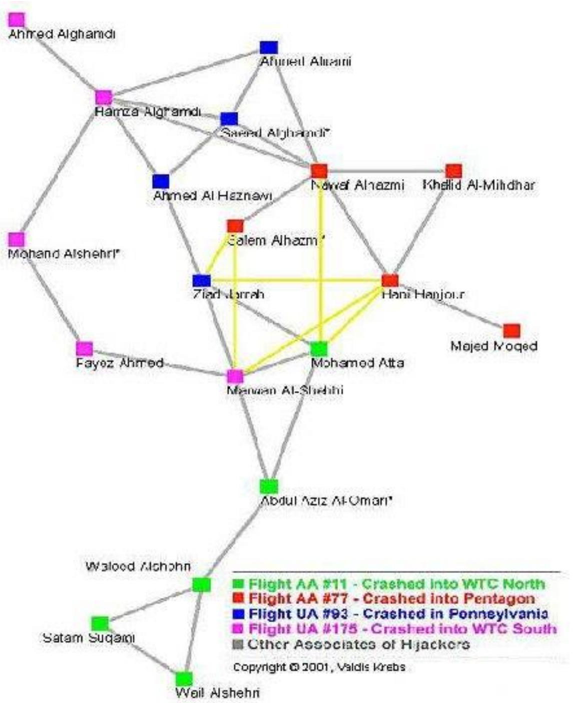
LEARNING FROM GRAPHS

Link analysis

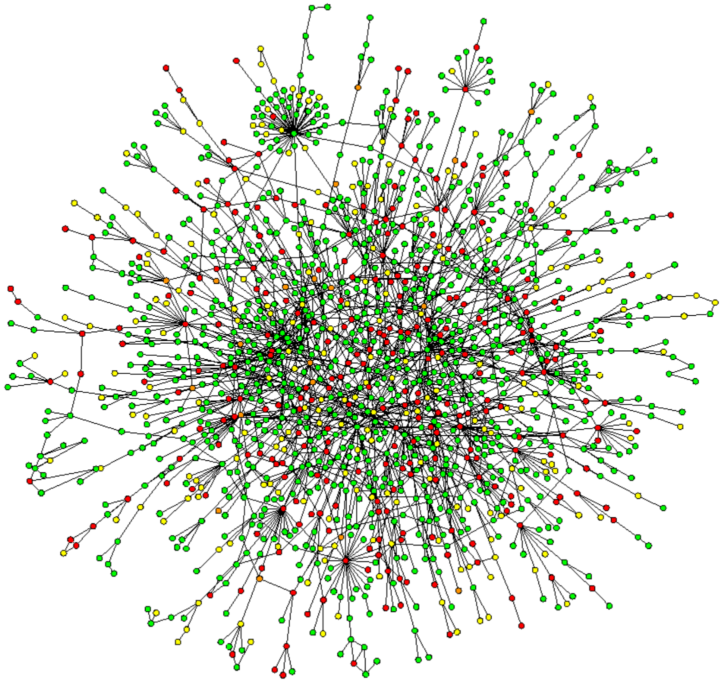
Motivation: who is most important?

- Goal: learn importance of graph nodes given only the structure of the graph



Copyright © 2001, Valdis Krebs

Terrorist network



Web graph

Theory: Random Walks on Graphs

- Random walk:
  - Start from a node chosen uniformly at random with probability $\frac{1}{n}$.
    - Pick one of the outgoing edges uniformly at random
    - Move to the destination of the edge
    - Repeat.

Random walk

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

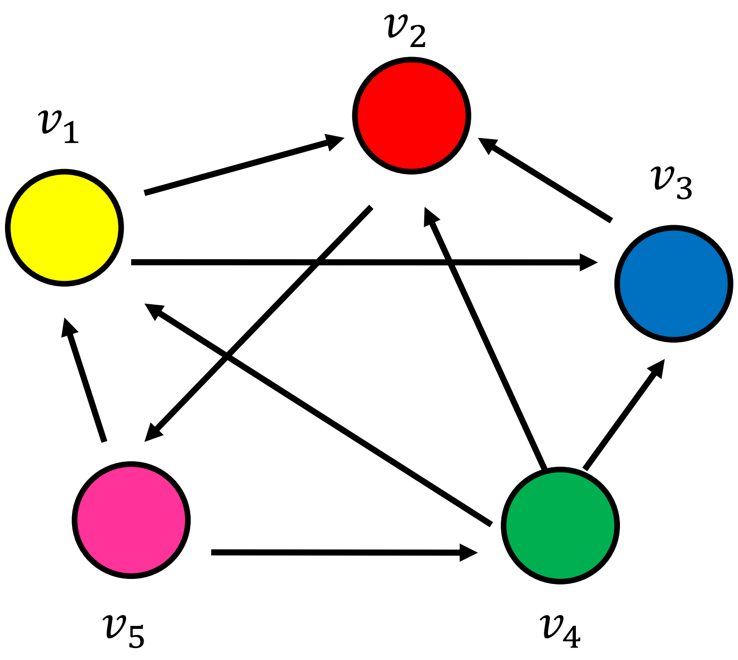
$$p_1^0 = \frac{1}{5} \qquad p_1^t = \frac{1}{3} p_4^{t-1} + \frac{1}{2} p_5^{t-1}$$

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

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

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

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



Stationary distribution

- After many-many steps ($t \rightarrow \infty$) the probabilities converge (updating the probabilities does not change the numbers)
- The converged probabilities define the stationary distribution of a random walk π
- 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.

Random walk with Restarts

- This is the random walk used by the PageRank algorithm
  - At every step with probability 1-α do a step of the random walk (follow a random link)
  - With probability α restart the random walk from a randomly selected node.
- The effect of the restart is that paths followed are never too long.
  - In expectation paths have length 1/α
- Restarts can also be from a specific node in the graph (always start the random walk from there)
- What is the effect of that?
  - The nodes that are close to the starting node have higher probability to be visited.

Why do we care?
Web Search is a huge IR system

- A Web crawler (robot) crawls the Web to collect all the pages.
- Servers establish a huge inverted indexing database and other indexing databases
- At query (search) time, search engines conduct different types of vector query matching.
  - There is an *Information Retrieval score* coming out of this.
- If many pages match the keyword search, we need to rank pages by a *reputation score*.

Google Page Ranking

“The Anatomy of a Large-Scale Hypertextual Web Search Engine”

by

Sergey Brin and Lawrence Page

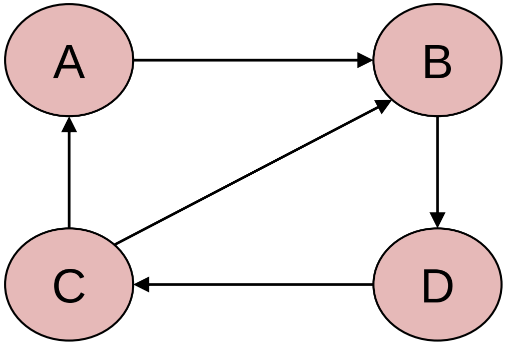
http://www-db.stanford.edu/~backrub/google.html

HOW TO EVALUATE A REPUTATION SCORE OF A NODE IN A GRAPH

Page Rank Algorithm

Ranking pages by Link Analysis: intuition

- Represent WEB pages by a directed graph
- Nodes are pages
- Edges are links
- To be clear: an arrow ending at a given page is a link into that page, and an arrow starting there is a link out to another webpage.



Ideas

- Idea 1: A webpage is important if it has many arrows pointing to it, i.e., many incoming links.

Why this is too naïve?

Ideas

- Idea 1: A webpage is important if it has many arrows pointing to it, i.e., many incoming links.

Why this is too naïve?

- Pages from any WEB site have links to the Home page, which will always be rated higher than individual pages

Ideas

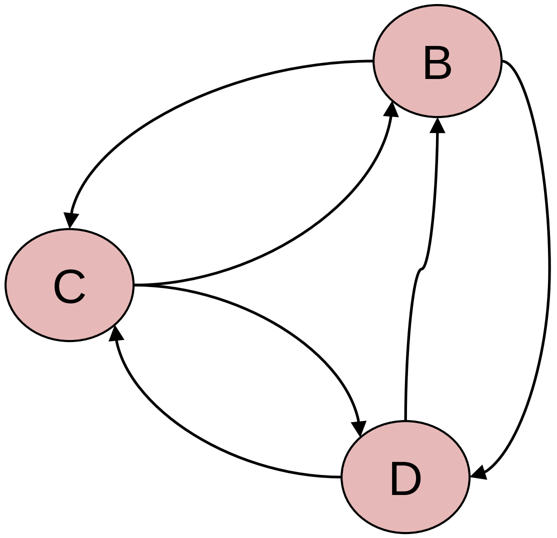
- Idea 2: a webpage is important if many important pages link to it.

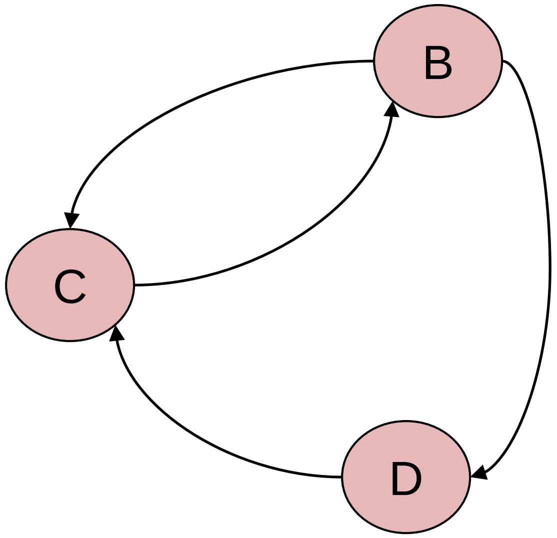
It seems that:

a problem now is the *self-referential* nature of this definition

if we follow this line of reasoning, we might find that the importance of a web page depends on itself.

Models of the WEB





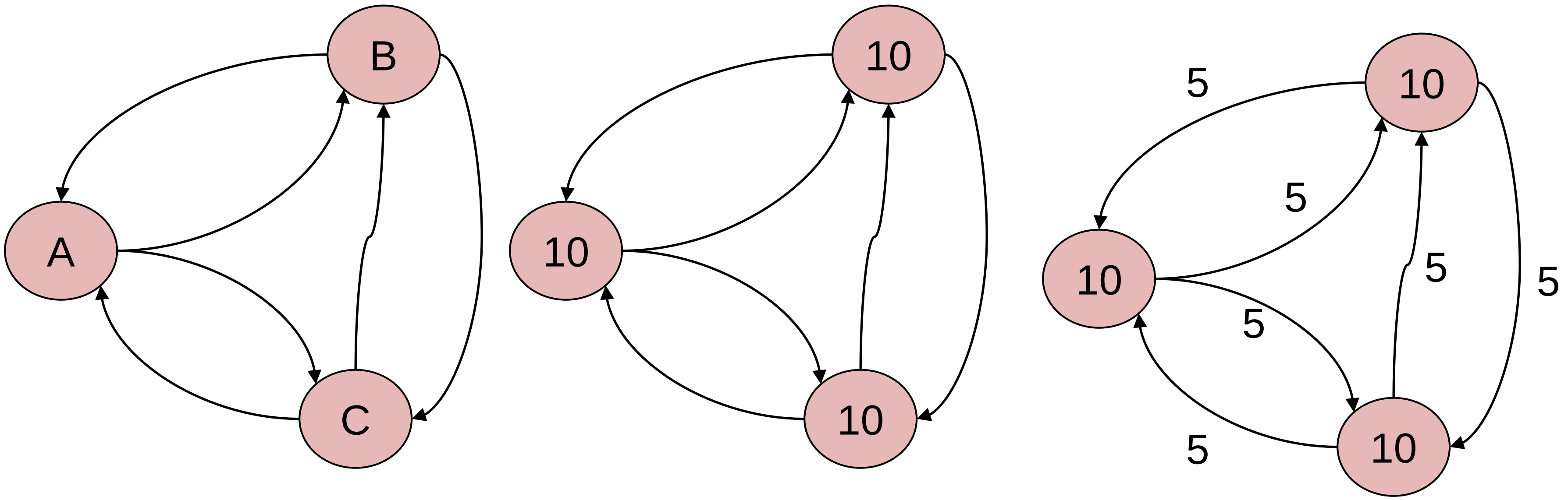
- What can we speculate about the relative importance of pages in each of these models, solely from the structure of the links (which is anyways the only information at hand)?

Traffic and mindless surfing.

- Assumptions:
  - The WEB site is important if it gets a lot of traffic.
  - Let us further assume that everyone is surfing spending a second on each page and then randomly following a link to a new page.
  - In this scheme it is convenient to make sure a surfer cannot get stuck, so we make the following STANDING ASSUMPTION:

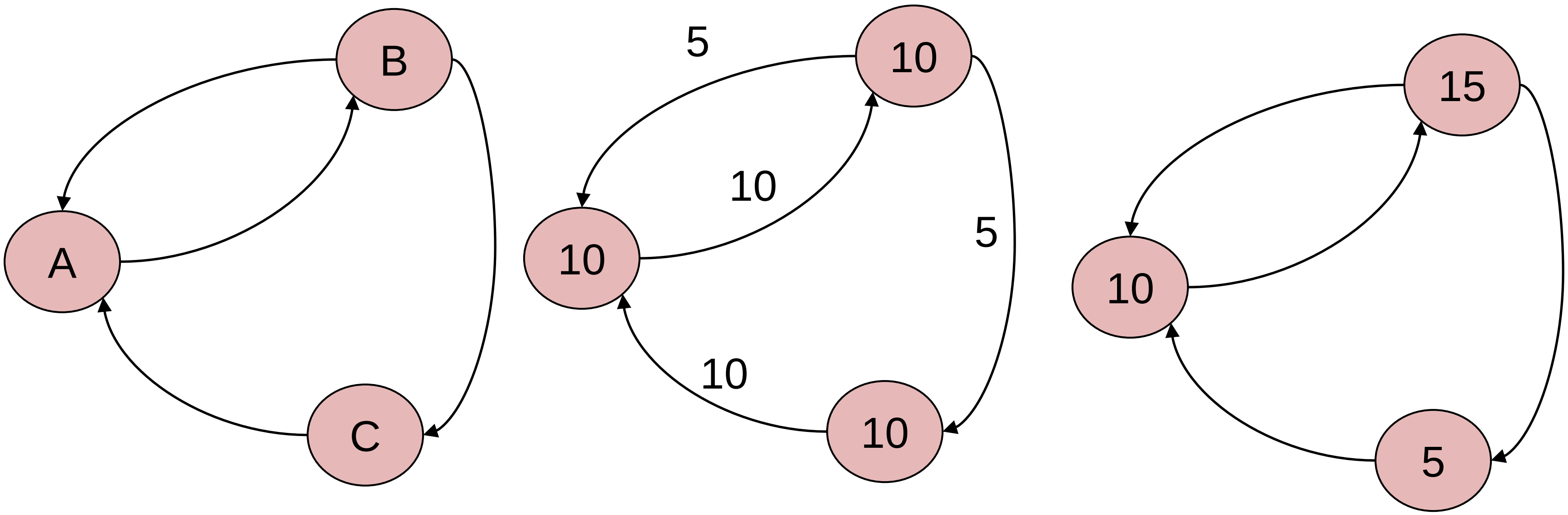
    Each page has at least one outgoing link.

Traffic and mindless surfing. Example 1



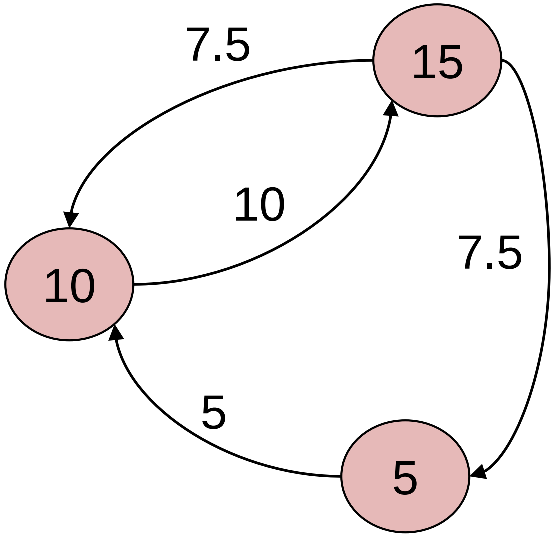
- We start with 10 surfers in each page
- At the first random click, 5 of the surfers at page A, say, go to page B, and the other 5 go to page C. So while each site sees all 10 of its visitors leave, it gets 5 + 5 incoming visitors to replace them: **So the amount of traffic at each page remains constant at 10.**

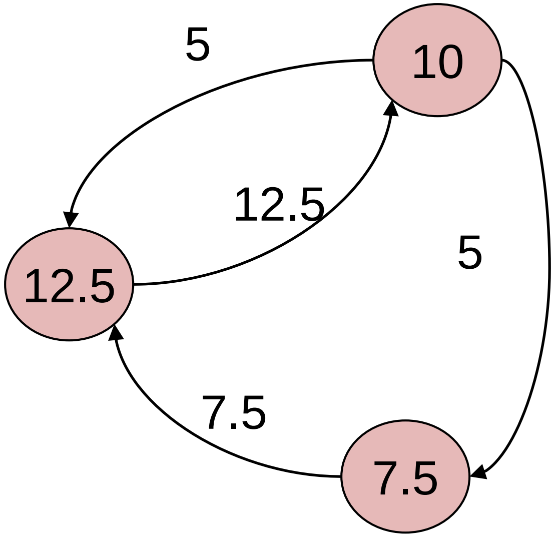
Traffic and mindless surfing. Example 2

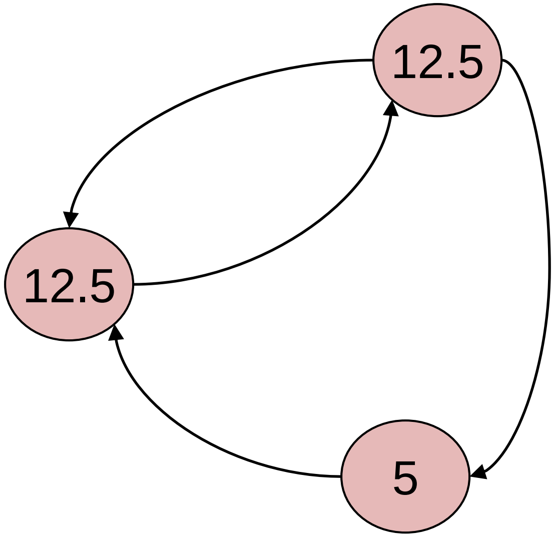


- We start with 10 surfers in each page
- After the first random click, 10 of the surfers at page A go to page B, since there is only 1 outgoing link from A etc…

Traffic and mindless surfing. Example 2

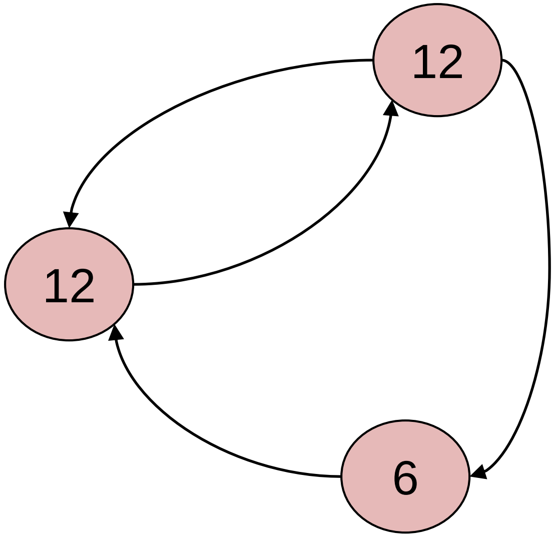






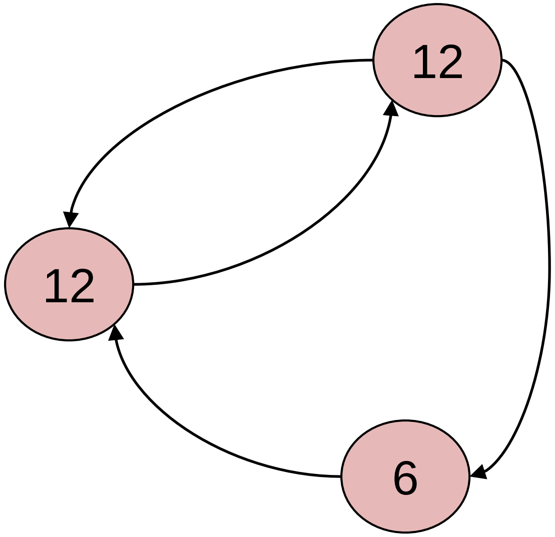
- After the two next clicks it becomes
- Where is this leading? Do we ever reach a stable configuration, as in the first model?

Traffic and mindless surfing. Example 2



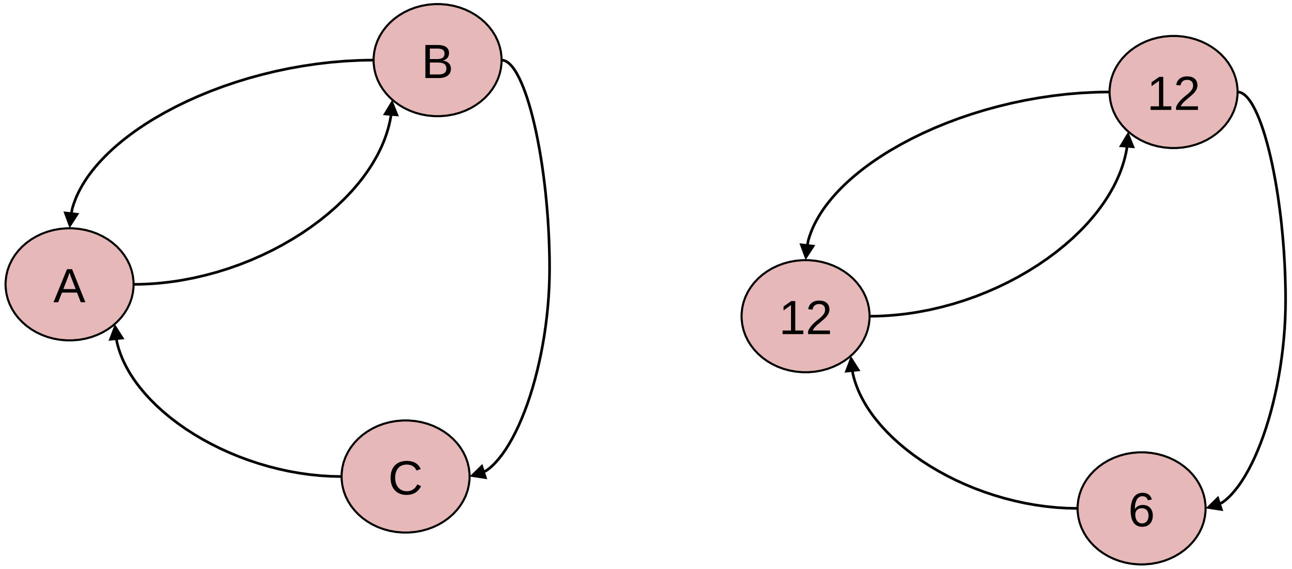
- While the answer is no, the process **converges** to the following distribution, which (you can check) remains the same going forward in time
- This precisely corresponds to *the random walk* on graphs and Markov chain property that the probability of the next step does not depend on the history of previous walks

Traffic and mindless surfing. Example 2



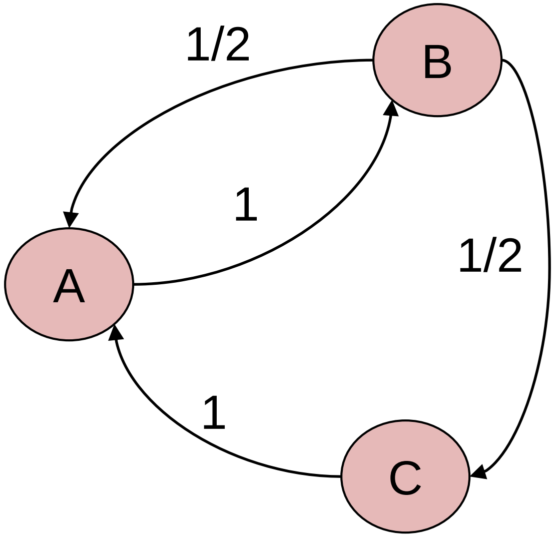
- This stable distribution is what the PageRank algorithm (in its most basic form) uses to assign a rank to each page:
  - The two pages with 12 visitors are equally important, and each more important than the remaining page having 6 visitors.

Traffic and mindless surfing.
Example 2



- How do we qualitatively explain why two of the pages in this model should be ranked equally, even though one has more incoming links than the other?

How to compute the stable distribution?



- Initiate matrix of probabilities to go from the current page to all other pages
- This gives the number of random surfers ending at each page
- We assume that this number indicates the importance of each page at time *i*

	A	B	C
A	0	1/2	1
B	1	0	0
C	0	1/2	0

To simultaneously re-evaluate page rank in each iteration *i*

Importance distribution from the previous iteration

Rank of the page in the previous iteration

$$\begin{pmatrix} R_{i+1}(A) \\ R_{i+1}(B) \\ R_{i+1}(C) \end{pmatrix} = \begin{pmatrix} 0 & 1/2 & 1 \\ 1 & 0 & 0 \\ 0 & 1/2 & 0 \end{pmatrix} * \begin{pmatrix} R_i(A) \\ R_i(B) \\ R_i(C) \end{pmatrix}$$

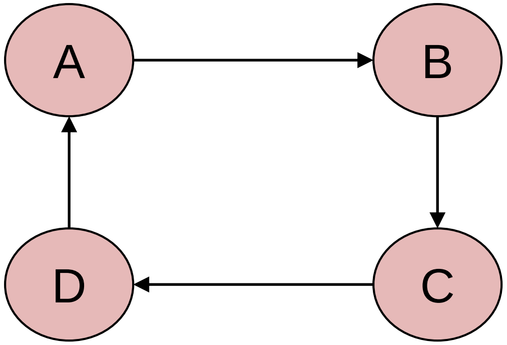
- All boils down to a sequence of matrix-vector multiplications
- *Map-reduce* was invented to do this fast in parallel

Coding is easy

- Code repository:

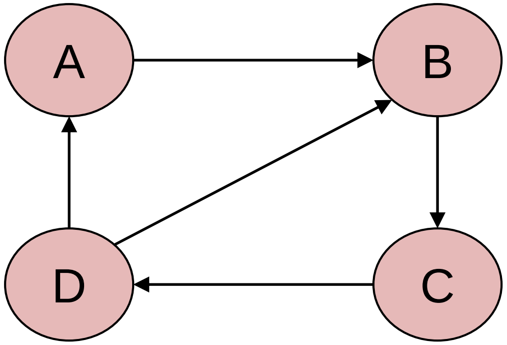
https://github.com/mgbarsky/page_rank_lab.git

PageRank exercise 1.



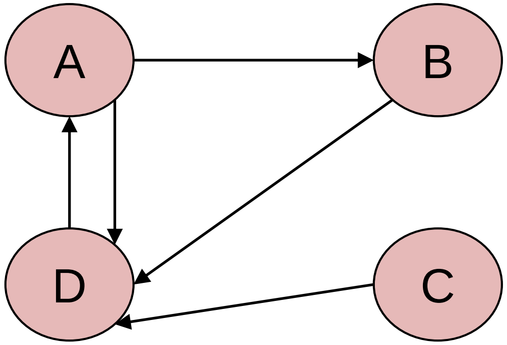
- Guess what pages in the given model got the highest rank
- Check your guess by running the program

PageRank exercise 2.



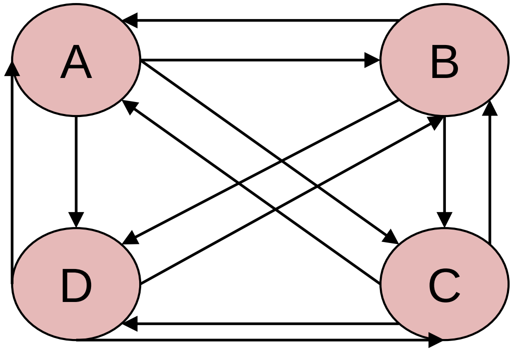
- Guess what pages in the given model got the highest rank
- Check your guess by running the program

PageRank exercise 3.



- Guess what pages in the given model got the highest rank
- Check your guess by running the program

PageRank exercise 4.



- Guess what pages in the given model got the highest rank
- Check your guess by running the program

Page Rank: General Formula

$PR(A) = PR(T_1)/C(T_1) + \ldots + PR(T_n)/C(T_n)$

1. **$PR(T_i)$** - Each page has a notion of its own self-importance, which is say 1 initially.
2. **$C(T_i)$** – Count of outgoing links from page T_i.
   1. Each page spreads its vote out evenly amongst all of it's outgoing links.
3. **$PR(T_i)/C(T_i)$** –
   a) Each page spreads its vote out evenly amongst all of it's outgoing links.
   b) So if our page (say page A) has a back link from page "*i*" the share of the vote page A will get from page "*i*" is "**$PR(T_i)/C(T_i)$**."

We re-estimate the rank for all pages **at the same time**

- The page rank (PR) of each page depends on the PR of the pages pointing to it.
  - We won’t know what PR those pages have until the pages pointing to them have their PR calculated and so on…

- Well, we just go ahead and calculate a page’s PR without knowing the final value of the PR of the other pages.
  - Each time we run the calculation we’re getting a closer estimate of the final value.
  - Repeat the calculations lots of times until the numbers converge.

Web Matrix

Capture the formula by the web matrix (M) that is:

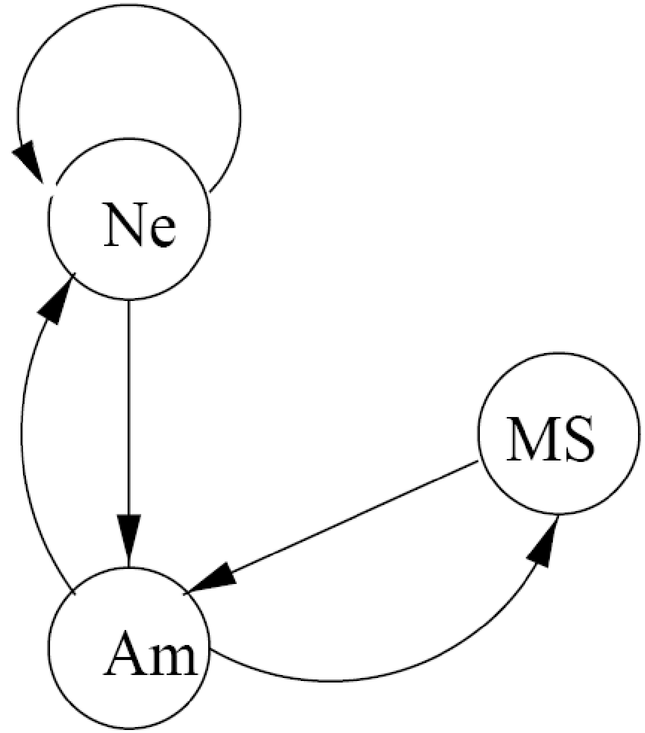
- If page *j* has *n* successors (links), then:
  - $M[i, j] = 1/n$ if page *i* is one of these *n* successors of page *j*, and
  - 0 otherwise.

Then, the importance vector containing the rank of each page is calculated by:

$$\text{Rank}_{\text{new}} = M \bullet \text{Rank}_{\text{old}}$$

Example

- Assume that in 1939, the Web consisted of only three pages: Netscape, Microsoft, and Amazon.



$$\begin{bmatrix} n_{new} \\ m_{new} \\ a_{new} \end{bmatrix} = \begin{bmatrix} 1/2 & 0 & 1/2 \\ 0 & 0 & 1/2 \\ 1/2 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} n_{old} \\ m_{old} \\ a_{old} \end{bmatrix}$$

For example, the first column of the Web matrix reflects the fact that Netscape divides its importance between itself and Amazon.

The second column indicates that Microsoft gives all its importance to Amazon.

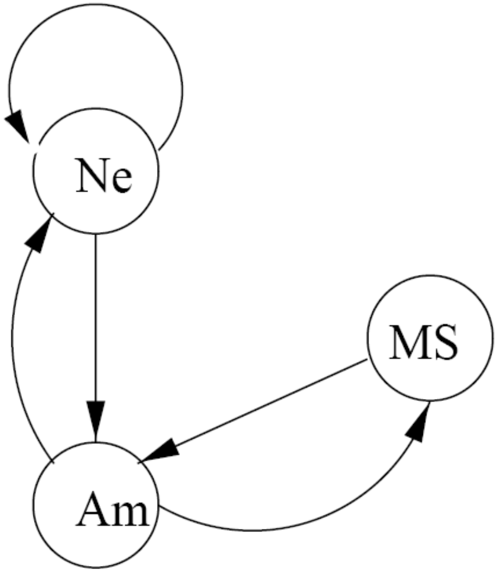
Start with n = m = a = 1, then do rounds of improvements.

Based on Jeff Ullman’s notes

Example

- The first four iterations give the following estimates:

n =	1	1	5/4	9/8	5/4
m =	1	1/2	3/4	1/2	11/16
a =	1	3/2	1	11/8	17/16



- In the limit, the solution is n = a = 6/5; m = 3/5.

- That is, Netscape and Amazon each have the same importance, and twice the importance of Microsoft (well this was 1839).

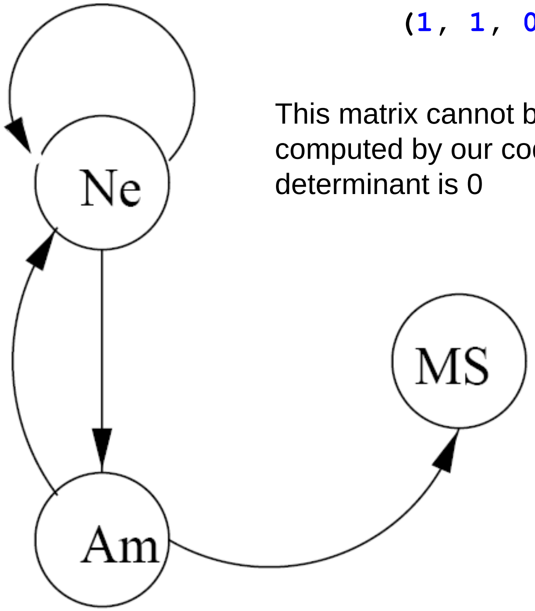
Real Web Graphs: dead ends

Dead ends: a page that has no successors has nowhere to send its importance.

Eventually, all importance will "leak out" of the Web.

Example: Suppose Microsoft tries to claim that it is a monopoly by removing all links from its site.

```
web = ((1, 0, 1),
       (0, 0, 0),
       (1, 1, 0))
```



This matrix cannot be computed by our code – determinant is 0

The new Web, and the rank vectors for the first 4 iterations are shown.

$$\begin{bmatrix} n_{new} \\ m_{new} \\ a_{new} \end{bmatrix} = \begin{bmatrix} 1/2 & 0 & 1/2 \\ 0 & 0 & 1/2 \\ 1/2 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} n_{old} \\ m_{old} \\ a_{old} \end{bmatrix}$$

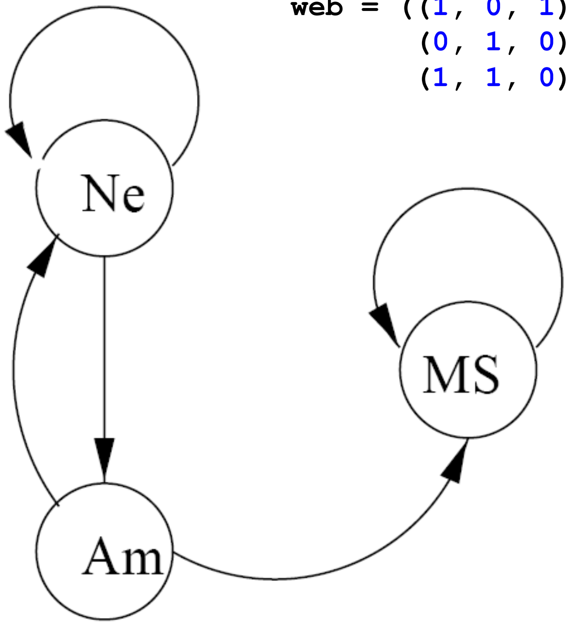
Eventually, each of *n*, *m*, and *a* become 0; i.e., all the importance will leak out.

$$\begin{array}{llllll} n = & 1 & 1 & 3/4 & 5/8 & 1/2 \\ m = & 1 & 1/2 & 1/4 & 1/4 & 3/16 \\ a = & 1 & 1/2 & 1/2 & 3/8 & 5/16 \end{array}$$

Real Web Graphs: spider traps

```
web = ((1, 0, 1),
       (0, 1, 0),
       (1, 1, 0))
```

Spider traps: a group of one or more pages that have no links out of the group will eventually accumulate all the importance of the Web.



Example: Angered by the decision, Microsoft decides it will link only to itself from now on. Now, Microsoft has become a spider trap.

$$\begin{bmatrix} n_{new} \\ m_{new} \\ a_{new} \end{bmatrix} = \begin{bmatrix} 1/2 & 0 & 1/2 \\ 0 & 1 & 1/2 \\ 1/2 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} n_{old} \\ m_{old} \\ a_{old} \end{bmatrix}$$

The new Web, and the rank vectors for the first 4 iterations are shown.

$n =$	1	1	3/4	5/8	1/2
$m =$	1	3/2	7/4	2	35/16
$a =$	1	1/2	1/2	3/8	5/16

Now, *m* converges to 3, and *n* = *a* = 0.

Google Solution to Dead Ends and Spider Traps

Stop the other pages having too much influence.

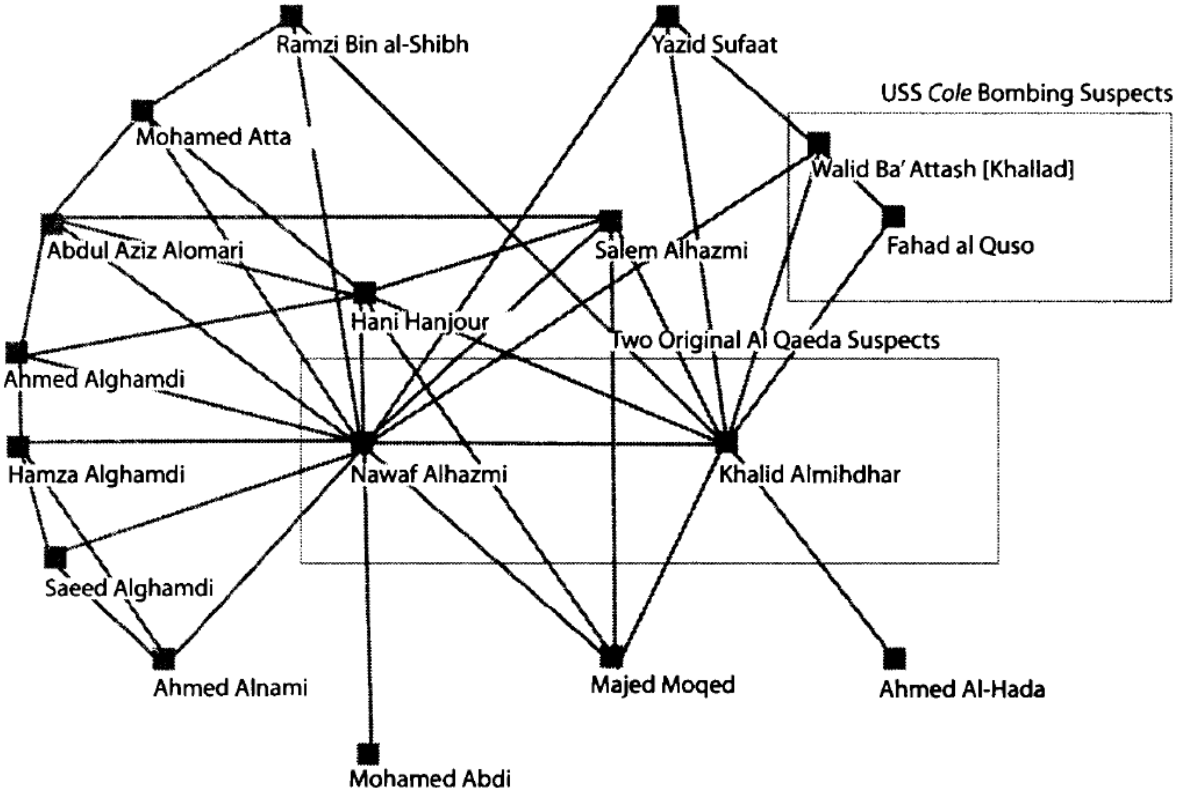
This total vote is “damped down” by multiplying it by a factor.

Example: If we use a 20% damp-down, the equation of previous example becomes:

$$\begin{bmatrix} n_{new} \\ m_{new} \\ a_{new} \end{bmatrix} = 0.80 \cdot \begin{bmatrix} 1/2 & 0 & 1/2 \\ 0 & 1 & 1/2 \\ 1/2 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} n_{old} \\ m_{old} \\ a_{old} \end{bmatrix} + 0.20 \cdot \begin{bmatrix} n_{old} \\ m_{old} \\ a_{old} \end{bmatrix}$$

The solution to this equation is n = 7/11; m = 21/11; a = 5/11.

Lab: most important terrorists?



Graph of the Al Qaeda group behind the September 11 attacks

Source: The Numbers Behind NUMB3RS