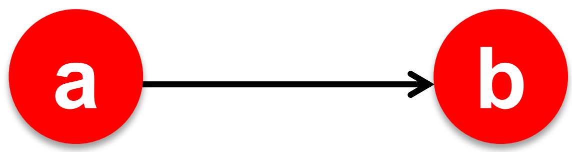
- **Example:**

Iteration: 0, 1, 2, 3…

$$\begin{matrix} \mathrm{r_a} \\ \mathrm{r_b} \end{matrix} = \begin{array}{c|c|c|c} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \end{array}$$

Does it converge to what we want?

- **The "Dead end" problem:**



$$r_j^{(t+1)} = \sum_{i \to j} \frac{r_i^{(t)}}{\mathrm{d_i}}$$

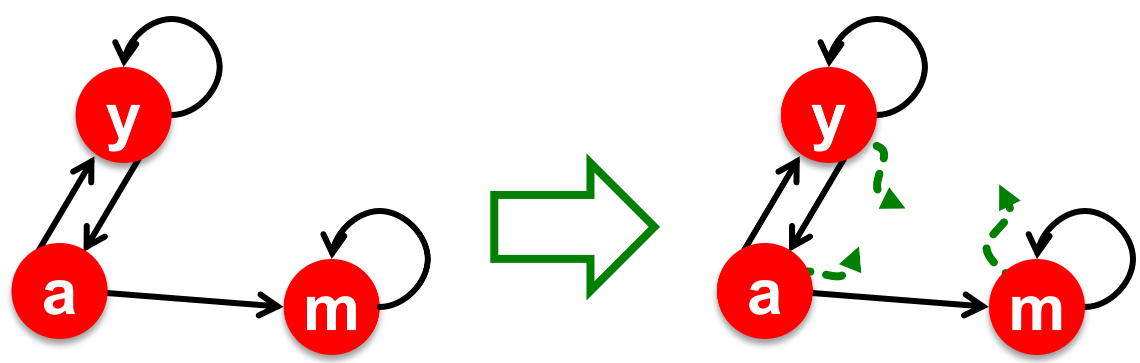
- **Example:**

Iteration: 0, 1, 2, 3...

$$\begin{matrix} \mathrm{r_a} \\ \mathrm{r_b} \end{matrix} = \begin{array}{c|c|c|c} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{array}$$

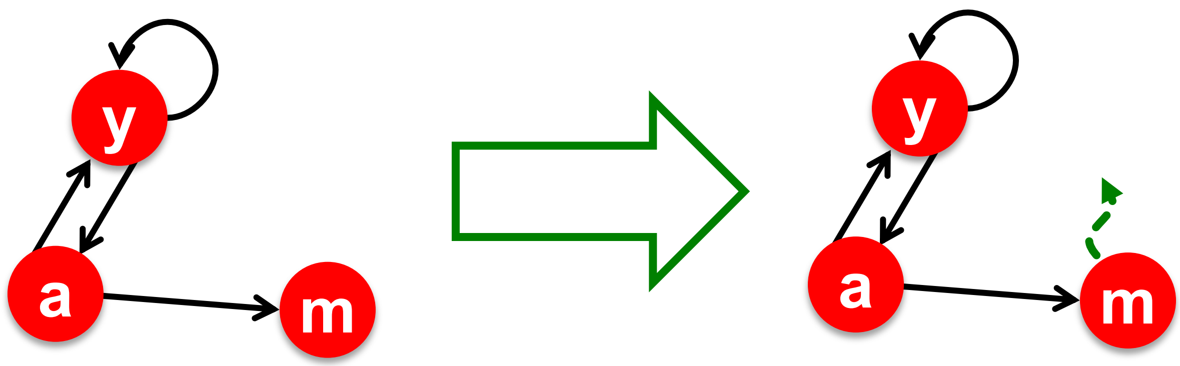
Solution to Spider Traps

- **The Google solution for spider traps: At each time step, the random surfer has two options**
  - With prob. β, follow a link at random
  - With prob. **1-β,** jump to a random page
  - Common values for β are in the range 0.8 to 0.9
- **Surfer will teleport out of spider trap within a few time steps**



Solution to Dead Ends

- **Teleports:** Follow random teleport links with probability **1.0** from dead-ends
  - Adjust matrix accordingly



	y	a	m
y	½	½	0
a	½	0	0
m	0	½	0

	y	a	m
y	½	½	⅓
a	½	0	⅓
m	0	½	⅓

Final PageRank Equation

- **Google's solution: At each step, random surfer has two options:**
  - With probability β, follow a link at random
  - With probability $1\text{-}\beta$, jump to some random page
- **PageRank equation** [Brin-Page, '98]

$$r_j = \sum_{i \rightarrow j} \beta \frac{r_i}{d_i} + (1-\beta)\frac{1}{n}$$

d_i ... out-degree of node i

The above formulation assumes that M has no dead ends. We can either preprocess matrix M **(bad!)** or explicitly follow random teleport links with probability 1.0 from dead-ends. See P. Berkhin, *A Survey on PageRank Computing*, Internet Mathematics, 2005.

The PageRank Algorithm

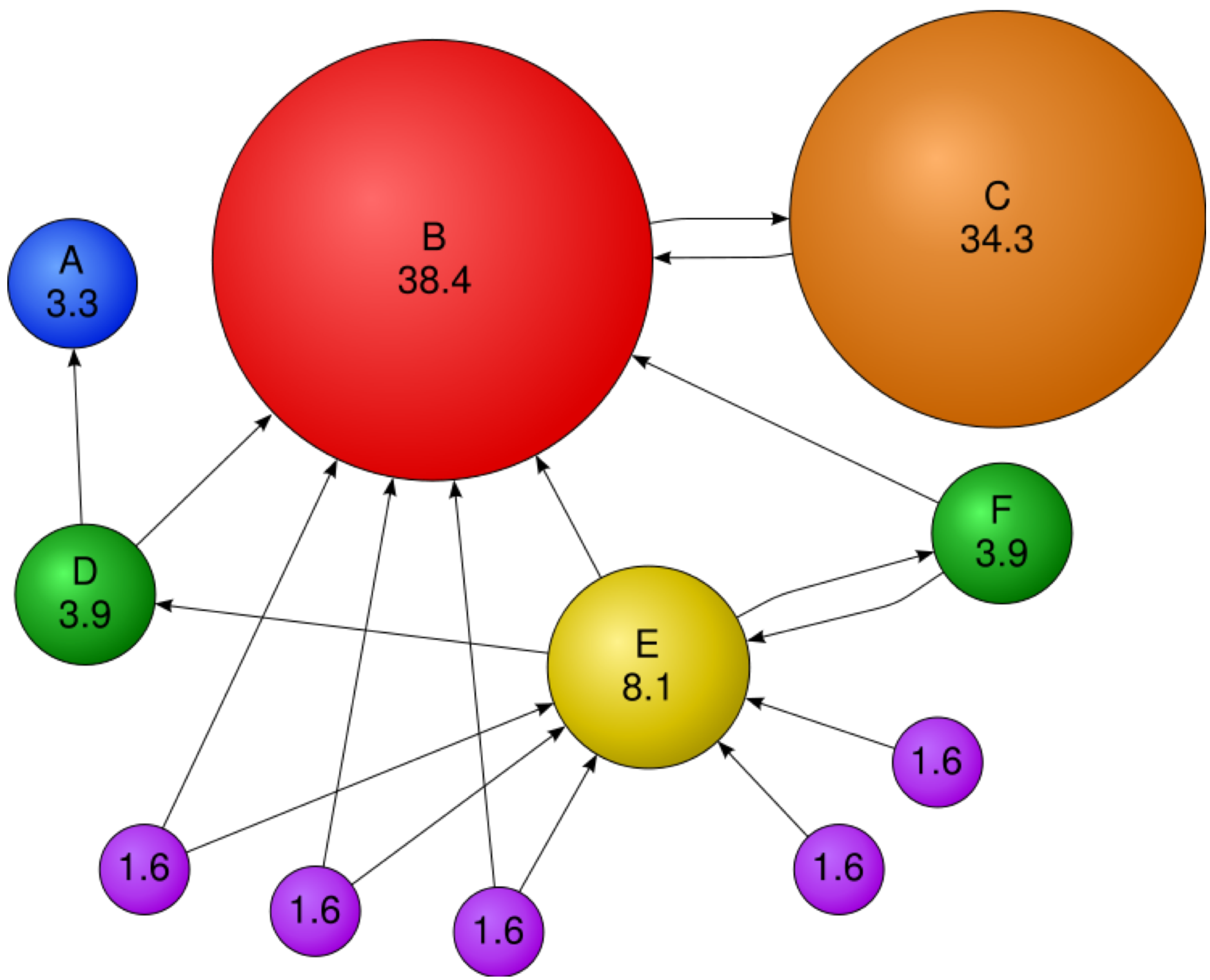
- **Input: Graph G and parameter β**
  - Directed graph G with **spider traps** and **dead ends**
  - Parameter β
- **Output: PageRank vector r**
  - **Set:** $r_j^{(0)} = \frac{1}{N}, \quad t = 1$
  - **do:**
    - $\forall j$: $\boldsymbol{r'}_j^{(t)} = \sum_{i \to j} \boldsymbol{\beta} \frac{\boldsymbol{r}_i^{(t-1)}}{\boldsymbol{d_i}}$

      $\boldsymbol{r'}_j^{(t)} = \mathbf{0}$ if in-deg. of $\boldsymbol{j}$ is $\mathbf{0}$
    - **Now re-insert the leaked PageRank:**

      $\forall \boldsymbol{j}$: $\boldsymbol{r}_j^{(t)} = \boldsymbol{r'}_j^{(t)} + \boxed{\frac{\mathbf{1-S}}{\boldsymbol{N}}}$ **where:** $S = \sum_j r'^{(t)}_j$
    - $\boldsymbol{t = t + 1}$
  - **while** $\sum_j \left| r_j^{(t)} - r_j^{(t-1)} \right| > \varepsilon$

Example

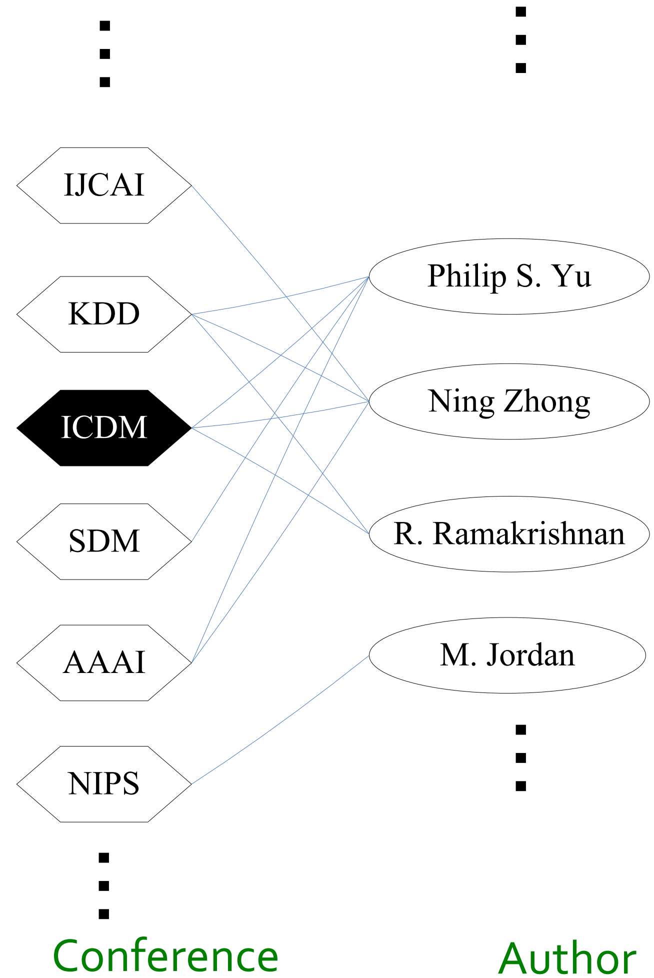
Node size proportional to the PageRank score



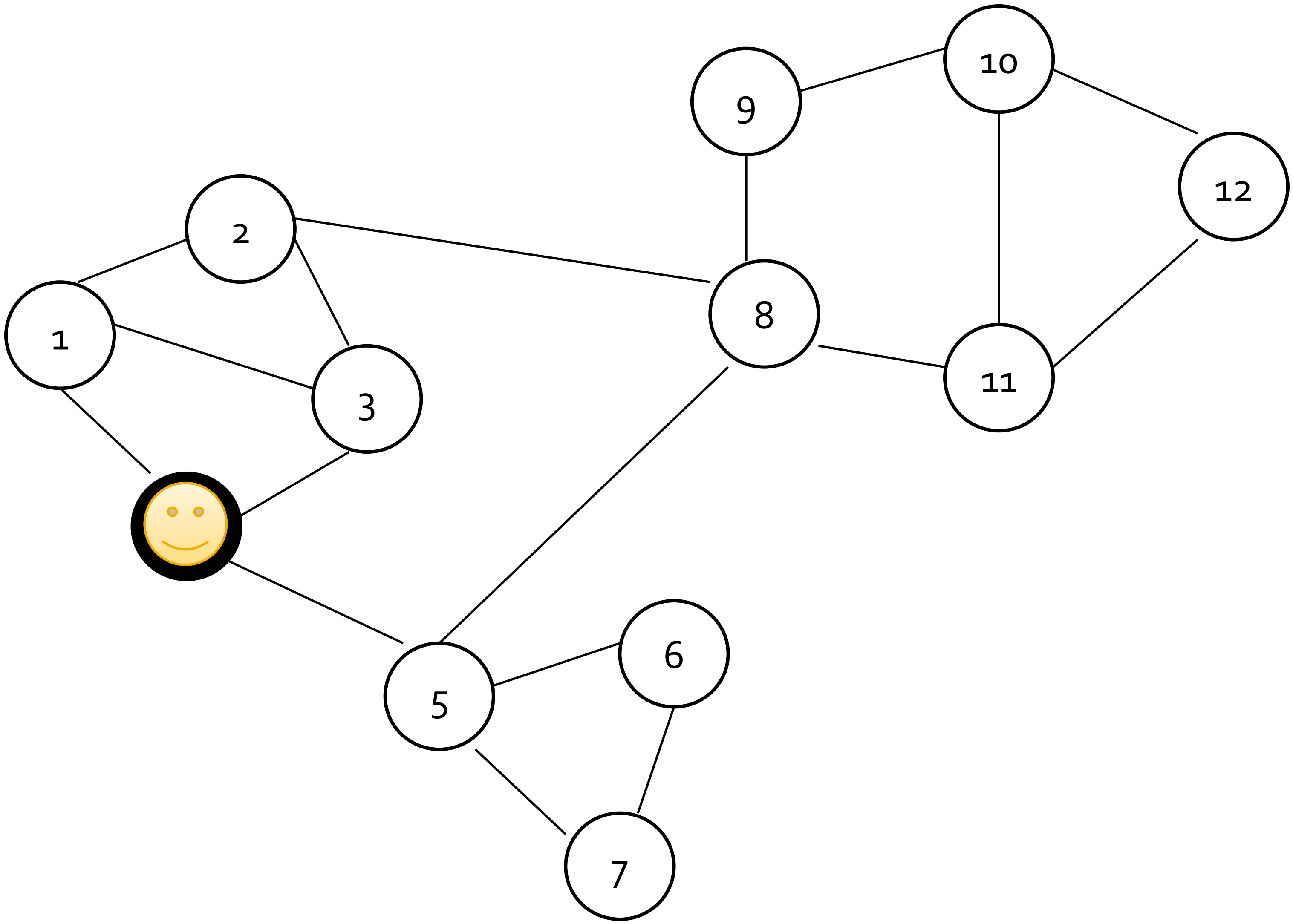
Random Walk with Restarts and Personalized PageRank

Example Application: Graph Search

- **Given:** Conferences-to-authors graph
- **Goal:** Proximity on graphs
  - Q: What is most related conference to ICDM?



Random Walk with Restarts



Personalized PageRank

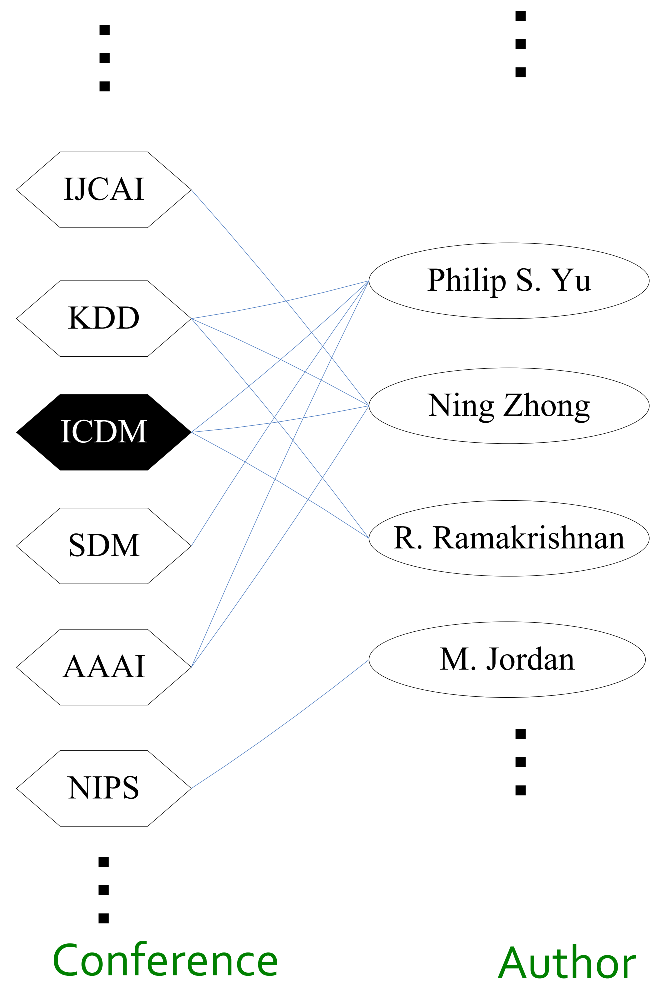
- **<u>Goal:</u>** Evaluate pages not just by popularity but by how close they are to the topic
- **Teleporting can go to:**
  - **Any page with equal probability**
    - PageRank (we used this so far)
  - **A topic-specific set of "relevant" pages**
    - Topic-specific (personalized) PageRank (S …teleport set)

$$
\begin{aligned}
M'_{ij} &= \beta\, M_{ij} + (1-\beta)/|S| \quad \text{if } i \in S \\
&= \beta\, M_{ij} \quad \text{otherwise}
\end{aligned}
$$

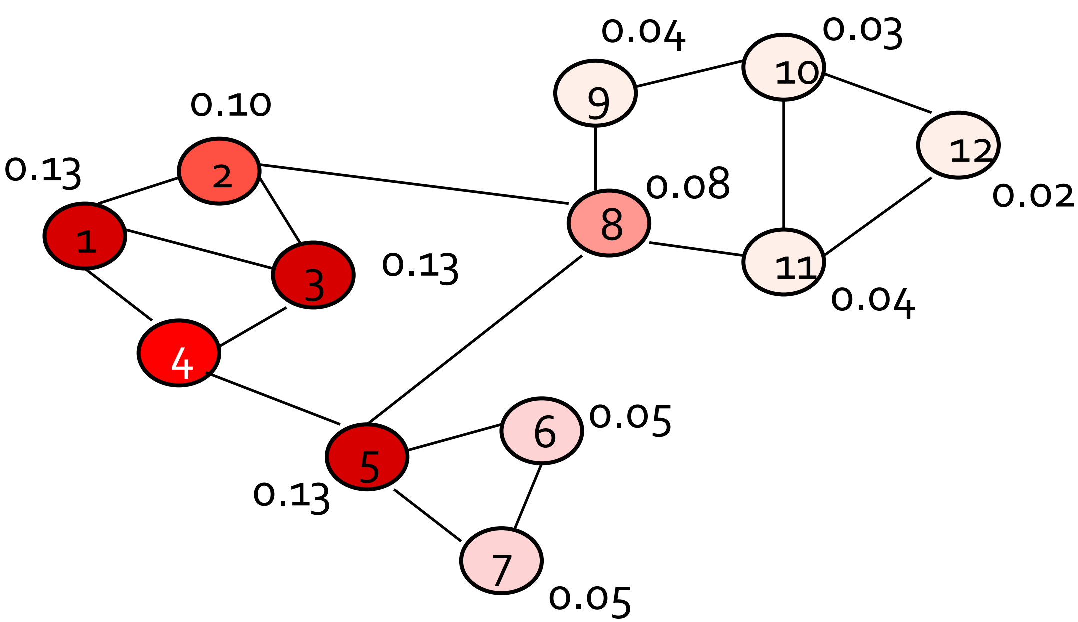
  - **A single page/node ($|S| = 1$),**
    - Random Walk with Restarts

PageRank: Applications

- **Graphs and web search:**
  - Ranks nodes by "importance"
- **Personalized PageRank:**
  - Ranks proximity of nodes to the teleport set S
- **Proximity on graphs:**
  - **Q:** What is most related conference to **ICDM**?
  - **Random Walks with Restarts**
    - Teleport back to the starting node: $S = \{ \text{ single node } \}$



Random Walk with Restarts



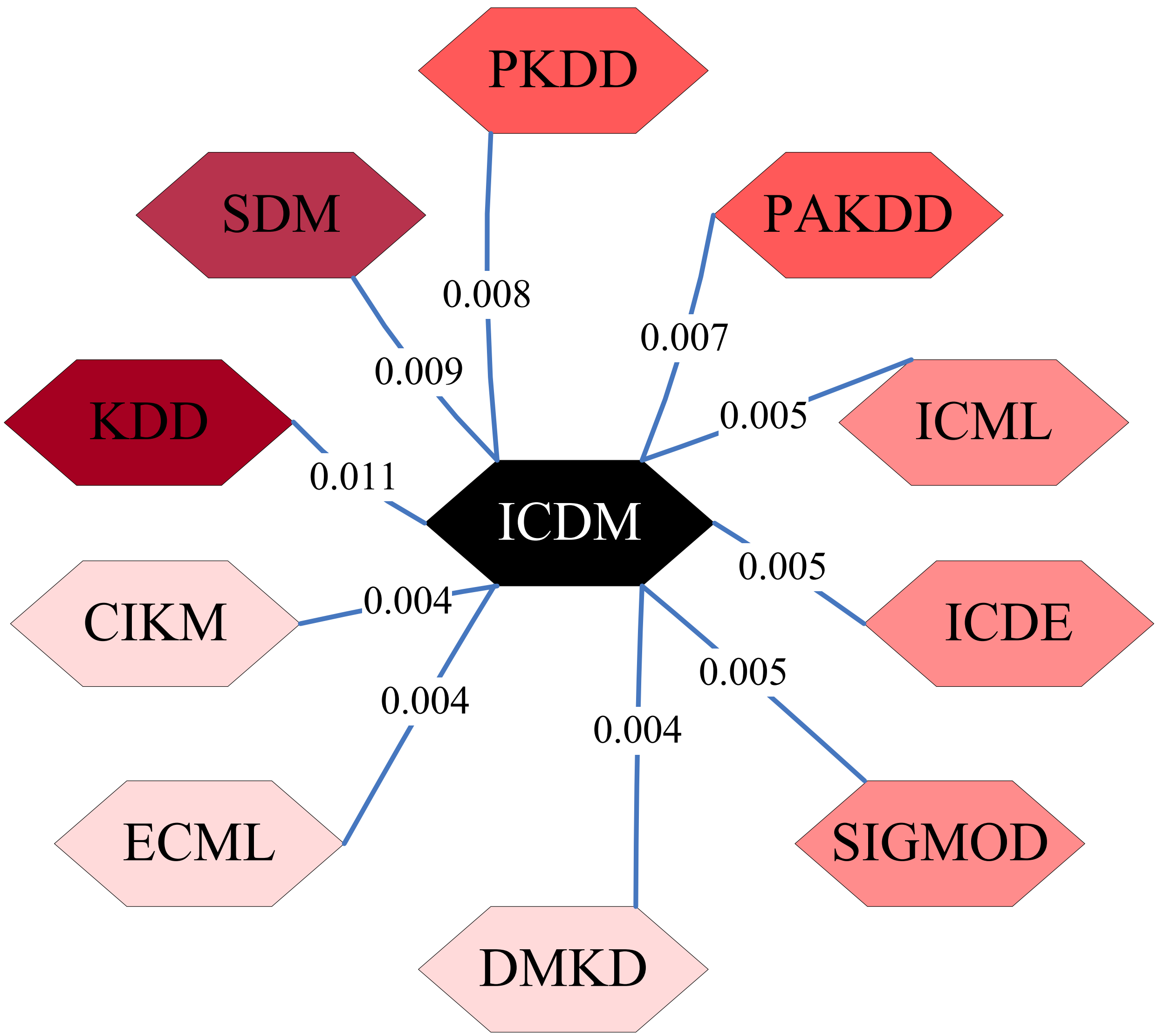
	Node 4
Node 1	0.13
Node 2	0.10
Node 3	0.13
Node 4	/
Node 5	0.13
Node 6	0.05
Node 7	0.05
Node 8	0.08
Node 9	0.04
Node 10	0.03
Node 11	0.04
Node 12	0.02

S={4}

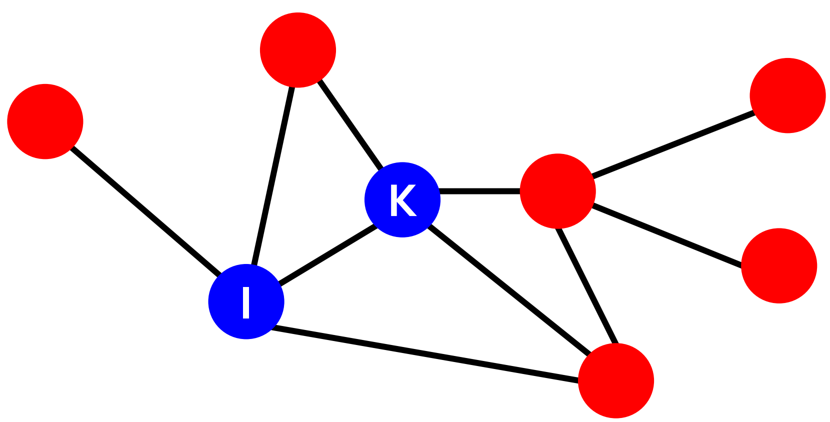
Notice: Nearby nodes have higher scores (are more red)

Ranking vector

Most related conferences to ICDM



Personalized PageRank



Graph of CS conferences

Q: Which conferences are closest to KDD & ICDM?

A: Personalized PageRank with teleport set *S*={KDD, ICDM}