Ranking

Internet Analytics (COM-308)

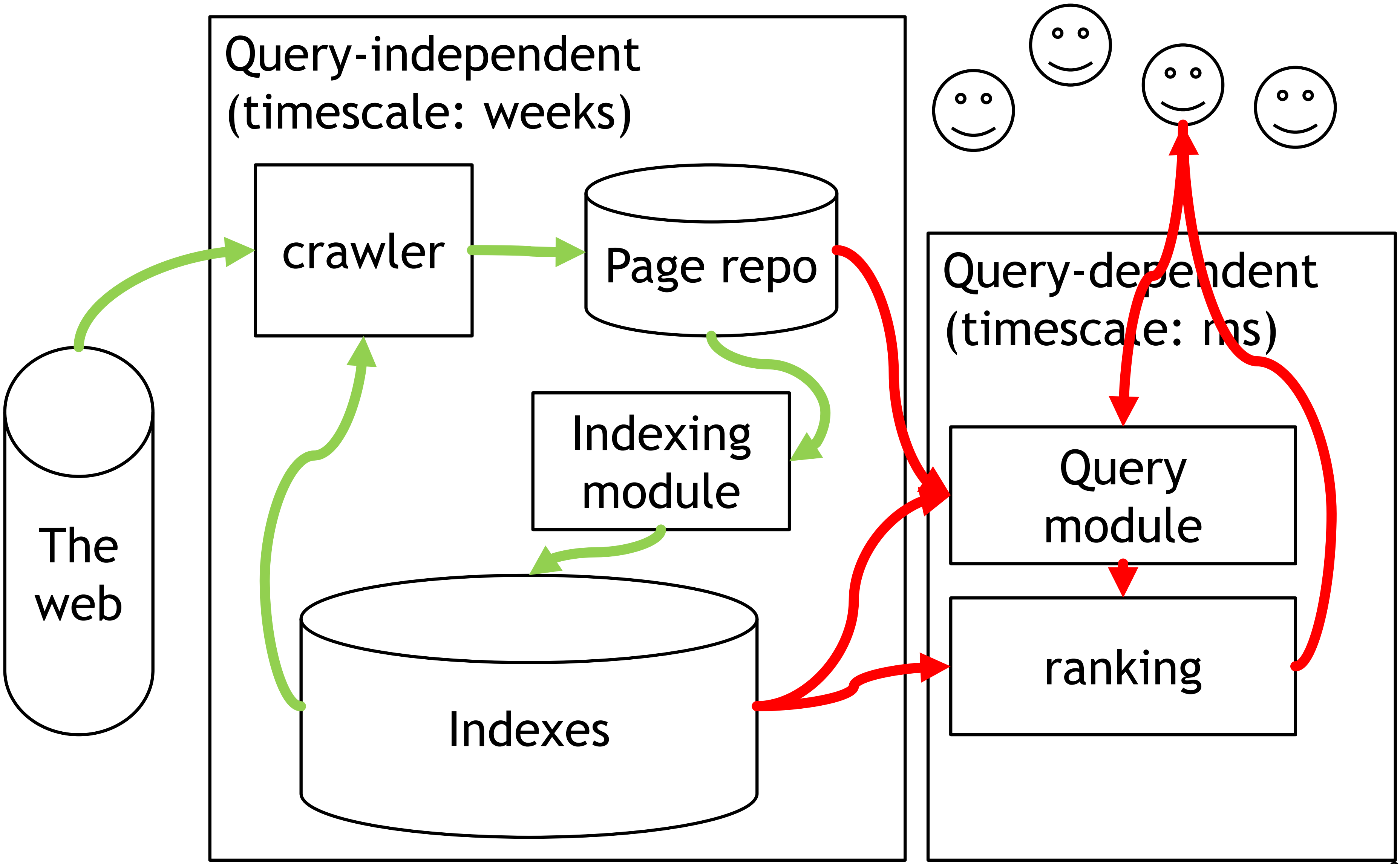
Prof. Matthias Grossglauser
School of Computer and Communication Sciences

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Overview

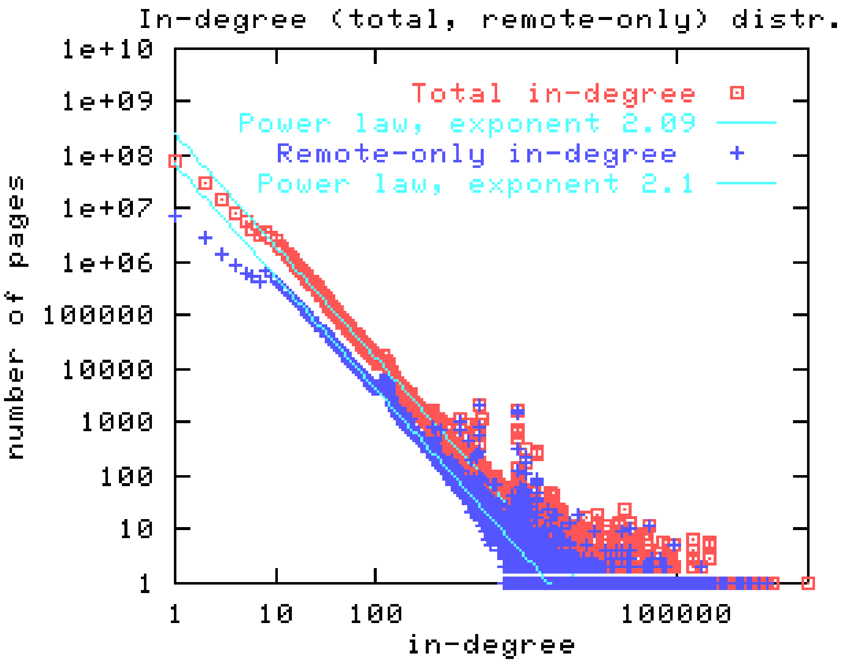
- Web search: result should be…
  - …relevant to the query
  - …of high quality/correctness/importance
- Importance: use network structure – hyperlinks
  - A link is a vote for the target of the link
- PageRank:
  - Graph eigenvector problem
  - Heuristic turning graph structure into a score
- Power method for efficient computation
- HITS: hubs and authorities variant
- Implementation and search-engine optimization

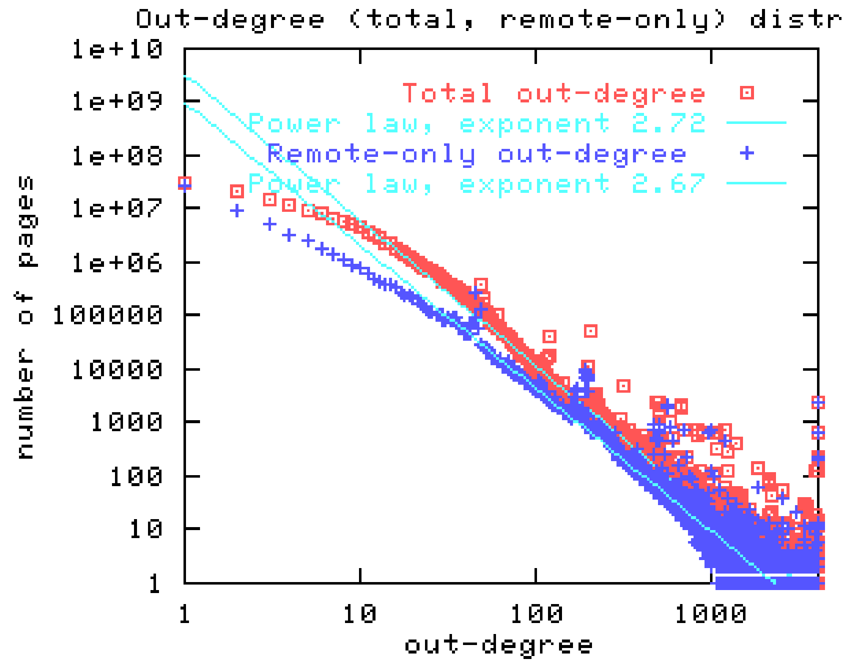
Architecture of a web search engine



In/out-degree on the web

- Link "physically resides" at the tail → constraint on out-degree, not on in-degree



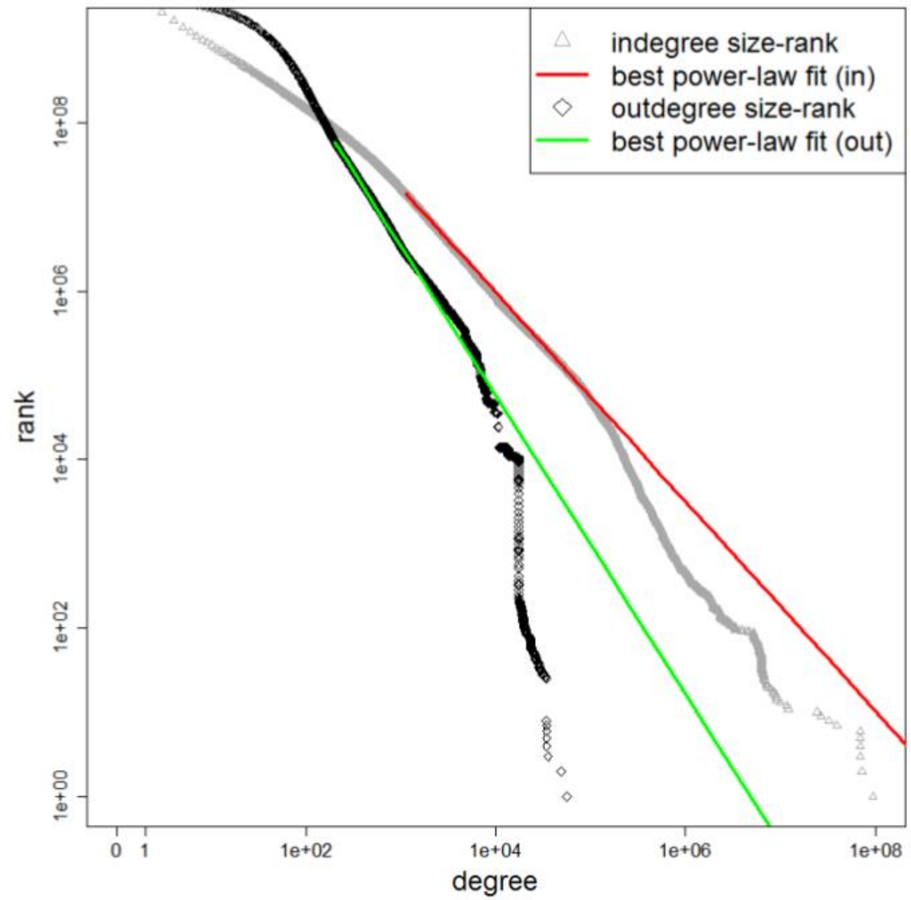


- In-degree more skewed (γ_{in} ~ 2.1 vs γ_{out}~2.7)

[Graph Structure in the Web, A. Broder, R. Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins, J. Wiener, WWW9, 2000]

In/out-degree on the web

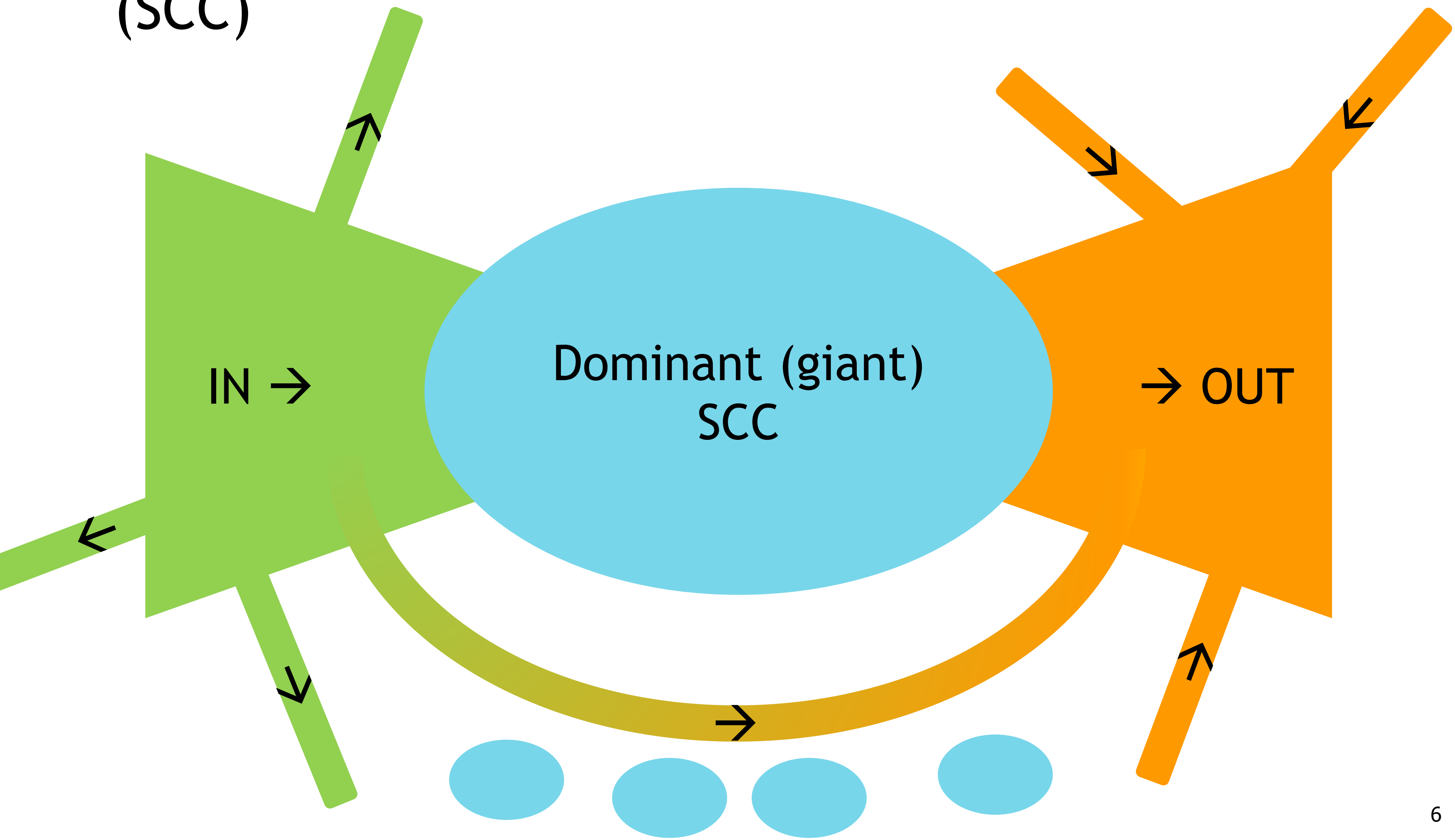
- More recent study (2015):



[The Graph Structure in the Web – Analyzed on Different Aggregation Levels, R. Meusel, S. Vigna, O. Lehmberg, and Ch. Bizer, J. Web Science, 2015, 1: 33–47]

Structure of the web

- Classification of strongly connected components (SCC)



Search → ranking

- Search query → ranked list of results
- Two ingredients:
  - Relevance score: how relevant is the result to the query (cf retrieval lectures)
  - Importance score: quality, importance of the result independent of query
- This lecture: importance score
- Key idea: importance ranking from hyperlinks

The Anatomy of a Large-Scale Hypertextual Web Search Engine

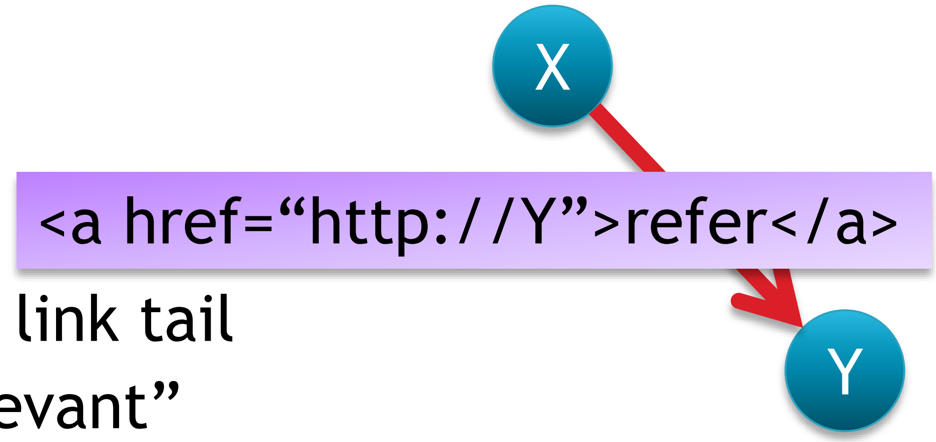
Sergey Brin and Lawrence Page

Computer Science Department,
Stanford University, Stanford, CA 94305, USA
sergey@cs.stanford.edu and page@cs.stanford.edu

Abstract

In this paper, we present Google, a prototype of a large-scale search engine which makes heavy use of the structure present in hypertext. Google is designed to crawl and index the Web efficiently

Hyperlink: intuition



- Links are asymmetric
  - Existence under control of link tail
    - Means “X considers Y relevant”
    - Does not necessarily mean “quality” or “agreement”
  - Represented as directed graph
- Note:
  - Very easy to extract out-links, but need to download entire web to extract in-links
  - Google “`link:`” search query

Turning hyperlink net into ranking

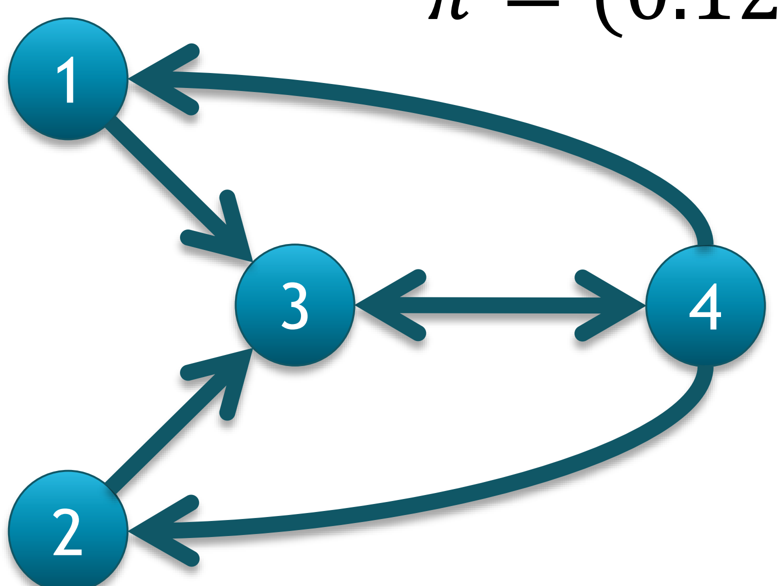
- Importance score of page u: π_u
- Approach 1: $\pi_u = i_u$ (in-degree)
  - More endorsements = more important
  - Problem: easy to spam (e.g., link-farm)

- Approach 2: take into account importance of endorser → circular
  - $\pi_u = \sum_{(v,u)} \pi_v$
  - More important endorsers = more important
  - Problem: a page pointing to a single other page should be stronger endorsement than e.g. a long list of links
- Approach 3:
  - $\pi_u = \sum_{(v,u)} \frac{\pi_v}{o_v}$

Example: basic PageRank

- Basic PageRank: $\pi_u = \sum_{(v,u)} \frac{\pi_v}{o_v}$
- Question: is there a $\{\pi\}$ that satisfies the above condition?

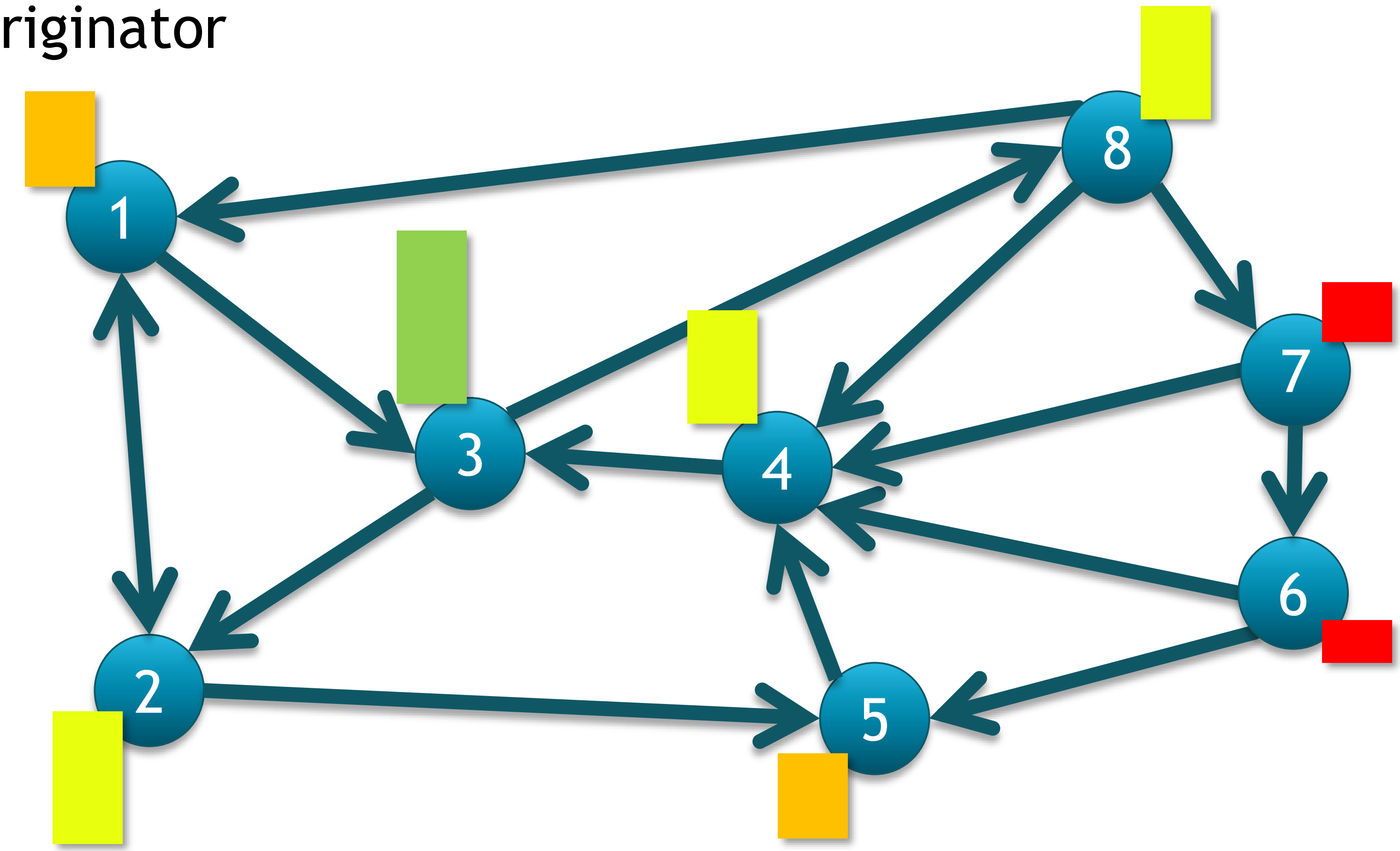
$$\pi = (0.125, 0.125, 0.375, 0.375)$$



Note: 3 and 4 have the same score, even though their in/out-degrees are different

Networked endorsements

- PageRank:
  - A hyperlink “endorses” the target
  - An endorsement depends on the “relevance” of the originator

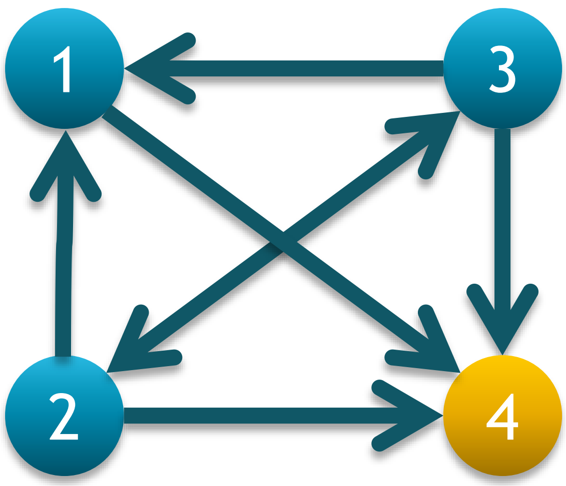


Score-flow matrix H

- Def: $H_{uv} = \begin{cases} \frac{1}{o_u} & (u, v) \in E \\ 0 & \text{otherwise} \end{cases}$
- Note: H is the transition matrix of a RW on the web
  - "random surfer": $P(\text{at } v \text{ at time } t+1) = \sum_u P(\text{at } u \text{ at time } t)/o_u$
  - $\pi_{t+1} = \pi_t \ H$
- If RW is ergodic, then $p_{ij}(t) \rightarrow \pi_j$
  - $\pi = \pi H$, i.e., solves the score-flow equation
  - Condition for ergodicity: graph has to be non-periodic and strongly connected → aperiodic and irreducible Markov chain

Problem: dangling nodes

- Dangling node = absorbing state of RW (not strongly connected)



$$H = \begin{bmatrix} 0 & 0 & 0 & 1 \\ \frac{1}{3} & 0 & \frac{1}{3} & \frac{1}{3} \\ \frac{1}{3} & \frac{1}{3} & 0 & \frac{1}{3} \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

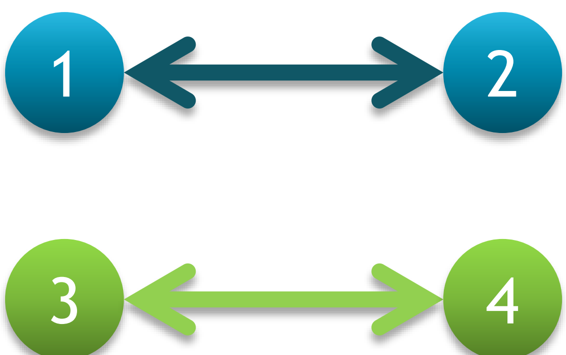
- There is no (non-zero) π that solves $\pi \quad = \pi \quad H$
- Note: setting $H_{44} = 1$ does not solve problem either → $\pi = (0,0,0,1)$

Dealing with dangling nodes

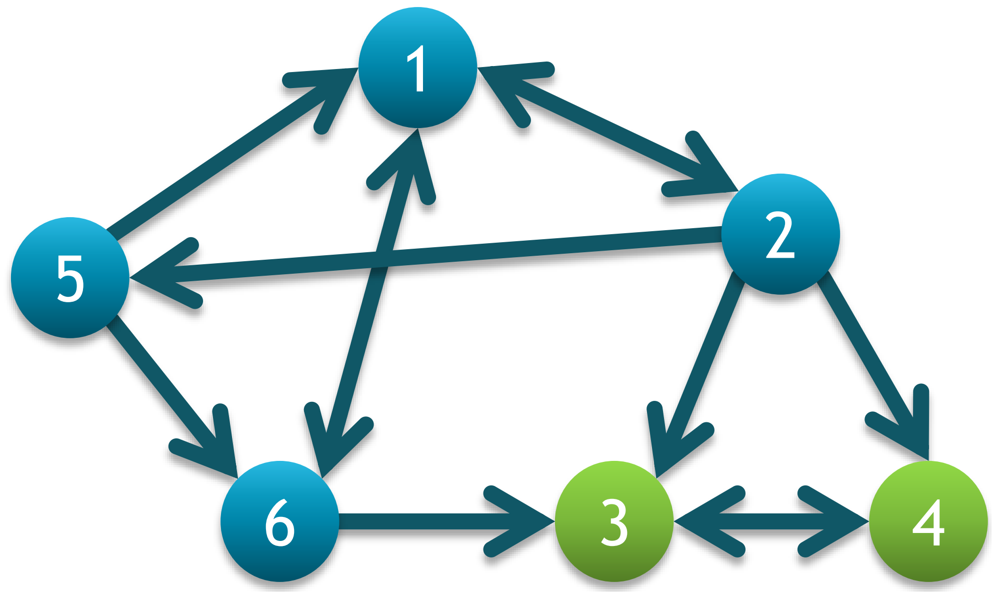
- Idea: if random surfer arrives at dangling node → go to any webpage uniformly at random
  - Or following some well-chosen distribution a over all nodes
- Def: w=indicator of dangling nodes
  - Example: $w = (0,0,0,1)$
- $\widehat{H} = H + \frac{1}{n}(w^T e)$ (stochastic matrix)
  - Example: $\widehat{H} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ \frac{1}{3} & 0 & \frac{1}{3} & \frac{1}{3} \\ \frac{1}{3} & \frac{1}{3} & 0 & \frac{1}{3} \\ 1/4 & 1/4 & 1/4 & 1/4 \end{bmatrix}$

The Google Matrix G

- Does $\widehat{H}$ define an ergodic RW = single score vector π? Not always…
- Dangling nodes = absorbing states are not the only classes we can get
- Examples:



Any $\pi = (x, x, y, y)$ is solution



3,4 not dangling, but $\{3{,}4\}$ is absorbing class

The Google Matrix G

- Solution: add randomization
  - At every iteration, coin flip: with prob. θ walk on the graph ($\widehat{H}$), with prob. $1-\theta$ jump to a random page
- $G = \theta \widehat{H} + (1-\theta)\frac{e^T e}{n}$

teleportation matrix

- Theorem:
  - If $\theta < 1$, $\pi = \pi G$ has exactly one solution for any network graph
  - $\theta = 0$ → π uniform
  - In practice: $0.8 \leq \theta \leq 0.9$, i.e., 5-10 steps on web graph between random jumps
- PageRank algorithm computes this solution

Random walk driven by Google matrix

- Irreducible:
  - Every page is directly connected to every other page
- Aperiodic:
  - $G_{ii} > 0$ (self-loops from teleportation matrix)
  - This is enough to avoid periodic patterns
- Irreducible + aperiodic = ergodic:
  - Single stationary distribution π
  - Long-term page frequency of random surfer

Generalization: non-uniform jumps

- Uniform jumps: crude
  - We can incorporate more information about the a-priori importance of web pages
    - Length of the URL
    - Words in the domain
    - Language
    - HTML tags
    - …
- Model: when randomizing, sample from a = distribution over all nodes
- $G = \theta H + (\theta w^T + (1 - \theta) e^T) a$

Computing scores

- Approach 1: simulate random walker
  - Stationary regime: $P(\text{walker at } u) = \pi_u$
  - Problem: with Θ(100bn) web pages: slow convergence, very costly
- Approach 2: linear-system method
  - Compute solution of $x(I - \theta H) \quad = a$
  - Normalized rank: $\pi = x/(xe^T)$
  - Efficient for small graphs
- Approach 3: power method
  - π is (left) dominant eigenvector (eigenvalue=1) of G
  - Iterating $\pi_{t+1} = \dfrac{\pi_t \ G}{\|\pi_t \ G\|}$

Approach 2: linear system equivalence

- Theorem: approach 2 produces PageRank vector
- Proof:
  - PageRank vector π: $\pi G = \pi$ and $\pi e^T = 1$
  - Want to show that $\pi(I - G) = 0 \Leftrightarrow x(I - G) = 0$
  - $x(I - G) = x(I - \theta H - \theta w^T a - (1 - \theta)e^T a) =$
  - $\quad = x(I - \theta H) - x(\theta w^T + (1 - \theta)e^T)a =$
  - $\quad = a - a = 0$
  - Last step used:
  - $1 = ae^T = x(I - \theta H)e^T =$
  - $\quad = xe^T - \theta xHe^T =$
  - $\quad = xe^T - \theta x(e - w)^T =$
  - $\quad = (1 - \theta)xe^T + \theta xw^T$

Convergence of power method

- Power method: obtaining dominant eigenvalue+eigenvector
- Why it works:
  - Assume G has n distinct eigenvalues $\lambda_1 = 1 > \lambda_2 > \cdots > \lambda_n$
  - The eigenvectors $(v_1 = \pi, v_2, \ldots, v_n)$ are orthogonal, form a basis
  - Write π_0 in this basis: $\pi_0 = \pi + \sum_2^n \alpha_i v_i$
  - $\pi_1 = (\pi + \sum_2^n \alpha_i v_i) \; G = \pi \; + \sum_{i=2}^n \alpha_i \lambda_i v_i$
  - $\pi_2 = (\pi + \sum_2^n \alpha_i \lambda_i v_i) \; G = \pi \; + \sum_{i=2}^n \alpha_i \lambda_i^2 v_i$
  - …
  - $\lambda_2 < 1 \rightarrow \pi_t \rightarrow \pi$
  - Can be generalized to non-distinct EVs

Speed of convergence of power method

- How many iterations are needed until PageRank score is close enough?
- Theorem:
  - If spectrum of $\widehat{H}$ is $\sigma(1, \lambda_2, \dots, \lambda_n)$, then spectrum of G is $\sigma(1, \theta\lambda_2, \dots, \theta\lambda_n)$
  - So convergence is at least $\propto \theta^k$
- Intuition:
  - We overlay over the real directed hyperlink graph a complete graph (with lower weight $1 - \theta$)
  - This ensures good conductance/good mixing/fast convergence of the power method
- In practice, 50-100 iterations are sufficient

Implementation of power method

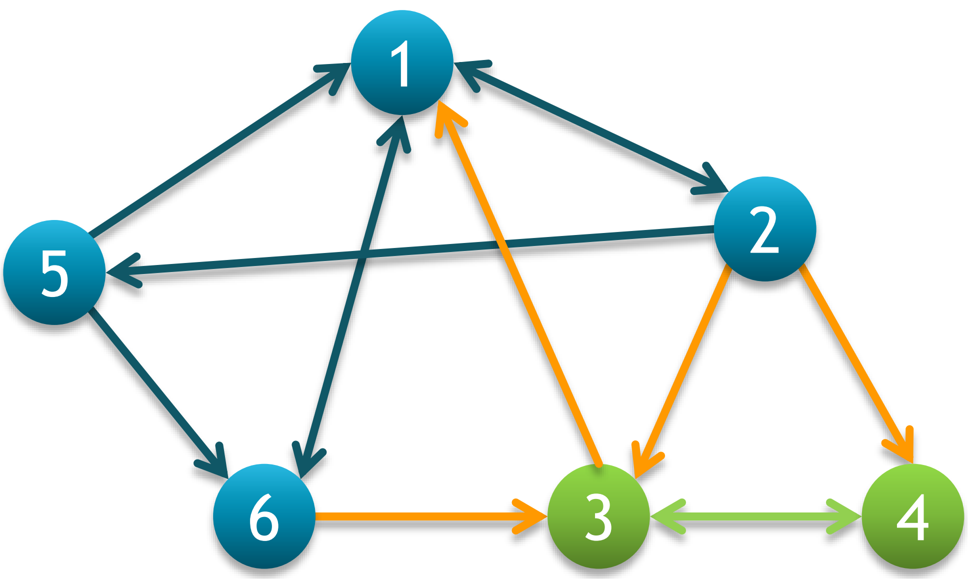
- H: has $O(10^{20})$ elements, but very sparse
- w: list of dangling nodes, probably a few bn
- π: dense, $O(10^{10})$, updated during PageRank
- a: dense, $O(10^{10})$, obtained while crawling, const.
- $e^T a$: teleportation matrix not computed&stored explicitly

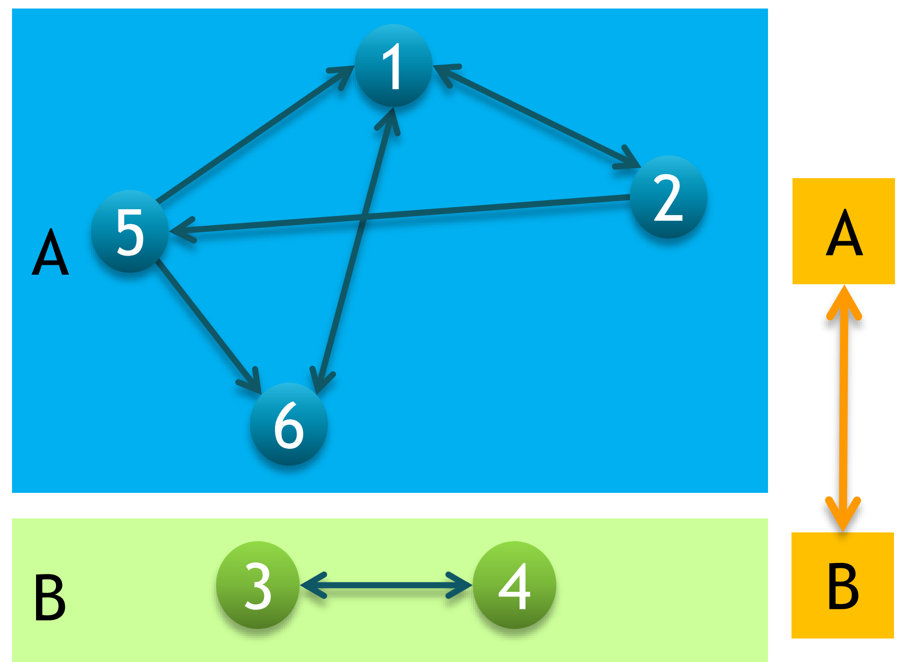
Computational optimizations, "tricks"

- Challenging scale:
  - 10s of bn of webpages, 100s of bn of links
- Large, but sparse matrix:
  - Sparse (adjacency) representation
- Ranking vs score:
  - Exact scores not needed, only rank order → stop early
- Node-specific convergence:
  - Most nodes converge fast → lock-in, iterate only rest
- Dangling nodes:
  - Remove or collapse
- Aggregate related pages:
  - Cluster related, hierarchical computation

Aggregate approximation

- Hierarchical decomposition of web graph (cf community detection)
- Conceptually: run random-surfing at each level



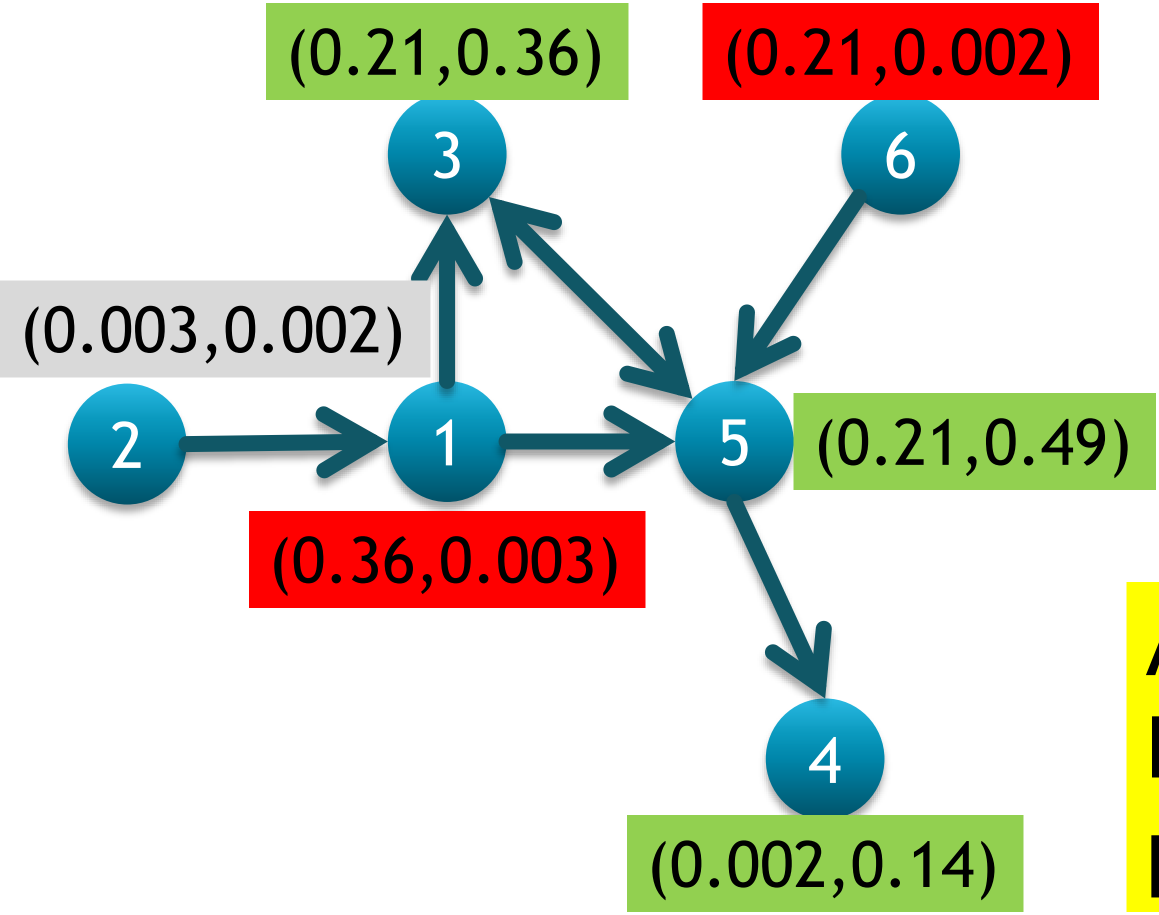


$$\pi(3) = \pi_B(3) \times \pi(B)$$

HITS algorithm

- PageRank:
  - Basic idea: An important page is pointed to by many other important pages
- HITS:
  - "Hypertext Induced Topic Search"
  - There are two importance scores for each node: hub and authority
  - Authority: contains important primary information
  - Hub: Points to a lot of primary information (directory)
  - Basic idea:
    - A hub points to many important authorities
    - An authority is pointed to by many important hubs

PageRank vs HITS



Auth: $X_{k+1} \propto A^T Y_k$
Hub: $Y_{k+1} \propto A X_{k+1}$
plus randomization

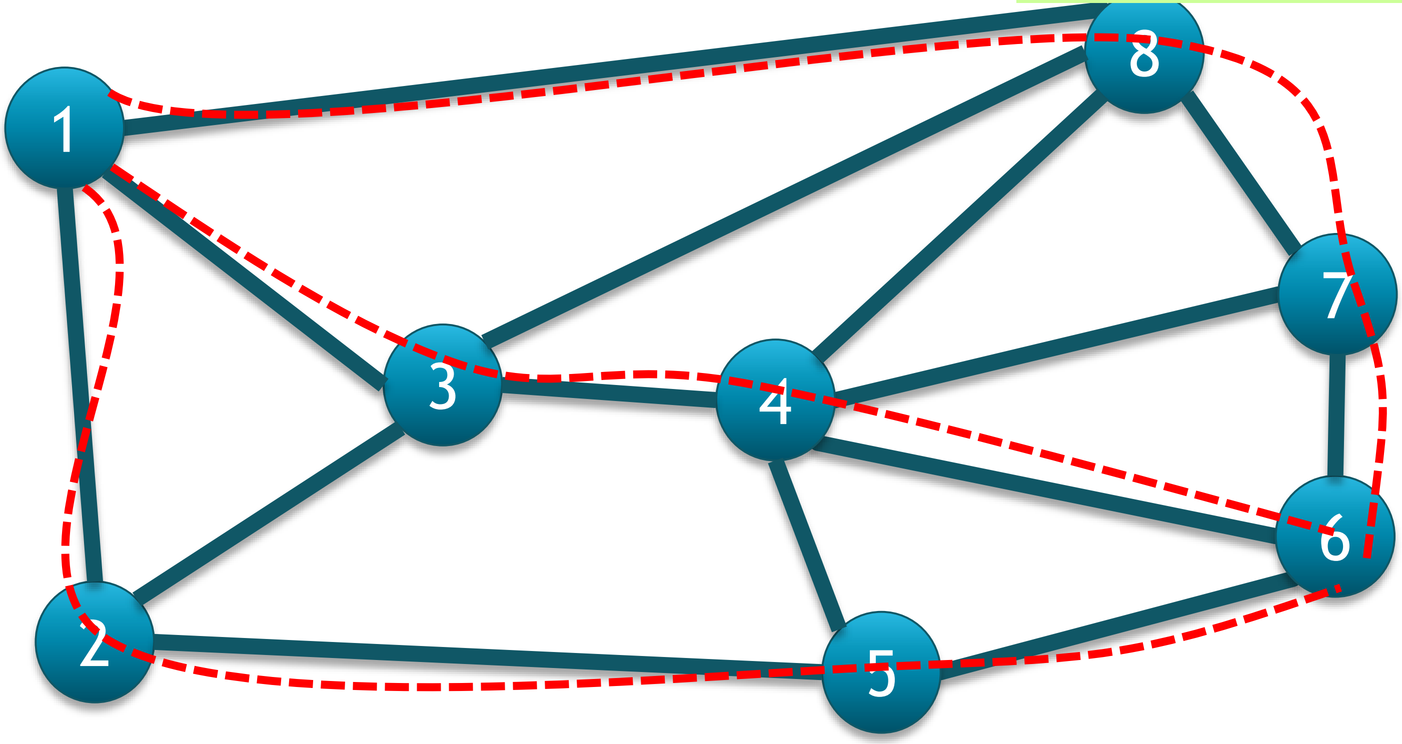
score=(hub,authority)

Other centrality measures

- Betweenness centrality:
  - $C_B(u) = \sum_{v,w \neq u} \frac{\sigma_{vw}(u)}{\sigma_{vw}}$

\# shortest paths between v, w going through u

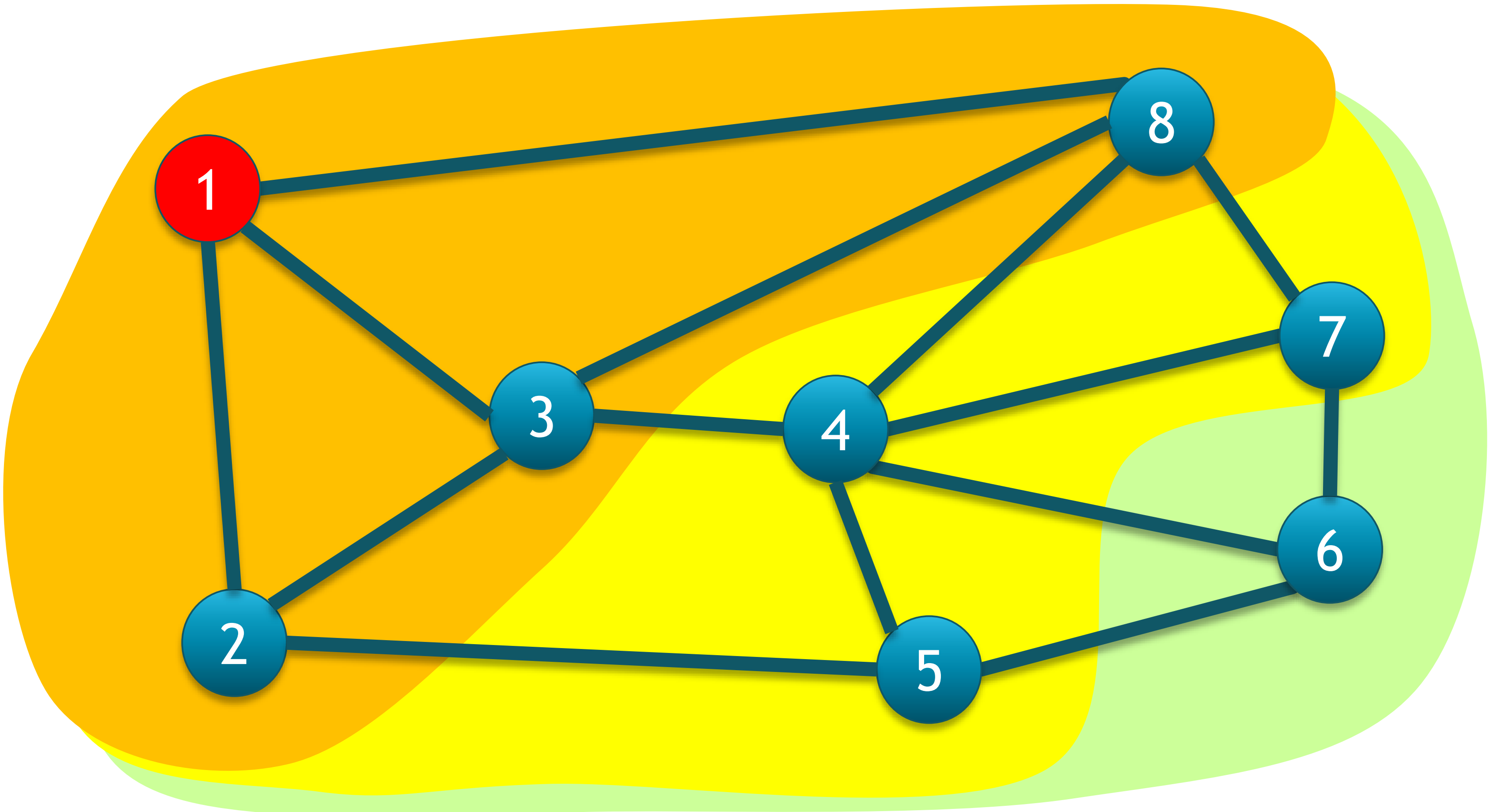
\# shortest paths between v, w



Other centrality measures

- Closeness centrality:
  - $C_C(u) = \sum_{v \neq u} \frac{1}{d(u,v)}$

Inverse of distance to other node v



SEO: Search Engine Optimization

- Cottage industry helping to increase rankings for a fee
  - Early search engines: term spamming and hiding (e.g., including terms that are invisible to user, but picked up by search engine)
  - Cloaking: sending different content to crawlers and users
- Link manipulation to raise PageRank score:
  - Trading links (I point to you if you point to me)
  - Link farms
- Google Dance: monthly crawl + fiddling with parameters by Google

Summary

- Search engines:
  - Big business (advertisement)
  - Highly specialized datacenters and methods, details are trade secrets
- PageRank:
  - Basic idea: interpret links as expressions of trust or endorsement
  - Turn into an importance score
  - Beautiful connections to random walk theory, spectral graph theory
- Related ideas can be applied to many other contexts
  - E.g., impact of scientific publications; importance of patents; social capital in social networks;…

References

- [M. Chiang, Networked Life, Cambridge, 2012 (chapter 3)]
- [A. N. Langville, C. D. Meyer, Google's PageRank and Beyond – The Science of Search Engine Rankings, Princeton U Press, 2006]
- [D. Easley, J. Kleinberg: Networks, Crowds, and Markets, Cambridge 2010 (chapter 14)]