Introduction to

Information Retrieval

ΠΛΕ70: Ανάκτηση Πληροφορίας

Διδάσκουσα: Ευαγγελία Πιτουρά

Διάλεξη 12: Ανάλυση Συνδέσμων.

Τι θα δούμε σήμερα

Πως διαφέρει η ανάκτηση πληροφορίας από το web από την ανάκτηση πληροφορίας από ποιο «παραδοσιακές συλλογές κειμένου;

Τι θα δούμε σήμερα

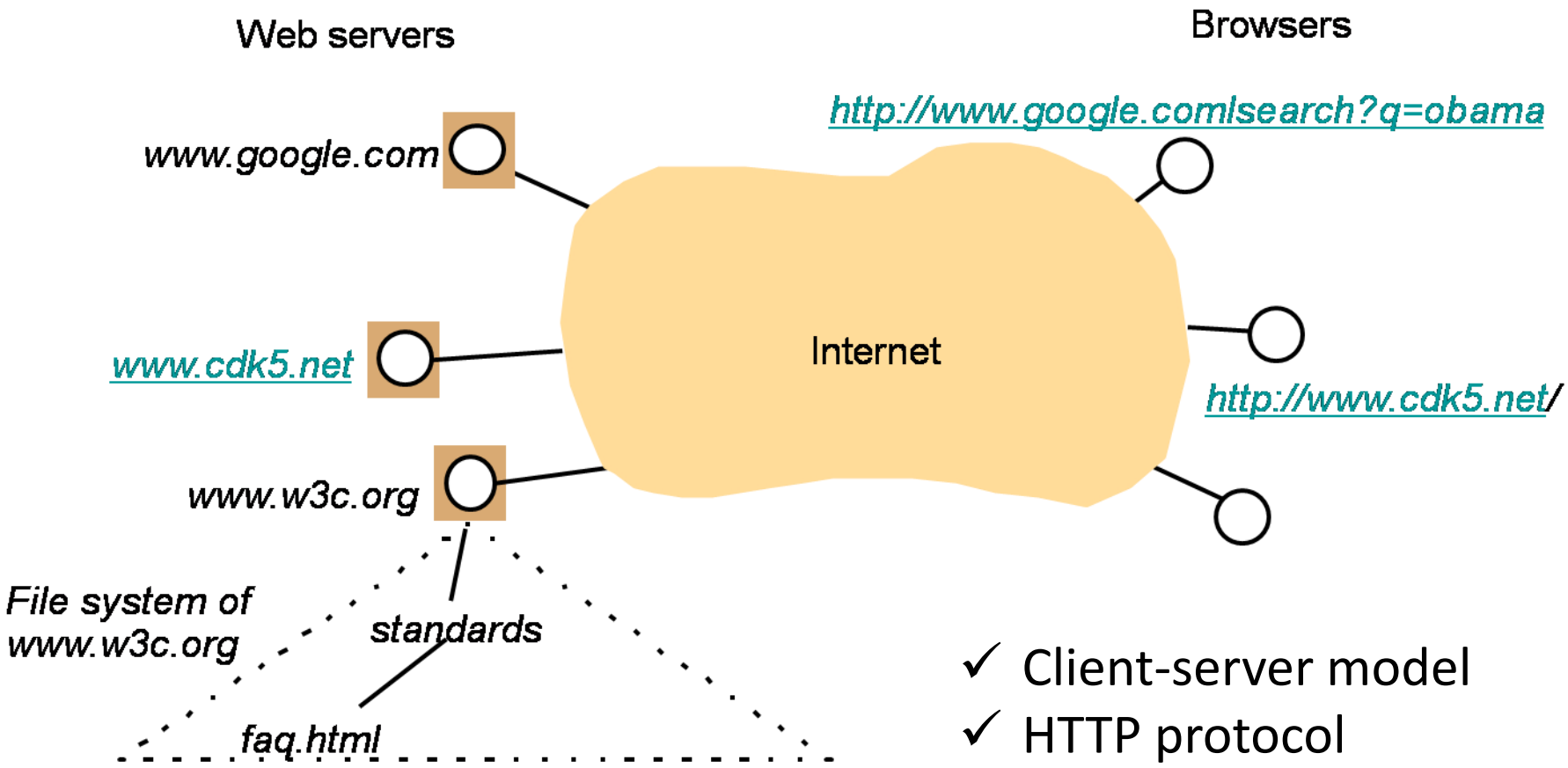
- Web: λίγη ιστορία και ο web γράφος
- Σημασία της άγκυρας (anchor text)
- Ανάλυση συνδέσμων
  - PageRank
  - HITS (Κομβικές σελίδες και σελίδες κύρους)

Web: τι είναι

Web (*World Wide Web, WWW, W3*) μια συλλογή από web σελίδες (ιστοσελίδες) που είναι έγγραφα κειμένου και άλλες πηγές συνδεδεμένα με hyperlinks και URLs
Μια εφαρμογή που τρέχει πάνω από το Internet

- 63 δισεκατομμύρια ιστοσελίδες
- 1 τρισεκατομμύριο διαφορετικές web διευθύνσεις

Web: η δομή του



- ✓ Client-server model
- ✓ HTTP protocol
- ✓ HTML
- ✓ URL/URI

Web (WWW): Ιστορία

Στο τεύχος του **Ιουνίου 1970** του περιοδικού *Popular Science*

Arthur C. Clarke

satellites would one day "bring the accumulated knowledge of the world to your fingertips" using a console that would combine the functionality of the Xerox, telephone, television and a small computer, allowing data transfer and video conferencing around the globe.

Web (WWW): Ιστορία

1980, Tim Berners-Lee (ENQUIRE)

November 1990, με τον *Robert Cailliau*, πρόταση για ένα "Hypertext project με το όνομα "WorldWideWeb" ("W3"): *"web" of "hypertext documents" to be viewed by "browsers" using a client–server architecture.*

Χριστούγεννα 1990, το πρώτο λειτουργικό σύστημα:

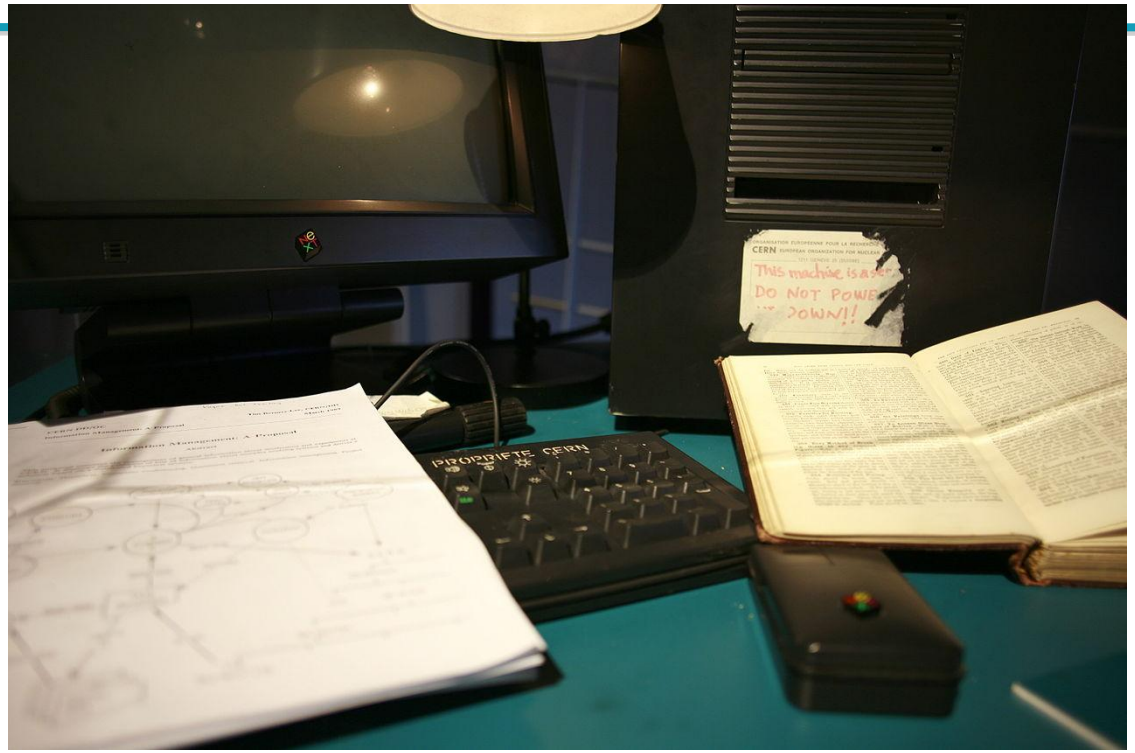
- ο πρώτος web browser (που ήταν και web editor);
- ο πρώτος web server και
- οι πρώτες ιστοσελίδες, που περιέγραφαν το ίδιο το project.

Αύγουστο 1991, post στο alt.hypertext newsgroup – νέο service στο ‘Ιντερνετ

Web (WWW): Ιστορία

Ο πρώτος web server (και πρώτος web browser): A NeXT Computer -

Η πρώτη φωτογραφία στο web το 1992 (CERN house band Les Horribles Cernettes)







logo by Robert Cailliau

Mosaic (1993) πρώτος graphical browser

Δυναμικές και στατικές σελίδες

Στατικές: σελίδες που το περιεχόμενο τους δεν αλλάζει από την μία αίτηση στην άλλη

Δυναμικές σελίδες: Hidden web – Deep web

- ✓ Παράδειγμα: προσωπική ιστοσελίδα vs σελίδα με την κατάσταση των πτήσεων σε ένα αεροδρόμιο

URL: συνήθως όχι κάποιο αρχείο αλλά κάποιο πρόγραμμα στον server

Input part of the GET, e.g., http//www.google.com/search?q=obama

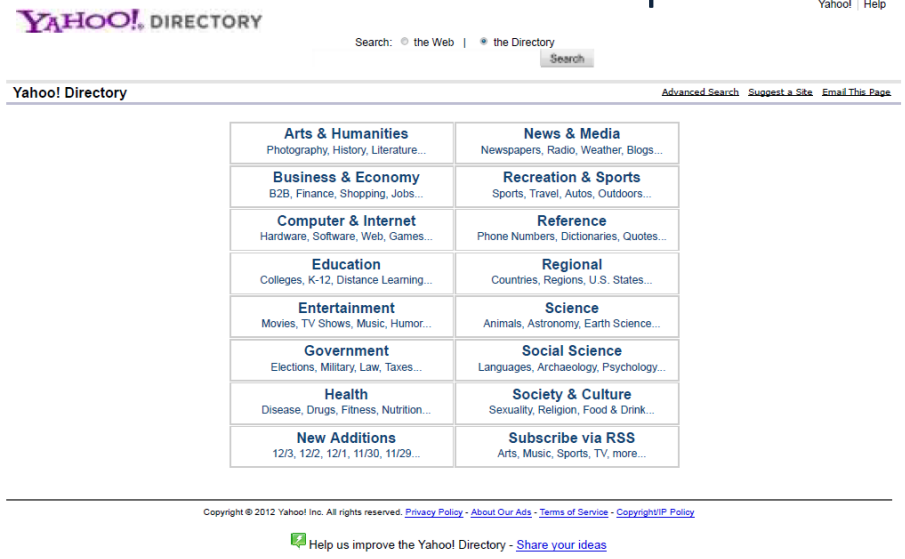


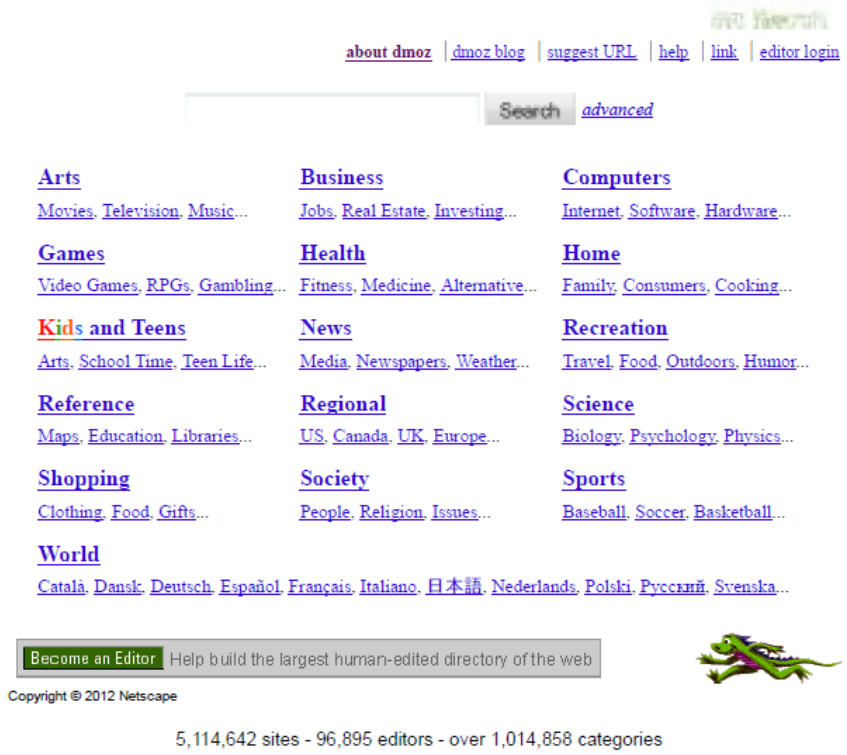
Εύρεση Πληροφορίας

- *Taxonomies* (Yahoo!) – browse through a hierarchical tree with category labels

  About.com

  DMOZ - Open Directory Project

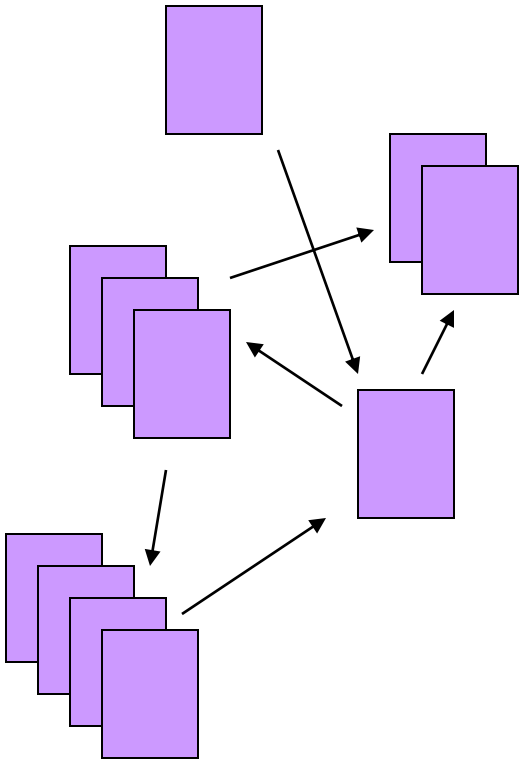




Εύρεση Πληροφορίας

- Full text search (Altavista, Excite, Infoseek)
- Η εποχή του Google: χρήση του web ως γράφου
  - Πέρασμα από τη συνάφεια στο κύρος (authoritativeness)
  - Δεν έχει μόνο σημασία μια σελίδα να είναι συναφής πρέπει να είναι και σημαντική στο web
- Για παράδειγμα, τι είδους αποτελέσματα θα θέλατε να πάρετε στην ερώτηση “greek newspapers”?

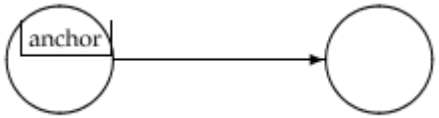
Η συλλογή εγγράφων του Web

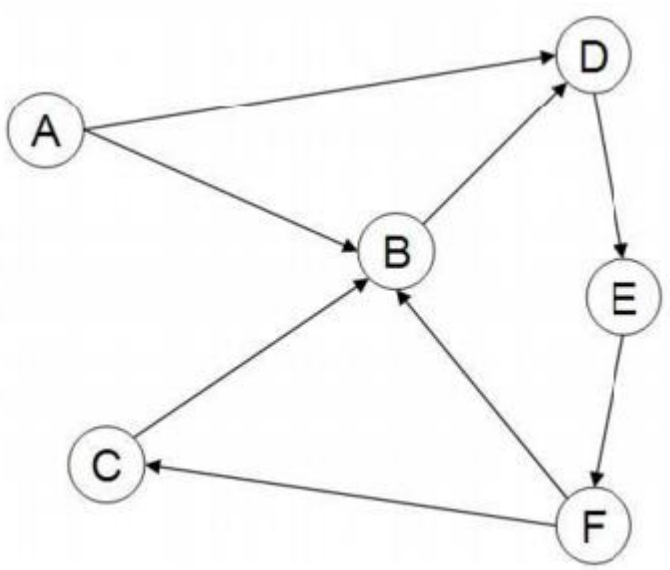


The Web

- No design/co-ordination
- Distributed content creation, linking, democratization of publishing
- Content includes truth, lies, obsolete information, contradictions …
- Unstructured (text, html, …), semi-structured (XML, annotated photos), structured (Databases)…
- Scale much larger than previous text collections … but corporate records are catching up
- Growth – slowed down from initial "volume doubling every few months" but still expanding
- Content can be *dynamically generated*

The Web graph





Anchor text <a>
In-links/Out-links
In-degree (8-15)
Out-degree

The Web Graph

- the distribution of in-degrees not Poisson distribution (if every web page were to pick the destinations of its links uniformly at random).

- Power law,

the total number of web pages with in-degree ***i*** is proportional to $1/i^{\alpha}$

α typically 2.1

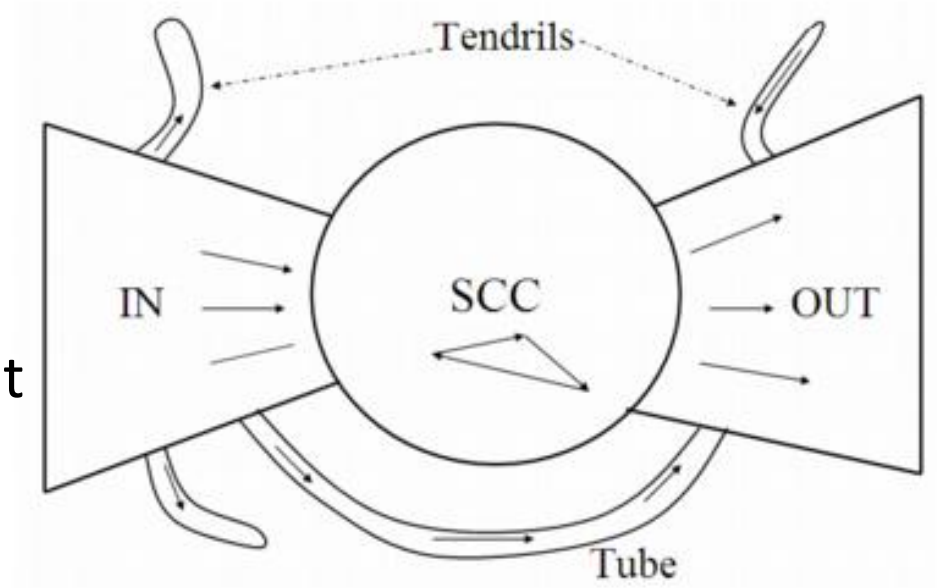
Που αλλού είδαμε παρόμοια κατανομή

The Web graph

Bow-tie shape

Τρεις κατηγορίες: **IN**, **OUT**, **SCC**

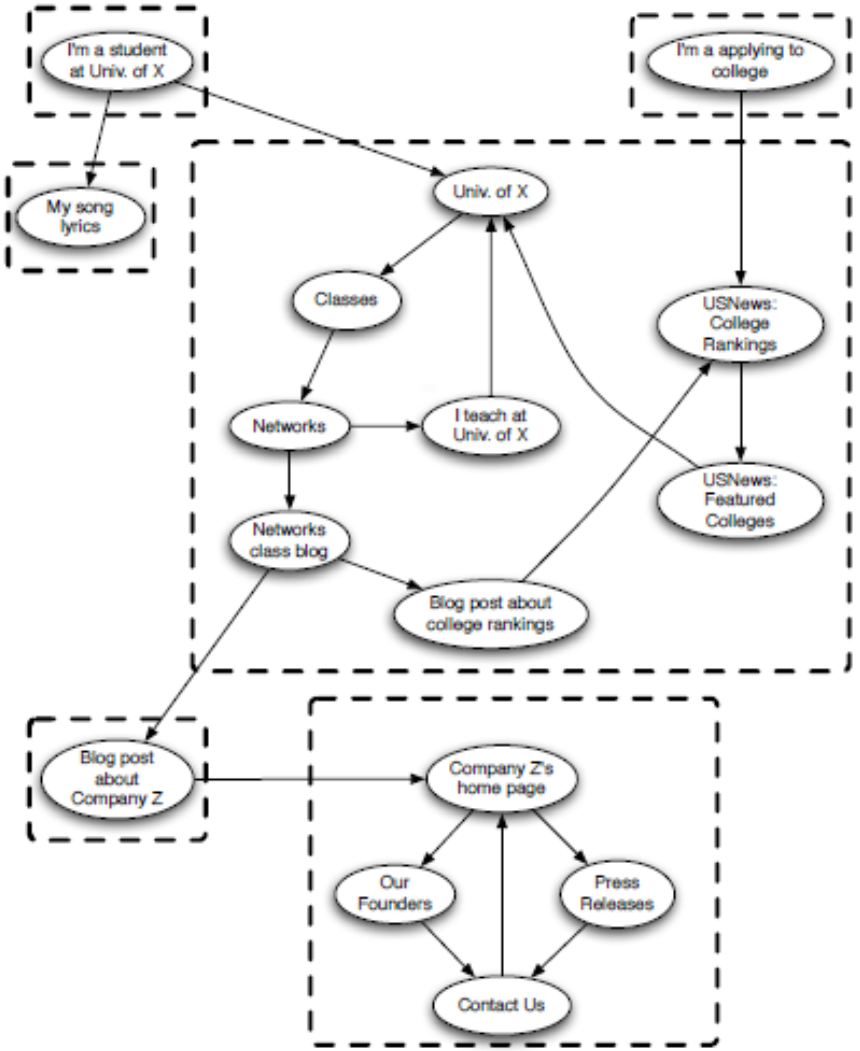
Περιέχει μια μεγάλη ισχυρά συνδεδεμένη συνιστώσα (Strongly Connected Component (SCC))



IN: Σελίδες που οδηγούν στο SCC αλλά όχι το ανάποδο

OUT: Σελίδες στις οποίες μπορούμε να φτάσουμε από το SCC αλλά δεν οδηγούν σε αυτό

The Web graph



From the book Networks, Crowds, and Markets: Reasoning a Highly Connected World. By David Easley and Jon Kleinberg. University Press, 2010. Complete preprint on-line at **http://www.cs.cornell.edu/home/kleinber/networks-book/**

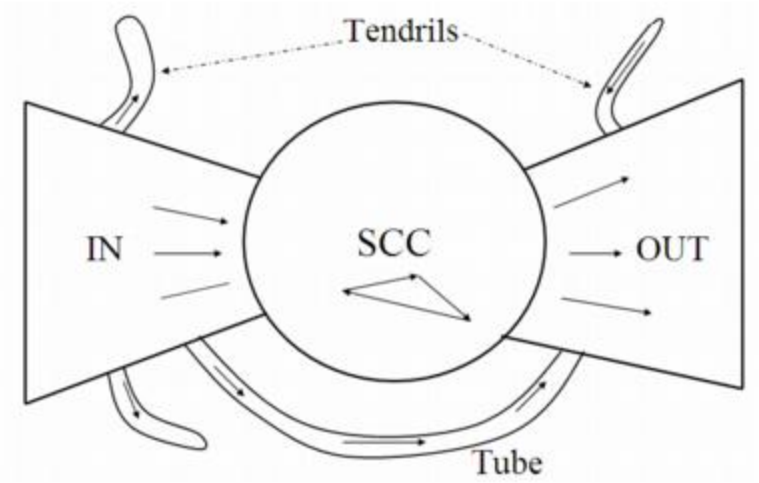
The Web graph

IN, OUT same size, SCC larger

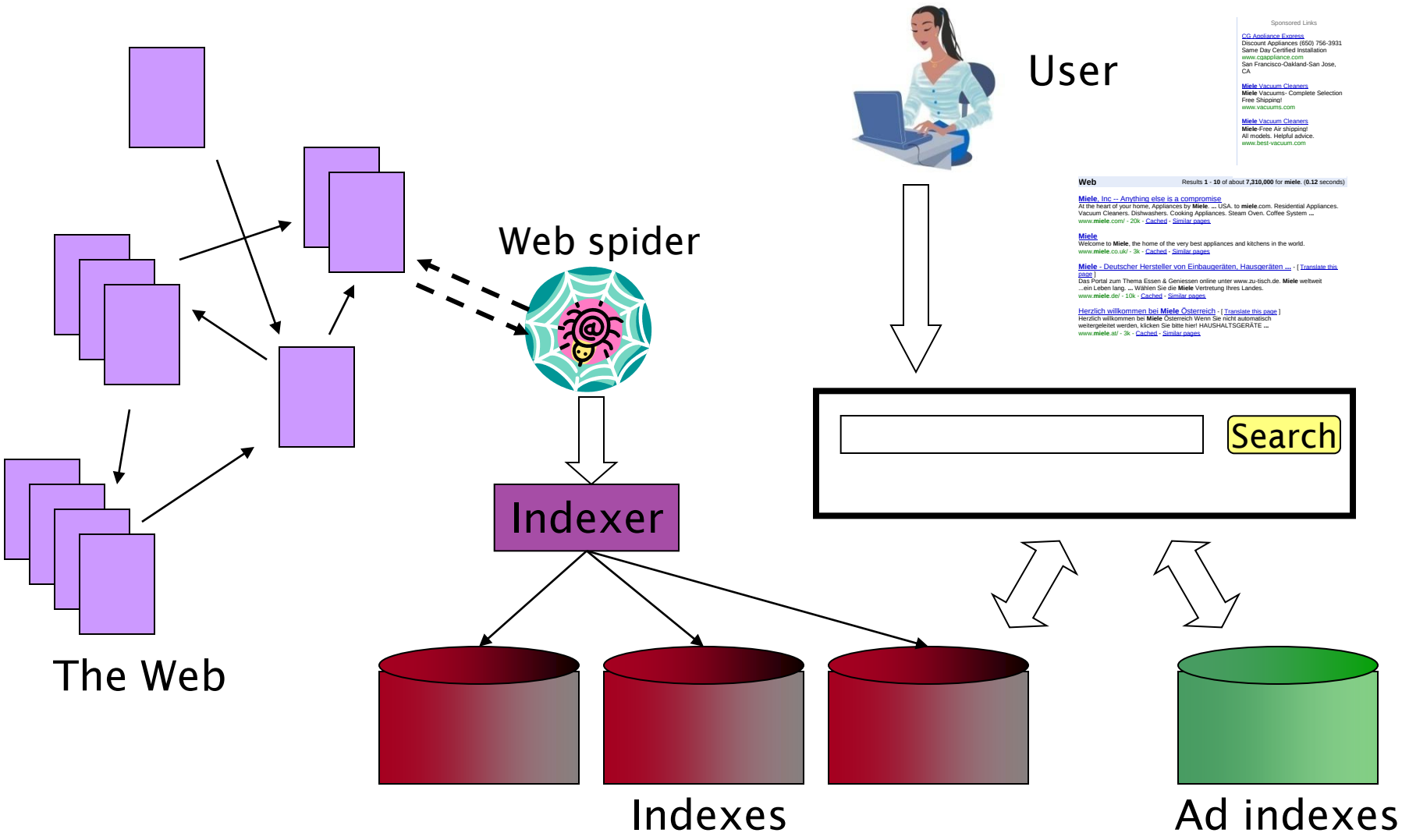
Remaining pages:

- Tubes: small sets of pages outside SCC that lead directly from IN to OUT,
- Tendrils: either lead nowhere from IN, or from nowhere to OUT.

Small disconnected components



Web search basics

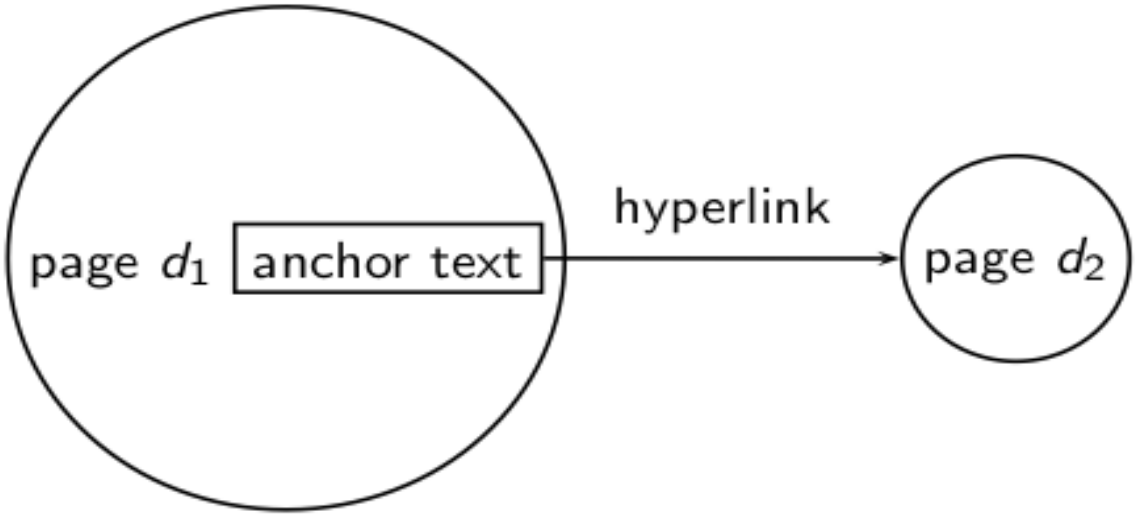


Τι (άλλο) θα δούμε σήμερα

Ανάλυση συνδέσμων (Link Analysis)

- Web: λίγη ιστορία και ο γράφος
- Σημασία της άγκυρας (anchor text)
- Ανάλυση συνδέσμων
  - PageRank
  - HITS (Κομβικές σελίδες και σελίδες κύρους)

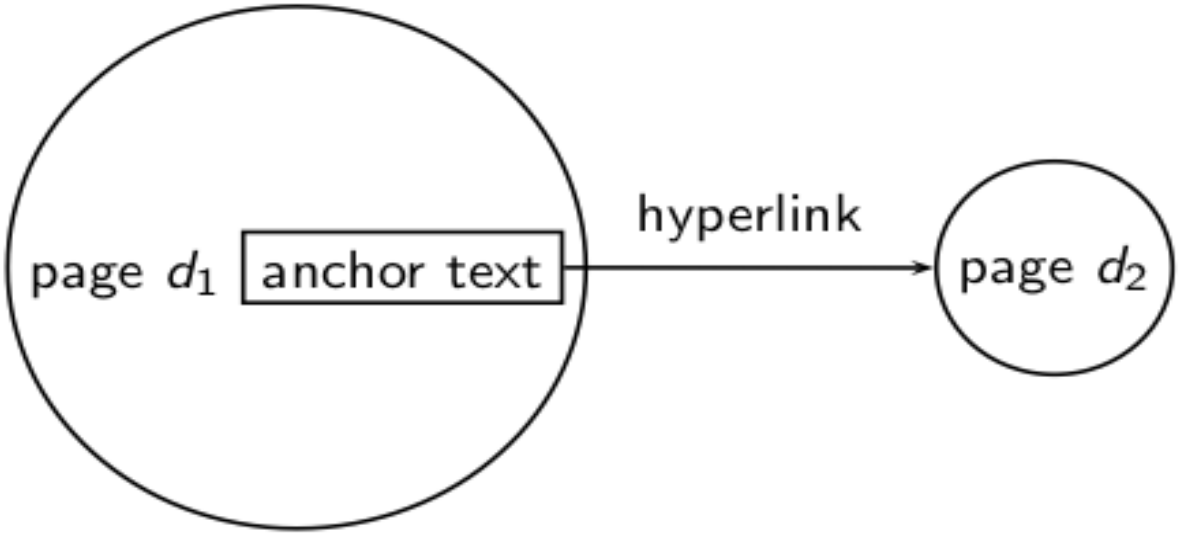
Κείμενο Άγκυρας



Anchor text (κείμενο άγκυρας) κείμενο που περιβάλει τον σύνδεσμο

- Παράδειγμα: “You can find cheap cars <a href =http://…>here . ”
- Anchor text: “You can find cheap cars here”

Σημασία των συνδέσεων



- 1η Υπόθεση: A hyperlink is a quality signal.
  - Η σύνδεση $d_1 \rightarrow d_2$ υποδηλώνει ότι ο συγγραφέας του d_1 θεωρεί το d_2 καλής ποιότητας και συναφές.
- 2η Υπόθεση: Το κείμενο της άγκυρας περιγράφει το περιεχόμενο του d_2.

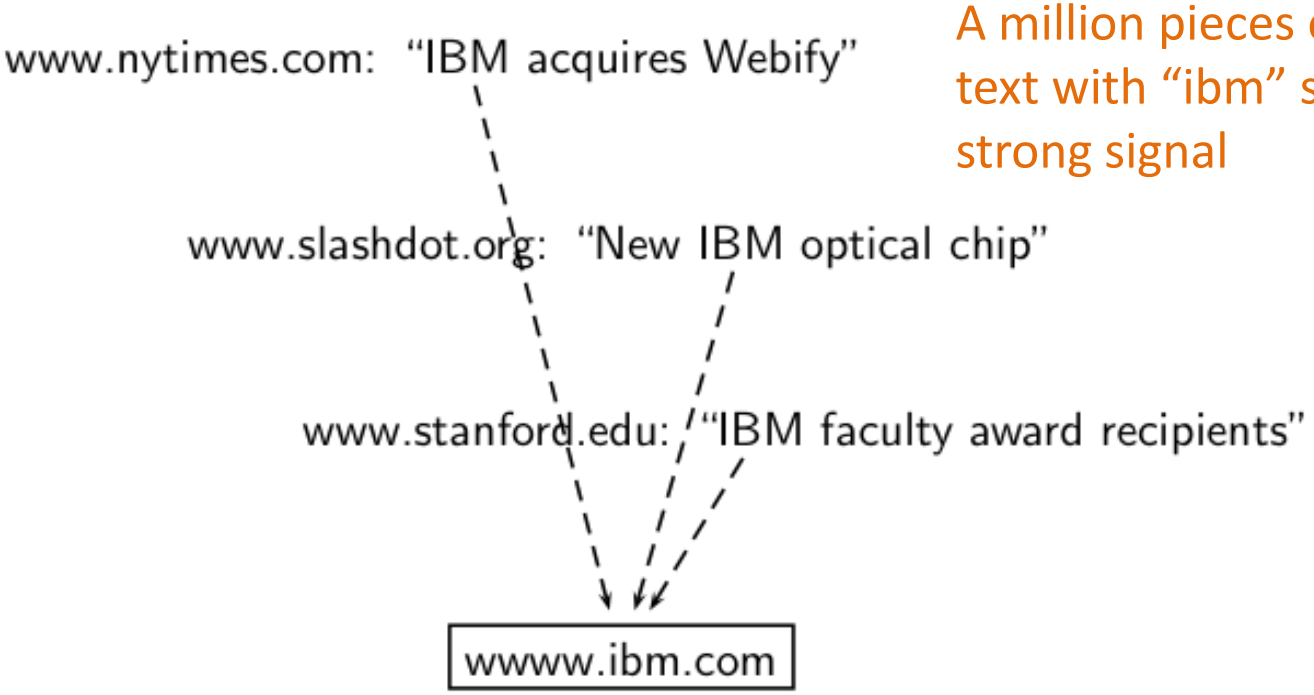
Κείμενο Άγκυρας

Χρήση μόνο **[text of d_2]** ή **[text of d_2] + [anchor text → d_2]**

- Αναζήτηση του [text of d_2] + [anchor text → d_2] συχνά πιο αποτελεσματική από την αναζήτηση μόνο του [text of d_2]
- Παράδειγμα: Ερώτημα *IBM*
  - Matches IBM's copyright page
  - Matches many spam pages
  - Matches IBM wikipedia article
  - May not match IBM home page! if IBM home page is mostly graphics

Κείμενο Άγκυρας

- Αναζήτηση με χρήση του [anchor text → d_2] καλύτερη για το ερώτημα IBM
  - Η σελίδα με τις περισσότερες εμφανίσεις του όρου *IBM* είναι η www.ibm.com

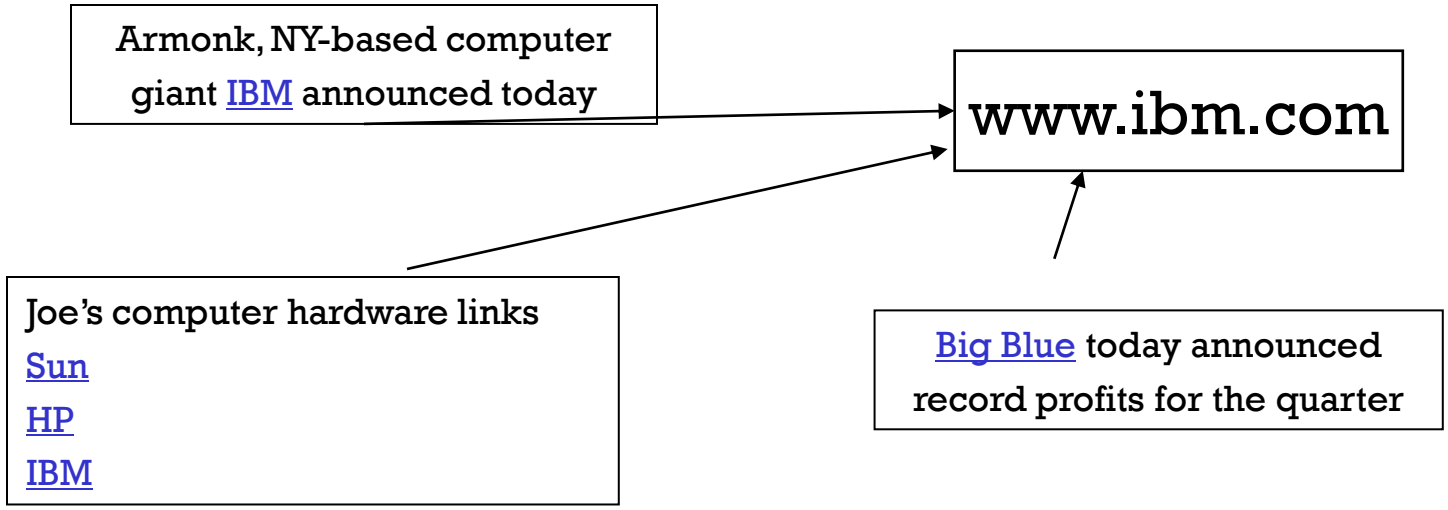


A million pieces of anchor text with "ibm" send a strong signal

Κείμενο Άγκυρας στο Ευρετήριο

Άρα: Το κείμενο στην άγκυρα αποτελεί καλύτερη περιγραφή του περιεχομένου της σελίδας από ότι το περιεχόμενο της

- Όταν κατασκευάζουμε το ευρετήριο για ένα έγγραφο *D*, συμπεριλαμβάνουμε (με κάποιο βάρος) και το κείμενο της άγκυρας των συνδέσεων που δείχνουν στο *D*.



✓ Weigted: Use idf for common words such as Click, Here
✓ Also, extended anchor text

Google Bombs

Google bomb: a search with "bad" results due to maliciously manipulated anchor text.

Google introduced a new weighting function in January 2007

✓ *Can score anchor text with weight depending on the authority of the anchor page's website*

E.g., if we were to assume that content from cnn.com or yahoo.com is authoritative, then trust the anchor text from them

- Miserable failure (Bush 2004)
- Still some remnants: [dangerous cult] on Google, Bing, Yahoo
  - Coordinated link creation by those who dislike the Church of Scientology
- Defused Google bombs: [dumb motherf…], [who is a failure?], [evil empire] [cheerful achievement]

Anchor Text

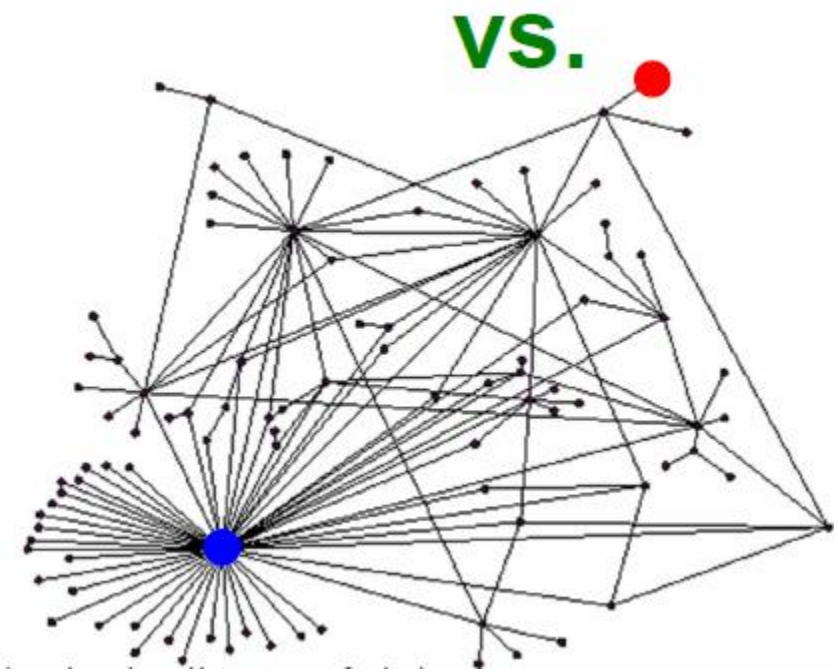
- Other applications
  - Weighting/filtering links in the graph
  - Generating page descriptions from anchor text

Υπόθεση 2: annotation of target



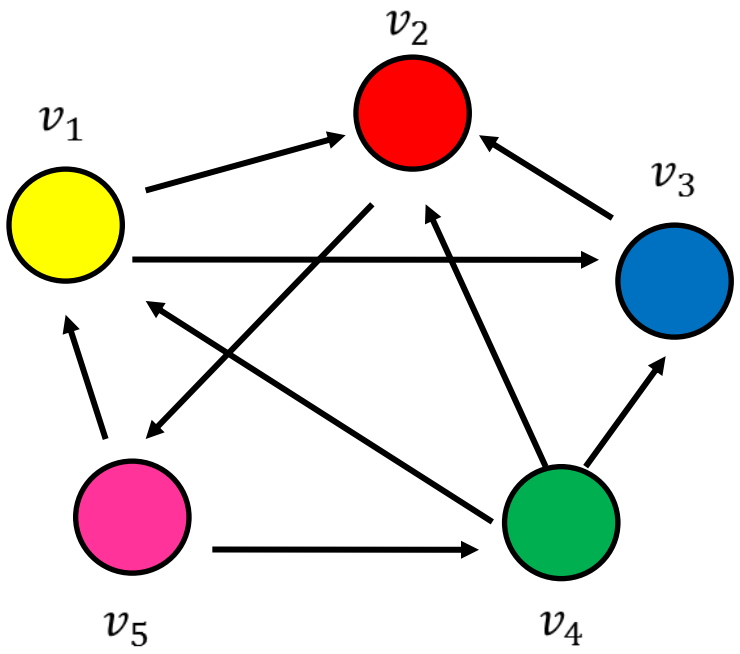
Ανάλυση Συνδέσμων - Link Analysis

Δεν είναι όλες οι σελίδες ίσες



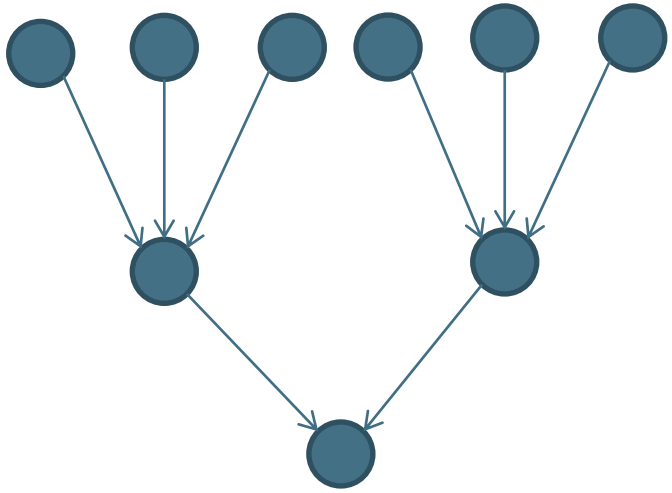
Διάταξη με βάση τη δημοτικότητα

- Διάταξη των σελίδων με βάσει τον αριθμό των εισερχόμενων ακμών (in-degree, degree centrality)



1. **Red Page**
2. **Yellow Page**
3. **Blue Page**
4. **Purple Page**
5. **Green Page**

Αρκεί η δημοτικότητα;



- Δεν είναι σημαντικό *πόσοι κόμβοι* δείχνουν σε μια σελίδα αλλά το *πόσο σημαντικοί* είναι αυτοί οι κόμβοι

PageRank

PageRank

- Βασική ιδέα: Μια σελίδα είναι σημαντική αν δείχνουν σε αυτήν σημαντικές σελίδες
(η αξία ενός κόμβου είναι το άθροισμα της αξίας των φίλων του)

- *Αναδρομικός* ορισμός!

- Πως υλοποιούμε το παραπάνω;

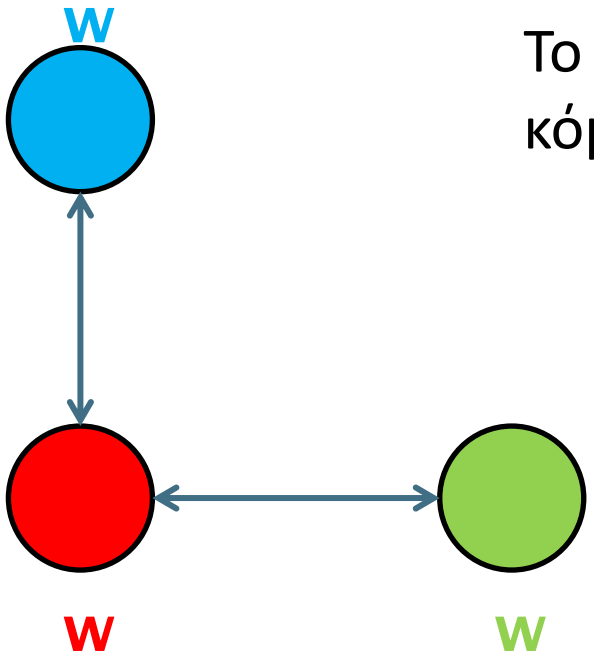
PageRank: Βασική ιδέα

Έχουμε μια «μονάδα κύρους» που τη λέμε PageRank και την μοιράζουμε στις σελίδες.

Κάθε σελίδα έχει ένα PageRank

- Κάθε σελίδα *μοιράζει το PageRank στις σελίδες που δείχνει*
- Το PageRank μιας σελίδας είναι το *άθροισμα των PageRank των σελίδων που δείχνουν σε αυτήν*

Ένα απλό παράδειγμα



Το συνολικό PageRank μοιράζεται στους 3 κόμβους

$w + w + w = 1$

$w = w + w$

$w = ½\ w$

$w = ½\ w$

- Solving the system of equations we get the authority values for the nodes
  - $w = ½$ $w = ¼$ $w = ¼$

Ακόμα ένα παράδειγμα

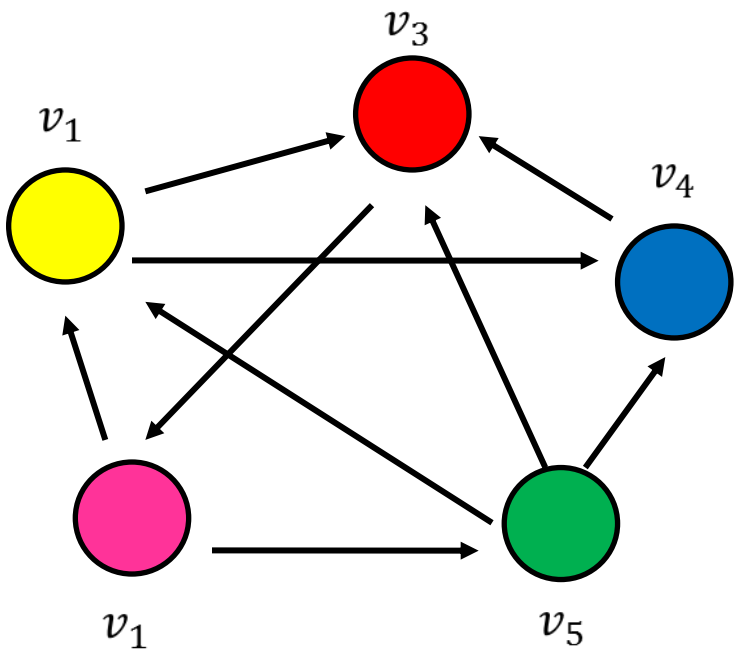
$w_1 = 1/3\ w_4 + 1/2\ w_5$

$w_2 = 1/2\ w_1 + w_3 + 1/3\ w_4$

$w_3 = 1/2\ w_1 + 1/3\ w_4$

$w_4 = 1/2\ w_5$

$w_5 = w_2$



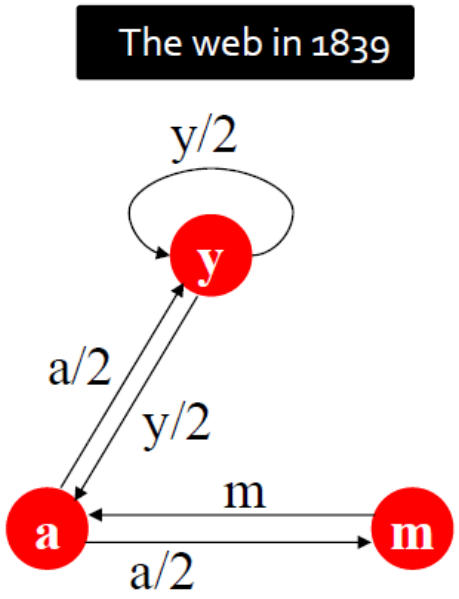
Και ακόμα ένα μαζί με τον ορισμό

Κάθε κόμβος (σελίδα) έχει ένα βαθμό (rank)

Ο βαθμός r_j για τον κόμβο j ισούται με

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

d_i … out-degree of node i



"Flow" equations:

$r_y = r_y/2 + r_a/2$

$r_a = r_y/2 + r_m$

$r_m = r_a/2$

PageRank: Αλγόριθμος

Σε ένα γράφο με *n* nodes, αναθέτουμε σε όλους το ίδιο αρχικό PageRank = 1/*n*.

- Εκτελούμε μια ακολουθία από *k* ενημερώσεις των PageRank τιμών με βάση των παρακάτω κανόνα:

1. Κάθε σελίδα μοιράζει την τρέχουσα PageRank τιμή της ισόποσα στις out-going *ακμές και τις περνά στους αντίστοιχους κόμβους*
2. Κάθε σελίδα ανανεώνει την PageRank τιμή της ώστε να είναι ίση με το άθροισμα τον ποσών που δέχεται μέσω των incoming ακμών της.

PageRank: Αλγόριθμος

Επαναληπτικός υπολογισμός

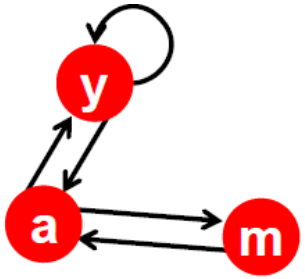
Initialize all PageRank weights to $\frac{1}{n}$

Repeat:

$$w_v = \sum_{u \to v} \frac{1}{d_{out}(u)} w_u$$

Until the weights do not change

Παράδειγμα



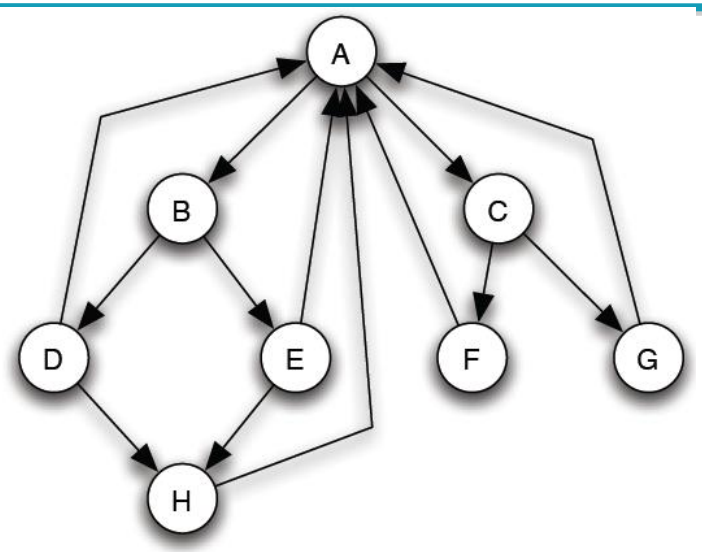
$$r_y = r_y/2 + r_a/2$$
$$r_a = r_y/2 + r_m$$
$$r_m = r_a/2$$

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

Iteration 0, 1, 2, …

Ένα μεγαλύτερο παράδειγμα

Αρχικά όλοι οι κόμβοι PageRank 1/8

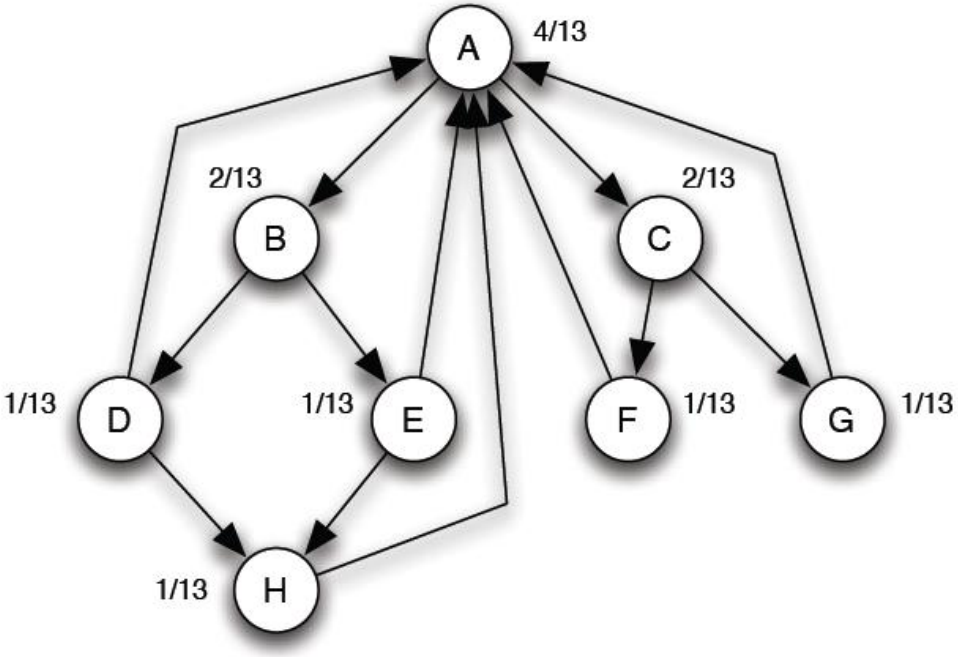


Step	A	B	C	D	E	F	G	H
1	1/2	1/16	1/16	1/16	1/16	1/16	1/16	1/8
2	3/16	1/4	1/4	1/32	1/32	1/32	1/32	1/16

✓ Ένα είδος ροής ("fluid") που κινείται στο δίκτυο

✓ Το συνολικό PageRank στο δίκτυο παραμένει σταθερό (δε χρειάζεται κανονικοποίηση)

Ισορροπία



- ✓ Ένας απλός τρόπος να ελέγξουμε αν σε ισορροπία (an equilibrium set of PageRank values): αθροίζουν σε 1 και δεν αλλάζουν αν εφαρμόσουμε τον κανόνα ενημερώσης
- ✓ Αν το δίκτυο ισχυρά συνεκτικό, υπάρχει ένα μοναδικό σύνολο τιμών ισορροπίας

PageRank: Διανυσματική αναπαράσταση

Stochastic Adjacency Matrix – Πίνακας Γειτνίασης *M*

Πίνακας M – πίνακας γειτνίασης του web

Αν j -> i, τότε $M_{ij} = 1/\text{outdegree}(j)$

Αλλιώς, $M_{ij} = 0$

Page Rank Vector *r*

Ένα διάνυσμα με μία τιμή για κάθε σελίδα (το PageRank της σελίδας)

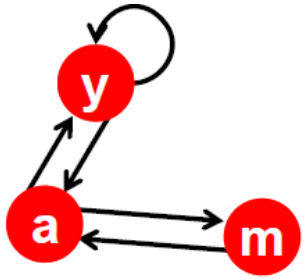
PageRank: Διανυσματική αναπαράσταση

- **Stochastic adjacency matrix M**
  - Let page j has d_j out-links
  - If $j \rightarrow i$, then $M_{ij} = \frac{1}{d_j}$ else $M_{ij} = 0$
    - M is a **column stochastic matrix**
      - Columns sum to 1
- **Rank vector r:** vector with an entry per page
  - r_i is the importance score of page i
  - $\sum_i r_i = 1$
- **The flow equations can be written**

$$r = M \cdot r \qquad r_j = \sum_{i \rightarrow j} \frac{r_i}{\mathrm{d_i}}$$

PageRank: Διανυσματική αναπαράσταση

- **Power Iteration:**
  - Set $r_j = 1$
  - $r_j = \sum_{i \to j} \frac{r_i}{d_i}$
    - And iterate



	y	a	m
y	½	½	0
a	½	0	1
m	0	½	0

$r_y = r_y/2 + r_a/2$
$r_a = r_y/2 + r_m$
$r_m = r_a/2$

- **Example:**

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

Iteration 0, 1, 2, …

PageRank: Διανυσματική αναπαράσταση

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

- Συγκλίνει;
- Συγκλίνει σε αυτό που θέλουμε;
- Ποια είναι η φυσική σημασία;

Τυχαίος Περίπατος (Random Walks)

Ο αλγόριθμος προσομοιώνει ένα τυχαίο περίπατο στο γράφο

Τυχαίος περίπατος (random walk)

- Ξεκίνα από κάποιον κόμβο επιλεγμένο uniformly at random με πιθανότητα 1/n
- Επέλεξε μια από τις εξερχόμενες ακμές του κόμβου uniformly at random
- Ακολούθησε την ακμή
- Επανέλαβε

Τυχαίος Περίπατος (Random Walks)

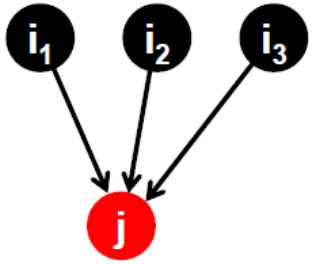
Claim: Η πιθανότητα να είσαι στη σελίδα X μετά από k βήματα του τυχαίου περιπάτου είναι το PageRank της σελίδας X μετά από k επαναλήψεις του υπολογισμού του PageRank

Το μοντέλου του Random Surfer

Του χρήστη που τριγυρνά στο web, ξεκινώντας από μια τυχαία σελίδα και ακολουθώντας τυχαία συνδέσεις

Και πιο τυπικά

- **Imagine a random web surfer:**
  - At any time t, surfer is on some page i
  - At time $t+1$, the surfer follows an out-link from i uniformly at random
  - Ends up on some page j linked from i
  - Process repeats indefinitely



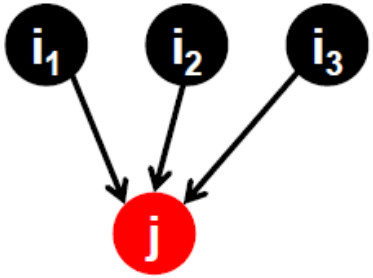
$$r_j = \sum_{i \to j} \frac{r_i}{\mathrm{d}_{\mathrm{out}}(\mathrm{i})}$$

- **Let:**
  - $\boldsymbol{p}(\boldsymbol{t})$ … vector whose i^{th} coordinate is the prob. that the surfer is at page i at time t
  - So, $\boldsymbol{p}(\boldsymbol{t})$ is a probability distribution over pages

Και πιο τυπικά

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

$$p(t+1) = M \cdot p(t)$$



- Suppose the random walk reaches a state

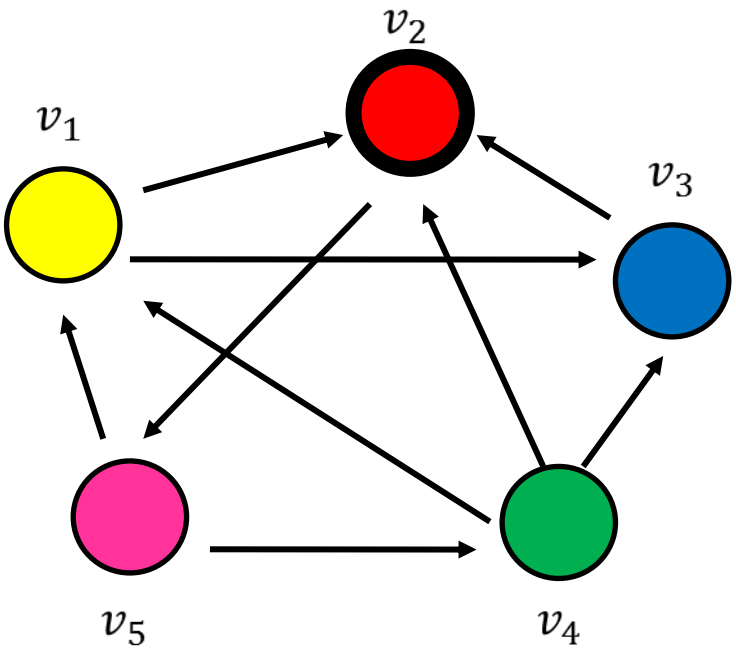
$$p(t+1) = M \cdot p(t) = p(t)$$

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**

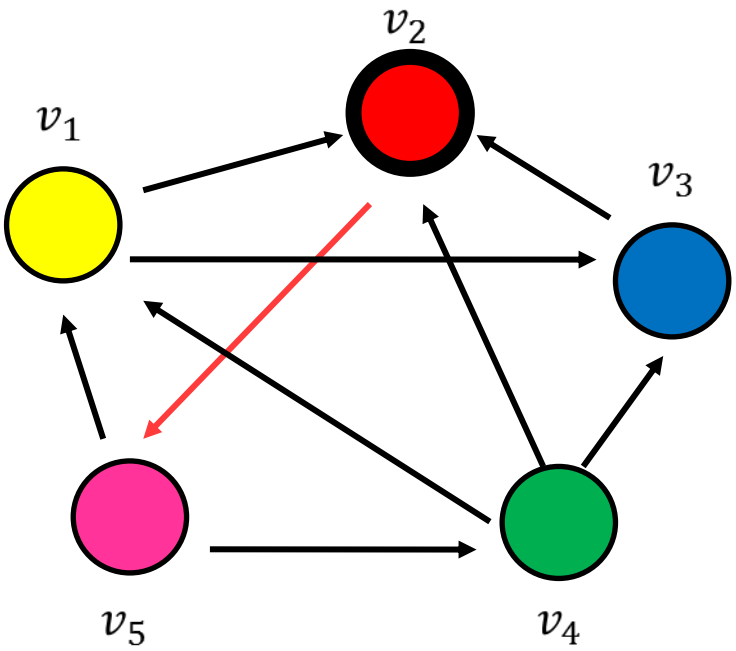
Example

- Step 0



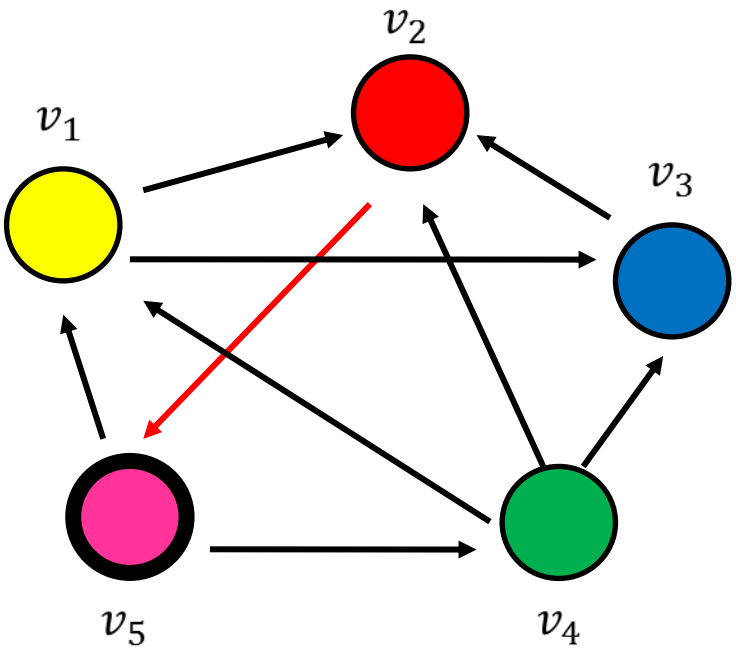
Example

- Step 0



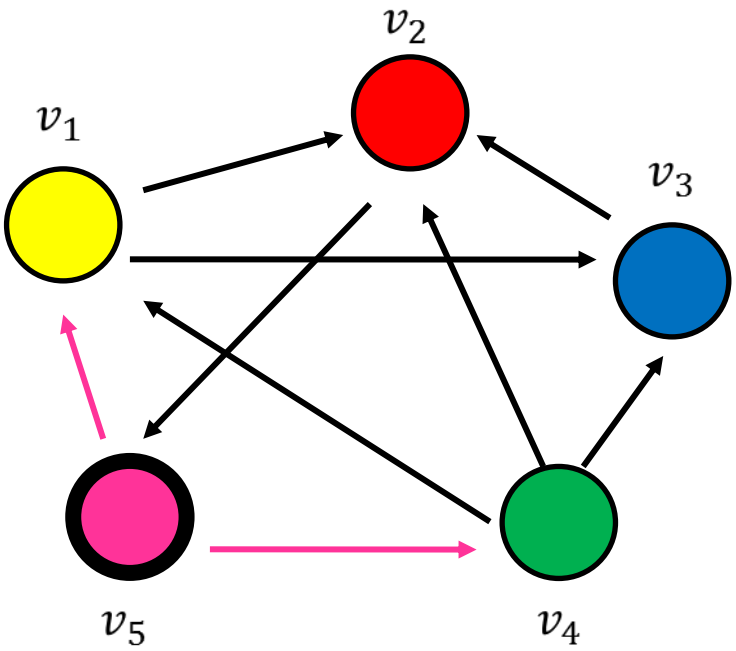
Example

- Step 1



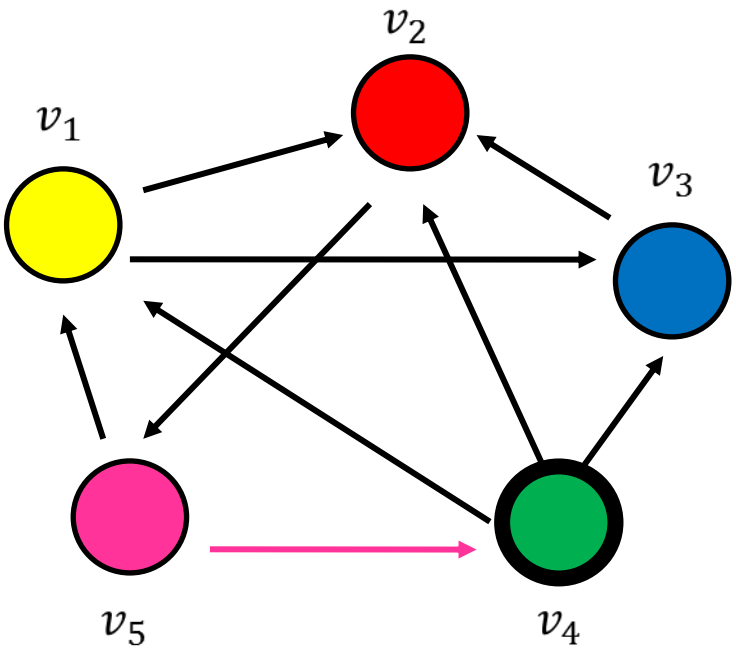
Example

- Step 1



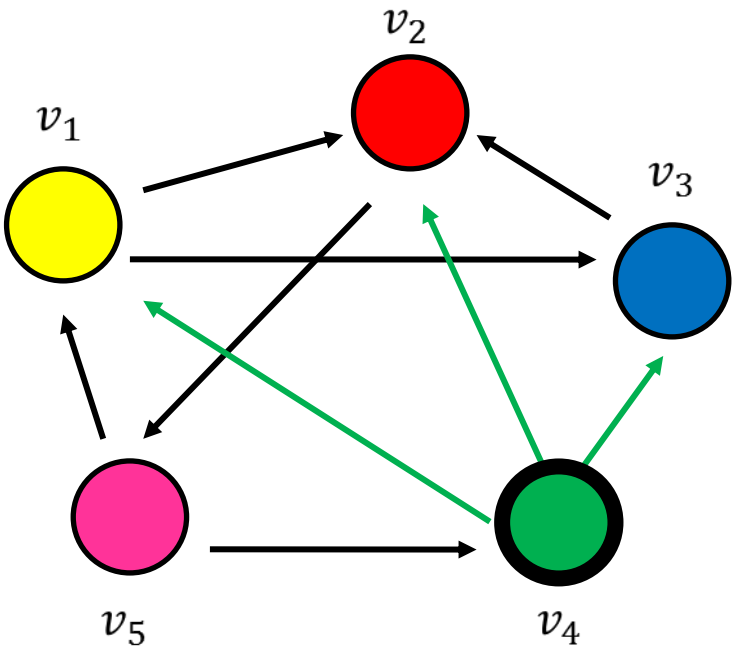
Example

- Step 2



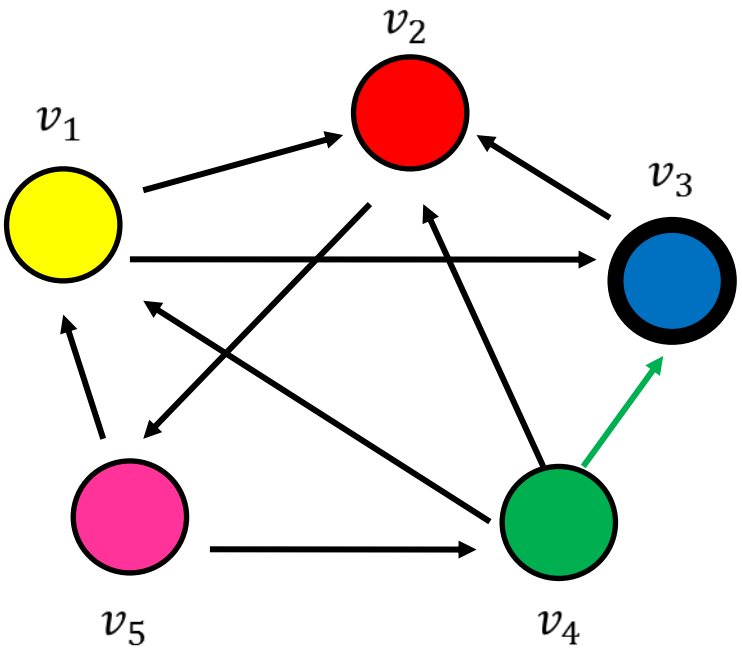
Example

- Step 2



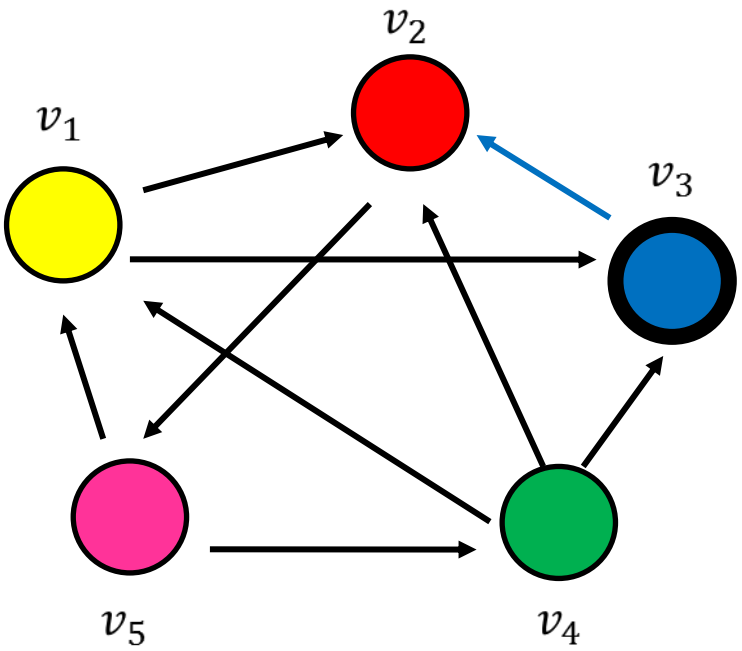
Example

- Step 3



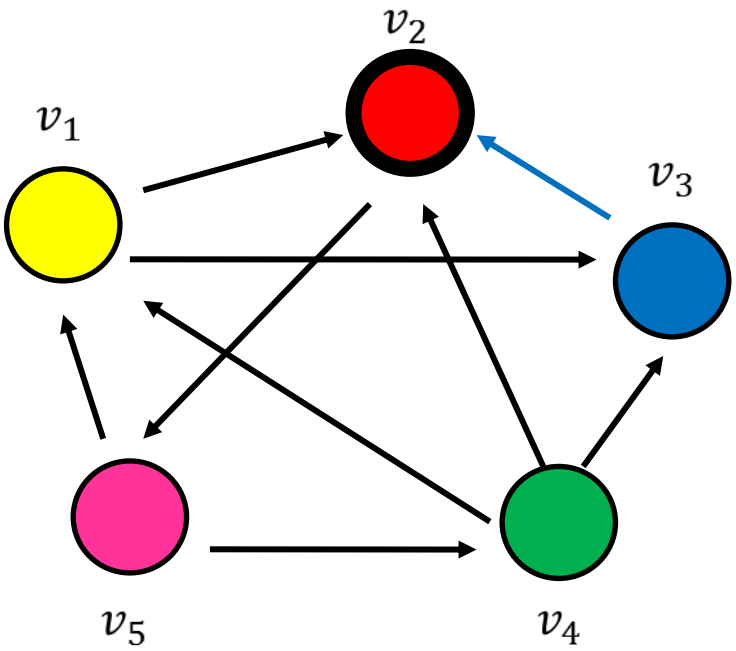
Example

- Step 3



Example

- Step 4…



Random walk

- Question: what is the probability p_i^t of being at node i after t steps?

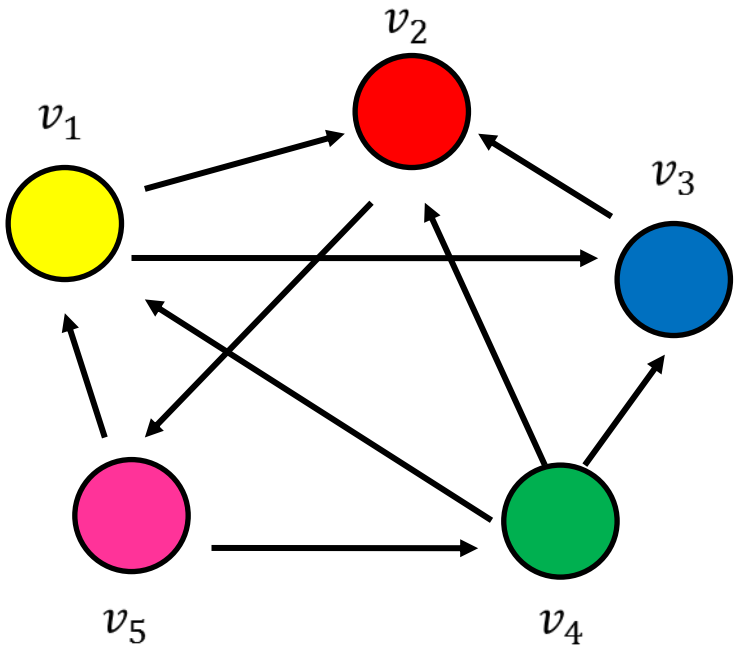
$$p_1^0 = \frac{1}{5} \qquad p_1^t = \frac{1}{3}p_4^{t-1} + \frac{1}{2}p_5^{t-1}$$

$$p_2^0 = \frac{1}{5} \qquad p_2^t = \frac{1}{2}p_1^{t-1} + p_3^{t-1} + \frac{1}{3}p_4^{t-1}$$

$$p_3^0 = \frac{1}{5} \qquad p_3^t = \frac{1}{2}p_1^{t-1} + \frac{1}{3}p_4^{t-1}$$

$$p_4^0 = \frac{1}{5} \qquad p_4^t = \frac{1}{2}p_5^{t-1}$$

$$p_5^0 = \frac{1}{5} \qquad p_5^t = p_2^{t-1}$$



PageRank: Επεκτάσεις

Δύο προβλήματα

1. Dead ends: σελίδες χωρίς εξερχόμενες ακμές

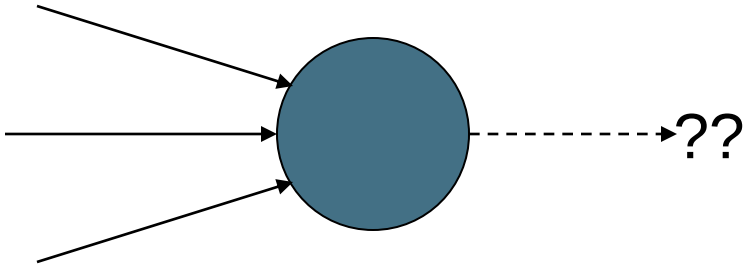
   Έχουν ως αποτέλεσμα να ξεφεύγει (leak out) το PageRank

2. Spider traps: Ομάδα σελίδων που όλες οι εξερχόμενες ακμές είναι μεταξύ τους

   Τελικά απορροφούν όλο το PageRank

PageRank: Αδιέξοδα

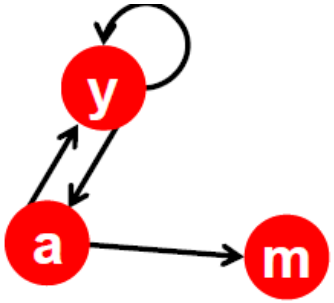
Αδιέξοδα (dead ends): σελίδες που δεν έχουν outlinks



Ο τυχαίος περίπατος μπορεί να κολλήσει σε ένα τέτοιον κόμβο

PageRank: Αδιέξοδα

- **Power Iteration:**
  - Set $r_j = 1$
  - $r_j = \sum_{i \to j} \frac{r_i}{d_i}$
    - And iterate



	y	a	m
y	½	½	0
a	½	0	0
m	0	½	0

$r_y = r_y/2 + r_a/2$
$r_a = r_y/2$
$r_m = r_a/2$

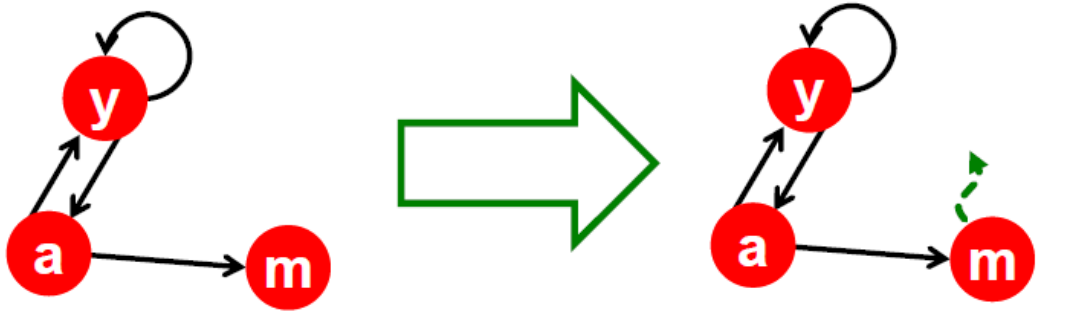
- **Example:**

$$\begin{bmatrix} r_y \\ r_a \\ r_m \end{bmatrix} = \begin{matrix} 1/3 & 2/6 & 3/12 & 5/24 & & 0 \\ 1/3 & 1/6 & 2/12 & 3/24 & \dots & 0 \\ 1/3 & 1/6 & 1/12 & 2/24 & & 0 \end{matrix}$$

Iteration 0, 1, 2, …

PageRank: Αδιέξοδα

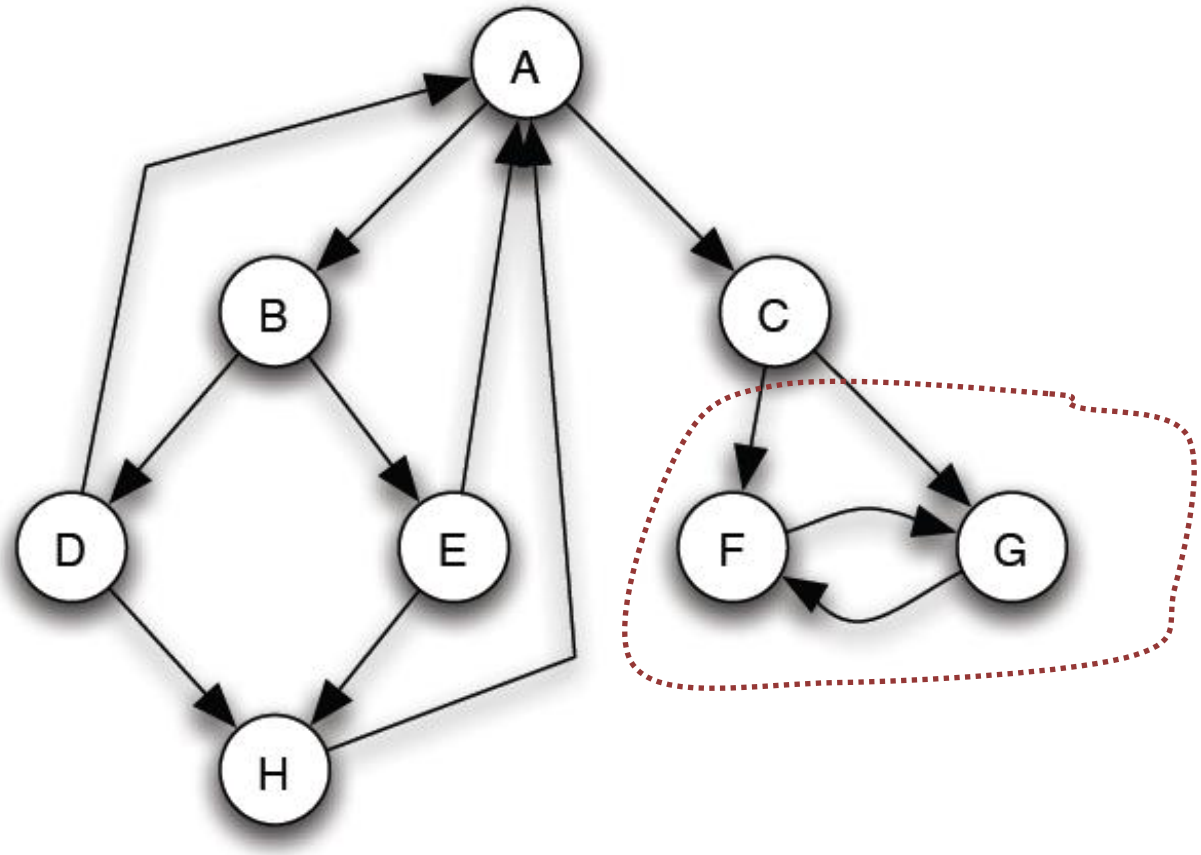
- **Teleports:** Follow random teleport links with probability 1.0 from dead-ends
  - Adjust matrix accordingly



	y	a	m
y	½	½	0
a	½	0	0
m	0	½	0

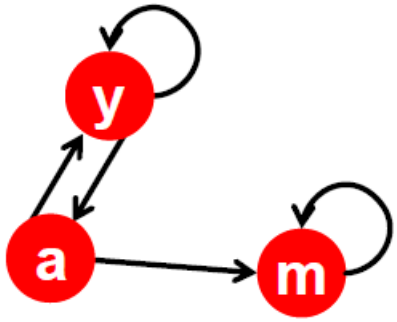
	y	a	m
y	½	½	⅓
a	½	0	⅓
m	0	½	⅓

PageRank: Spider Traps



PageRank: Spider Traps

- **Power Iteration:**
  - Set $r_j = 1$
  - $r_j = \sum_{i \to j} \frac{r_i}{d_i}$
    - And iterate



	y	a	m
y	½	½	0
a	½	0	0
m	0	½	1

$r_y = r_y/2 + r_a/2$
$r_a = r_y/2$
$r_m = r_a/2 + r_m$

- **Example:**

$$\begin{pmatrix} r_y \\ r_a \\ r_m \end{pmatrix} = \begin{matrix} 1/3 & 2/6 & 3/12 & 5/24 & & 0 \\ 1/3 & 1/6 & 2/12 & 3/24 & \ldots & 0 \\ 1/3 & 3/6 & 7/12 & 16/24 & & 1 \end{matrix}$$

Iteration 0, 1, 2, …

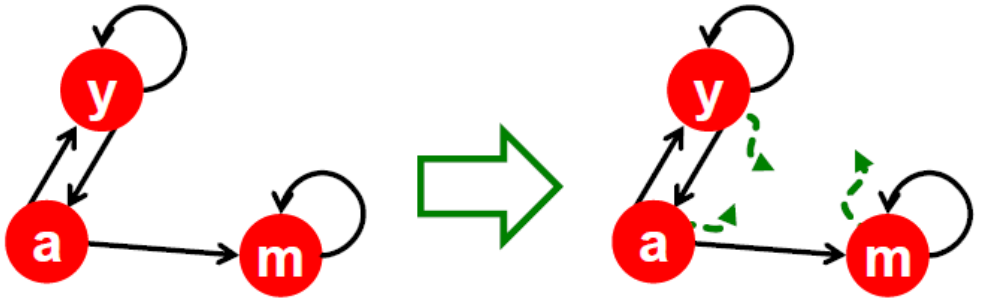
PageRank: Spider Traps

Τυχαία περίπατοι με «άλματα»

Με πιθανότητα β, ο περιπατητής ακολουθεί μια τυχαία εξερχόμενη ακμή όπως πριν και με πιθανότητα 1-β επιλέγει (jumps) σε μια τυχαία σελίδα στο δίκτυο, επιλεγμένη με ίση πιθανότητα (1/n)

PageRank: 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 some page uniformly at random
  - 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**



PageRank: random walks with jumps

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

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

d_i ... out-degree of node i

PageRank και αλυσίδες Markov

Markov chains

- A Markov chain describes a discrete time stochastic process over a set of states

$$S = \{s_1, s_2, \ldots, s_n\}$$

according to a transition probability matrix $P = \{P_{ij}\}$

  - P_{ij} = probability of moving to state j when at state i

- Matrix P has the property that the entries of all rows sum to 1

$$\sum_j P[i,j] = 1$$

A matrix with this property is called stochastic

- State probability distribution: The vector $p^t = (p_i^t, p_2^t, \ldots, p_n^t)$ that stores the probability of being at state s_i after t steps

- Memorylessness property: The next state of the chain depends only at the current state and not on the past of the process (first order MC)
  - Higher order MCs are also possible

- Markov Chain Theory: After infinite steps the state probability vector converges to a unique distribution if the chain is irreducible and aperiodic

Markov chains

Irreducible: ensures that there is a sequence of transitions of non-zero probability from any state to any other

Aperiodicity: ensures that the states are not partitioned into sets such that all state transitions occur cyclically from one set to another.

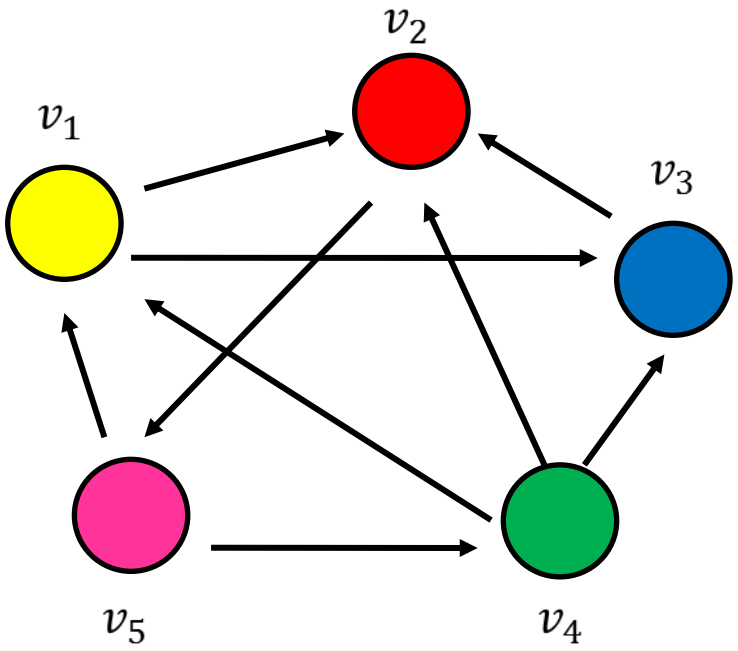
Random walks

- Random walks on graphs correspond to Markov Chains
  - The set of states S is the set of nodes of the graph G
  - The transition probability matrix is the probability that we follow an edge from one node to another

$$P[i,j] = 1/\deg_{out}(i)$$

An example

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



$$P = \begin{bmatrix} 0 & 1/2 & 1/2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 1/3 & 1/3 & 1/3 & 0 & 0 \\ 1/2 & 0 & 0 & 1/2 & 0 \end{bmatrix}$$

Node Probability vector

- The vector $p^t = (p_i^t, p_2^t, \dots, p_n^t)$ that stores the probability of being at node v_i at step t

- p_i^0 = the probability of starting from state i (usually) set to uniform

- We can compute the vector p^t at step t using a vector-matrix multiplication

$$p^t = p^{t-1} P$$

An example

$$P = \begin{bmatrix} 0 & 1/2 & 1/2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 1/3 & 1/3 & 1/3 & 0 & 0 \\ 1/2 & 0 & 0 & 1/2 & 0 \end{bmatrix}$$

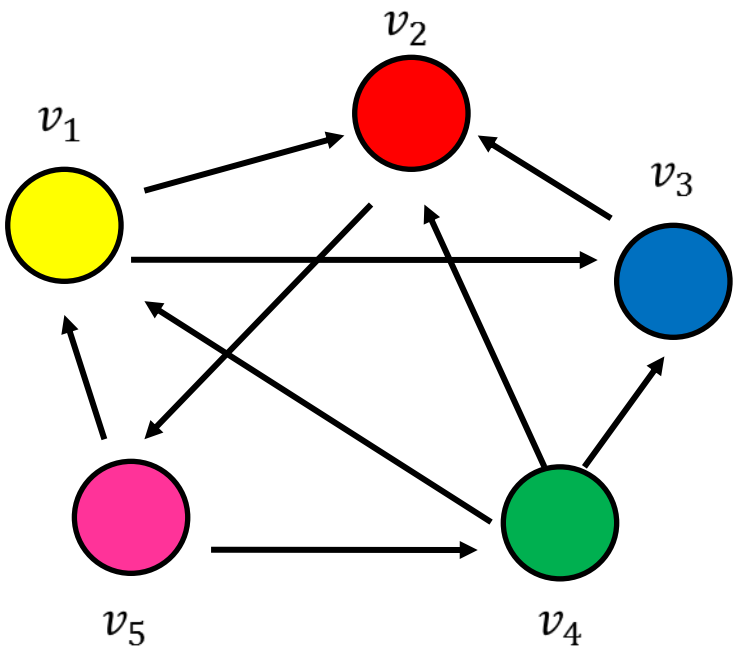
$$p_1^t = \frac{1}{3}p_4^{t-1} + \frac{1}{2}p_5^{t-1}$$

$$p_2^t = \frac{1}{2}p_1^{t-1} + p_3^{t-1} + \frac{1}{3}p_4^{t-1}$$

$$p_3^t = \frac{1}{2}p_1^{t-1} + \frac{1}{3}p_4^{t-1}$$

$$p_4^t = \frac{1}{2}p_5^{t-1}$$

$$p_5^t = p_2^{t-1}$$



Stationary distribution

- The stationary distribution of a random walk with transition matrix P, is a probability distribution π, such that $\pi = \pi P$
- The stationary distribution is an eigenvector of matrix P
  - the principal left eigenvector of P – stochastic matrices have maximum eigenvalue 1
- The probability π_i is the fraction of times that we visited state i as $t \rightarrow \infty$
- Markov Chain Theory: The random walk converges to a unique stationary distribution independent of the initial vector if the graph is strongly connected, and not bipartite.

Computing the stationary distribution

- The Power Method

> Initialize q^0 to some distribution
> Repeat
> $q^t = q^{t-1}P$
> Until convergence

- After many iterations $q^t \rightarrow \pi$ regardless of the initial vector q^0
- Power method because it computes $q^t = q^0 P^t$

- Rate of convergence
  - determined by the second eigenvalue $\lambda_2{}^t$

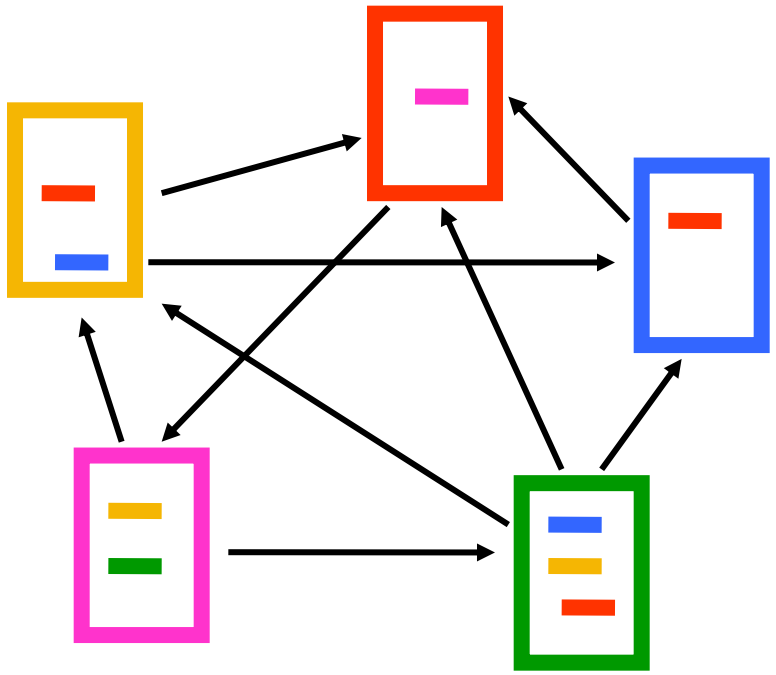
The stationary distribution

- What is the meaning of the stationary distribution π of a random walk?
- $\pi(i)$: the probability of being at node i after very large (infinite) number of steps
- $\pi = p_0 P^\infty$, where P is the transition matrix, p_0 the original vector
  - $P(i,j)$: probability of going from i to j in one step
  - $P^2(i,j)$: probability of going from i to j in two steps (probability of all paths of length 2)
  - $P^\infty(i,j) = \pi(j)$: probability of going from i to j in infinite steps – starting point does not matter.

The PageRank random walk

- Vanilla random walk
  - make the adjacency matrix stochastic and run a random walk

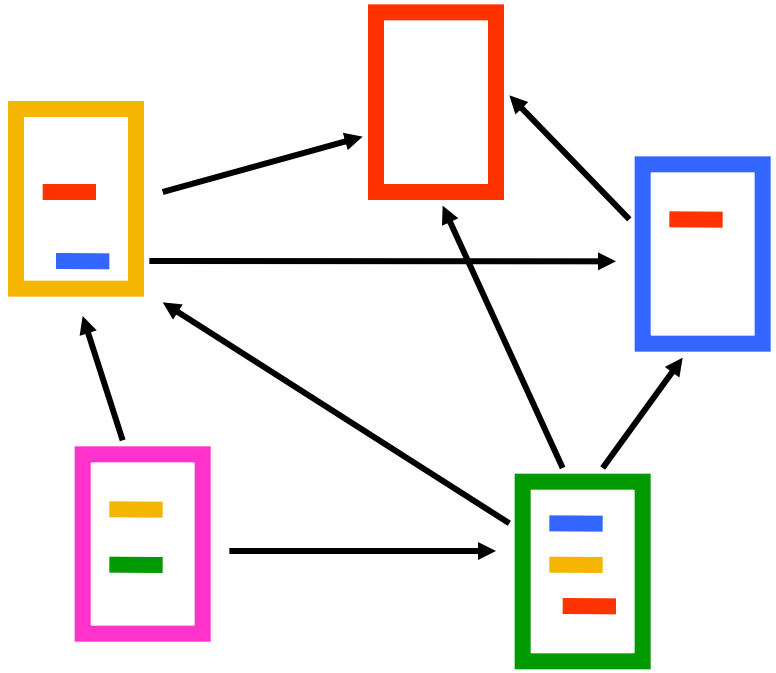
$$P = \begin{bmatrix} 0 & 1/2 & 1/2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 1/3 & 1/3 & 1/3 & 0 & 0 \\ 1/2 & 0 & 0 & 1/2 & 0 \end{bmatrix}$$



The PageRank random walk

- What about sink nodes?
  - what happens when the random walk moves to a node without any outgoing inks?

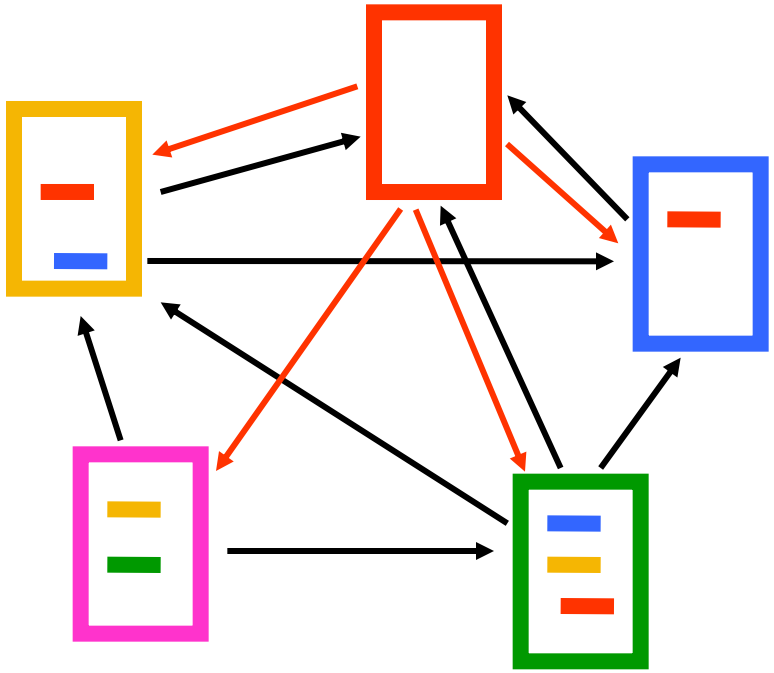
$$P = \begin{bmatrix} 0 & 1/2 & 1/2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1/3 & 1/3 & 1/3 & 0 & 0 \\ 1/2 & 0 & 0 & 1/2 & 0 \end{bmatrix}$$



The PageRank random walk

- Replace these row vectors with a vector v
  - typically, the uniform vector

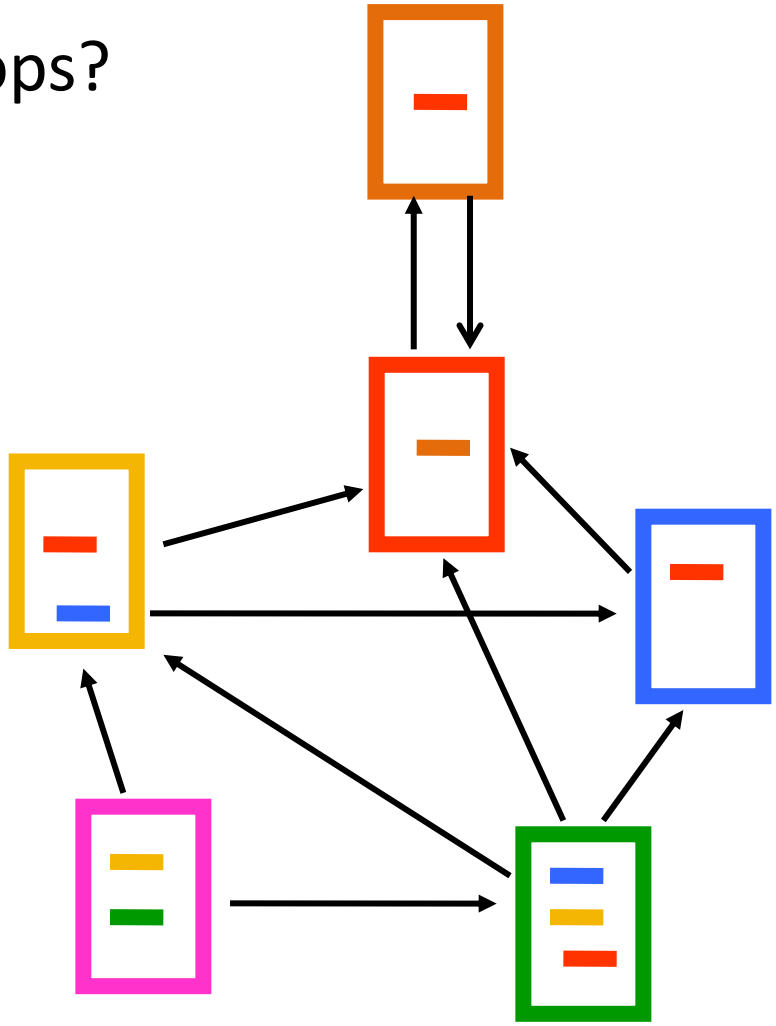
$$P' = \begin{bmatrix} 0 & 1/2 & 1/2 & 0 & 0 \\ 1/5 & 1/5 & 1/5 & 1/5 & 1/5 \\ 0 & 1 & 0 & 0 & 0 \\ 1/3 & 1/3 & 1/3 & 0 & 0 \\ 1/2 & 0 & 0 & 1/2 & 0 \end{bmatrix}$$



$$P' = P + dv^T \qquad d = \begin{cases} 1 & \text{if } i \text{ is sink} \\ 0 & \text{otherwise} \end{cases}$$

The PageRank random walk

- What about loops?
  - Spider traps



The PageRank random walk

- Add a random jump to vector v with prob 1-α
  - typically, to a uniform vector
- Restarts after 1/(1-α) steps in expectation
  - Guarantees irreducibility, convergence

$$P'' = \alpha \begin{bmatrix} 0 & 1/2 & 1/2 & 0 & 0 \\ 1/5 & 1/5 & 1/5 & 1/5 & 1/5 \\ 0 & 1 & 0 & 0 & 0 \\ 1/3 & 1/3 & 1/3 & 0 & 0 \\ 1/2 & 0 & 0 & 0 & 1/2 \end{bmatrix} + (1-\alpha) \begin{bmatrix} 1/5 & 1/5 & 1/5 & 1/5 & 1/5 \\ 1/5 & 1/5 & 1/5 & 1/5 & 1/5 \\ 1/5 & 1/5 & 1/5 & 1/5 & 1/5 \\ 1/5 & 1/5 & 1/5 & 1/5 & 1/5 \\ 1/5 & 1/5 & 1/5 & 1/5 & 1/5 \end{bmatrix}$$

$P'' = \alpha P' + (1-\alpha)uv^T$, where u is the vector of all 1s

Random walk with restarts

PageRank: Spectral Analysis

- **PageRank as a principal eigenvector**

  $r = M \cdot r$ or equivalently $r_j = \sum_i \frac{r_i}{d_i}$

- **But we really want:**

  $$r_j = \beta \sum_i \frac{r_i}{d_i} + (1-\beta)\frac{1}{n}$$

- **Let's define:**

  $$M'_{ij} = \beta M_{ij} + (1-\beta)\frac{1}{n}$$

- **Now we get what we want:**

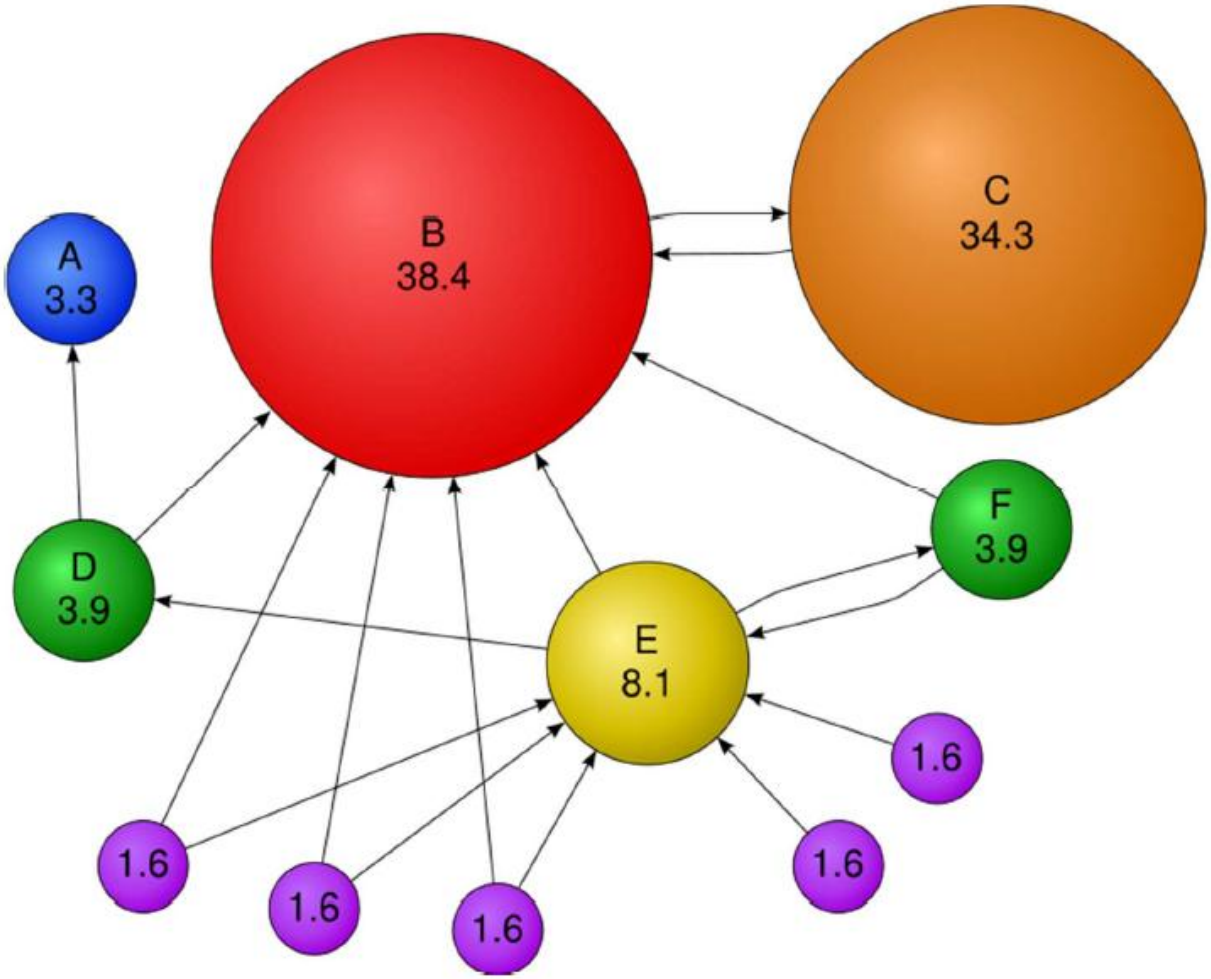
  $\boldsymbol{r = M' \cdot r}$

- **What is $1-\beta$?**
  - In practice 0.15 (5 links and jump)

d_i ... out-degree of node i

Note: M is a sparse matrix but M' is dense (all entries $\neq 0$). In practice we never "materialize" M but rather we use the "sum" formulation

PageRank: Example



Personalized PageRank

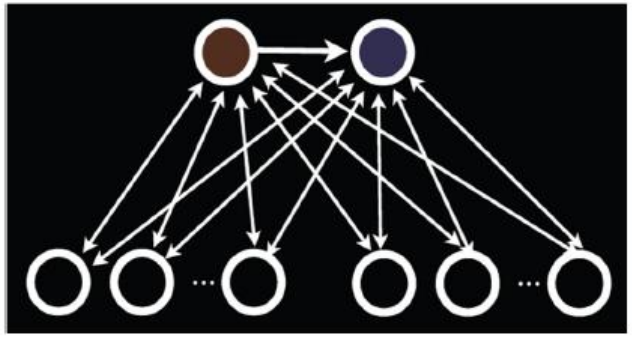
- **Goal:** Evaluate pages not just by popularity but by how close they are to the topic
- Teleporting can go to:
  - Any page with equal probability
    - (we used this so far)
  - A topic-specific set of "relevant" pages
    - Topic-specific (personalized) PageRank (S …teleport set)

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

    - Useful for measuring "proximity" of other nodes to S

PageRank: Trust Rank

- **Link Farms:** networks of millions of pages design to focus PageRank on a few undeserving webpages
- To minimize their influence use a teleport set of trusted webpages
  - E.g., homepages of universities





Pagerank summary

- Preprocessing:
  - Given graph of links, build matrix **P**.
  - From it compute **a** – left eigenvector of **P**.
  - The entry a_i is a number between 0 and 1: the pagerank of page *i*.
- Query processing:
  - Retrieve pages meeting query.
  - Rank them by their pagerank.
  - But this rank order is query-*independent*

The reality

- Pagerank is used in google and other engines, but is hardly the full story of ranking
  - Many sophisticated features are used
  - Some address specific query classes
  - Machine learned ranking heavily used
- Pagerank still very useful for things like crawl policy

Google's official description of PageRank

PageRank reflects our view of the importance of web pages by considering more than 500 million variables and 2 billion terms. Pages that believe are important pages receive a higher PageRank and are more likely to appear at the top of the search results.

HITS

HITS

Την ίδια εποχή με το PageRank

Δύο βασικές διαφορές

- Κάθε σελίδα έχει δύο βαθμούς:
  - ένα βαθμό κύρους (authority rank) και
  - ένα κομβικό βαθμό (hub rank)

- Οι βαθμοί είναι θεματικοί

HITS

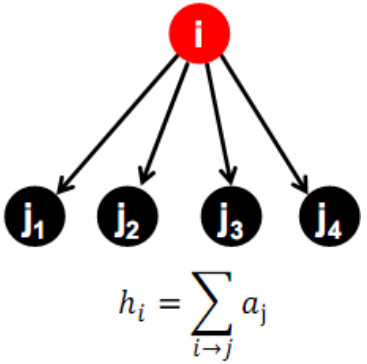
- Authorities: pages containing useful information (the prominent, highly endorsed answers to the queries)
  - Newspaper home pages
  - Course home pages
  - Home pages of auto manufacturers
- Hubs: pages that link to authorities (highly value lists)
  - List of newspapers
  - Course bulletin
  - List of US auto manufacturers

✓ A good hub links to many good authorities
✓ A good authority is linked from many good hubs

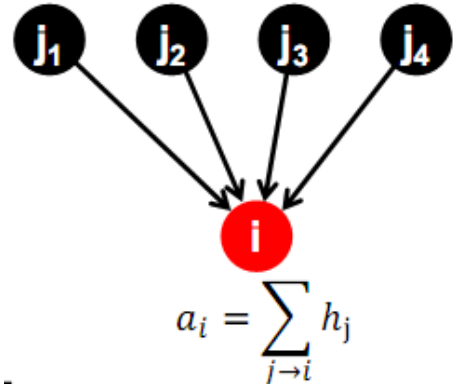
HITS: Algorithm

Each page p, has two scores

- A **hub score** (h) quality as an expert
  - Total sum of authority scores that it points to

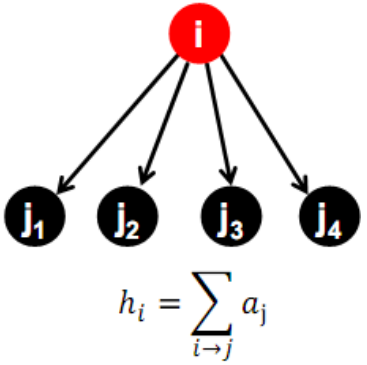


- An **authority score** (a) quality as content
  - Total sum of hub scores that point to it

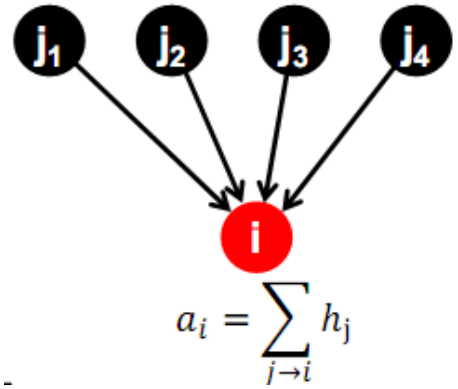


HITS: Algorithm

Authority Update Rule: For each page i, update a(i) to be the sum of the hub scores of all pages that point to it.



Hub Update Rule: For each page i, update h(i) to be the sum of the authority scores of all pages that it points to.



HITS: Algorithm

- Start with all hub scores and all authority scores equal to 1.
- Perform a sequence of k hub-authority updates. For each node:

  - *First, apply the Hub Update Rule* to the current set of scores.
  - *Then, apply the Authoroty Update Rule* to the resulting set of scores.

  - At the end, hub and authority scores may be very large.

  Normalize: divide each authority score by the sum of all authority scores, and each hub score by the sum of all hub scores.

High-level scheme

- Extract from the web a base set of pages that *could* be good hubs or authorities.
- From these, identify a small set of top hub and authority pages;

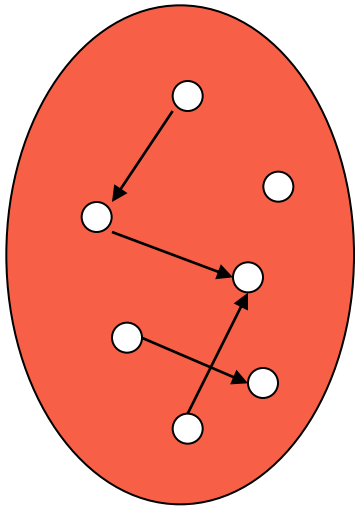
  →iterative algorithm.

Base set

- Given text query (say ***browser***), use a text index to get all pages containing ***browser.***
  - Call this the root set of pages.
- Add in any page that either
  - points to a page in the root set, or
  - is pointed to by a page in the root set.
- Call this the base set.

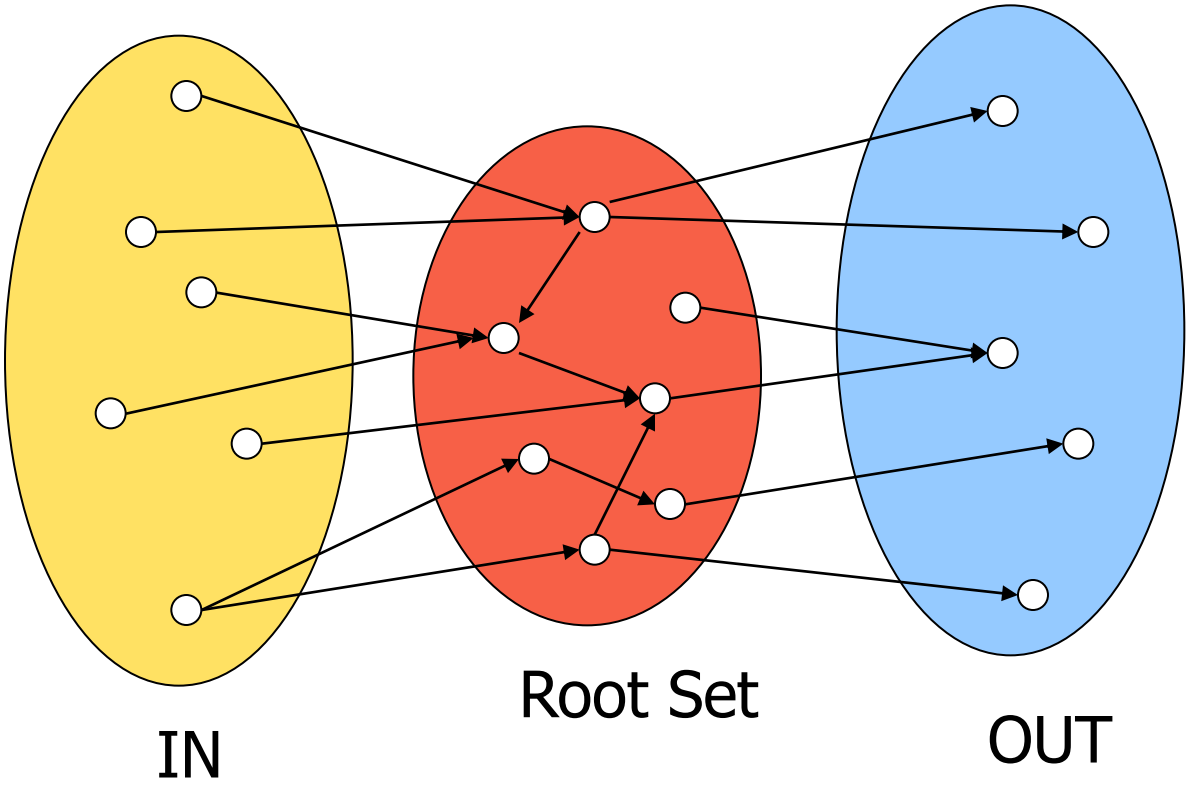
Query dependent input

Root set obtained from a text-only search engine

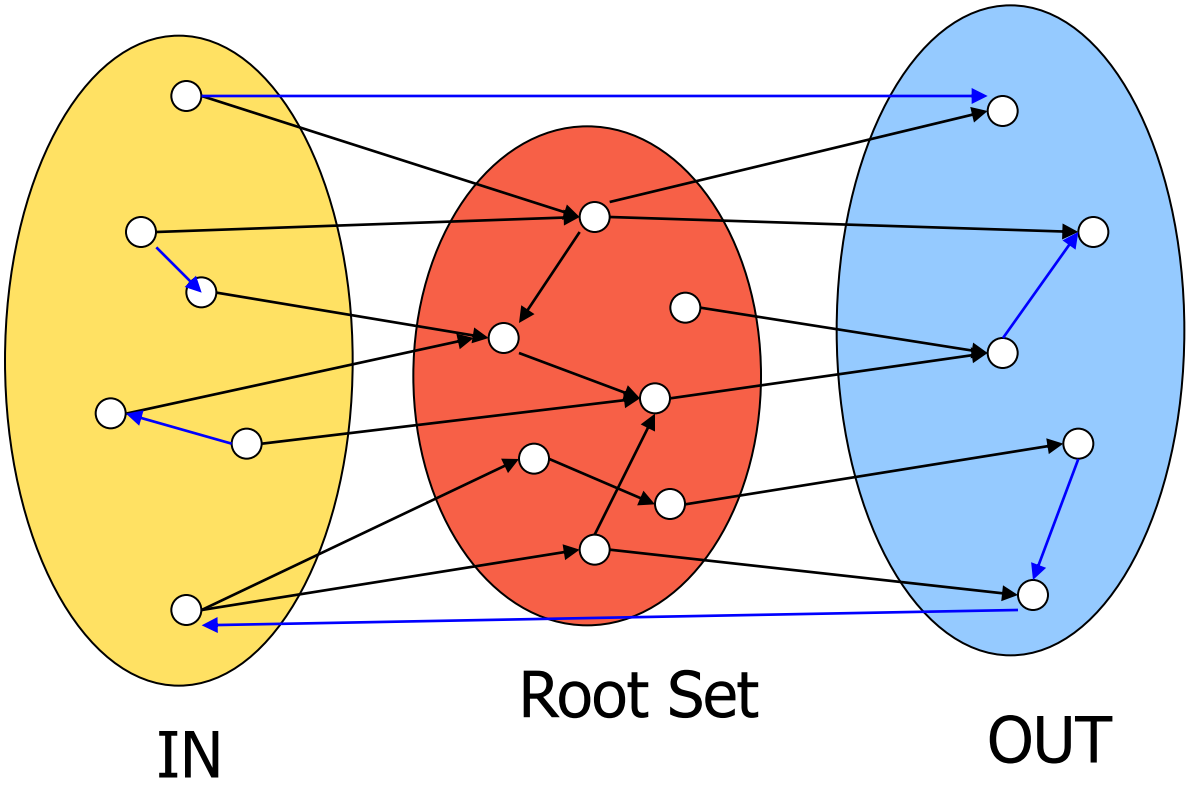


Root Set

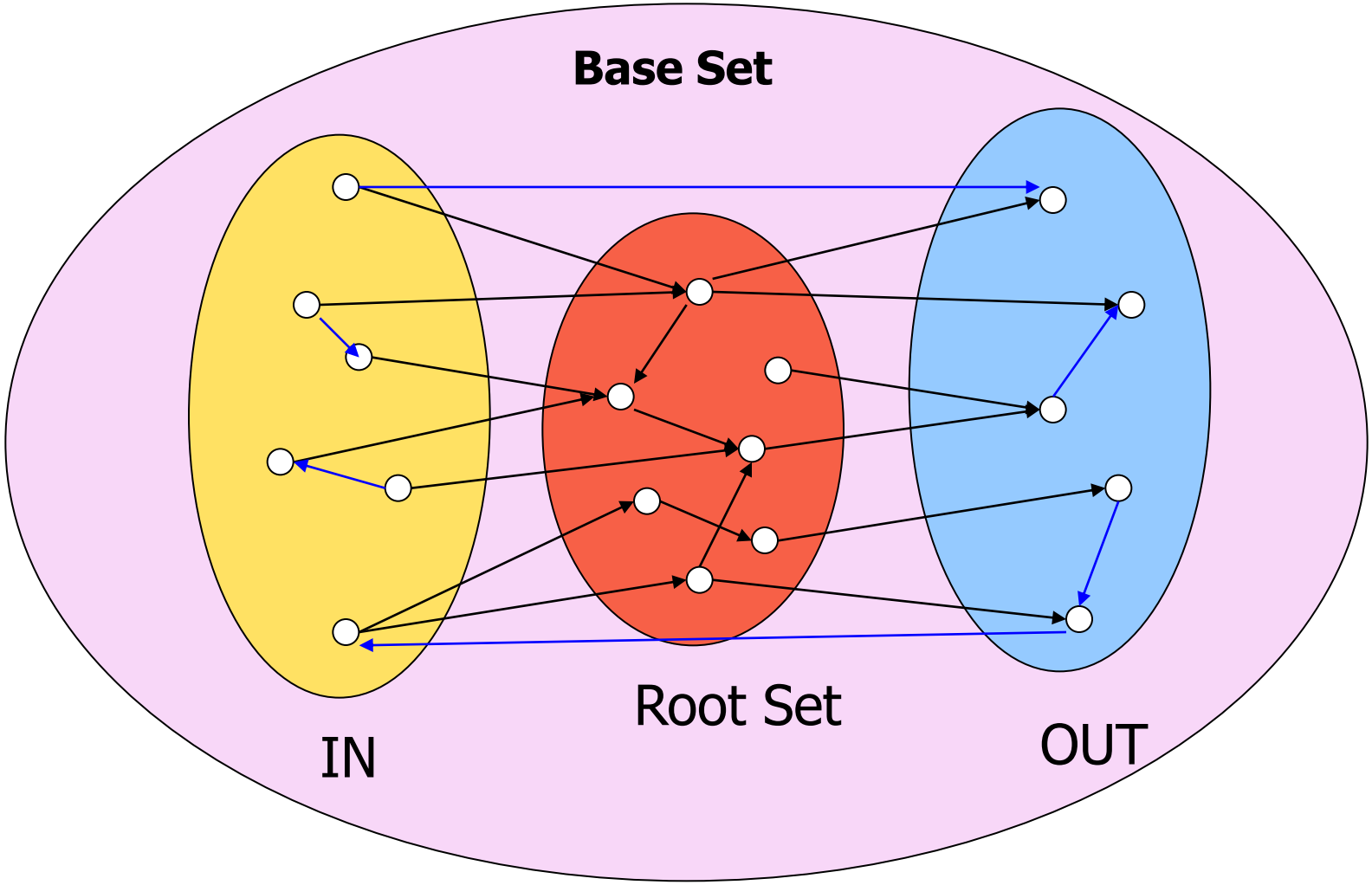
Query dependent input



Query dependent input



Query dependent input



Distilling hubs and authorities

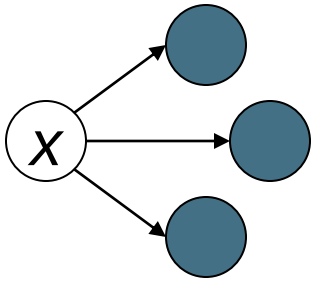
- Compute, for each page *x* in the base set, a hub score *h(x)* and an authority score *a(x).*
- Initialize: for all *x*, *h(x)←1; a(x) ←1;*
- Iteratively update all *h(x), a(x)*; ← Key
- After iterations
  - output pages with highest *h()* scores as top hubs
  - highest *a()* scores as top authorities.

Iterative update

- Repeat the following updates, for all x:

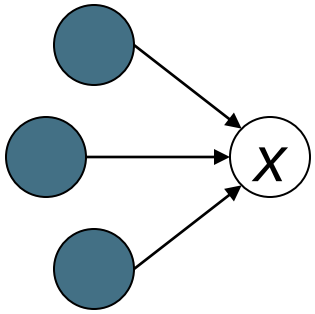
I operation

$$h(x) \leftarrow \sum_{x \mapsto y} a(y)$$



O operation

$$a(x) \leftarrow \sum_{y \mapsto x} h(y)$$

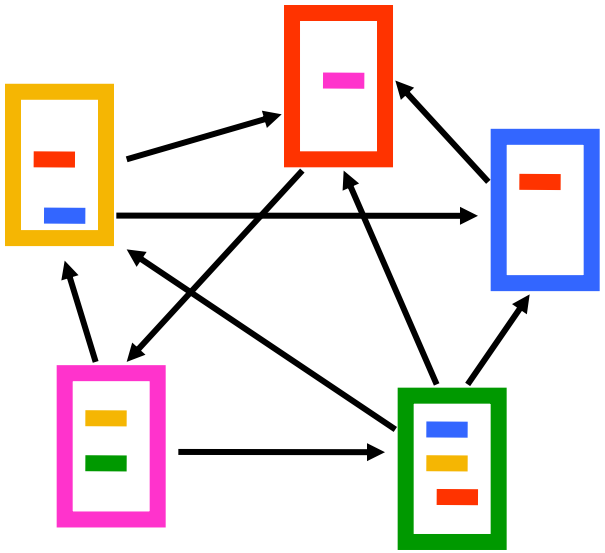


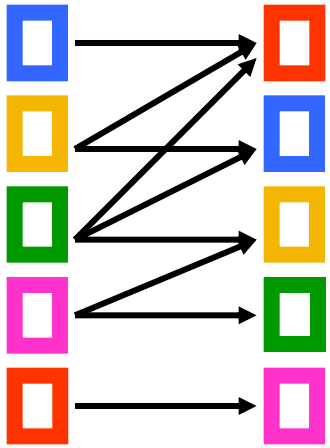
Normalize

Scaling

- To prevent the *h()* and *a()* values from getting too big, can scale down after each iteration.
- Scaling factor doesn't really matter:
  - we only care about the *relative* values of the scores.

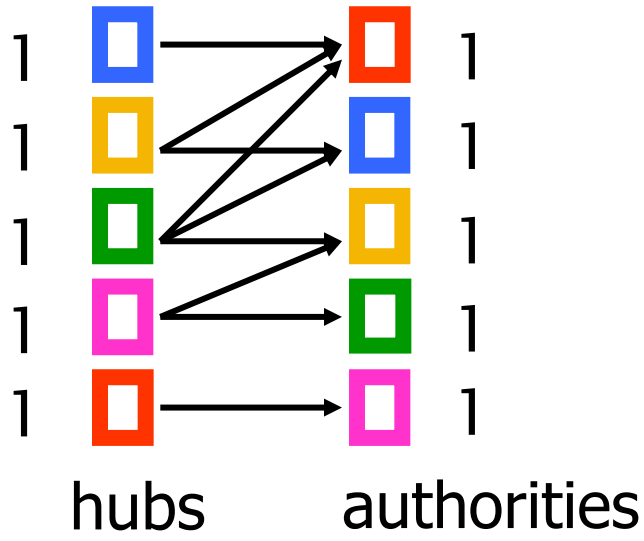
Example





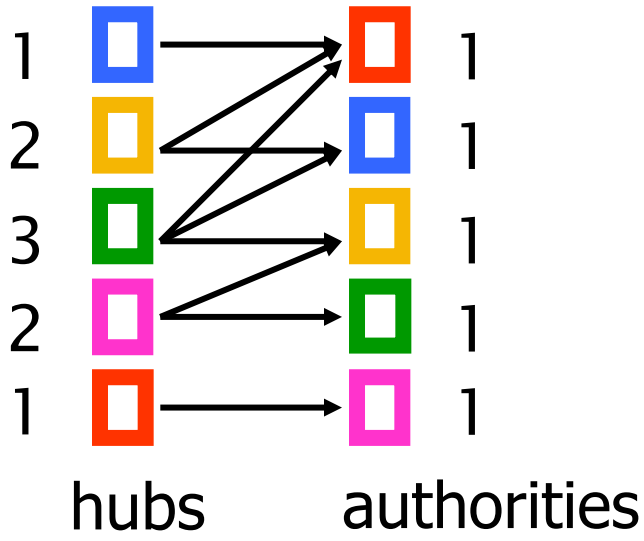
Example

Initialize



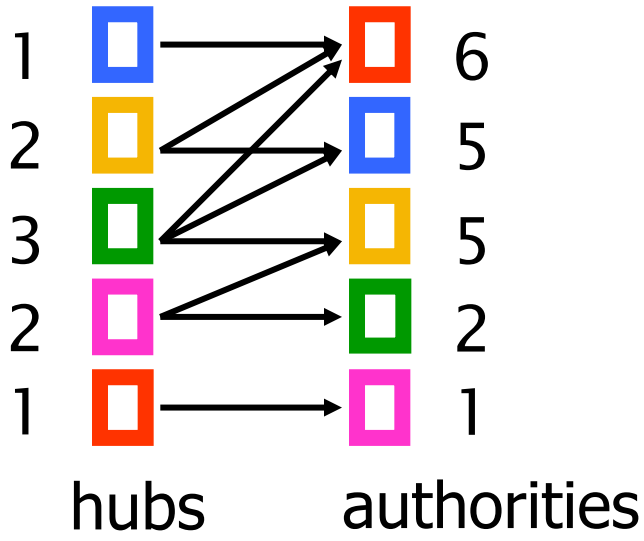
Example

Step 1: O operation



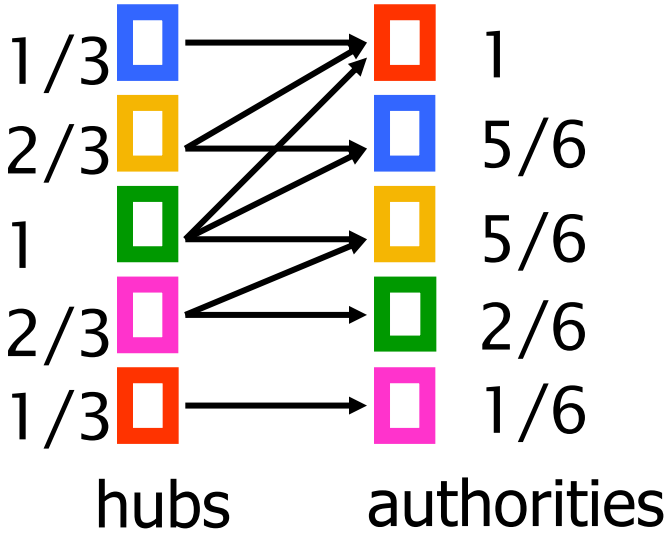
Example

Step 1: I operation



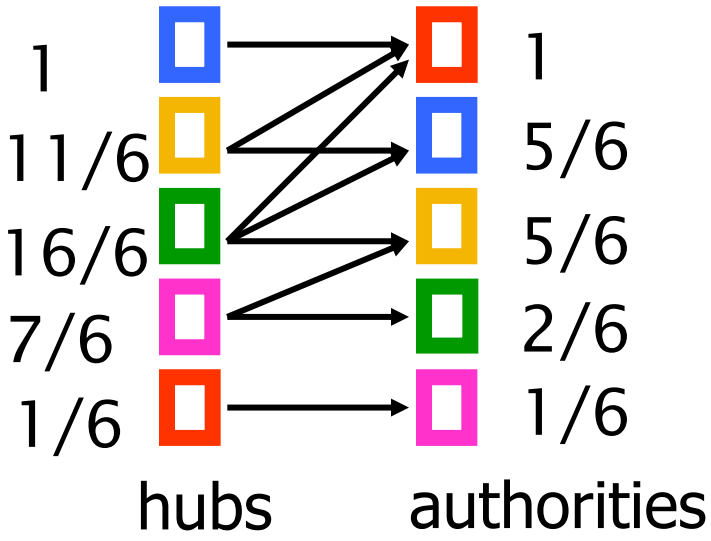
Example

Step 1: Normalization (Max norm)



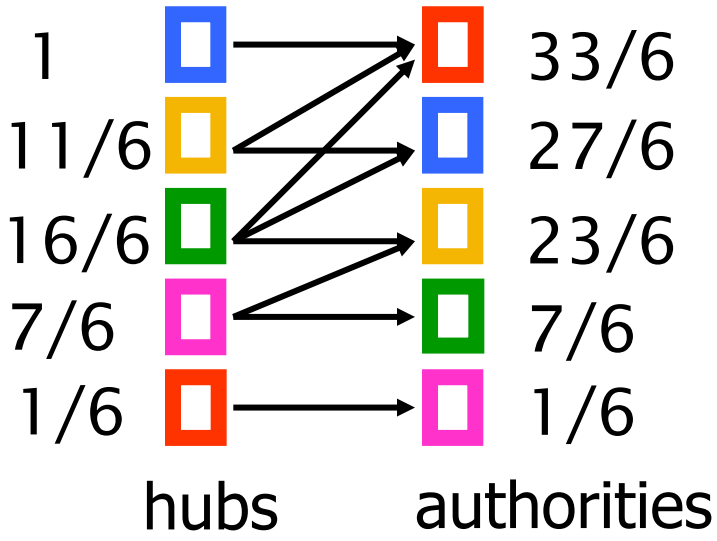
Example

Step 2: O step



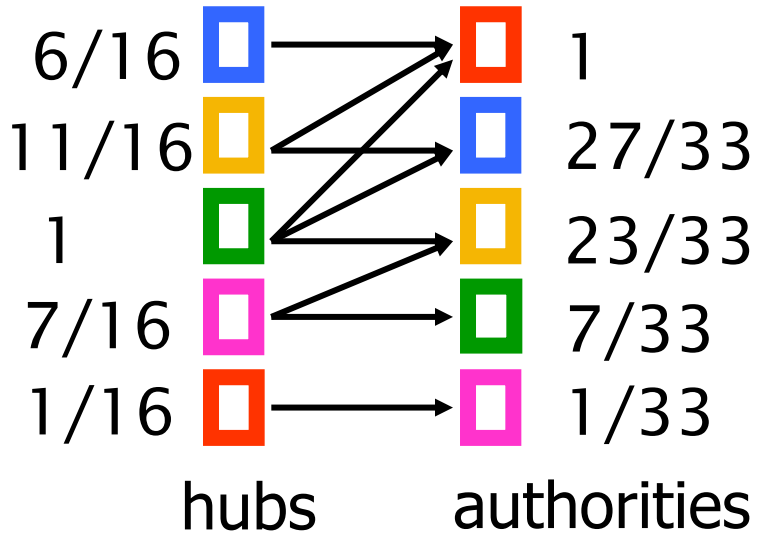
Example

Step 2: I step



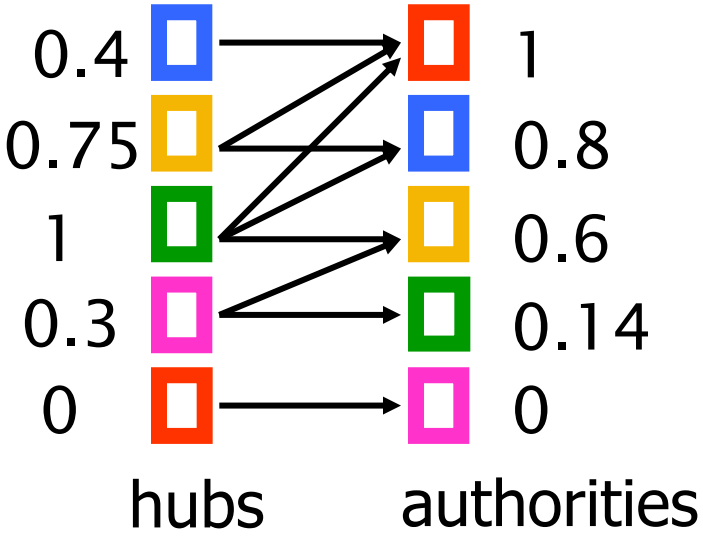
Example

Step 2: Normalization



Example

Convergence



How many iterations?

- Claim: relative values of scores will converge after a few iterations:
  - in fact, suitably scaled, *h()* and *a()* scores settle into a steady state!

- In practice, ~5 iterations get you close to stability.

Japan Elementary Schools

Hubs

- schools
- LINK Page-13
- “ú–{‚ÌŠwZ
- a‰„¬ŠwZƒz[ƒƒy[ƒW
- 100 Schools Home Pages (English)
- K-12 from Japan 10/...rnet and Education)
- http://www...iglobe.ne.jp/~IKESAN
- ‚l‚f‚j¬ŠwZ‚U”N‚P‘g•¨Œê
- ÒŠ—’¬— § ÒŠ—“Œ¬ŠwZ
- Koulutus ja oppilaitokset
- TOYODA HOMEPAGE
- Education
- Cay's Homepage(Japanese)
- –y“ì¬ŠwZ‚Ìƒz[ƒƒy[ƒW
- UNIVERSITY
- ‰J—³¬ŠwZ DRAGON97-TOP
- Â‰ª¬ŠwZ‚T”N‚P‘gƒz[ƒƒy[ƒW
- ¶µ° é¼ÂÁ© ¥á¥Ë¥å¡¼ ¥á¥Ë¥å¡¼

Authorities

- The American School in Japan
- The Link Page
- ‰ªèš— § ˆä“c¬ŠwZƒz[ƒƒy[ƒW
- Kids' Space
- ˆÀéš— § ˆÀé¼•”¬ŠwZ
- ‹{é‹³ˆç‘åŠw•‘®¬ŠwZ
- KEIMEI GAKUEN Home Page (Japanese)
- Shiranuma Home Page
- fuzoku-es.fukui-u.ac.jp
- welcome to Miasa E&J school
- _“ÞìŒ § E‰¡•ls— § ’†ì¼¬ŠwZ‚Ìƒy
- http://www...p/~m_maru/index.html
- fukui haruyama-es HomePage
- Torisu primary school
- goo
- Yakumo Elementary,Hokkaido,Japan
- FUZOKU Home Page
- Kamishibun Elementary School...

Things to note

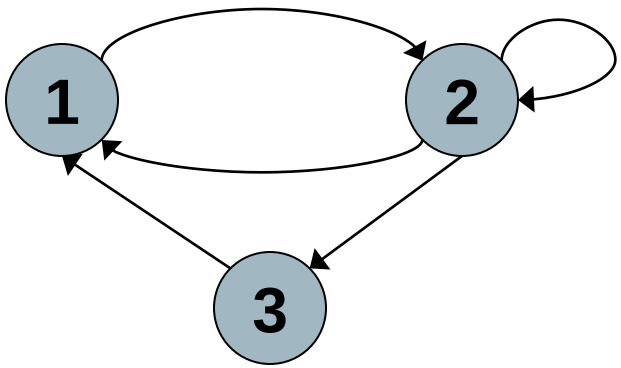
- Pulled together good pages regardless of *language* of *page content*.
- Use *only* link analysis after base set assembled
  - iterative scoring is query-independent.
- Iterative computation after text index retrieval - significant overhead.

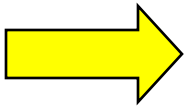
Issues

- Topic Drift
  - Off-topic pages can cause off-topic "authorities" to be returned
    - E.g., the neighborhood graph can be about a "super topic"
- Mutually Reinforcing Affiliates
  - Affiliated pages/sites can boost each others' scores
    - Linkage between affiliated pages is not a useful signal

Πίνακας γειτνίασης

- *n*×*n* adjacency matrix **A**:
  - each of the *n* pages in the base set has a row and column in the matrix.
  - Entry $A_{ij} = 1$ if page *i* links to page *j*, else = 0.





	1	2	3
1	0	1	0
2	1	1	1
3	1	0	0

Hub/authority vectors

- View the hub scores *h()* and the authority scores *a()* as vectors with *n* components.
- Recall the iterative updates

$$h(x) \leftarrow \sum_{x \mapsto y} a(y)$$

$$a(x) \leftarrow \sum_{y \mapsto x} h(y)$$

HITS: Διανυσματική Αναπαράσταση

- **HITS converges to a single stable point**
- **Notation:**
  - Vector $a = (a_1 \ldots, a_n), \quad h = (h_1 \ldots, h_n)$
  - Adjacency matrix $\boldsymbol{A}$ (n x n): $A_{ij} = 1$ if $i \rightarrow j$
- **Then** $h_i = \sum_{i \rightarrow j} a_j$
  can be rewriten as $h_i = \sum_j A_{ij} \cdot a_j$
- **So:** $\boldsymbol{h} = \boldsymbol{A} \cdot \boldsymbol{a}$
- **And likewise:** $\boldsymbol{a} = \boldsymbol{A}^T \cdot \boldsymbol{h}$

Rewrite in matrix form

- **h**=**Aa**.
- **a**=**A**^{t}**h**.

Recall A^t is the transpose of A.

Substituting, **h**=**AA**^{t}**h** and **a**=**A**^{t}**Aa**.

Thus, **h** is an eigenvector of **AA**^{t} and **a** is an eigenvector of **A**^{t}**A**.

Further, our algorithm is a particular, known algorithm for computing eigenvectors: the *power iteration* method.

Guaranteed to converge.

HITS: Spectral Analysis

- **Definition:**
  - Let $R \cdot x = \lambda \cdot x$
    for some scalar λ, vector x, matrix R
  - Then $\boldsymbol{x}$ **is an eigenvector**, and λ **is its eigenvalue**
- **Fact:**
  - If R is symmetric ($R_{ij} = R_{ji}$)
    (in our case $R = A^T \cdot A$ and $\mathrm{R} = A \cdot A^T$ are symmetric)
  - Then R has n orthogonal unit eigenvectors
    $w_1 \ldots w_n$ that form a basis (coordinate system) with
    eigenvalues $\lambda_1 \ldots \lambda_n \quad (|\lambda_i| \geq |\lambda_{i+1}|)$

HITS: Διανυσματική Αναπαράσταση

- **HITS algorithm in vector notation:**
  - Set: $a_i = h_i = \frac{1}{\sqrt{n}}$

  Repeat until convergence:
  - $h = A \cdot a$
  - $a = A^T \cdot h$
  - Normalize a and h

Convergence criterion:

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

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

- **Then:** $\boldsymbol{a = A^T \cdot (\underbrace{A \cdot a}_{\text{new } h})}$ (new a)
- **Thus, in $2k$ steps:**

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

a is updated (in 2 steps):

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

h is updated (in 2 steps):

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

Repeated matrix powering

Rewrite in matrix form

- The HITS algorithm is a power-method eigenvector computation
- In vector terms
  - $a^t = A^T h^{t-1}$ and $h^t = A a^{t-1}$
  - $a^t = A^T A a^{t-1}$ and $h^t = A A^T h^{t-1}$
  - Repeated iterations will converge to the eigenvectors
- The authority weight vector a is the eigenvector of $A^T A$ and the hub weight vector h is the eigenvector of $A A^T$
- The vectors a and h are called the singular vectors of the matrix A

Why does the Power Method work?

- If a matrix R is real and symmetric, it has real eigenvalues and eigenvectors: $(\lambda_1, w_1), (\lambda_2, w_2), \ldots, (\lambda_r, w_r)$
  - r is the rank of the matrix
  - $|\lambda_1| \geq |\lambda_2| \geq \cdots \geq |\lambda_r|$
- For any matrix R, the eigenvectors $w_1, w_2, \ldots, w_r$ of R define a basis of the vector space
  - For any vector x, $Rx = \alpha_1 w_1 + a_2 w_2 + \cdots + a_r w_r$
- After t multiplications we have:
  - $R^t x = \lambda_1^{t-1} \alpha_1 w_1 + \lambda_2^{t-1} a_2 w_2 + \cdots + \lambda_2^{t-1} a_r w_r$
- Normalizing leaves only the term w_1.

HITS: Spectral Analysis

$$Rx = \lambda x$$

- Let's write x in coordinate system $w_1 \dots w_n$

  $$x = \lambda_i \alpha_i w_i$$

  - x has coordinates $(\alpha_1, \dots, \alpha_n)$
- **Suppose:** $\lambda_1 \dots \lambda_n \quad (|\lambda_1| \geq \dots \geq |\lambda_n|)$
- **Then:** $R^k x = \lambda^k x = \sum_i \lambda_i^k \alpha_i w_i$
- **As** $k \to \infty$, if we normalize

  $$R^k x \to \lambda_1 \alpha_1 w_1$$

  (contribution of all other coordinates $\to 0$)

$$\lim_{k\to\infty} \frac{{\lambda_1}^k}{{\lambda_2}^k} = \left(\frac{\lambda_1}{\lambda_2}\right)^k \to \infty$$

- So, **authority** a is eigenvector of $\mathrm{R} = A^T A$ associated with largest eigenvalue λ_1
- Similarly: **hub** h is eigenvector of $\mathrm{R} = A A^T$

PageRank vs HITS

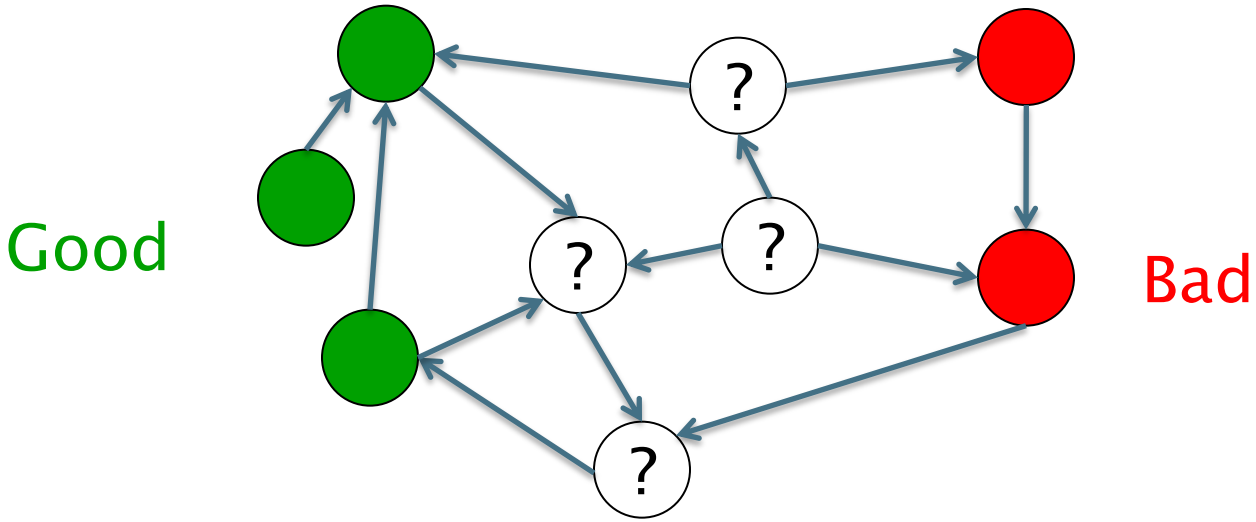
- PageRank can be *precomputed*, HITS has to be computed at *query time*.
  - HITS is too expensive in most application scenarios.
- PageRank and HITS two different design choices: (1) the eigenproblem formalization (2) the set of pages to apply the formalization. They are orthogonalQ
  - *We could also apply HITS to the entire web and PageRank to a small base set*.
- Claim: On the web, a good hub almost always is also a good authority.
  - Actual difference between PageRank and HITS ranking not as large

Περίληψη

- Anchor text: What exactly are links on the web and why are they important for IR?
- PageRank: the original algorithm that was used for link-based ranking on the web
- Hubs & Authorities: an alternative link-based ranking algorithm

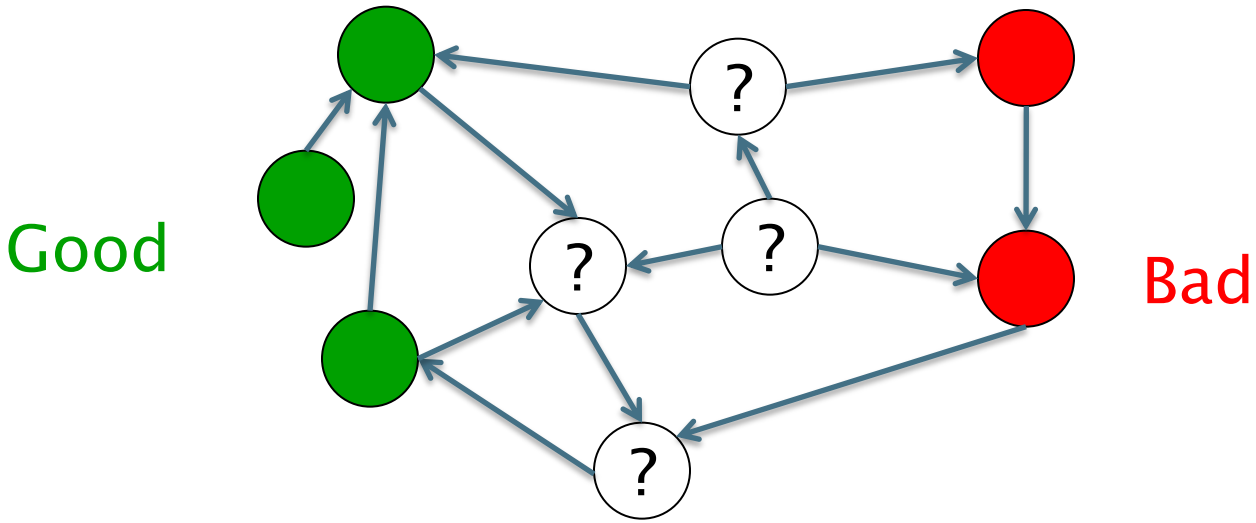
Οι σύνδεσμοι είναι παντού!

- Powerful sources of authenticity and authority
  - Mail spam – which email accounts are spammers?
  - Host quality – which hosts are “bad”?
  - Phone call logs
- The Good, The Bad and The Unknown



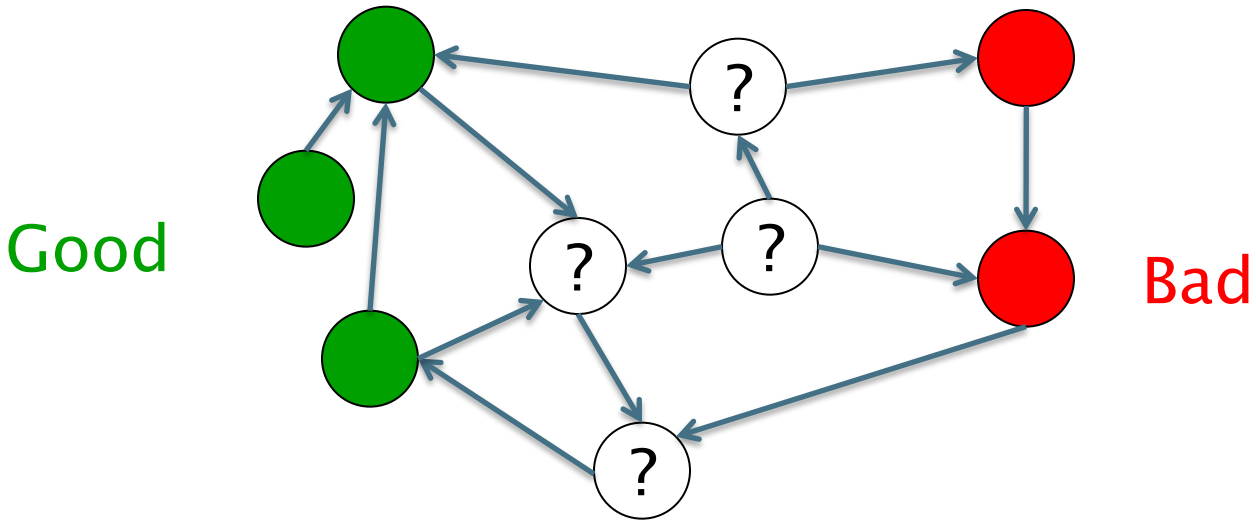
Simple iterative logic

- The Good, The Bad and The Unknown
  - Good nodes won't point to Bad nodes
  - All other combinations plausible



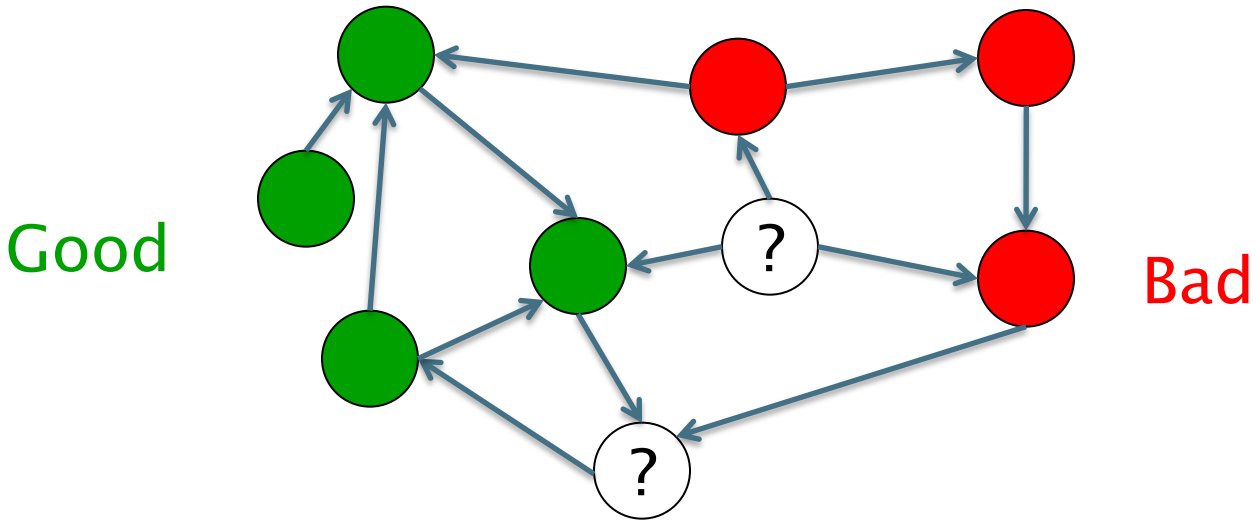
Simple iterative logic

- Good nodes won't point to Bad nodes
  - If you point to a Bad node, you're Bad
  - If a Good node points to you, you're Good



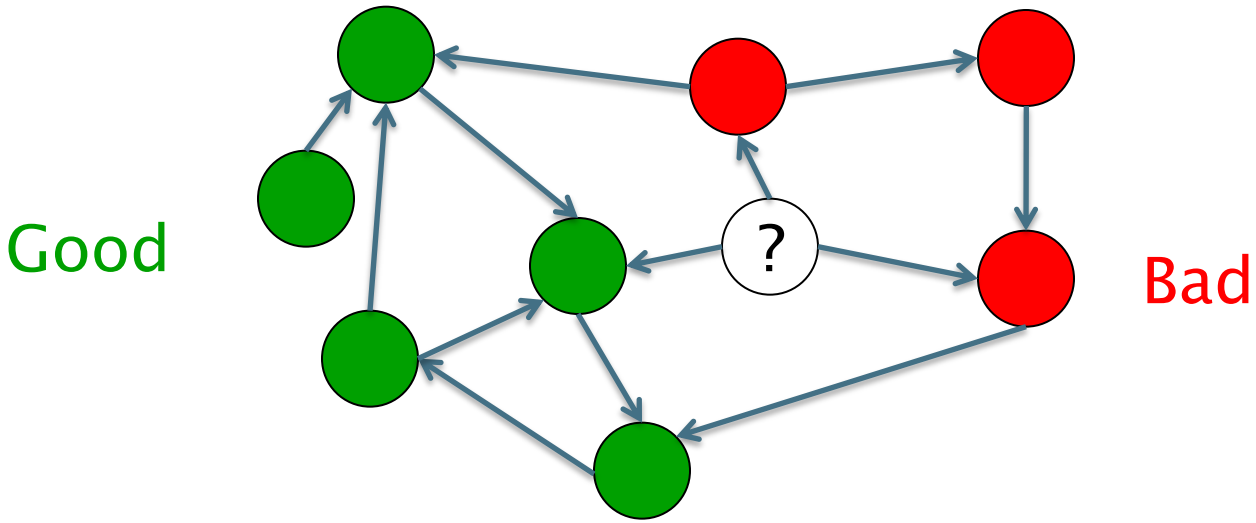
Simple iterative logic

- Good nodes won't point to Bad nodes
  - If you point to a Bad node, you're Bad
  - If a Good node points to you, you're Good



Simple iterative logic

- Good nodes won't point to Bad nodes
  - If you point to a Bad node, you're Bad
  - If a Good node points to you, you're Good



Many other examples of link analysis

- Social networks are a rich source of grouping behavior
- E.g., Shoppers' affinity – Goel+Goldstein 2010
  - Consumers whose friends spend a lot, spend a lot themselves
  - http://www.cs.cornell.edu/home/kleinber/networks-book/

Bibliometrics
e.g., citation analysis

ΤΕΛΟΣ 12ου Μαθήματος

Ερωτήσεις?

Χρησιμοποιήθηκε κάποιο υλικό από:
- ✓*Pandu Nayak and Prabhakar Raghavan, CS276:Information Retrieval and Web Search (Stanford)*
- ✓*Hinrich Schütze and Christina Lioma, Stuttgart IIR class*
- ✓ *Τις αντίστοιχες διαλέξεις του μεταπτυχιακού μαθήματος «Κοινωνικά Δίκτυα και Μέσα»*