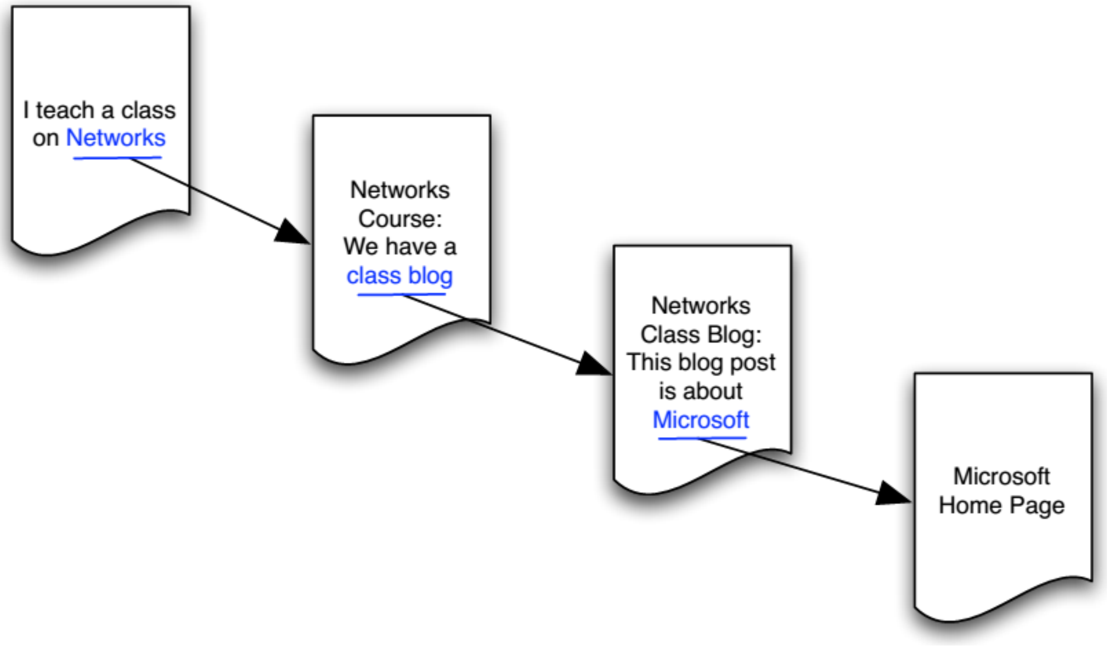
Link Analysis

Alfan F. Wicaksono
Fakultas Ilmu Komputer, Universitas Indonesia

Hypertext

- There is a crucial design principle embedded in the Web
- This is what turns the **set of Web pages** into the **“web” of Web pages**.



The idea of using Network Metaphor is due to the concept of **Hypertext**, in **which any portion of the text can link directly to any other part.**

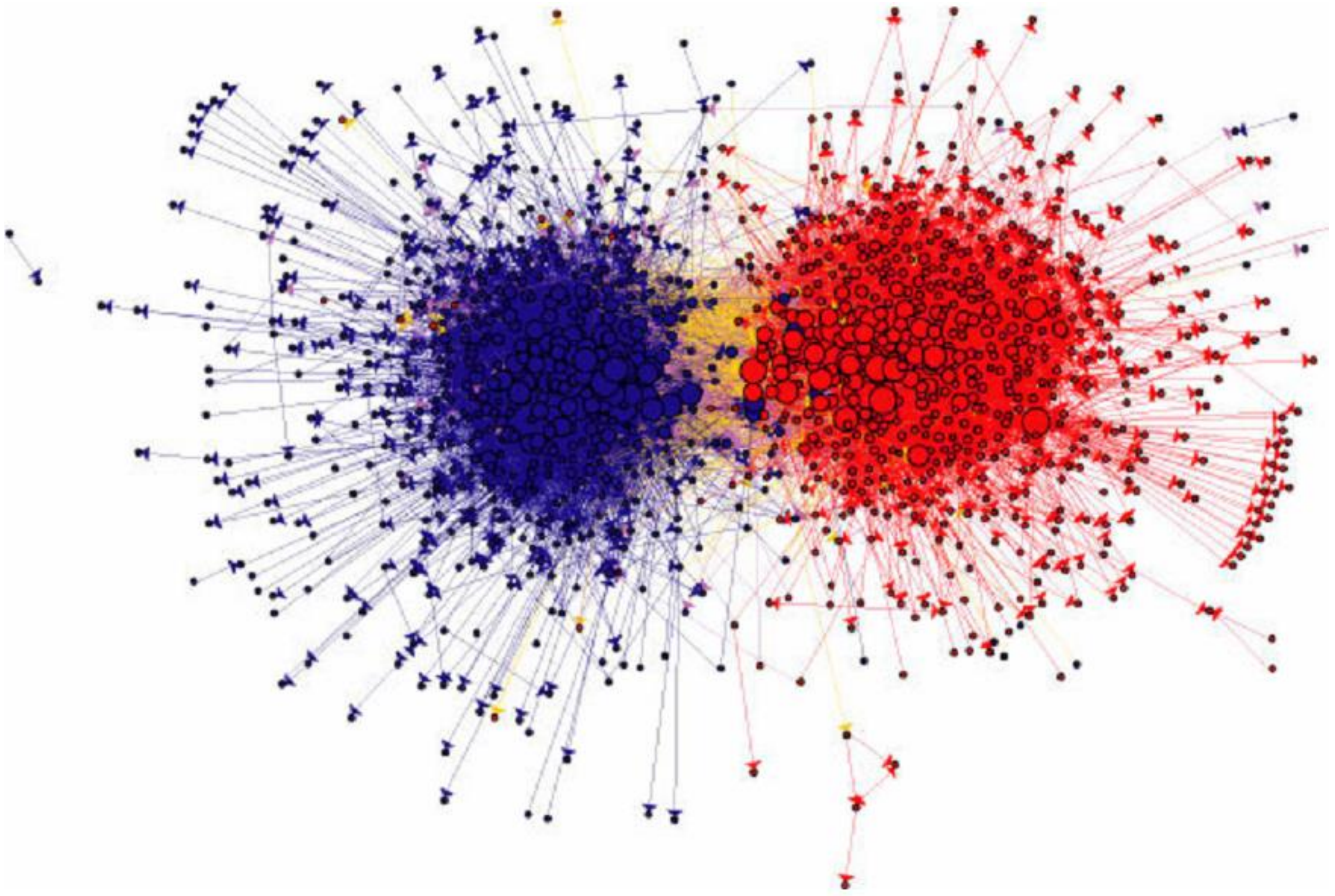
Information on the Web is organized using a **network metaphor**: The links among Web pages turn the Web into a directed graph.

Information Networks

A network of links among political blogs before 2004 U.S. elections.

- Type of network, in which **the basic units being connected are pieces of information**, and links join pieces of information that are related to each other in some fashion.

- Links among Web pages, for example, can help us to understand how these pages are related, how they are grouped into different communities, and which pages are the most prominent or important.



Authenticity & Authority

- Authenticity
  - Halaman web spam **vs** non-spam
  - Halaman web asli **vs** palsu (penipuan)

- Authority
  - Apakah situs terkait informasi tertentu "resmi"?
  - Query: "universitas indonesia"
    - Web www.ui.ac.id **lebih "resmi"** dibandingkan halaman Wikipedia UI https://id.wikipedia.org/wiki/Universitas_Indonesia

Apakah cukup Score(Q,D) hanya melihat content?

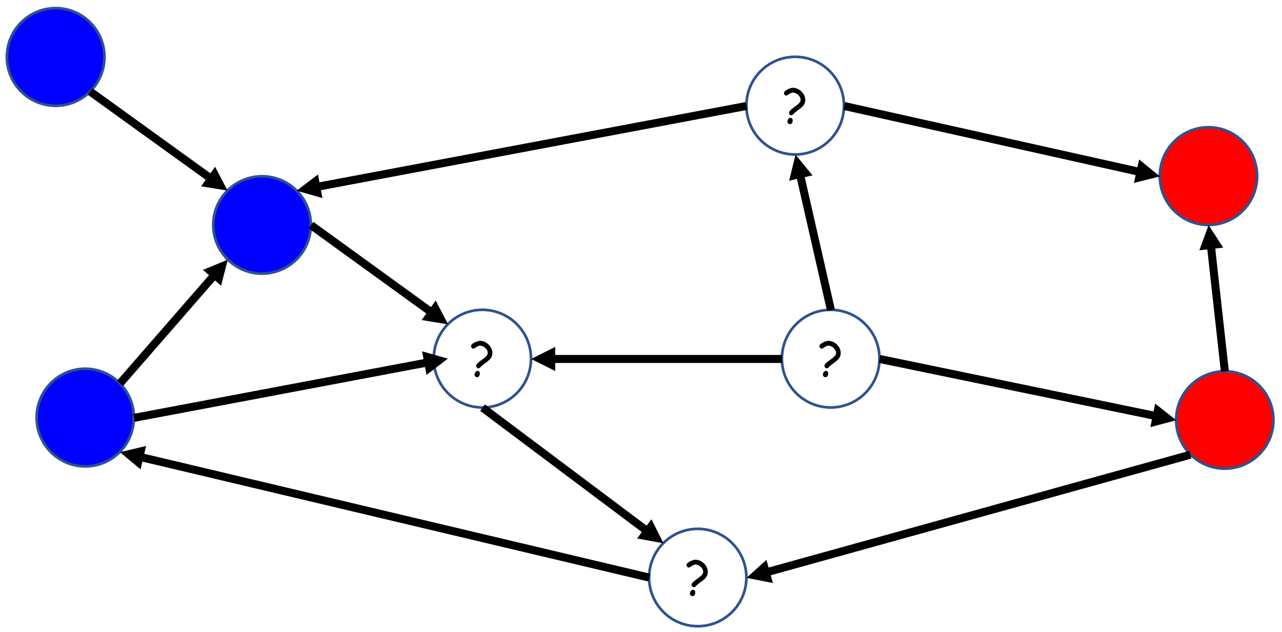
- Content-based Ranking is not Sufficient

- Most useful webpage don't have the keyword
  - Query: ``Harvard"
  - 49 "Harvard" in www.harvard.edu
  - 357 "Harvard" in http://en.wikipedia.org/wiki/Harvard_University

- Pages are not sufficiently descriptive
  - "automobile manufacturers" in Honda or Toyota

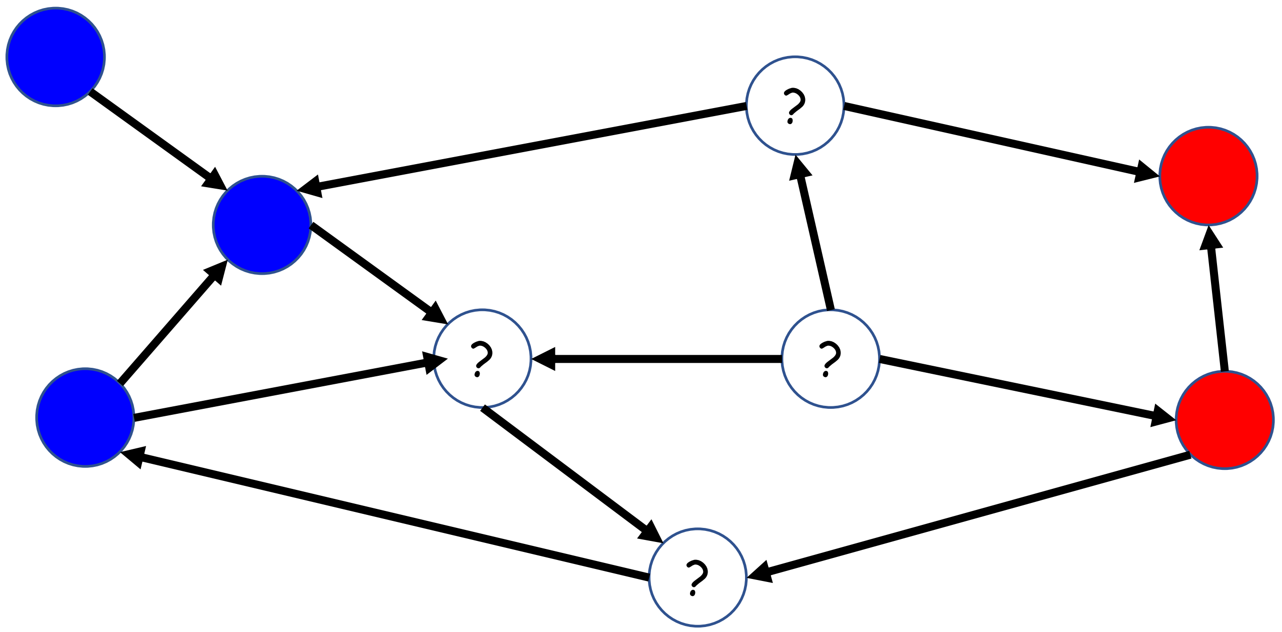
Links: sources of authenticity & authority

- **Biru**: good web pages
- **Merah**: bad web pages
- Coba tebak apakah yang "**unknown ?**" baik atau buruk?



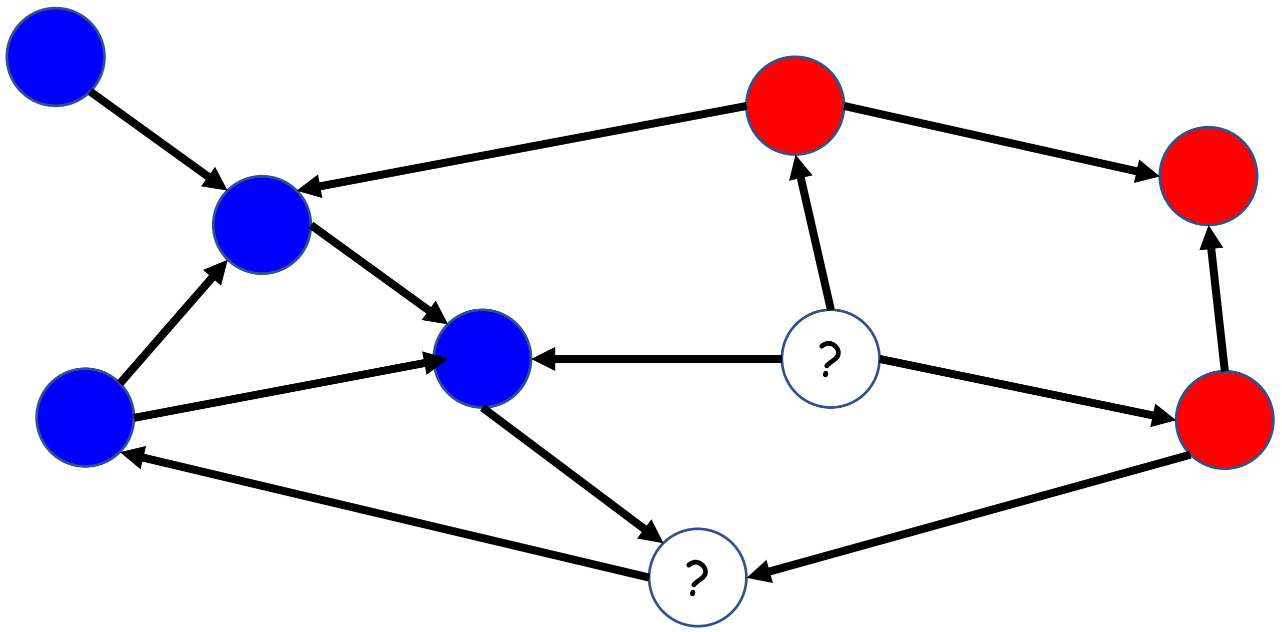
Links: sources of authenticity & authority

- "Jika Anda *endorse* (menunjuk) sesuatu yang **buruk**, Anda juga **buruk**."
- "Jika sesuatu yang **baik** *endorse* Anda, Anda orang **baik**."



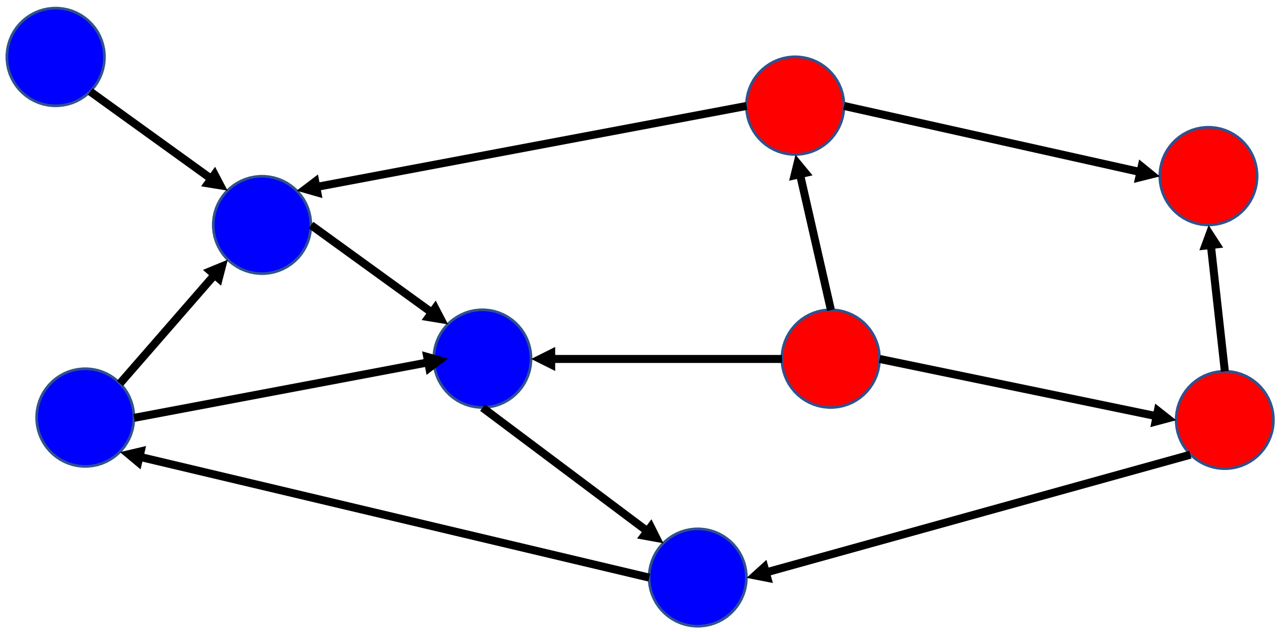
Links: sources of authenticity & authority

- **Biru**: good web pages
- **Merah**: bad web pages
- Coba tebak apakah yang "**unknown ?**" baik atau buruk?



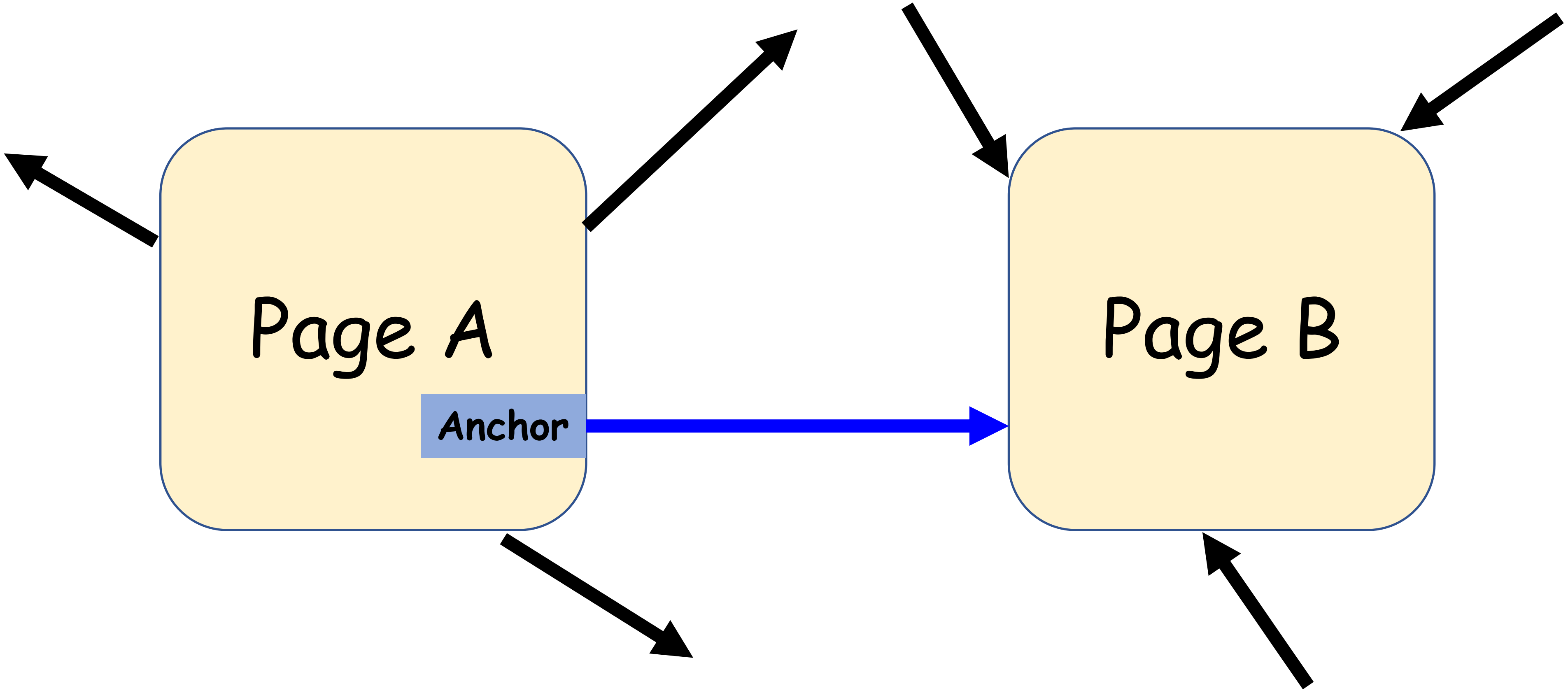
Links: sources of authenticity & authority

- **Biru**: good web pages
- **Merah**: bad web pages
- Coba tebak apakah yang **"unknown ?"** baik atau buruk?



Hipotesis

A hyperlink between pages denotes a conferral of authority (quality signal).



PageRank

- Based on Sergey Brin and Larry Page's academic paper …
- You know them well …

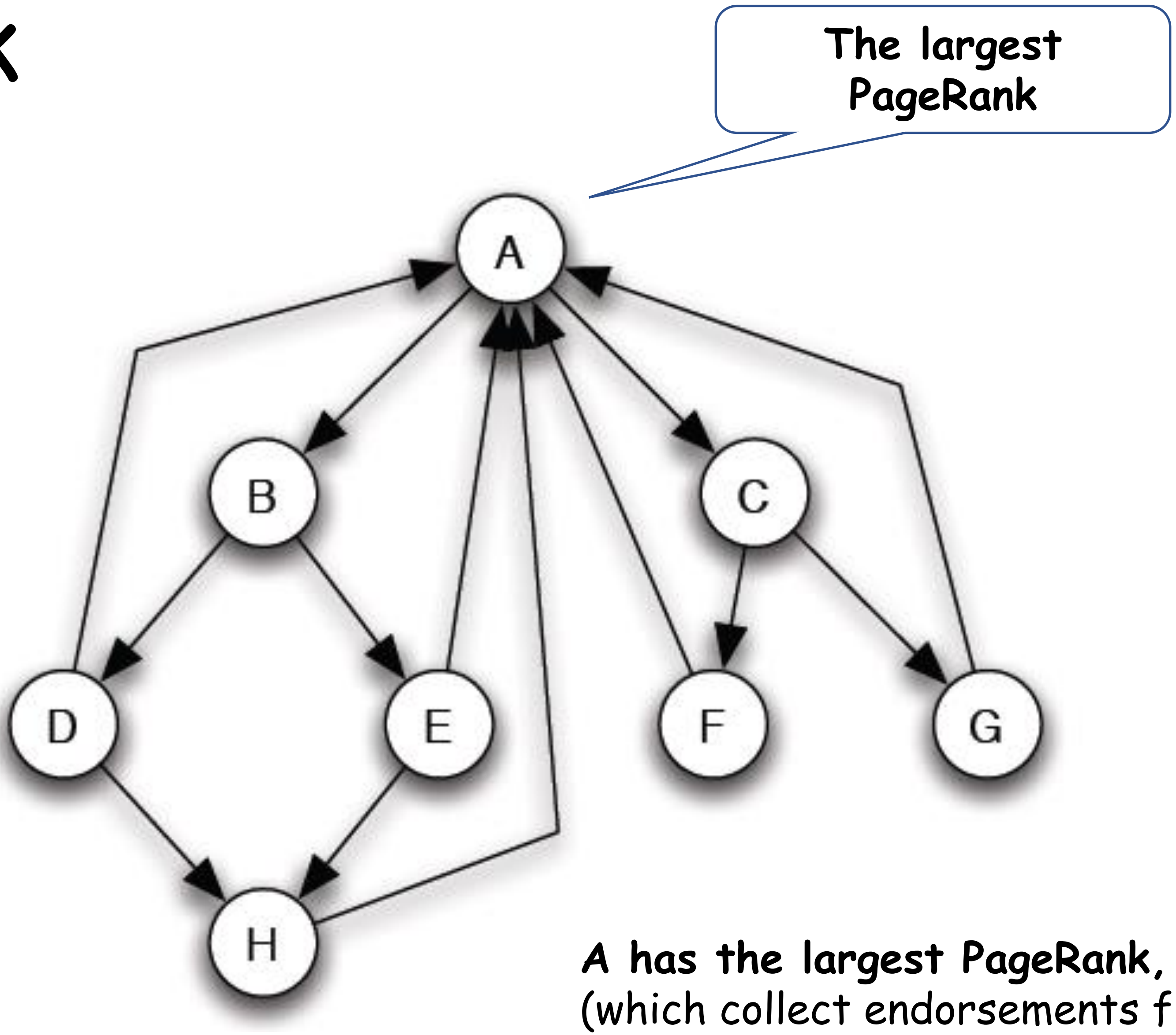
- Larry Page (~Rank)
  - BS in CE from UMich, MS from Stanford

- Sergey Brin
  - BS in Math&CS from UMD, MS from Stanford

PageRank

- In other settings on the Web, **endorsement is best viewed as passing directly from one prominent page to another**.
- This is often dominant mode of endorsement, for example,
  - Among academic papers
  - Among bloggers
  - Among personal pages
- This mode of endorsement is the basis for PageRank to compute the importance of a node!

PageRank



A has the largest PageRank, followed by B and C
(which collect endorsements from A).

PageRank

- Intuitively, we can think of PageRank as a kind of **"fluid"** that circulates through the network ...

- passing from node to node across edges, and pooling at the nodes that are the most important.

PageRank

Notice that the total PageRank in the network will remain constant as we apply these steps: since each page takes its PageRank, divides it up, and passes it along links !

Specifically, **Basic PageRank** is computed as follows:

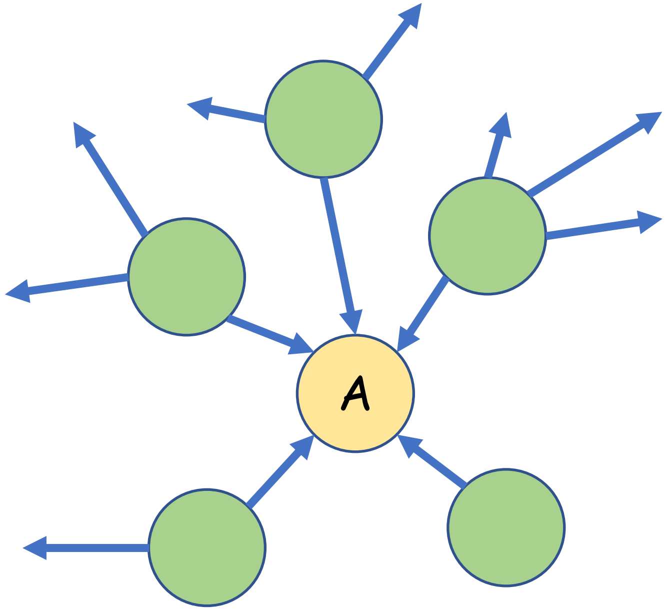
(1) In a network with **n nodes**, we assign all nodes the same initial PageRank, set to be **1/n**.

(2) We choose a number of **steps k**.

(3) We then perform **a sequence of k updates** to the PageRank values:

- **Basic PageRank Update Rule**: **Each page divides its current PageRank equally** across its out-going links, and passes these equal shares to the pages it points to. Each page updates its new PageRank to be the sum of the shares it receives.

Update nilai pagerank node A:



T_1, ..., T_n adalah semua tetangga yang menunjuk ke A

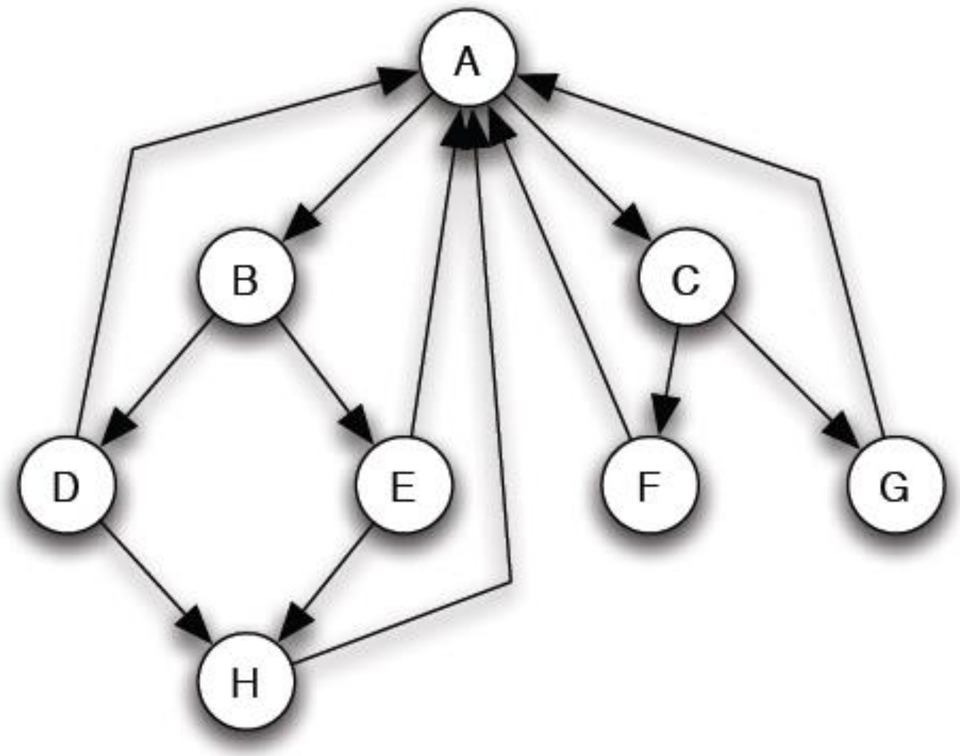
total semua rank harus 1, nanti

PageRank

$$PR(A) = \frac{PR(T_1)}{C(T_1)} + \frac{PR(T_2)}{C(T_2)} + \cdots + \frac{PR(T_n)}{C(T_n)}$$

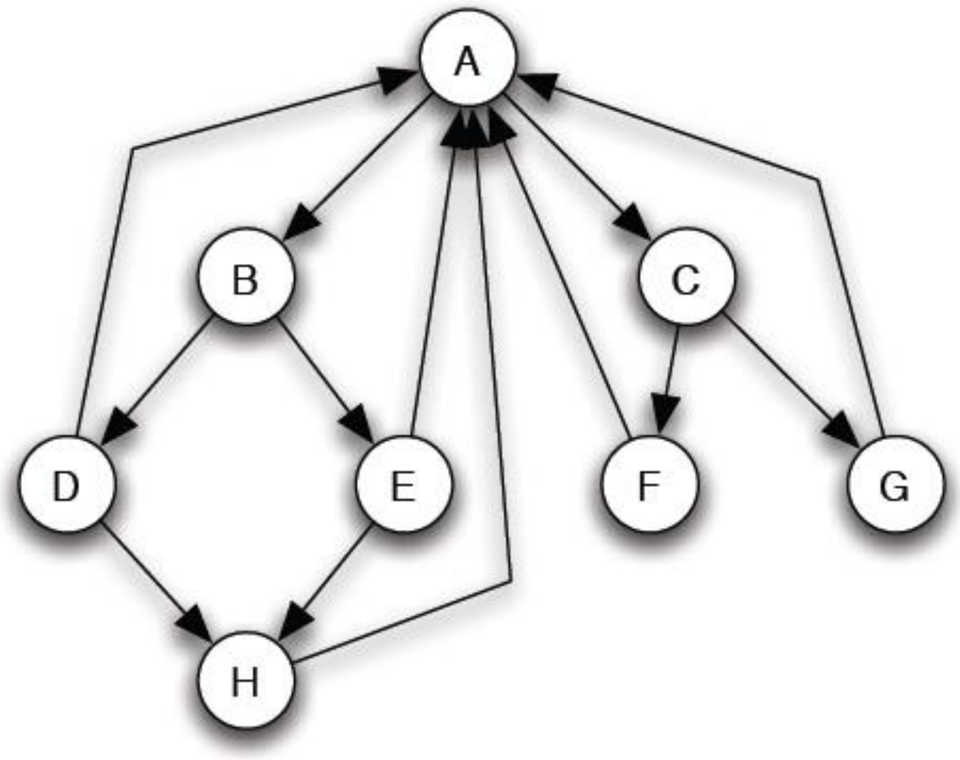
Banyaknya out-links dari tetangga

PageRank



Step	A	B	C	D	E	F	G	H
0	1/8	1/8	1/8	1/8	1/8	1/8	1/8	1/8

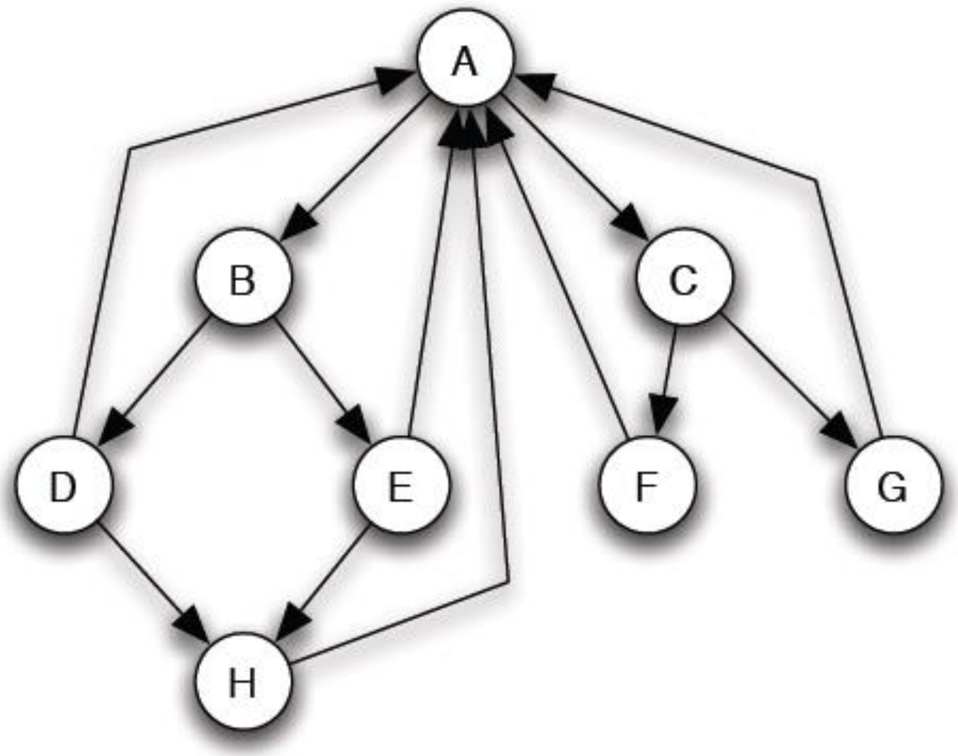
PageRank



We update from A to H, consecutively!

Step	A	B	C	D	E	F	G	H
0	1/8	1/8	1/8	1/8	1/8	1/8	1/8	1/8
1	1/2	1/16	1/16	1/16	1/16	1/16	1/16	1/8

PageRank

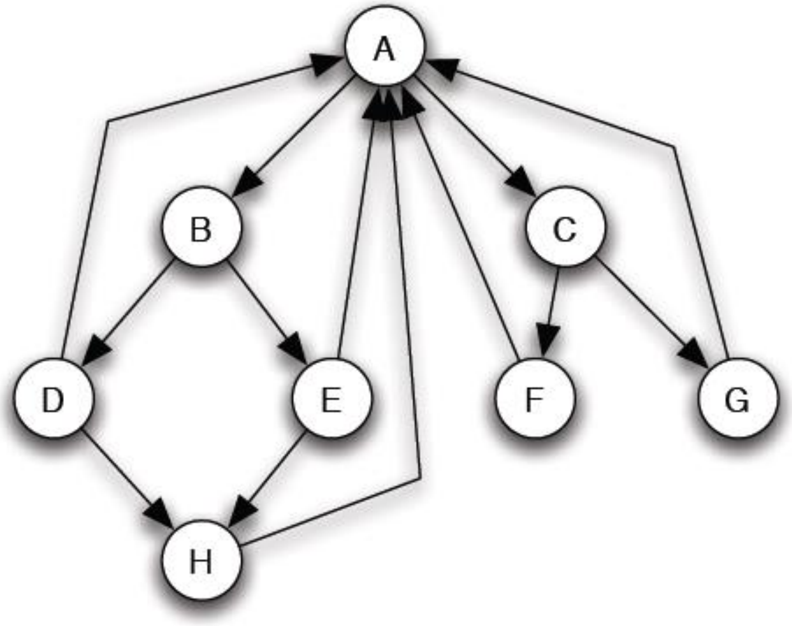


We update from A to H, consecutively!

Step	A	B	C	D	E	F	G	H
0	1/8	1/8	1/8	1/8	1/8	1/8	1/8	1/8
1	1/2	1/16	1/16	1/16	1/16	1/16	1/16	1/8
2	5/16	1/4	1/4	1/32	1/32	1/32	1/32	1/16

PageRank

Cara lain, dengan **Power Iteration**, yang melibatkan **Transition Matrix P**:

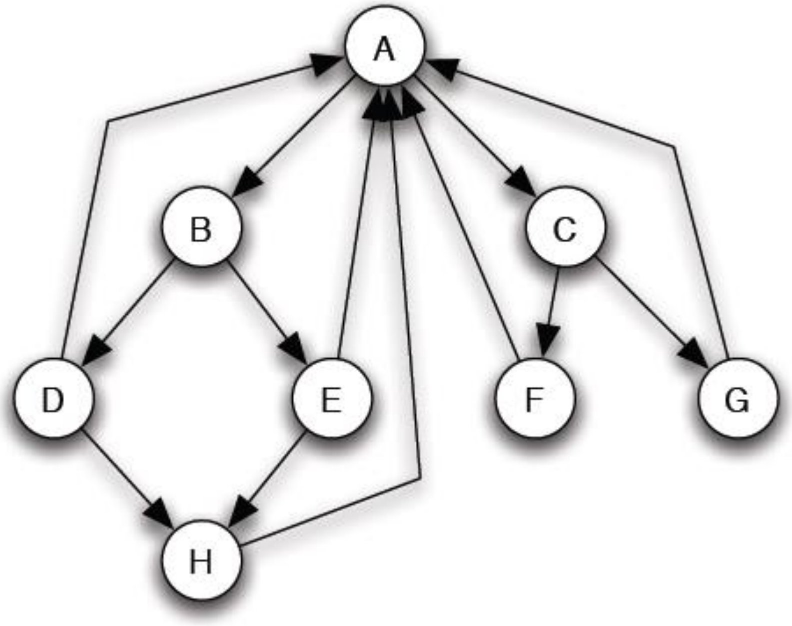



```
import numpy as np

P = np.array([[0, 1/2, 1/2, 0, 0, 0, 0, 0],
              [0, 0, 0, 1/2, 1/2, 0, 0, 0],
              [0, 0, 0, 0, 0, 1/2, 1/2, 0],
              [1/2, 0, 0, 0, 0, 0, 0, 1/2],
              [1/2, 0, 0, 0, 0, 0, 0, 1/2],
              [1, 0, 0, 0, 0, 0, 0, 0],
              [1, 0, 0, 0, 0, 0, 0, 0],
              [1, 0, 0, 0, 0, 0, 0, 0]])


# initial pagerank untuk semua nodes
x0 = np.array([[1/8, 1/8, 1/8, 1/8, 1/8, 1/8, 1/8, 1/8]])
```

PageRank



Cara lain, dengan **Power Iteration**, yang melibatkan **Transition Matrix P**:

```
>>> x0 @ P
array([[0.5   , 0.0625, 0.0625, 0.0625,
                     0.0625, 0.0625, 0.0625, 0.125 ]])

>>> x0 @ P @ P
array([[0.3125 , 0.25   , 0.25   , 0.03125,
                     0.03125, 0.03125, 0.03125, 0.0625 ]])

>>> x0 @ P @ P @ P
array([[0.15625, 0.15625, 0.15625, 0.125,
                     0.125   , 0.125   , 0.125 , 0.03125]])

>>> x0 @ P @ P @ P @ P @ P @ P @ P @ P @ P
array([[0.23632812, 0.18554688, 0.18554688, 0.0859375,
0.0859375 , 0.0859375 , 0.0859375 , 0.04882812]])
```

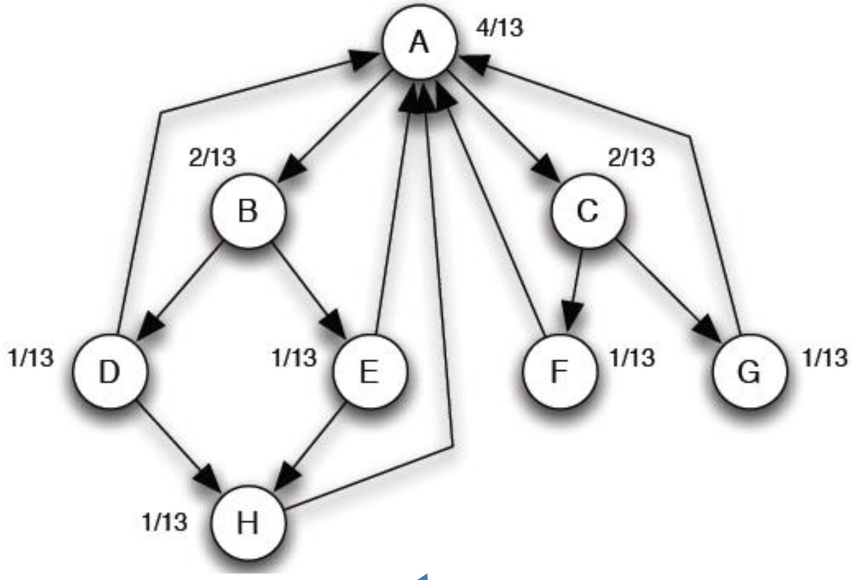
PageRank

Dengan syarat Transition Matrix-nya merefleksikan **Ergodic Markov Chain**

Equilibrium Values of PageRank (Steady-State Prob)

- One can prove that the PageRank **values of all nodes converge to limiting values as the number of update steps k goes to infinity**.

- The **limiting PageRank values** exhibit kind of ***equilibrium***: if we take the limiting PageRank values and apply one step of the Basic PageRank Update Rule, then the values at every node remain the same.



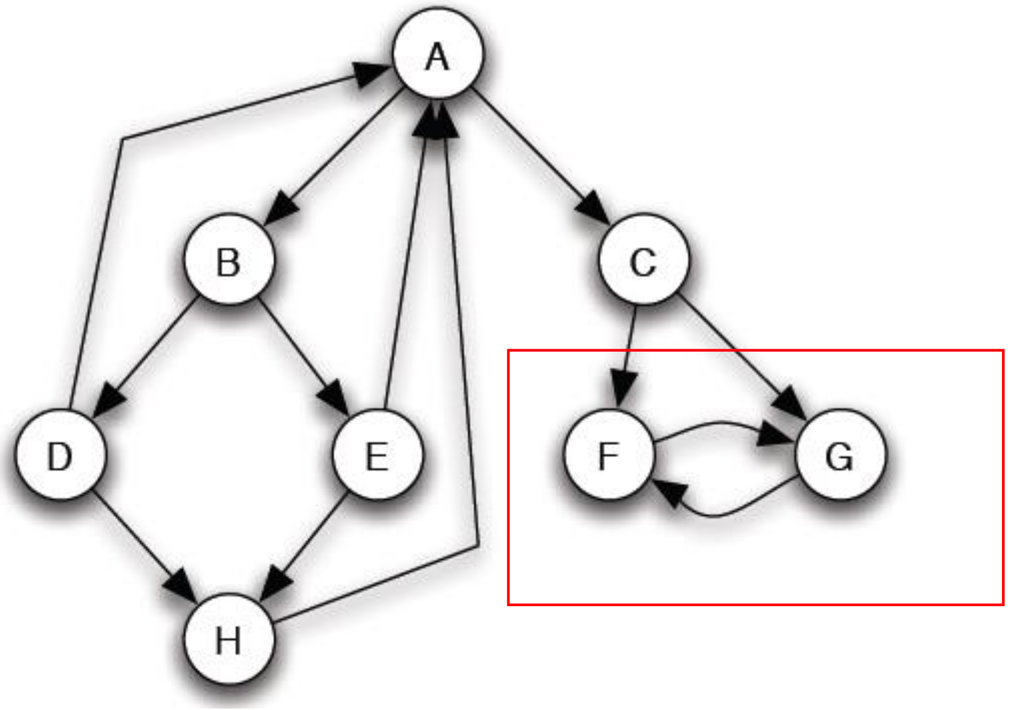
Example of Equilibrium PageRank values

Ergodic Markov Chain

A Markov chain is called an *ergodic chain* if it is possible to go from every state to every state (not necessarily in one move).

Jika tidak Ergodic ???

The Problem with **Basic PageRank Algorithm**



PageRank that flows from C to F and G can never circulate back into the rest of the network!

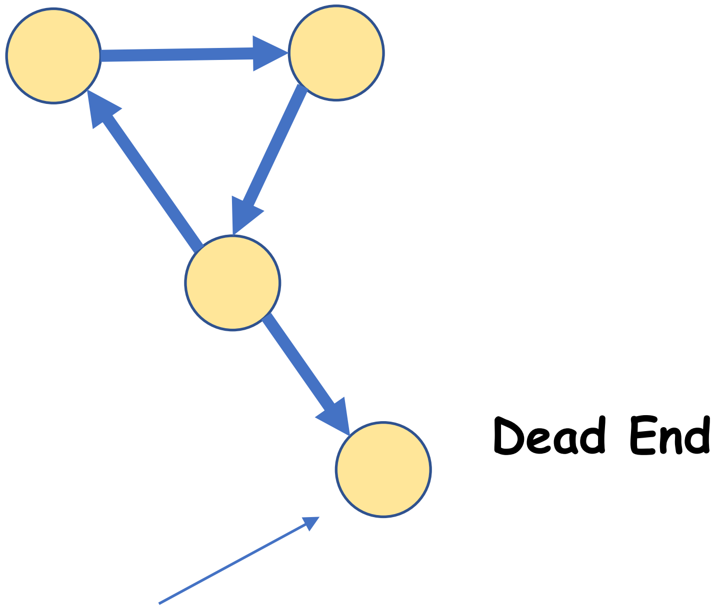
so the links out of C function as a kind of **"slow leak"** that eventually causes all the PageRank to end up at F and G.

You can check that by repeatedly running the Basic PageRank Update Rule, we converge to PageRank **values of 1/2 for each of F and G, and 0 for all other nodes**!!

Teleportation Probability d

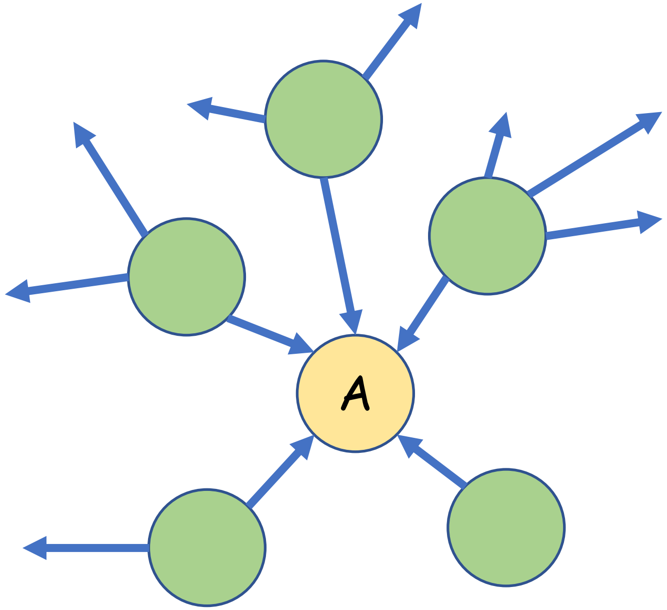
Untuk mengatasi kasus **"dead end"**, PageRank algorithm menggunakan konsep teleportation probability.

Probabilitas bahwa "seorang user akan lompat ke halaman web random lain tanpa melalui out-links"



Jika user sudah sampai sini, ia punya probabilitas **d** untuk teleport ke node lainnya (walau tidak ada out-link kesana).

Biasanya d = 0.1 atau 0.15



Basic PageRank:

$$PR(A) = \frac{PR(T_1)}{C(T_1)} + \frac{PR(T_2)}{C(T_2)} + \cdots + \frac{PR(T_n)}{C(T_n)}$$

Original PageRank:

Teleportation probability

$$PR(A) = \frac{d}{N} + (1-d)\left[\frac{PR(T_1)}{C(T_1)} + \frac{PR(T_2)}{C(T_2)} + \cdots + \frac{PR(T_n)}{C(T_n)}\right]$$

N = Banyaknya node/dokumen

Biasanya d = 0.1 atau 0.15

Algorithm

Symbol	Meaning
P	A web page
d	Damping factor—the probability that a user opens a new web page to begin a new random walk
$PR(P)$	PageRank of page P
$deg(P)^-$	The number of links coming into a page P (in-degree of P)
$deg(P)^+$	The number of links going out of a page P (out-degree of P)
$N(P)^-$	The set of pages that point to P (the in-neighborhood of P)
$N(P)^+$	The set of pages a web page P points to (the out-neighborhood of P)

Algorithm: PageRank calculation of a single graph

Input: G—Directed graph of N web pages
d—Damping factor
Output: $PR[1 \dots N]$, where $PR[P_i]$ is the PageRank of page P_i

Let $PP[1 \dots N]$ denote a spare array of size N
Let d denote the probability of reaching a particular node by a random jump either from a vertex with no outlinks or with probability $(1-d)$
Let $N(P_u)^+$ denote the set of pages with at least one outlink
foreach P_i in N pages of G **do**
    $PR[P_i] = \frac{1}{N}$
    $PP[i] = 0$
end
while PR not converging **do**
    foreach P_i in N pages of G **do**
        foreach P_j in $N(P_i)^+$ **do**
            $PP[P_j] = PP[P_j] + \frac{PR[P_i]}{deg(P_i)^+}$
        end
    end
    foreach P_i in N pages of G **do**
        $PR[P_i] = \frac{d}{N} + (1-d)(PP[P_i])$
        $PP[P_i] = 0$
    end
    Normalize $PR[P_i]$ so that $\sum_{P_i \in N} PR[P_i] = 1$
end

Catatan Penting

- Score PageRank bukan satu-satunya Score untuk ranking dokumen!

- Score PageRank hanyalah "salah satu komponen" untuk scoring dan perlu digabung dengan mekanisme scoring lain seperti BM25 atau yang menggunakan Machine Learning.