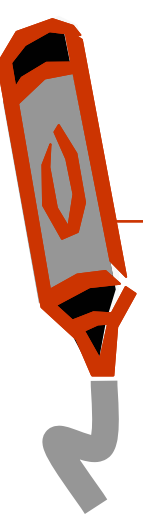


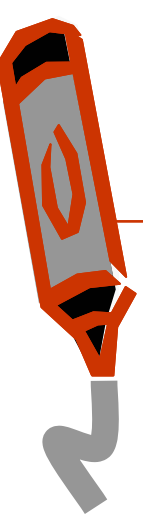
www IR



www IR

- world wide web
- google, page rank
- markov chains
- HITS link analysis
- behavior-based web search
- crawling, indexing the web
- duplicates, mirrors and spam
- www infrastructure
- www size
- cache, hardware, systems

top online activities



(a) Source: Jupiter Communications.



US users (2002)

- Total Internet users = 111 M
- Do a search on any given day = 33 M
- Have used Internet to search = 85%

[//www.pewinternet.org/reports/toc.asp?Report=64](http://www.pewinternet.org/reports/toc.asp?Report=64)



search on the web

Corpus: The publicly accessible Web: static + dynamic

- **Goal:** Retrieve high quality results relevant to the user's need
 - (not docs!)
- **Need**
 - Informational – want to learn about something (~40%)
 - Low hemoglobin**
 - Navigational – want to go to that page (~25%)
 - United Airlines**
 - Transactional – want to do something (web-mediated) (~35%)
 - Access a service **Tampere weather**
 - Downloads **Mars surface images**
 - Shop **Nikon CoolPix**
 - Gray areas
 - Find a good hub **Car rental Finland**
 - Exploratory search “see what’s there”



results

- Static pages (documents)
 - text, mp3, images, video, ...
- Dynamic pages = generated on request
 - data base access
 - “the invisible web”
 - proprietary content, etc.

terminology



URL = Universal Resource Locator

`http://www.cism.it/cism/hotels_2001.htm`

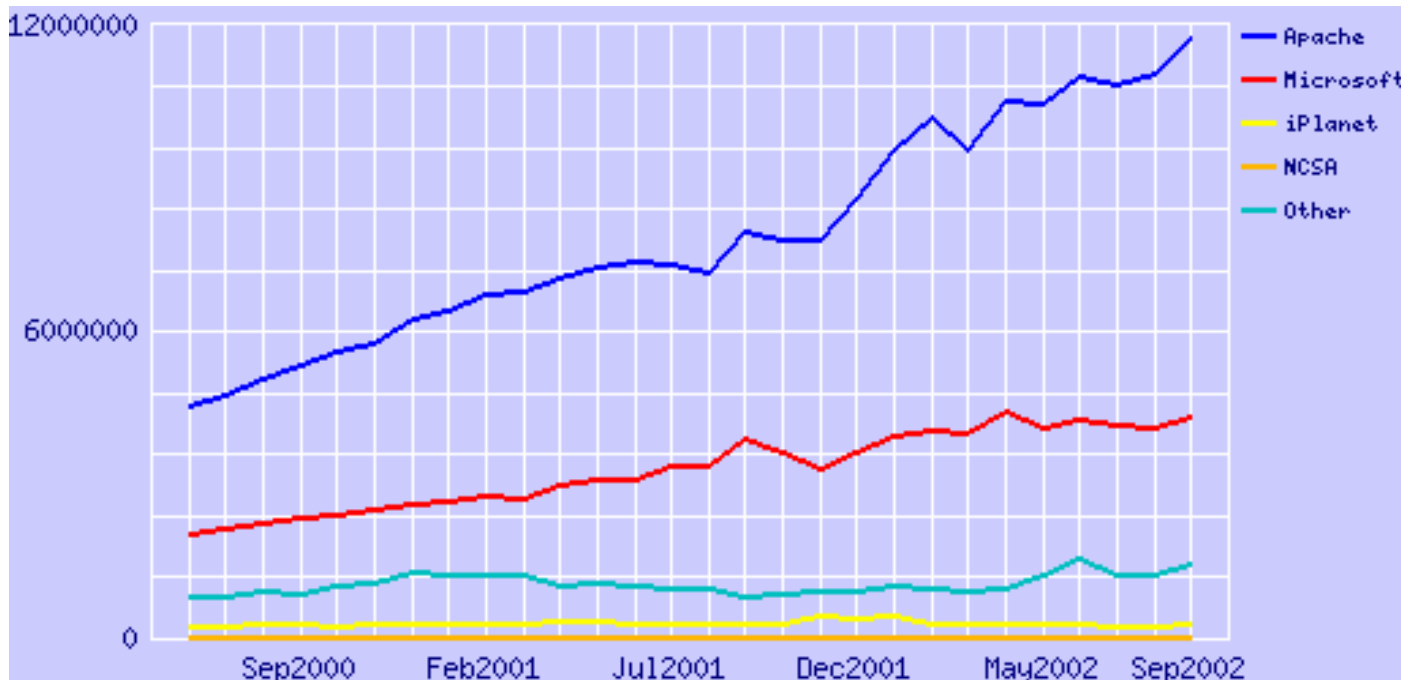
Access method

Host name

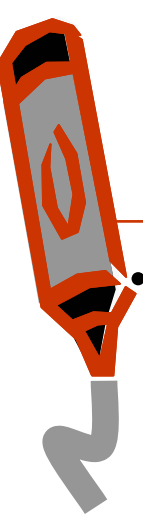
Page name

scale

- Immense amount of content
 - 2-10B static pages, doubling every 8-12 months
 - Lexicon Size: 10s-100s of millions of words
- Authors galore (1 in 4 hosts run a web server)



diversity



- Languages/Encodings
 - Hundreds (thousands ?) of languages, W3C encodings: 55 (Home pages (1997): English 82%, Next 15: 13% [Babe97])
 - Google (mid 2001): English: 53%
- Document & query topic
 - Popular Query Topics (from 1 mil Google queries, 06/2000)

Arts	14.6%	Arts: Music	6.1%
Computers	13.8%	Regional: North America	5.3%
Regional	10.3%	Adult: Image Galleries	4.4%
Society	8.7%	Computers: Software	3.4%
Adult	8%	Computers: Internet	3.2%
Recreation	7.3%	Business: Industries	2.3%
Business	7.2%	Regional: Europe	1.8%
...

rate of change

720K pages from 270 popular sites sampled daily from Feb 17 – Jun 14, 1999

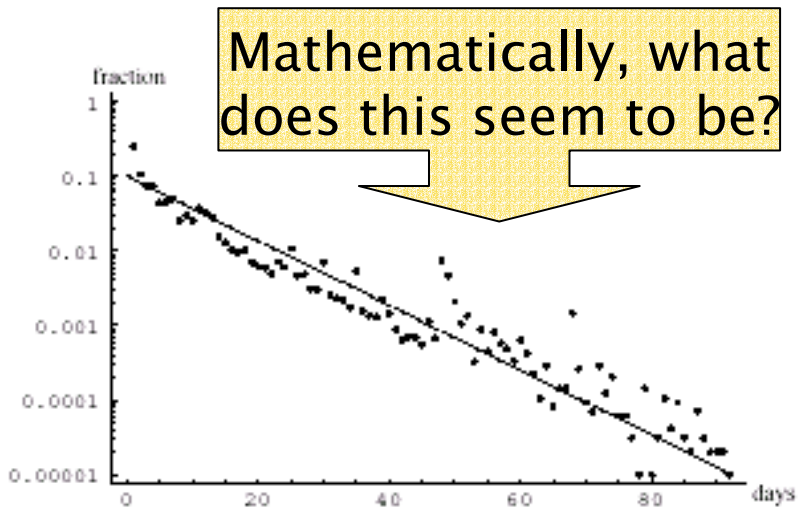


Figure 11: Change intervals for pages with the average change interval of 10 days

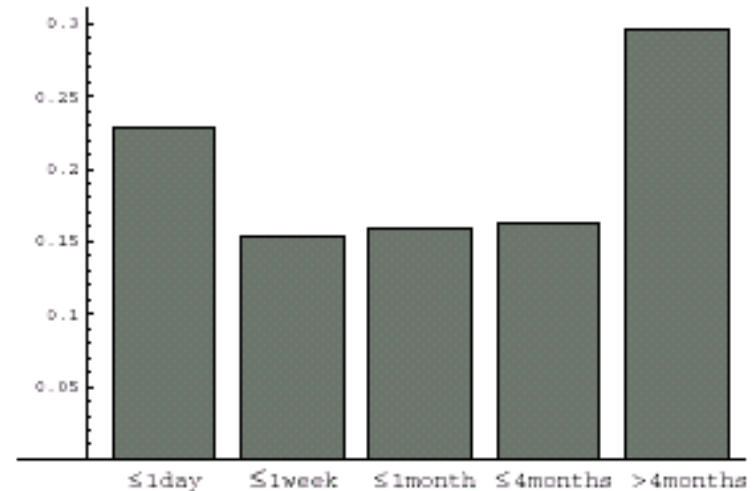


Figure 12: Percentage of pages with given average interval of change



web idiosyncrasies

- Distributed authorship
 - Millions of people creating pages with their own style, grammar, vocabulary, opinions, facts, falsehoods ...
 - Not all have the purest motives in providing high-quality information - commercial motives drive “spamming” - 100s of millions of pages.
 - The open web is largely a marketing tool.
 - IBM’s home page does not contain *computer*.

WWW

- Significant duplication
 - Syntactic - 30%-40% (near) duplicates [Brod97, Shiv99b]
 - Semantic
- High linkage
 - ~ 8 links/page in the average
- Complex graph topology
 - Not a small world; bow-tie structure [Brod00]
- More on these corpus characteristics later
 - how do we measure them?



web search users

- ill-defined queries
 - Short
 - AV 2001: 2.54 terms avg, 80% ; 3 words)
 - Imprecise terms
 - Sub-optimal syntax (80% queries without operator)
 - Low effort
- specific behavior
 - 85% look over one result screen only (mostly above the fold)
 - 78% of queries are not modified (one query/session)
 - Follow links – “the scent of information” ...
- wide variance in
 - Needs
 - Expectations
 - Knowledge
 - Bandwidth



evolution of search engines

- First generation -- use only “on page”, text data

- Word frequency, language

1995-1997 AV,
Excite, Lycos, etc

- Second generation -- use off-page, web-specific data

- Link (or connectivity) analysis

- Click-through data (What results people click on)

- Anchor-text (How people refer to this page)

From 1998-2003.
Made popular by
Google

- Third generation -- answer “the need behind the query”

- Semantic analysis -- what is this about?

- Focus on user need, rather than on query

- Context determination

- Helping the user

- Integration of search and text analysis

present



first generation ranking

- Extended Boolean model
 - Matches: exact, prefix, phrase,...
 - Operators: AND, OR, AND NOT, NEAR, ...
 - Fields: TITLE:, URL:, HOST:,...
 - AND is somewhat easier to implement, maybe preferable as default for short queries
- Ranking
 - TF like factors: TF, explicit keywords, words in title, explicit emphasis (headers), etc
 - IDF factors: IDF, total word count in corpus, frequency in query log, frequency in language



second generation search engine

- Ranking -- use off-page, web-specific data
 - Link (or connectivity) analysis
 - Click-through data (What results people click on)
 - Anchor-text (How people refer to this page)
- Crawling
 - Algorithms to create the best possible corpus



connectivity analysis

- Idea: mine hyperlink information in the Web
- Assumptions:
 - Links often connect related pages
 - A link between pages is a recommendation “people vote with their links”



third generation search engine

- Query language determination
- Different ranking
 - (if query Japanese do not return English)
- Hard & soft matches
 - Personalities (triggered on names)
 - Cities (travel info, maps)
 - Medical info (triggered on names and/or results)
 - Stock quotes, news (triggered on stock symbol)
 - Company info, ...
- Better integration of Search and Text Analysis



context determination

- Context determination
 - spatial (user location/target location)
 - query stream (previous queries)
 - personal (user profile)
 - explicit (vertical search, family friendly)
 - implicit (use AltaVista from AltaVista France)
- Context use
 - Result restriction
 - Ranking modulation



spatial context: geo-search

- Two aspects
 - Geo-coding
 - encode geographic coordinates to make search effective
 - Geo-parsing
 - the process of identifying geographic context.
- Geo-coding
 - Geometrical hierarchy (squares)
 - Natural hierarchy (country, state, county, city, zip-codes, etc)
- Geo-parsing
 - Pages (infer from phone nos, zip, etc). About 10% feasible.
 - Queries (use dictionary of place names)
 - Users
 - From IP data
 - Mobile phones
 - In its infancy, many issues (display size, privacy, etc)

AV barry bonds

Search for:

[Help](#) | [Customize Settings](#) | Family Filter is **off**

Related Searches:

- [who is barry bonds](#)

- [pictures of Barry Bonds](#)

- [barry bonds giants t shirt](#)

AltaVista Recommends

Barry Bonds



- [Player Page | Log | Stats](#)
- [Player News and Outlook](#)
- [Batter vs. Pitcher](#)
- [San Francisco Giants Team Page](#)

Find Results In:

15,048 pages found.

[Products](#)

[News](#)

[Business](#)

[Web Pages](#)

[Images](#)

[MP3/Audio](#)

[Video](#)

[Directories](#)

Lycos palo alto

[Track this Search](#)

Results for

Go Get It![®]

Search within these results

[NEW SEARCH](#)

[SEARCH GUARD](#)

[ADVANCED SEARCH](#)

Find [On the Prairie of Palo Alto](#)
by Charles Haecker

[✈ Book a room in Palo Alto](#)

POPULAR

[[POPULAR](#) | [WEB SITES](#) | [NEWS ARTICLES](#) | [SHOPPING](#)]

2 of the Web sites reviewed by Lycos Editors match your search

City Guide: Travel info about [Palo Alto](#)

Reservations: Book a [flight](#) or [rental car](#)

Lodging: Find [places to stay](#) in Palo Alto

Maps: [Palo Alto](#) map and [driving directions](#)

Weather: 5-day forecast for [Palo Alto](#)

Dining Out: [Palo Alto](#) restaurant listings

Yellow Pages: Find Palo Alto [colleges](#) and [apartments](#)

1. [California Travel Guide](#) - Things to see and do, hotels, maps, and other useful information.

http://travel.lycos.com/Destinations/North_America/USA/California/
[[Translate](#)]

[Book a room in Palo Alto](#)

2. [EAST PALO ALTO/Human Crosswalk Part Of Safe Walking Day](#) - SF Gate SF Gate Home Today's News Sports Entertainment Technology Live Views Traffic Weather Health Business Bay Area Travel Columnists Classifieds Conferences Search Index Jump to EAST **PALO ALTO** Huma
[More Articles](#) about **palo alto** from [sfgate.com](#)
[[Translate](#)]

WEB SITES

[[POPULAR](#) | [WEB SITES](#) | [NEWS ARTICLES](#) | [SHOPPING](#)]

Geo-search example

Edit your search below.

What

What info, service, or product would you like to search for?

Search for

Where

Where do you want to search from?

Street address

City, State

Zip code USA or Canada

How far

How far do you want to search?

Approx. radius

GeoSearch found 710 items in 575 sources for:

[Edit this search](#) [? Tips](#)

1. [Internet Cafés Louisiana](#)

Matching addresses on this page:

225 Baronne Street, New Orleans, LA 70112 0.3 miles

116 Bourbon Street, New Orleans, LA 70130 0.4 miles

621 Royal St, New Orleans, LA 70119 0.6 miles

507 Dumine Street, New Orleans, LA 70116 0.8 miles

[Show a map of these addresses](#)

[More results](#) from this site

2. [Royal Access Internet Cafe](#)

Matching addresses on this page:

Baton Rouge, LA 70140 (504)525-0401 0.6 miles

[Show a map of these addresses](#)

3. [Cyber Cafes](#)

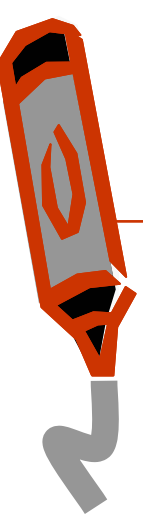
Matching addresses on this page:

621 Royal St, New Orleans, LA 70130 0.6 miles

605-607 Toulouse Street, New Orleans, LA 70130 0.6 miles

[Show a map of these addresses](#)

[More results](#) from this site



helping the user

- UI
- spell checking
- query refinement
- query suggestion
- context transfer ...



context sensitive spell check

[Advanced Search](#)[Preferences](#)[Language Tools](#)[Search Tips](#)[Web](#) | [Images](#) | [Groups](#) | [Directory](#)

Searched the web for **andrey broder**. Results 1 - 10 of about 160. Search took 0.10 seconds.

Did you mean: [andrei broder](#)

[CREEB CONFERENCE 6](#)

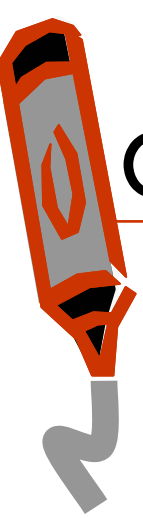
... **Broder** Dittschar, CREEB, 'Standardization versus adaptation in financial services: foreign ... companies in Romania'. Oleg Martinenko, Ludmila Kaverzina and **Andrey** ...

www.bcuc.ac.uk/business/creeb2.htm - 23k - [Cached](#) - [Similar pages](#)



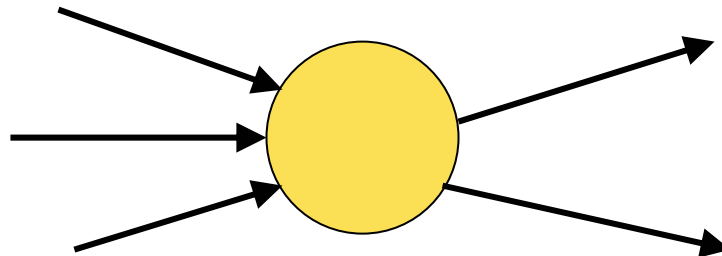
citation analysis

- Citation frequency
- Co-citation coupling frequency
 - Cocitations with a given author measures “impact”
 - Cocitation analysis [Mcca90]
- Bibliographic coupling frequency
 - Articles that co-cite the same articles are related
- Citation indexing
 - Who is a given author cited by? (Garfield [Garf72])



query-independent ordering

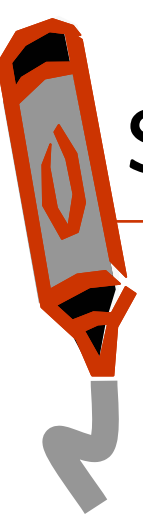
- First generation: using link counts as simple measures of popularity.
- Two basic suggestions:
 - Undirected popularity:
 - Each page gets a score = the number of in-links plus the number of out-links ($3+2=5$).
 - Directed popularity:
 - Score of a page = number of its in-links (3).





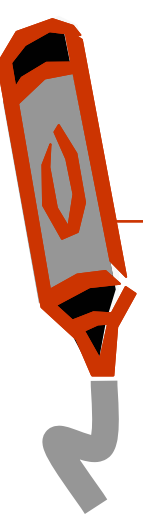
query processing

- First retrieve all pages meeting the text query (say *venture capital*).
- Order these by their link popularity (either variant on the previous page).



spamming simple popularity

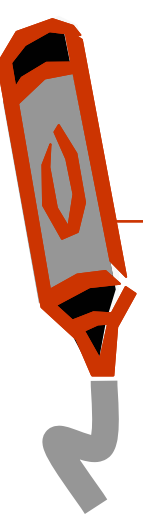
- *Exercise:* How do you spam each of the following heuristics so your page gets a high score?
- Each page gets a score = the number of in-links plus the number of out-links.
- Score of a page = number of its in-links.



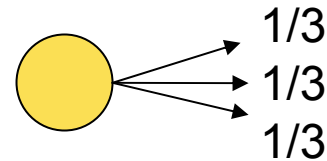
WWW IR

- world wide web
- google, page rank
- markov chains
- HITS link analysis
- behavior-based web search
- crawling, indexing the web
- duplicates, mirrors and spam
- www infrastructure
- www size
- cache, hardware, systems

pagerank scoring



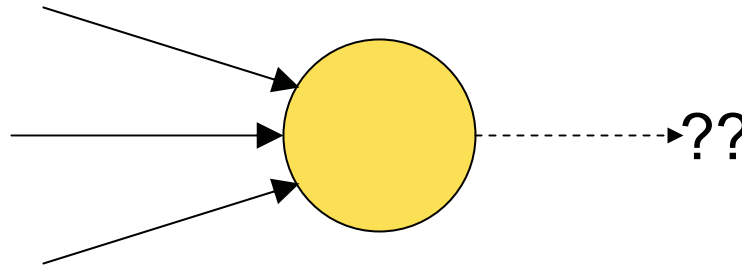
- Imagine a browser doing a random walk on web pages:
 - Start at a random page
 - At each step, go out of the current page along one of the links on that page, equiprobably



- “In the steady state” each page has a long-term visit rate - use this as the page’s score.

Not quite enough

- The web is full of dead-ends.
 - Random walk can get stuck in dead-ends.
 - Makes no sense to talk about long-term visit rates.



Teleporting

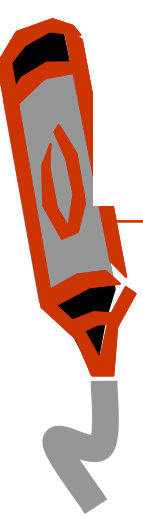
- At each step, with probability 10%, jump to a random web page.
- With remaining probability (90%), go out on a random link.
 - If no out-link, stay put in this case.



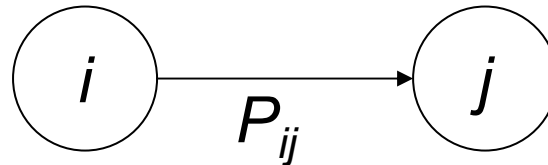
Result of teleporting

- Now cannot get stuck locally.
- There is a long-term rate at which any page is visited (not obvious, will show this).
- How do we compute this visit rate?

Markov chains

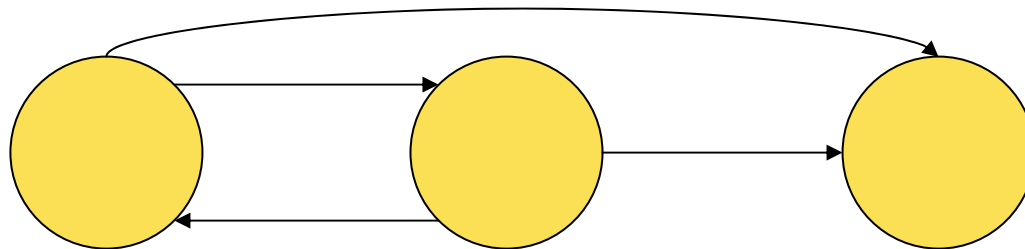


- A Markov chain consists of n states, plus an $n \times n$ transition probability matrix \mathbf{P} .
- At each step, we are in exactly one of the states.
- For $1 \leq i, j \leq n$, the matrix entry P_{ij} tells us the probability of j being the next state, given we are currently in state i .



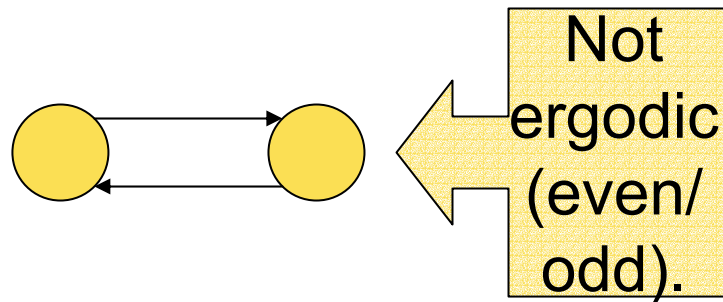
Markov chains

- Clearly, for all i ,
$$\sum_{j=1}^n P_{ij} = 1.$$
- Markov chains are abstractions of random walks.
- Exercise:* represent the teleporting random walk from 3 slides ago as a Markov chain, for this case:

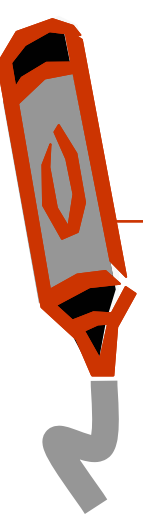


Ergodic Markov chains

- A Markov chain is ergodic if
 - you have a path from any state to any other
 - you can be in any state at every time step, with non-zero probability.

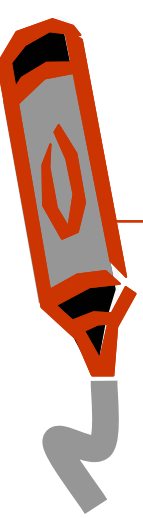


Ergodic Markov chains



- For any ergodic Markov chain, there is a unique long-term visit rate for each state.
 - *Steady-state distribution.*
- Over a long time-period, we visit each state in proportion to this rate.
- It doesn't matter where we start.

Probability vectors



- A probability (row) vector $\mathbf{x} = (x_1, \dots, x_n)$ tells us where the walk is at any point.
- E.g., $(\underset{1}{000}\dots\underset{i}{1}\dots\underset{n}{000})$ means we're in state i .

▪ More generally, the vector $\mathbf{x} = (x_1, \dots, x_n)$ means the walk is in state i with probability x_i .

$$\sum_{i=1}^n x_i = 1.$$

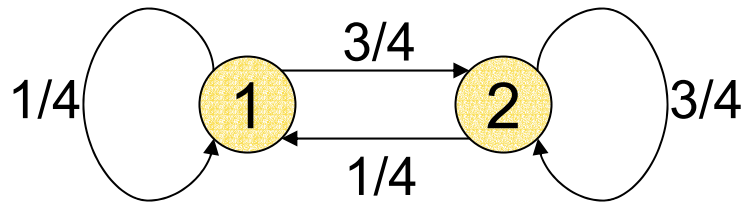


Change in probability vector

- If the probability vector is $\mathbf{x} = (x_1, \dots, x_n)$ at this step, what is it at the next step?
- Recall that row i of the transition prob. Matrix \mathbf{P} tells us where we go next from state i .
- So from \mathbf{x} , our next state is distributed as \mathbf{xP} .

Computing the visit rate

- The steady state looks like a vector of probabilities $\mathbf{a} = (a_1, \dots, a_n)$:
 - a_i is the probability that we are in state i .



For this example, $a_1=1/4$ and $a_2=3/4$.

How do we compute this vector?

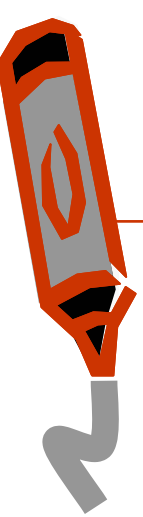
- Let $\mathbf{a} = (a_1, \dots, a_n)$ denote the row vector of steady-state probabilities.
- If we our current position is described by \mathbf{a} , then the next step is distributed as \mathbf{aP} .
- But \mathbf{a} is the steady state, so $\mathbf{a} = \mathbf{aP}$.
- Solving this matrix equation gives us \mathbf{a} .
 - So \mathbf{a} is the (left) eigenvector for \mathbf{P} .
 - (Corresponds to the “principal” eigenvector of \mathbf{P} with the largest eigenvalue.)



One way of computing \mathbf{a}

- Recall, regardless of where we start, we eventually reach the steady state \mathbf{a} .
- Start with any distribution (say $\mathbf{x}=(10\dots0)$).
- After one step, we're at \mathbf{xP} ;
- after two steps at \mathbf{xP}^2 , then \mathbf{xP}^3 and so on.
- “Eventually” means for “large” k , $\mathbf{xP}^k = \mathbf{a}$.
- Algorithm: multiply \mathbf{x} by increasing powers of \mathbf{P} until the product looks stable.

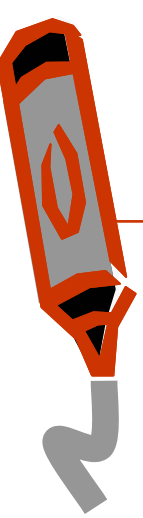
Pagerank summary

- 
- Preprocessing:
 - Given graph of links, build matrix P .
 - From it compute a .
 - 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.
 - Order is *query-independent*.

In Reality

- Pagerank is used in google, but so are many other clever heuristics
 - more on these heuristics later.

Pagerank



- Pagerank computation
 - Random walk on the web graph
 - Teleport operation to get unstuck from dead ends
 - ⇒ Steady state visit rate for each web page
 - Call this its pagerank score
 - computed from an eigenvector computation (linear system solution)

Pagerank recap



- Pagerank usage
 - Get pages matching text query
 - Return them in order of pagerank scores
 - This order is query-independent
 - Can combine arithmetically with text-based scores
- Pagerank is a global property
 - Your pagerank score depends on “everybody” else
 - Harder to spam than simple popularity counting

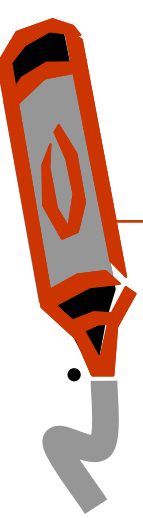


Topic Specific Pagerank [Have02]

Conceptually, we use a random surfer who teleports, with say 10% probability, using the following rule:

- Selects a category (say, one of the 16 top level ODP categories) based on a query & user -specific distribution over the categories
 - Teleport to a page uniformly at random within the chosen category
- Sounds hard to implement: can't compute PageRank at query time!

Topic Specific Pagerank [Have02]



Implementation

- **offline:** Compute pagerank distributions wrt to *individual* categories

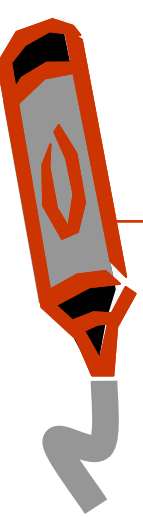
Query independent model as before

Each page has multiple pagerank scores – one for each ODP category, with teleportation only to that category

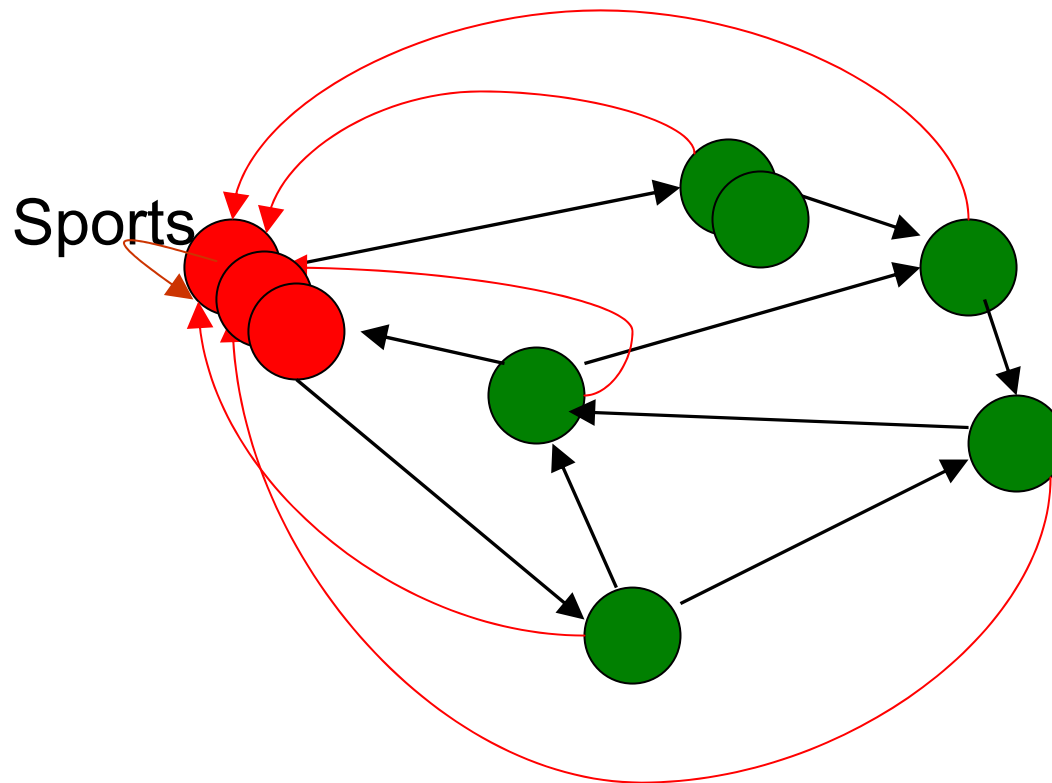
- **online:** Distribution of weights over categories computed by query context classification

Generate a dynamic pagerank score for each page - weighted sum of category-specific pageranks

Influencing PageRank ("Personalization")

- 
- Input:
 - Web graph W
 - influence vector v
 $v : (\text{page} \rightarrow \text{degree of influence})$
 - Output:
 - Rank vector r : (page \rightarrow page importance wrt v)
 - $r = \text{PR}(W, v)$

Non-uniform Teleportation



Teleport with 10% probability to a Sports page



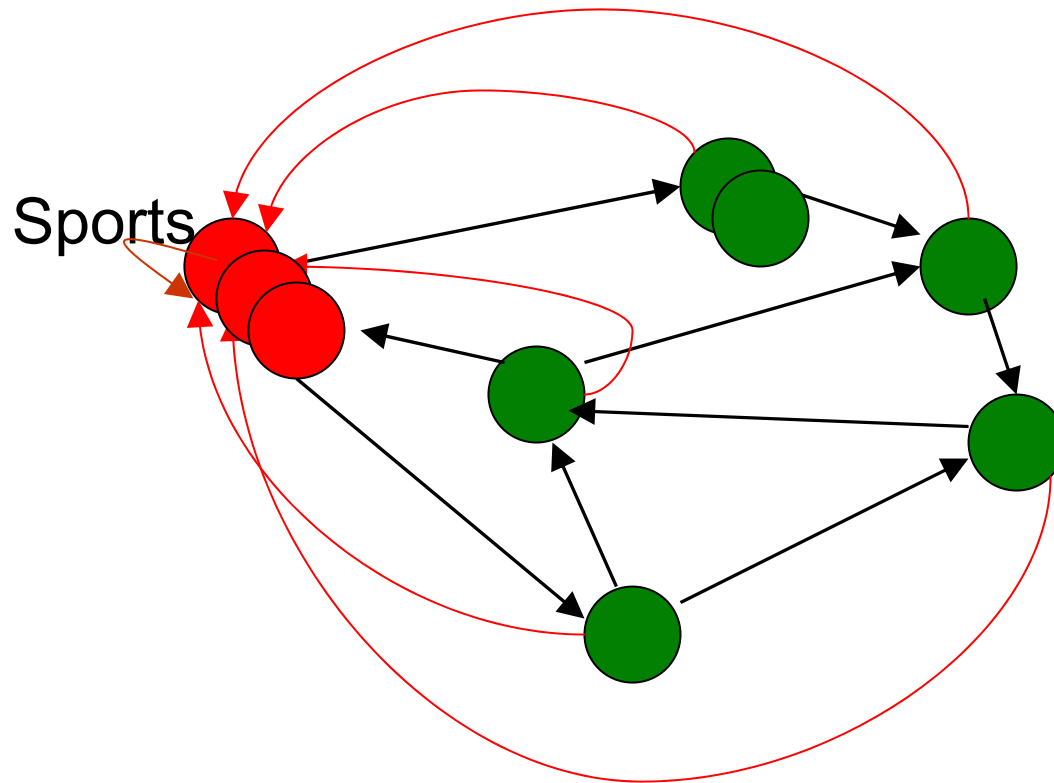
Interpretation of Composite Score

- For a set of personalization vectors $\{v_j\}$

$$\sum_j [w_j \cdot \text{PR}(W, v_j)] = \text{PR}(W, \sum_j [w_j \cdot v_j])$$

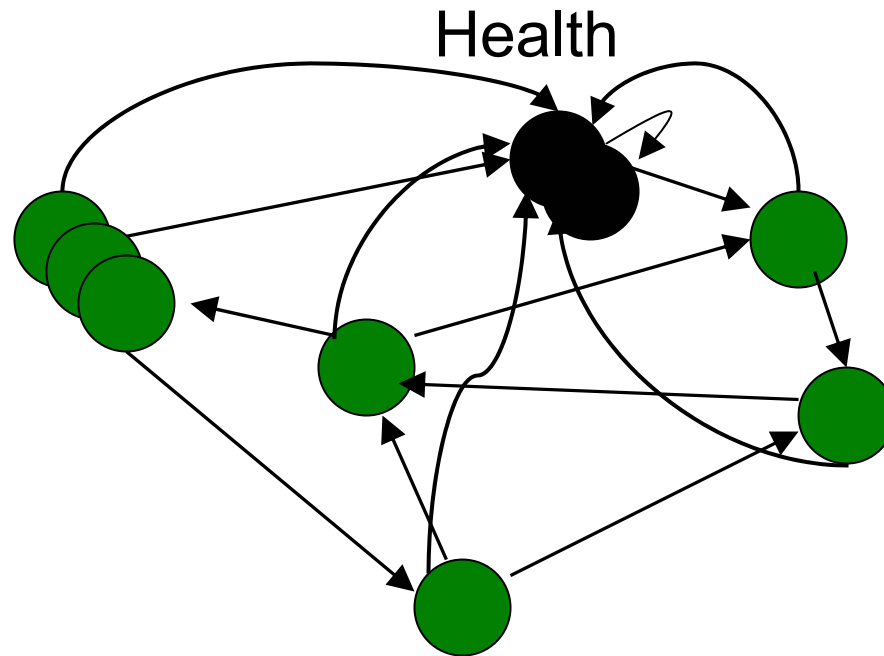
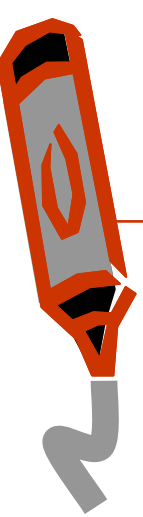
- Weighted sum of rank vectors itself forms a valid rank vector, because $\text{PR}()$ is linear wrt v_j

Interpretation



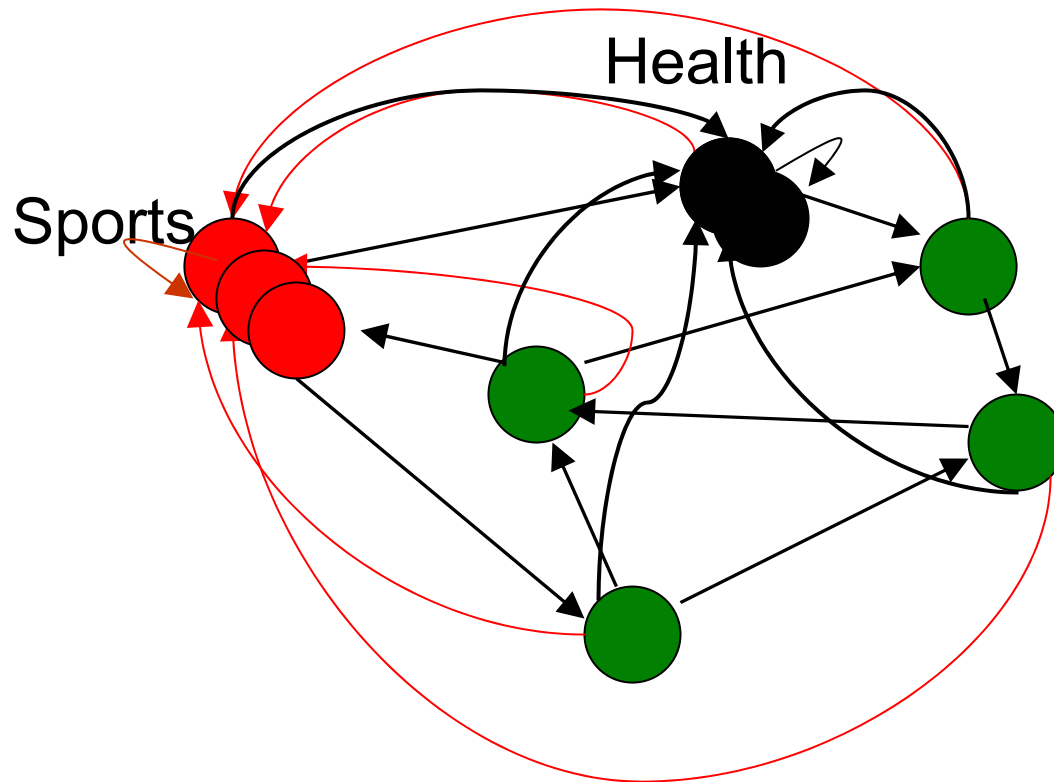
10% Sports teleportation

Interpretation

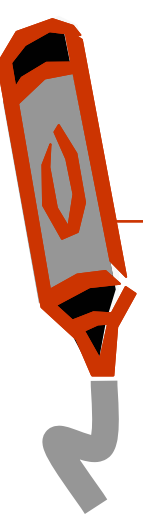


10% Health teleportation

Interpretation



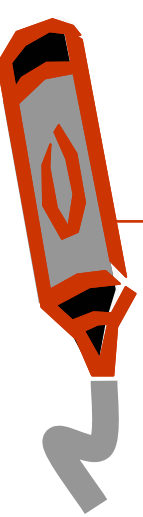
$pr = (0.9 PR_{\text{sports}} + 0.1 PR_{\text{health}})$ gives you:
9% sports teleportation, 1% health teleportation



WWW IR

- world wide web
- google, page rank
- markov chains
- HITS link analysis
- behavior-based web search
- crawling, indexing the web
- duplicates, mirrors and spam
- www infrastructure
- www size
- cache, hardware, systems

Hyperlink-Induced Topic Search (HITS) - Klei98

- 
- In response to a query, instead of an ordered list of pages each meeting the query, find two sets of inter-related pages:
 - *Hub pages* are good lists of links on a subject.
 - e.g., “Bob’s list of cancer-related links.”
 - *Authority pages* occur recurrently on good hubs for the subject.
 - Best suited for “broad topic” queries rather than for page-finding queries.
 - Gets at a broader slice of common *opinion*.

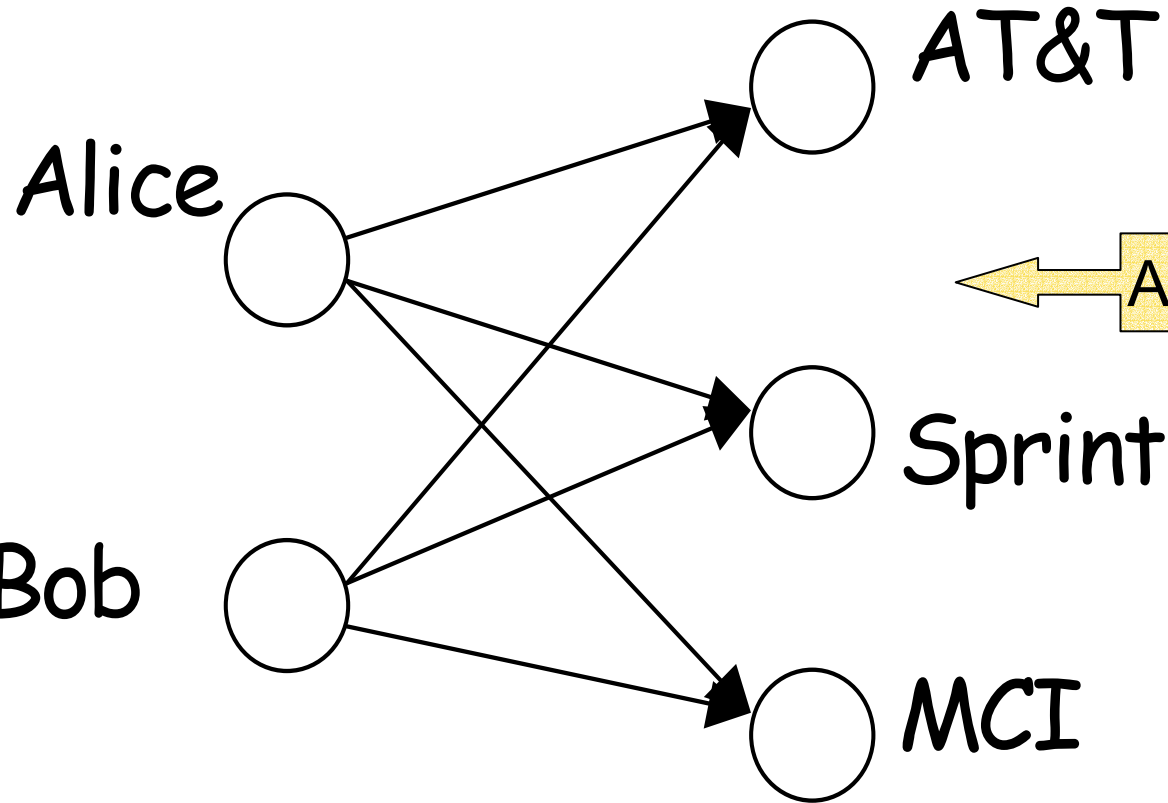


Hubs and Authorities

- Thus, a good hub page for a topic *points* to many authoritative pages for that topic.
- A good authority page for a topic is *pointed* to by many good hubs for that topic.
- Circular definition - will turn this into an iterative computation.

The hope

Hubs →



← Authorities

Long distance telephone companies



High-level scheme

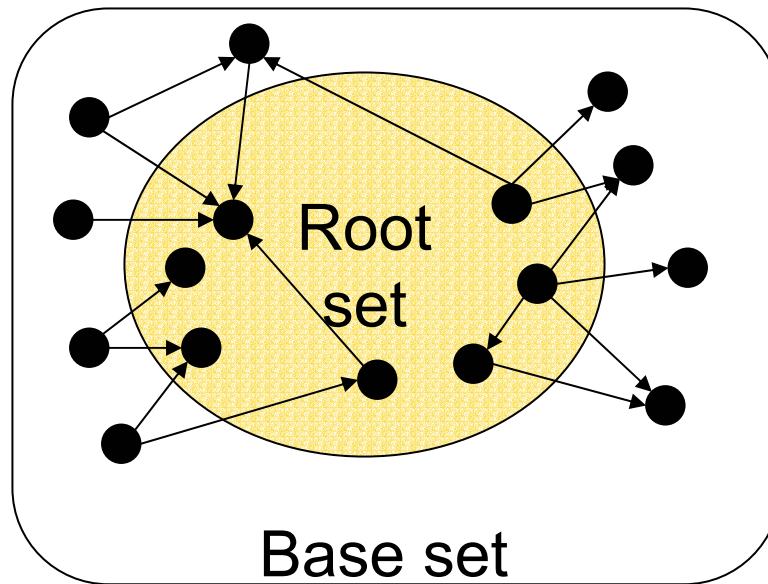
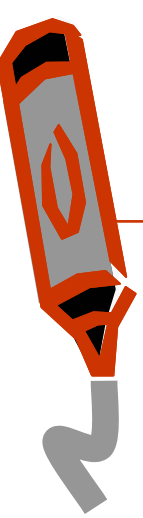
- Extract from the web a base set of pages that *could* be good hubs or authorities.
- From these, identify a small set of top hub and authority pages;
→ [iterative algorithm](#).



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.

Visualization





Assembling the base set

- Root set typically 200-1000 nodes.
- Base set may have up to 5000 nodes.
- How do you find the base set nodes?
 - Follow out-links by parsing root set pages.
 - Get in-links (and out-links) from a *connectivity server*.
 - (Actually, suffices to text-index strings of the form *href*="URL" to get in-links to URL.)



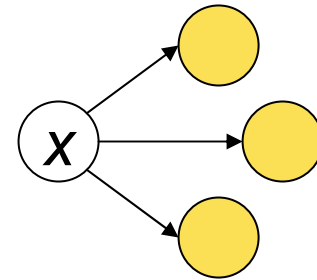
Distilling hubs and authorities

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

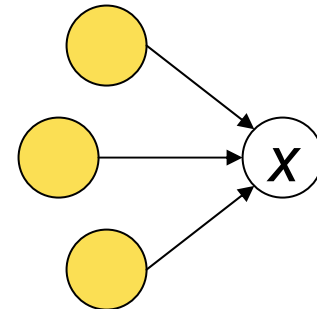
Iterative update

- Repeat the following updates, for all x :

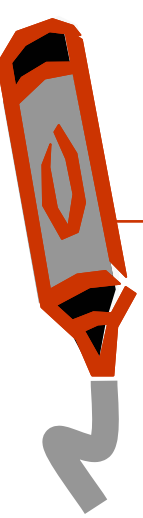
$$h(x) \leftarrow \sum_{x \mapsto y} a(y)$$



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



Scaling



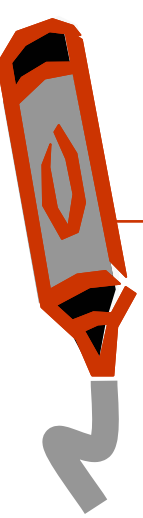
- To prevent the $h()$ and $a()$ values from getting too big, can scale down after each iteration.
- Scaling factor doesn't really matter:
 - we only care about the *relative* values of the scores.



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!
 - proof of this comes later.
- We only require the relative orders of the $h()$ and $a()$ scores - not their absolute values.
- In practice, ~ 5 iterations get you close to stability.

Things to note

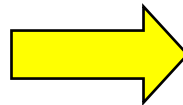
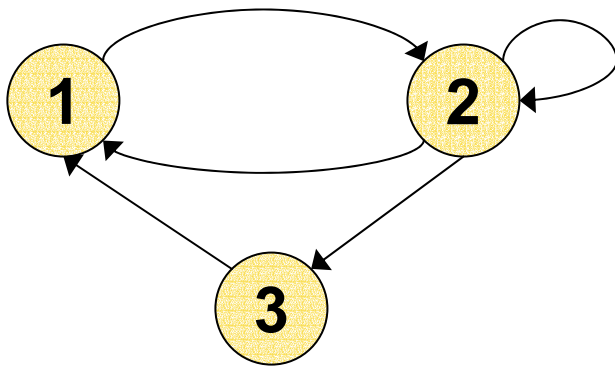


- Pulled together good pages regardless of language of page content.
- Use *only* link analysis after base set assembled
 - *iterative scoring is query-independent.*
- Iterative computation after text index retrieval - significant overhead.

Proof of convergence

$n \times n$ 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



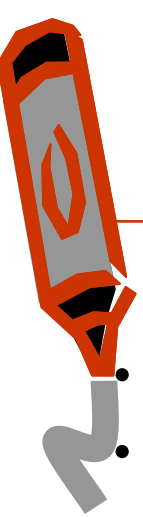
Hub/authority vectors

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

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

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

Rewrite in matrix form



- $h = Aa.$
- $a = A^t h.$

Recall A^t
is the
transpose
of $A.$

Substituting, $h = AA^t h$ and $a = A^t A a.$

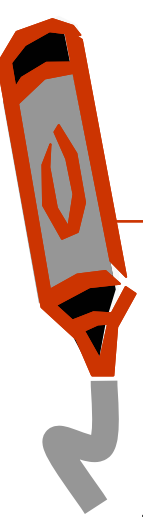
Thus, h is an eigenvector of AA^t and
 a is an eigenvector of $A^t A.$



Tag/position heuristics

- Increase weights of terms
 - in titles
 - in tags
 - near the beginning of the doc, its chapters and sections

Anchor text



Tiger image

Here is a great picture
of a tiger

Cool tiger webpage



The text in the vicinity of a hyperlink is descriptive of the page it points to.

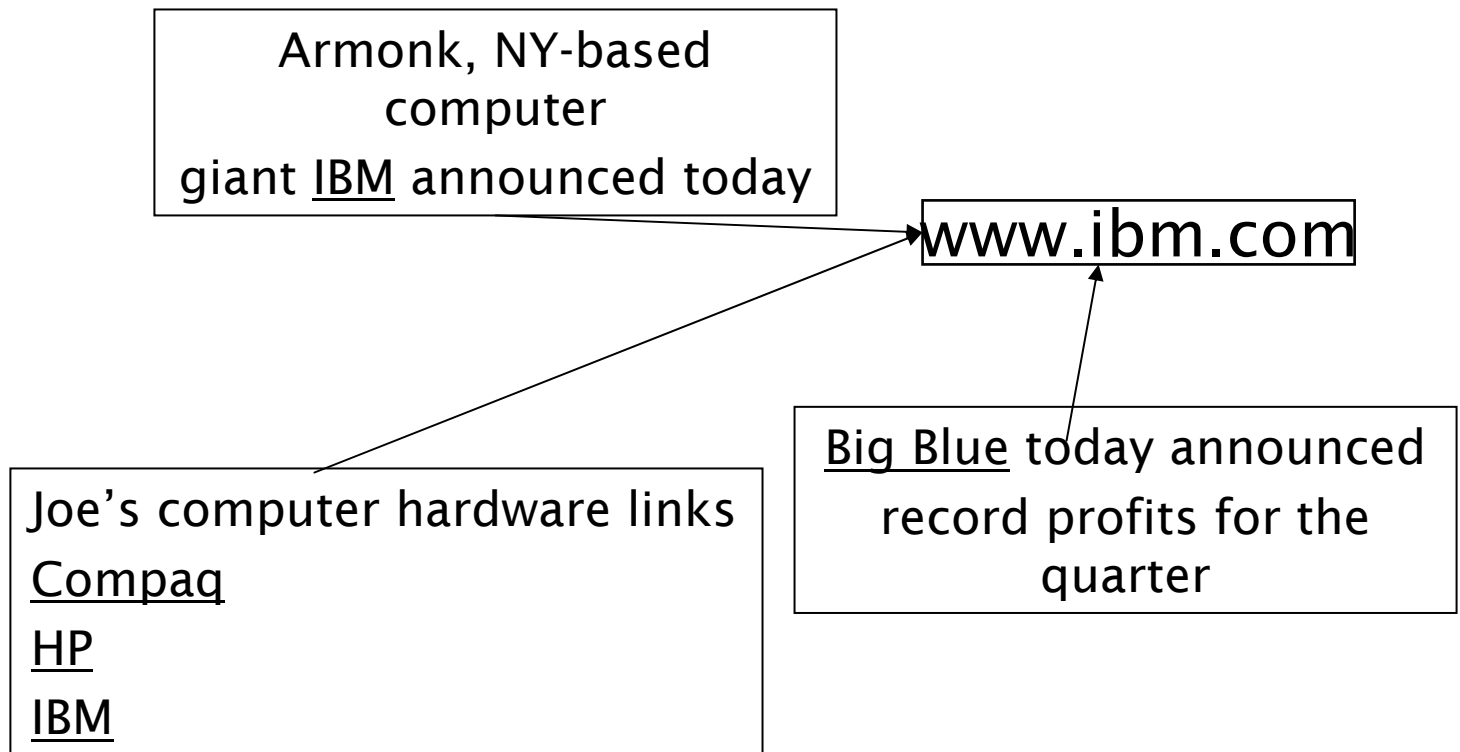


Anchor text

- When indexing a page, also index the anchor text of links pointing to it.
 - Retrieve a page when query matches its anchor text.
- To weight links in the hubs/authorities algorithm.
- Anchor text usually taken to be a window of 6-8 words around a link anchor.

indexing anchor text

When indexing a document D , include anchor text from links pointing to D .



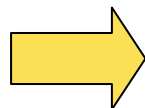
- Can sometimes have unexpected side effects - *e.g., evil empire.*
- Can index anchor text with less weight.

weighting anchor text

- In hub/authority link analysis, can match anchor text to query, then weight link.

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

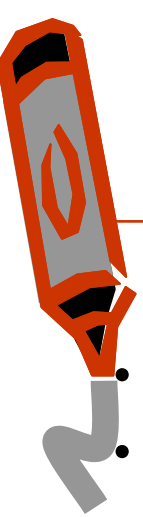
$$a(x) \leftarrow \sum_{y \mapsto x} h(y)$$



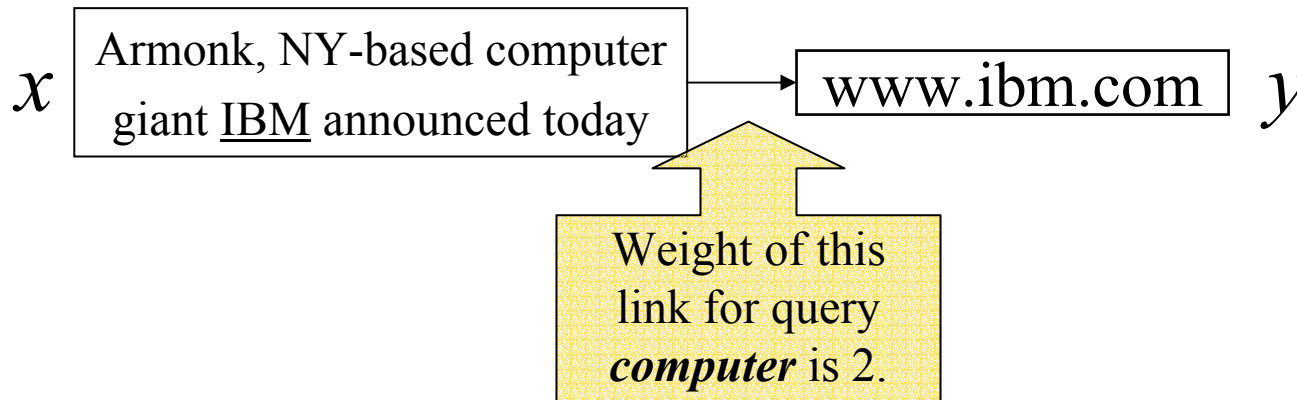
$$h(x) = \sum_{x \mapsto y} w(x, y) \cdot a(y)$$

$$a(x) = \sum_{y \mapsto x} w(x, y) \cdot h(y)$$

weighting anchor text



- What is $w(x,y)$?
- Should increase with the number of query terms in anchor text.
 - E.g.: $1 + \text{number of query terms}$.

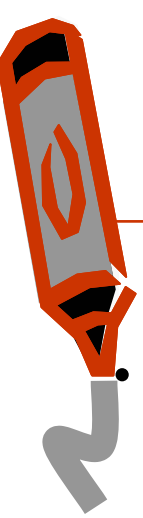




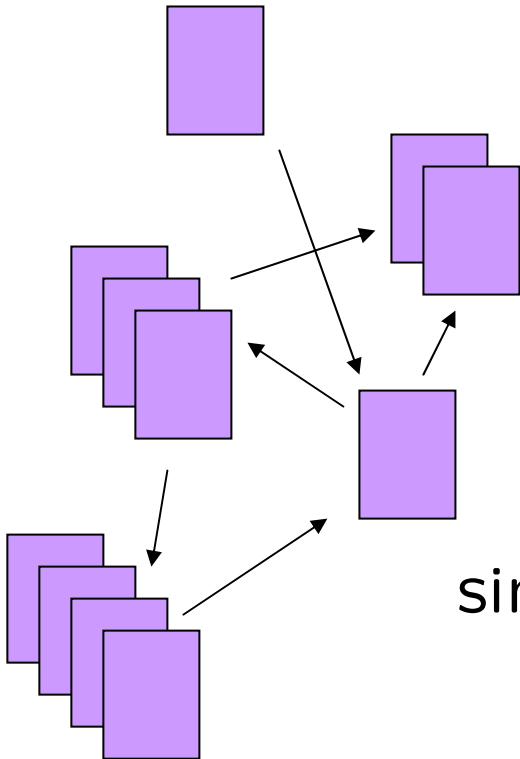
Weighted hub/authority computation

- Recall basic algorithm:
 - Iteratively update all $h(x)$, $a(x)$;
 - After iteration, output pages with
 - highest $h()$ scores as top hubs
 - highest $a()$ scores as top authorities.
- Now use weights in iteration.
- Raises scores of pages with “heavy” links.

Web sites, not pages



- Lots of pages in a site give varying aspects of information on the same topic.



Treat portions of web-sites as a single entity for score computations.



Link neighborhoods

- Links on a page tend to point to the same topics as neighboring links.
 - Break pages down into *pagelets* (say separate by tags)
 - compute a hub/authority score for each pagelet.
- **Example**
 - Ron Fagin's links
 - Logic links
 - Moshe Vardi's logic page
 - International logic symposium
 - Paper on modal logic
-
 - My favorite football team
 - The 49ers
 - Why the Raiders suck
 - Steve's homepage
 - The NFL homepage

comparison



Pagerank

Pros

- Hard to spam
- Computes quality signal for all pages

Cons

- Non-trivial to compute
- Not query specific
- Doesn't work on small graphs

Proven to be effective for general purpose ranking

HITS & Variants

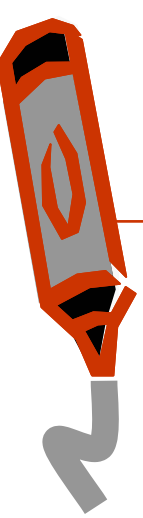
Pros

- Easy to compute, real-time execution is hard [Bhar98b, Stat00]
- Query specific
- Works on small graphs

Cons

- Local graph structure can be manufactured (spam!)
- Provides a signal only when there's direct connectivity (e.g., home pages)

Well suited for supervised directory construction



www IR

- world wide web
- google, page rank
- markov chains
- HITS link analysis
- behavior-based web search
- crawling, indexing the web
- duplicates, mirrors and spam
- www infrastructure
- www size
- cache, hardware, systems

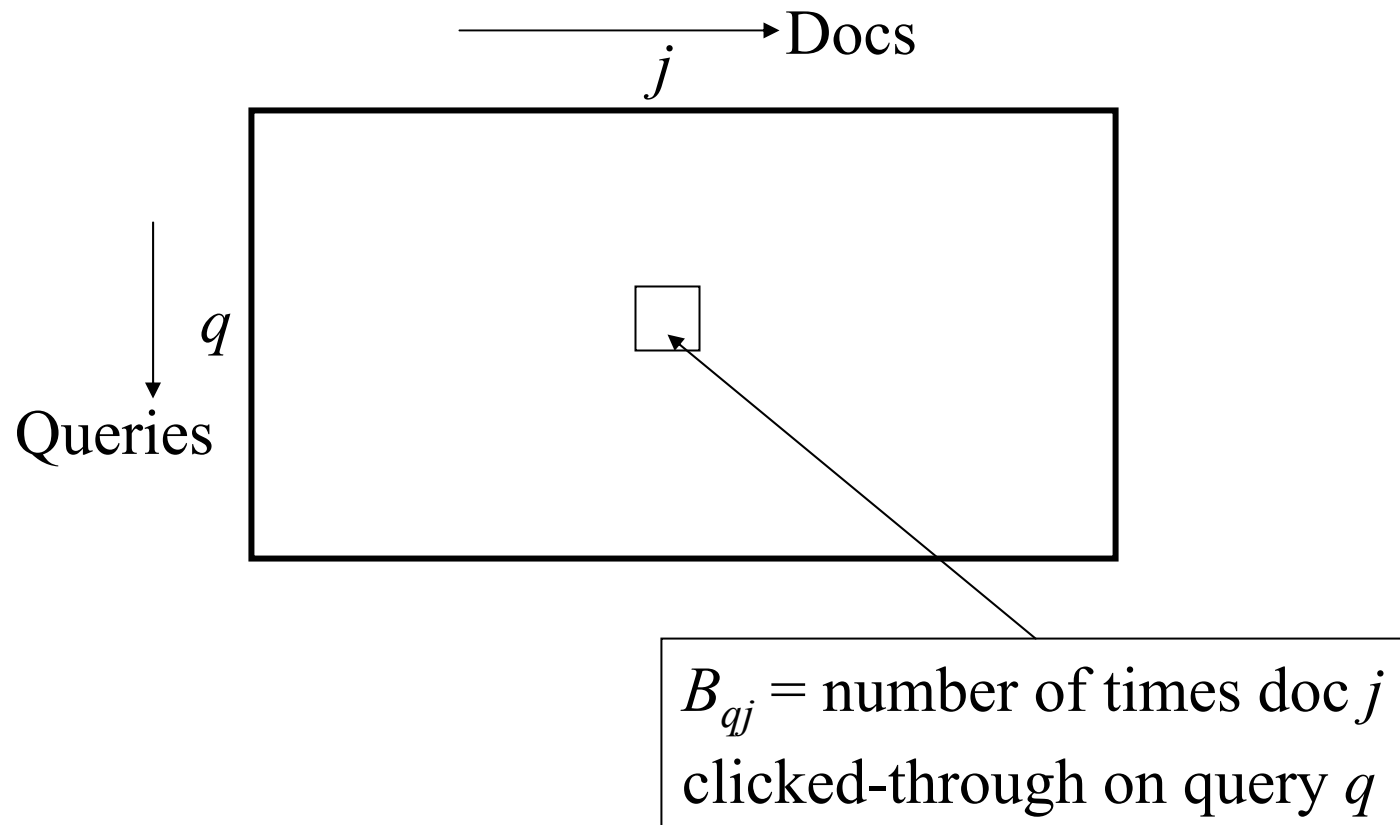


behavior-based ranking

- For each query Q , keep track of which docs in the results are clicked on
- On subsequent requests for Q , re-order docs in results based on click-throughs
- First due to DirectHit → AskJeeves
- Relevance assessment based on
 - Behavior/usage
 - vs. content



Query-doc popularity matrix B



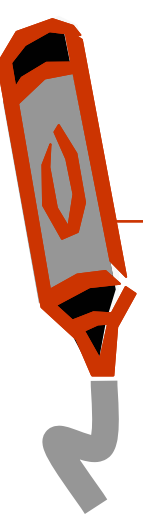
When query q is issued again, order docs by B_{qj} values.



vector space implementation

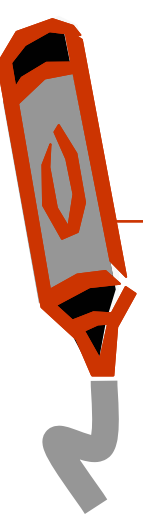
- Maintain a term-doc popularity matrix C
 - as opposed to query-doc popularity
 - initialized to all zeros
- Each column represents a doc j
 - If doc j clicked on for query q , update $C_j \leftarrow C_j + \varepsilon q$ (here q is viewed as a vector).
- On a query q' , compute its cosine proximity to C_j for all j .
- Combine this with the regular text score.

Issues

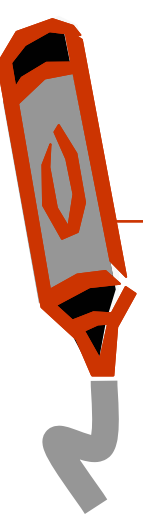


- Normalization of C_j after updating
- Assumption of query compositionality
 - “white house” document popularity derived from “white” and “house”
- Updating - live or batch?
- Basic assumption
 - Relevance can be directly measured by number of click throughs
 - Valid?
 - Click through to docs that turn out to be non-relevant: what does a click mean?
 - Self-perpetuating ranking
 - Spam
 - All votes count the same
 - More on this in recommendation systems

Variants

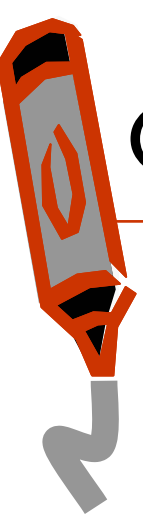


- Time spent viewing page
 - Difficult session management
 - Inconclusive modeling so far
- Does user back out of page?
- Does user stop searching?
- Does user transact?



www IR

- world wide web
- google, page rank
- markov chains
- HITS link analysis
- behavior-based web search
- crawling, indexing the web
- duplicates, mirrors and spam
- www infrastructure
- www size
- cache, hardware, systems



Crawling and Corpus Construction

- Crawl order
- Filtering duplicates
- Mirror detection



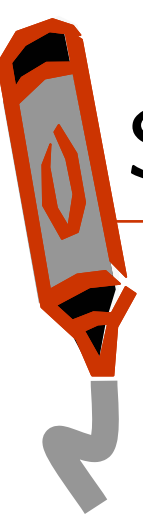
Crawling Issues

- How to crawl?
 - *Quality*: “Best” pages first
 - *Efficiency*: Avoid duplication (or near duplication)
 - *Etiquette*: Robots.txt, Server load concerns
- How much to crawl? How much to index?
 - *Coverage*: How big is the Web? How much do we cover?
 - *Relative Coverage*: How much do competitors have?
- How often to crawl?
 - *Freshness*: How much has changed?
 - How much has really changed? (why is this a different question?)



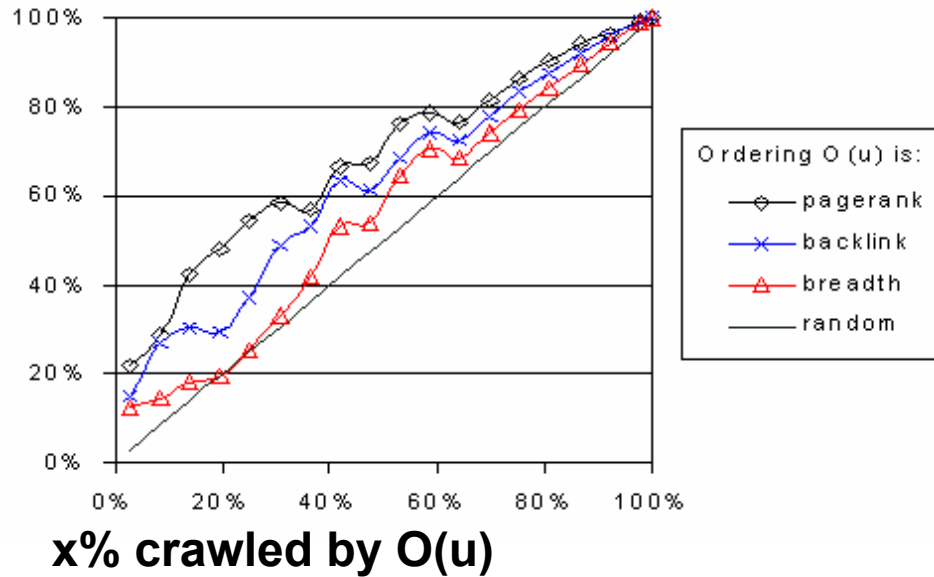
Crawl Order

- Best pages first
 - Potential quality measures:
 - Final Indegree
 - Final Pagerank
 - Crawl heuristic:
 - BFS
 - Partial Indegree
 - Partial Pagerank
 - Random walk

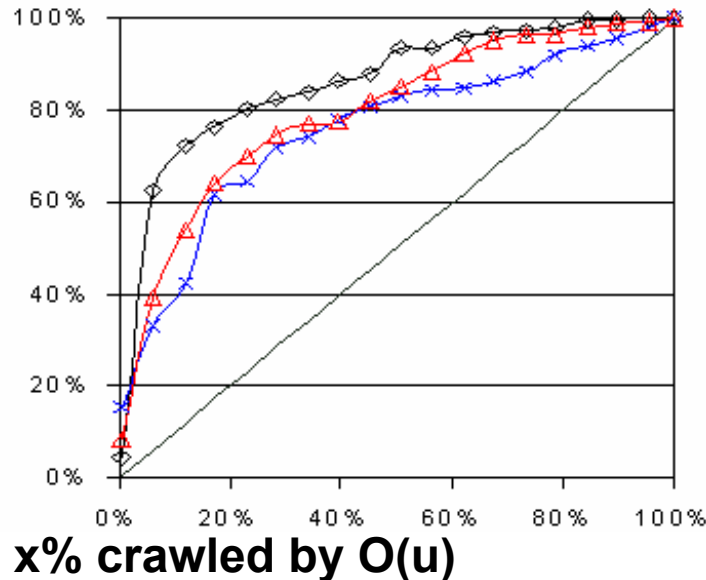


Stanford Web Base (179K, 1998)

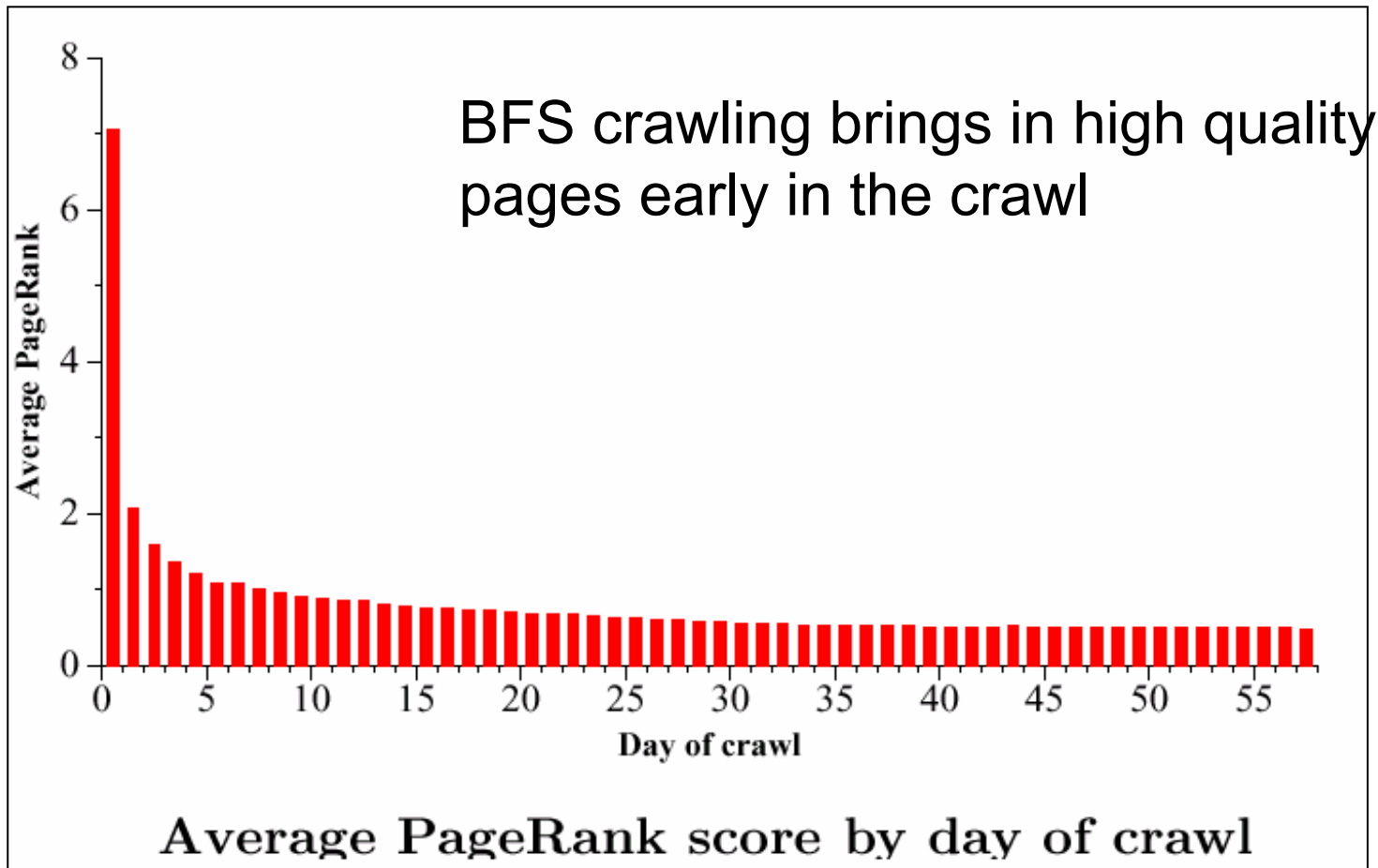
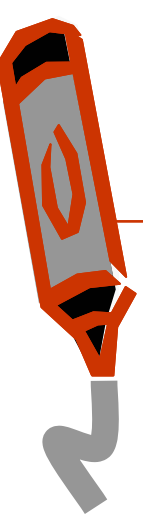
**Perc.
overlap
with
best
x%
by
indegree**



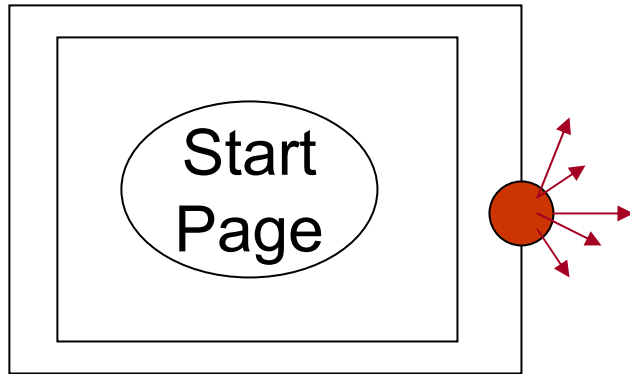
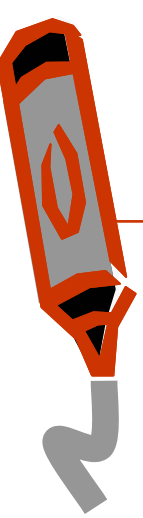
**Perc.
overlap
with
best
x%
by
pagerank**



Web Wide Crawl (328M pages, 2000)



BFS and spam

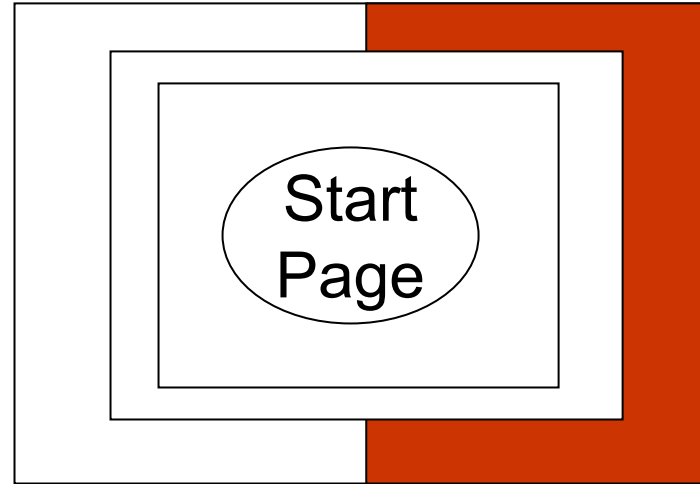


BFS depth = 2

Normal avg outdegree = 10

100 URLs on the queue
including a spam page.

Assume the spammer is
able to generate dynamic
pages with 1000 outlinks



BFS depth = 3

2000 URLs on the queue
50% belong to the spammer

BFS depth = 4

1.01 million URLs on the
queue
99% belong to the spammer

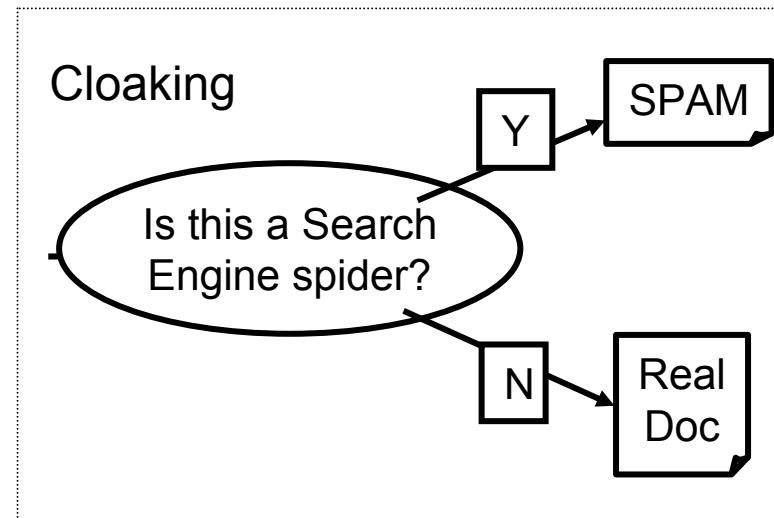


Adversarial IR (Spam)

- Motives
 - Commercial, political, religious, lobbies
 - Promotion funded by advertising budget
- Operators
 - Contractors (Search Engine Optimizers) for lobbies, companies
 - Web masters
 - Hosting services
- Forum
 - Web master world (www.webmasterworld.com)
 - Search engine specific tricks
 - Discussions about academic papers 😊

A few spam technologies

- Cloaking
 - Serve fake content to search engine robot
 - *DNS cloaking*: Switch IP address. Impersonate
- Doorway pages
 - Pages optimized for a single keyword that re-direct to the real target page
- Keyword Spam
 - Misleading meta-keywords, excessive repetition of a term, fake “anchor text”
 - Hidden text with colors, CSS tricks, etc.
- Link spamming
 - Mutual admiration societies, hidden links, awards
 - *Domain flooding*: numerous domains that point or re-direct to a target page
- Robots
 - Fake click stream
 - Fake query stream
 - Millions of submissions via Add-Url



Meta-Keywords =

“... London hotels, hotel, holiday inn, hilton, discount, booking, reservation, sex, mp3, britney spears, viagra, ...”

Can you trust words on the page?

auctions.hitsoffice.com/



**Pornographic
Content**

www.ebay.com/



Examples from July 2002



- Editors Chunder On
- JavaScript Weenie
- Windows Weenie
- Wacky HTML
- Site Search
- Ecommerce
- Web Tools
- Web Audio
- Propheads
- Ponytails
- Suits

The latest tips.

New Search Engine Marketing Practices

by David Gikandi

A study by Berrier Associates indicates that people who spend five or more hours a week online spend about 71% of their time searching for information. That goes to show the power search engines still wield over traffic. To keep you up to date on what online marketing professionals are now doing to win the **search engine wars** here is a brief look at some of the latest strategies being employed.

August 2, 2000

Search Engine Optimization I
Adversarial IR
("search engine wars")

...got a
**COMPUTER
QUESTION?**



Tutorial: Cloaking and Stealth Technology

Featured as an ongoing multi part section in our newsletter, we are offering you all the stuff you need to know, straight from the horse's mouth. Learn the secrets of the pros – subscription terminated anytime you wish.

“Stealth, Cloaking, Phantom Tech

FAQ

- [What are Ghost Pages?](#)
- [What are Doorway Pages, then?](#)
- [And Hallway Pages?](#)
- [How are cloaked pages submitted?](#)
- [How about changing stealth pages?](#)
- [What are the mechanics of cloaking?](#)
- [What's a key switch?](#)
- [Isn't this really simple redirection technique?](#)
- [What about penalization?](#)

fantomas **go!**
spiderSpy™

The botBase

Don't risk nasty surprises from spiders sneaking on your site under wraps!

Sure, they tend to add and switch engines, IPs and User Agents almost all the time, and keeping up with their antics is a grueling task at best. But it's also a fact that professional traffic evaluation, stealthing technology and even page submission management depend on reliable search engine reference data, if you don't want to waste your valuable resources on inventing the wheel over and

Search Engine Optimization II
Tutorial on
Cloaking & Stealth
Technology



The war against spam

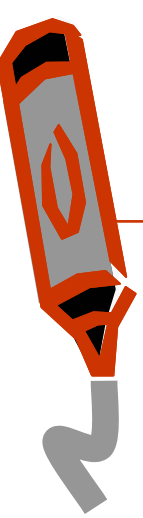
- Quality signals - Prefer authoritative pages based on:
 - Votes from authors (linkage signals)
 - Votes from users (usage signals)
- Policing of URL submissions
 - Anti robot test
- Limits on meta-keywords
- Robust link analysis
 - Ignore statistically implausible linkage (or text)
 - Use link analysis to detect spammers (guilt by association)
- Spam recognition by machine learning
 - Training set based on known spam
- Family friendly filters
 - Linguistic analysis, general classification techniques, etc.
 - For images: flesh tone detectors, source text analysis, etc.
- Editorial intervention
 - Blacklists
 - Top queries audited
 - Complaints addressed



duplicates/near duplicates detection

- *Duplication*: Exact match with fingerprints
- *Near-Duplication*: Approximate match
 - Compute syntactic similarity with an edit-distance measure
 - Use similarity threshold to detect near-duplicates
 - E.g., Similarity > 80% => Documents are “near duplicates”
 - Not transitive though sometimes used transitively

near similarity



– Features:

- Segments of a document (natural or artificial breakpoints) [Brin95]
- *Shingles* (Word N-Grams) [Brin95, Brod98]

“a rose is a rose is a rose” =>

a_rose_is_a

rose_is_a_rose

is_a_rose_is

– Similarity Measure

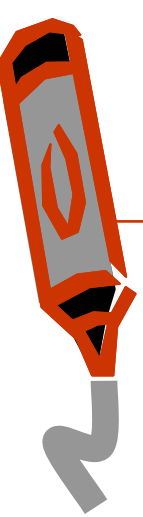
- TFIDF [Shiv95]
- Set intersection [Brod98]
(Specifically, $\text{Size_of_Intersection} / \text{Size_of_Union}$)



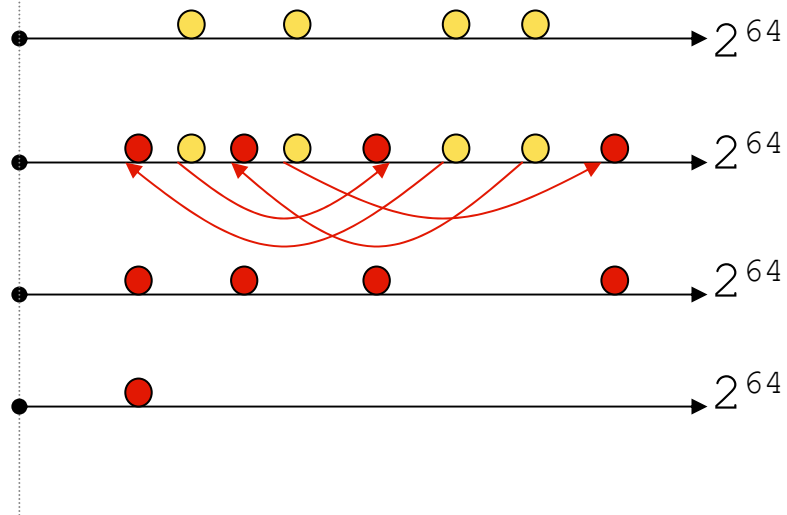
Shingles + Set Intersection

- Computing exact set intersection of shingles between all pairs of documents is expensive and infeasible
 - Approximate using a cleverly chosen subset of shingles from each (a sketch)
- Estimate `size_of_intersection / size_of_union` based on a short sketch ([Brod97, Brod98])
 - Create a “sketch vector” (e.g., of size 200) for each document
 - Documents which share more than t (say 80%) corresponding vector elements are `similar`
 - For doc D , `sketch[i]` is computed as follows:
 - Let f map all shingles in the universe to $0..2^m$ (e.g., $f =$ fingerprinting)
 - Let p_i be a specific random permutation on $0..2^m$
 - Pick `sketch[i] := MIN p_i (f(s))` over all shingles s in D

Computing Sketch[i] for doc1



Document 1

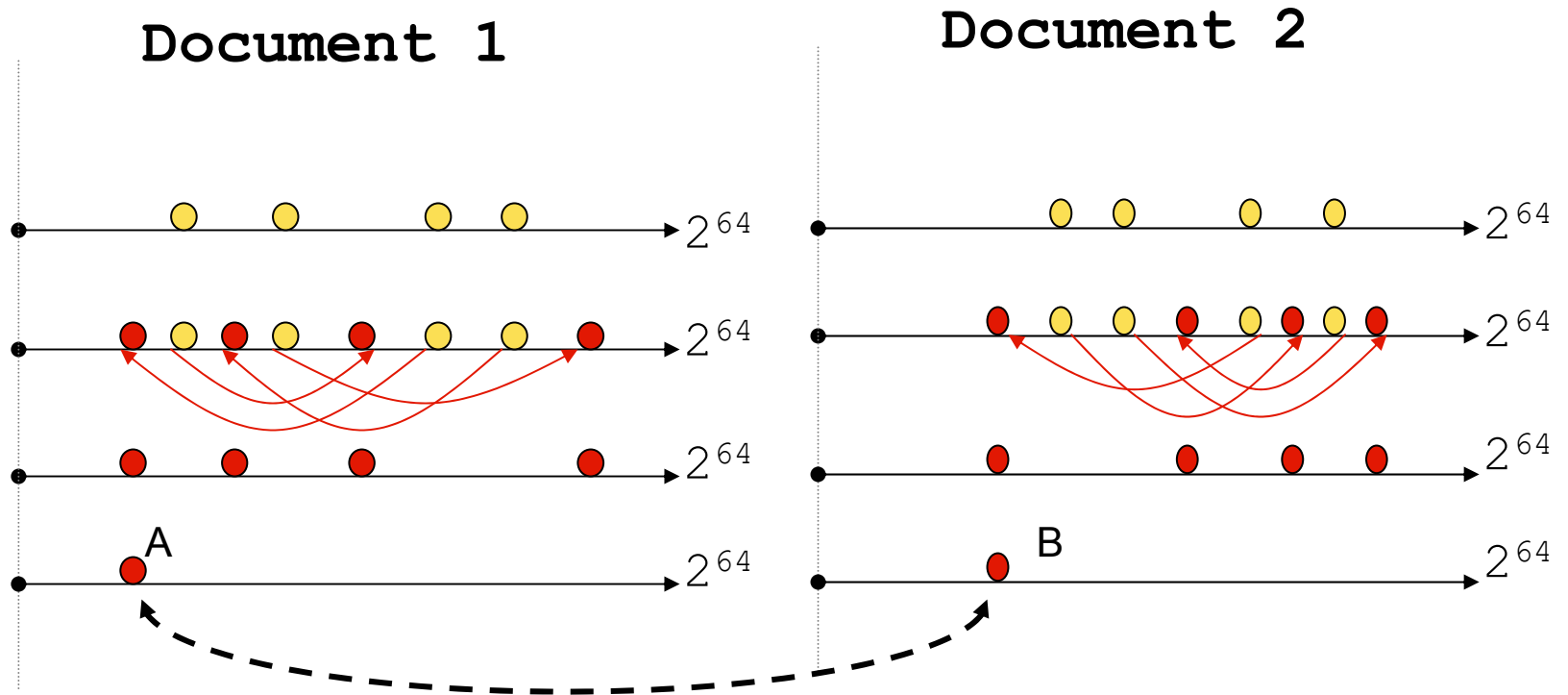
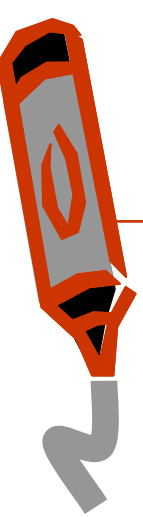


Start with 64 bit shingles

Permute on the number line
with π_i

Pick the min value

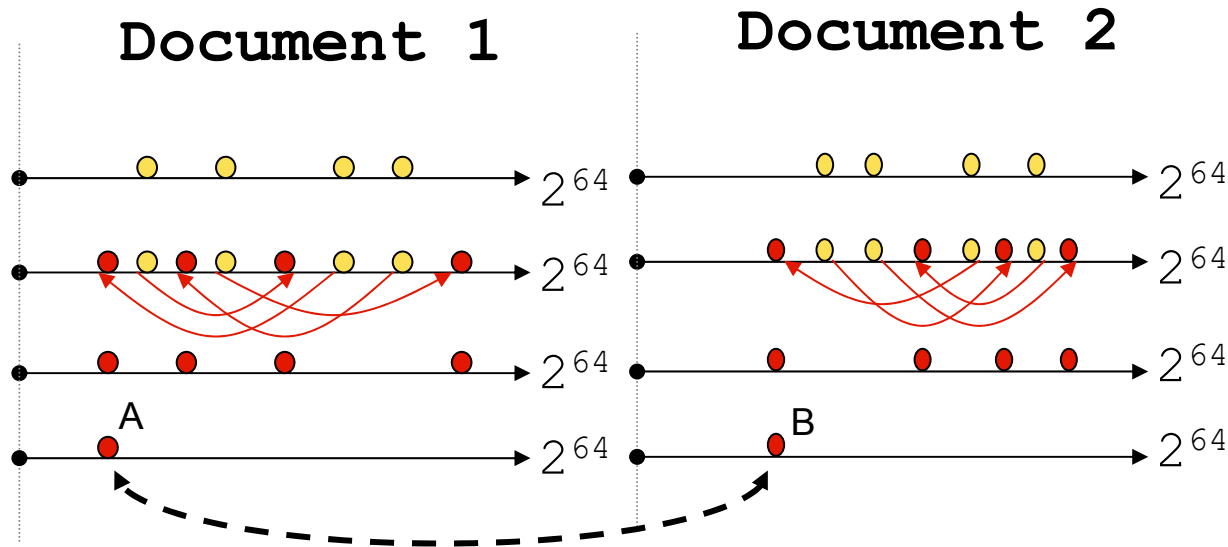
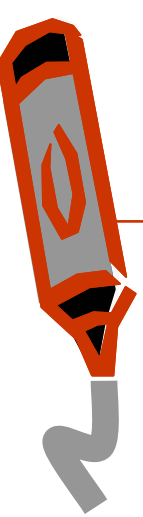
Sketch comparison



Are these equal?

Test for 200 random permutations: $\pi_1, \pi_2, \dots, \pi_{200}$

Sketch comparison



$A = B$ iff the shingle with the MIN value in the union of Doc1 and Doc2 is common to both (i.e., lies in the intersection)

This happens with probability:

$$\text{Size_of_intersection} / \text{Size_of_union}$$

mirrors



- Mirroring is systematic replication of web pages across hosts.
 - Single largest cause of duplication on the web
- **Host1/a** and **Host2/b** are mirrors iff
 - For all (or most) paths p such that when
 - $\text{http://Host1/ a / p}$ exists
 - $\text{http://Host2/ b / p}$ exists as wellwith identical (or near identical) content, and vice versa.



mirror detection

- <http://www.elsevier.com/> and <http://www.elsevier.nl/>
- Structural Classification of Proteins
 - <http://scop.mrc-lmb.cam.ac.uk/scop>
 - <http://scop.berkeley.edu/>
 - <http://scop.wehi.edu.au/scop>
 - <http://pdb.weizmann.ac.il/scop>
 - <http://scop.protres.ru/>

mirrors: repackaged

Auctions.msn.com

Auctions.lycos.com

Location: <http://auctions.msn.com/HTML/Cat17065/Page1.htm?CatNo=9>

Antiques
select parameters below to search antiques listings.

sort by

[Can't find it? Try the Auction Age](#)

[Narrow Your Search](#)

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 [Next](#)>

Title	Status	Bids	Price
~Flow Blue Cake Plate With Pedestal~Gorgeous!!!	A	5	\$50.00
~Flow Blue Taureen With Soup Spoon~Gorgeous~ All Porcelain~*...	R	3	\$55.00
Vintage Swiss Silver Case Pocket Watch by Remontoir	R	1	\$30.00
One Nina & Three Rara Kuyu Paintings	A	-	\$20.00
0b2150502 / GORGEOUS HANDICRAFT TEAKWOOD ELEPHANT NCS152	A	-	\$75.98
0b2151103 / BEAUTIFUL HAND MADE TEAKWOOD ELEPHANT NCS152	A	-	\$75.98

Bookmarks Location: <http://auctions.lycos.com/HTML/Cat8835/Page1.htm?CatNo=9>

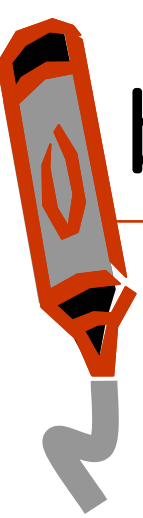
Antiques

Featured Items

	~Flow Blue Cake Plate With Pedestal~Gorgeous!!!	Current Bid: \$50.00	Auction Ends 8/18/01 11:00 PM
	~Flow Blue Taureen With Soup Spoon~Gorgeous~ All Porcelain~*	Current Bid: \$55.00	Auction Ends 8/18/01 10:40 PM
	Vintage Swiss Silver Case Pocket Watch by Remontoir	Current Bid: \$30.00	Auction Ends 8/18/01 1:00 AM
	One Nina & Three Rara Kuyu Paintings	Current Bid: \$20.00	Auction Ends 8/17/01 11:00 PM
	0b2150502 / GORGEOUS HANDICRAFT TEAKWOOD ELEPHANT NCS152	Current Bid: \$75.98	Auction Ends 8/18/01 1:00 AM

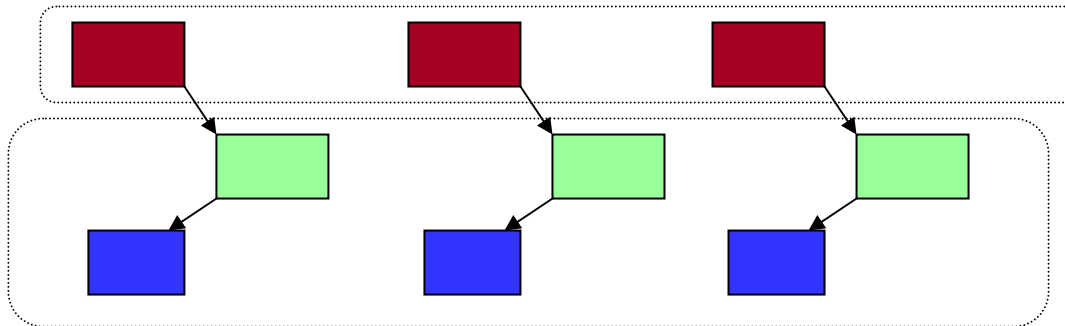
mirrors

- Why detect mirrors?
 - Smart crawling
 - Fetch from the fastest or freshest server
 - Avoid duplication
 - Better connectivity analysis
 - Combine inlinks
 - Avoid double counting outlinks
 - Redundancy in result listings
 - “If that fails you can try: ;mirror;/samepath”
 - Proxy caching



bottom up mirror detection

- Maintain clusters of subgraphs
- Initialize clusters of trivial subgraphs
 - Group near-duplicate single documents into a cluster
- Subsequent passes
 - Merge clusters of the same cardinality and corresponding linkage



- Avoid decreasing cluster cardinality
- To detect mirrors we need:
 - Adequate path overlap
 - Contents of corresponding pages within a small time range



top down mirror detection

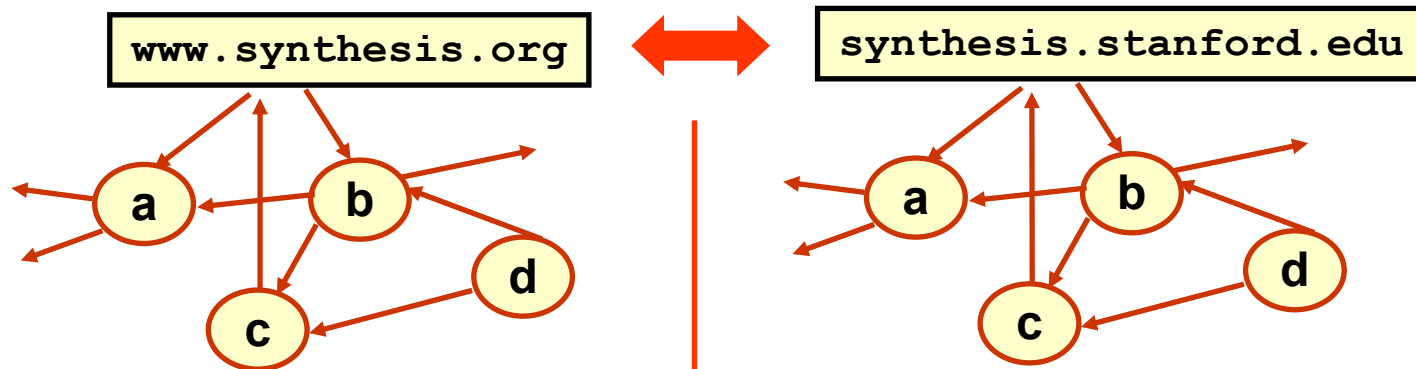
E.g.,

www.synthesis.org/Docs/ProjAbs/synsys/synanalysis.html

synthesis.stanford.edu/Docs/ProjAbs/synsys/quant-dev-new-teach.html

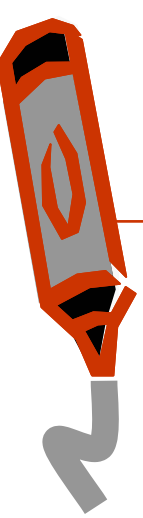
- What features could indicate mirroring?
 - Hostname similarity:
 - word unigrams and bigrams: – `www`, `www.synthesis`, `synthesis`, ...”
 - Directory similarity:
 - Positional path bigrams – `0:Docs/ProjAbs`, `1:ProjAbs/synsys`, ... ”
 - IP address similarity:
 - 3 or 4 octet overlap
 - Many hosts sharing an IP address => virtual hosting by an ISP
 - Host outlink overlap
 - Path overlap
 - Potentially, path + sketch overlap

mirror detection by urls



www.synthesis.org/Docs/ProjAbs/synsys/synanalysis.html
www.synthesis.org/Docs/ProjAbs/synsys/visual-semi-quant.html
www.synthesis.org/Docs/annual.report96.final.html
www.synthesis.org/Docs/cicee-berlin-paper.html
www.synthesis.org/Docs/myr5
www.synthesis.org/Docs/myr5/cicee/bridge-gap.html
www.synthesis.org/Docs/myr5/cs/cs-meta.html
www.synthesis.org/Docs/myr5/mech/mech-intro-mechatronics.html
www.synthesis.org/Docs/myr5/mech/mech-take-home.html
www.synthesis.org/Docs/myr5/synsys/experiential-learning.html
www.synthesis.org/Docs/myr5/synsys/mm-mech-dissec.html
www.synthesis.org/Docs/yr5ar
www.synthesis.org/Docs/yr5ar/assess
www.synthesis.org/Docs/yr5ar/cicee
www.synthesis.org/Docs/yr5ar/cicee/bridge-gap.html
www.synthesis.org/Docs/yr5ar/cicee/comp-integ-analysis.html

synthesis.stanford.edu/Docs/ProjAbs/deliv/high-tech-...
synthesis.stanford.edu/Docs/ProjAbs/mech/mech-enhanced-...
synthesis.stanford.edu/Docs/ProjAbs/mech/mech-intro-...
synthesis.stanford.edu/Docs/ProjAbs/mech/mech-mm-case-...
synthesis.stanford.edu/Docs/ProjAbs/synsys/quant-dev-new-...
synthesis.stanford.edu/Docs/annual.report96.final.html
synthesis.stanford.edu/Docs/annual.report96.final_fn.html
synthesis.stanford.edu/Docs/myr5/assessment
synthesis.stanford.edu/Docs/myr5/assessment/assessment-...
synthesis.stanford.edu/Docs/myr5/assessment/mm-forum-kiosk-...
synthesis.stanford.edu/Docs/myr5/assessment/neato-ucb.html
synthesis.stanford.edu/Docs/myr5/assessment/not-available.html
synthesis.stanford.edu/Docs/myr5/cicee
synthesis.stanford.edu/Docs/myr5/cicee/bridge-gap.html
synthesis.stanford.edu/Docs/myr5/cicee/cicee-main.html
synthesis.stanford.edu/Docs/myr5/cicee/comp-integ-analysis.html



www IR

- world wide web
- google, page rank
- markov chains
- HITS link analysis
- behavior-based web search
- crawling, indexing the web
- duplicates, mirrors and spam
- www infrastructure
- www size
- cache, hardware, systems



www infrastructure

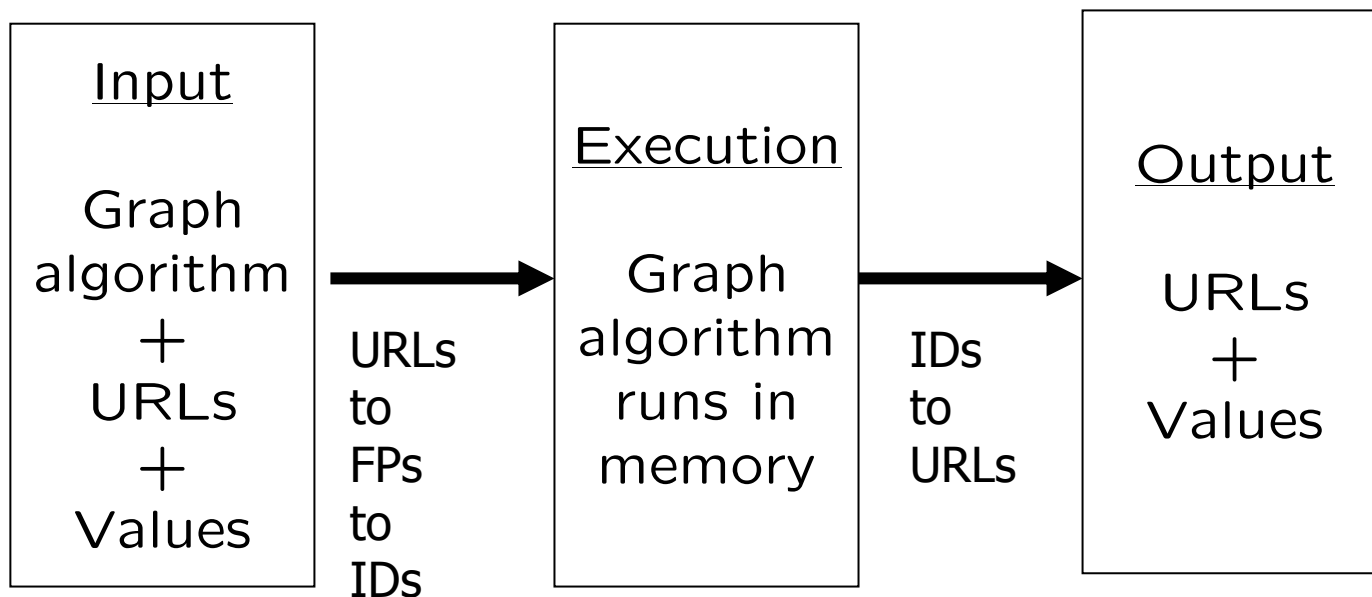
- Connectivity Server
 - Fast access to links to support for link analysis
- Term Vector Database
 - Fast access to document vectors to augment link analysis



connectivity server

- Fast web graph access to support connectivity analysis
- Stores mappings in memory from
 - URL to outlinks, URL to inlinks
- Applications
 - HITS, Pagerank computations
 - Crawl simulation
 - Graph algorithms: web connectivity, diameter etc.
 - more on this later
 - Visualizations

connectivity server



Translation Tables on Disk

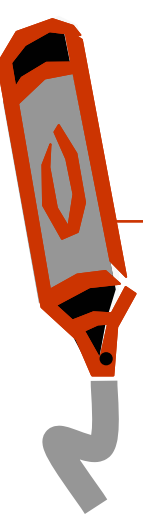
URL text: 9 bytes/URL (compressed from ~80 bytes)

FP(64b) -> ID(32b): 5 bytes

ID(32b) -> FP(64b): 8 bytes

ID(32b) -> URLs: 0.5 bytes

ID assignment



Partition URLs into 3 sets,
sorted lexicographically

- High: Max degree > 254
- Medium: $254 > \text{Max degree} > 24$
- Low: remaining (75%)

- IDs assigned in sequence
(densely)

Adjacency lists

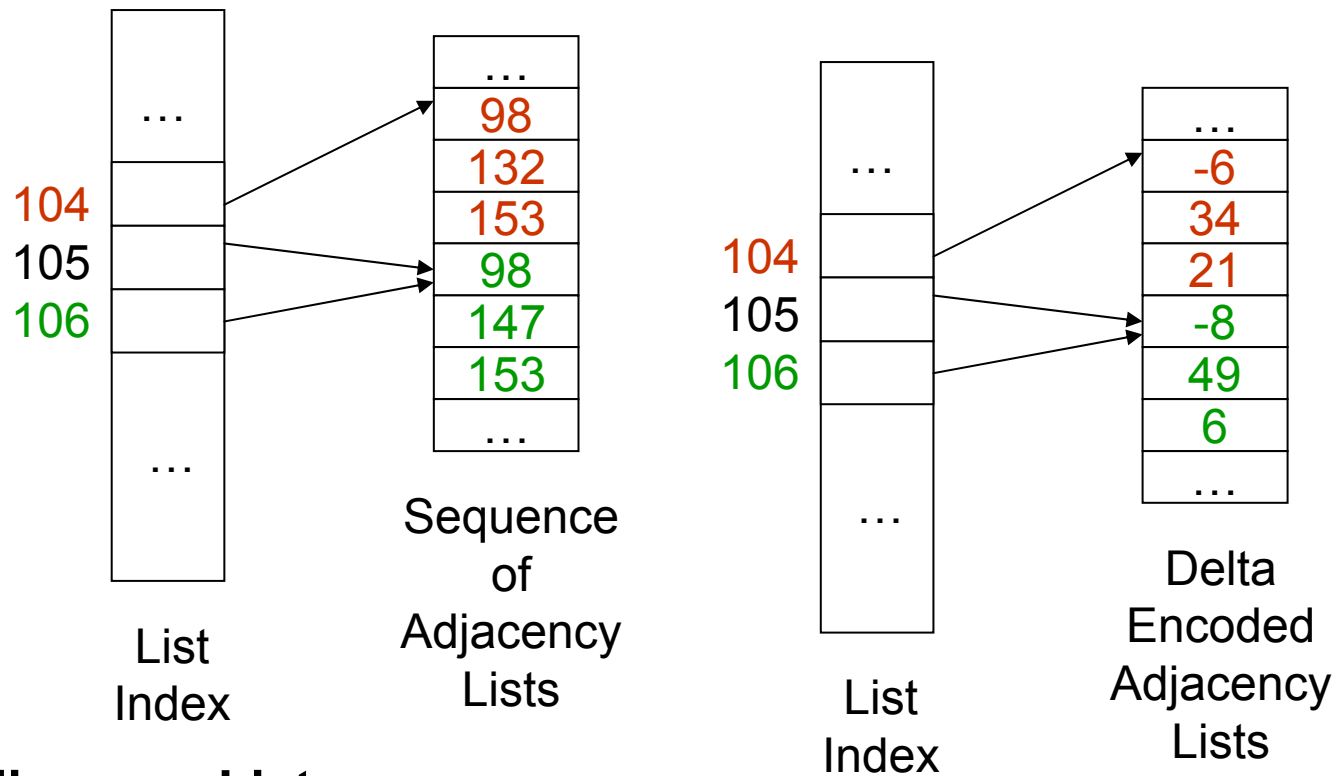
- In memory tables for Outlinks,
Inlinks
- List index maps from a Source
ID to start of adjacency list

E.g., HIGH IDs:

Max(indegree , outdegree) > 254

ID	URL
...	
9891	www.amazon.com/
9912	www.amazon.com/jobs/
...	
9821878	www.geocities.com/
...	
40930030	www.google.com/
...	
85903590	www.yahoo.com/

Adjacency List Compression - I



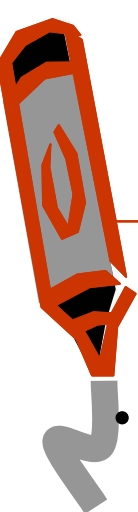
- **Adjacency List:**

- Smaller delta values are exponentially more frequent (80% to same host)
- Compress deltas with variable length encoding (e.g., Huffman)

- **List Index pointers:** 32b for high, Base+16b for med, Base+8b for low

- Avg = 12b per pointer

Adjacency List Compression - II



- Inter List Compression

- Basis: Similar URLs may share links
 - Close in ID space => adjacency lists may overlap
- Approach
 - Define a representative adjacency list for a block of IDs
 - Adjacency list of a reference ID
 - Union of adjacency lists in the block
 - Represent adjacency list in terms of deletions and additions *when it is cheaper to do so*
- Measurements
 - Intra List + Starts: 8-11 bits per link (580M pages/16GB RAM)
 - Inter List: 5.4-5.7 bits per link (870M pages/16GB RAM.)



Term Vector Database

- Fast access to 50 word term vectors for web pages
 - Term Selection:
 - Restricted to middle 1/3rd of lexicon by document frequency
 - Top 50 words in document by TF.IDF.
 - Term Weighting:
 - Deferred till run-time (can be based on term freq, doc freq, doc length)
- Applications
 - Content + Connectivity analysis (e.g., Topic Distillation)
 - Topic specific crawls
 - Document classification
- Performance
 - Storage: 33GB for 272M term vectors
 - Speed: 17 ms/vector on AlphaServer 4100 (latency to read a disk block)



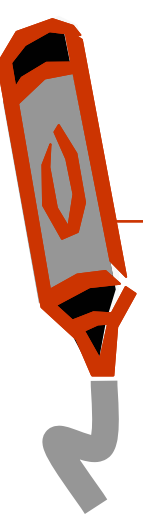
the size of the web

- Issues
 - The web is really infinite
 - Dynamic content, e.g., calendar
 - Static web contains syntactic duplication, mostly due to mirroring (~20-30%)
 - Some servers are seldom connected
- Who cares?
 - Media, and consequently the user
 - Engine design
 - Engine crawl policy. Impact on recall.



what to measure

- The relative size of search engines
 - The notion of a page being indexed is *still* reasonably well defined.
 - Already there are problems
 - Document extension: e.g. Google indexes pages not yet crawled by indexing anchor text.
 - Document restriction: Some engines restrict what is indexed (first n words, only relevant words, etc.)
- The coverage of a search engine relative to another particular crawling process.
- The ultimate coverage associated to a particular crawling process and a given list of seeds.



www size: statistical measures

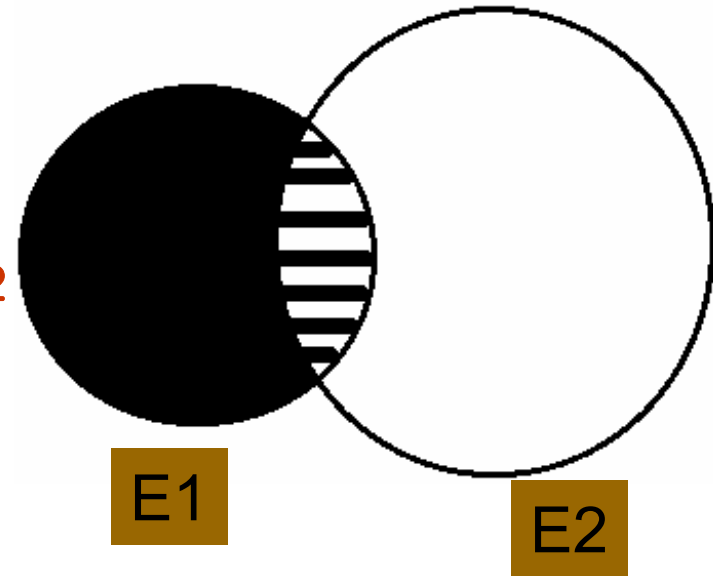
- Random queries
- Random searches
- Random IP addresses
- Random walks

URL sampling via Random Queries

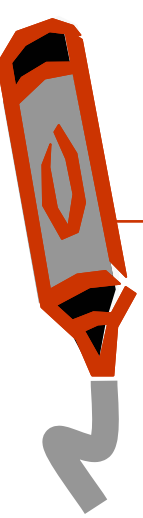
- Ideal strategy: Generate a random URL and check for containment in each index.
- Problem: **Random URLs are hard to find!**
- Sample URLs randomly from each engine
 - 20,000 random URLs from each engine
 - Issue random conjunctive query with ≥ 200 results
 - Select a random URL from the top 200 results
- Test if present in other engines.
 - Query with 8 rarest words. Look for URL match
- Compute intersection & size ratio

Intersection = $x\%$ of E1 = $y\%$ of E2

$$E1/E2 = y/x$$



- Issues
 - Random narrow queries may bias towards long documents (Verify with disjunctive queries)
 - Other biases induced by process

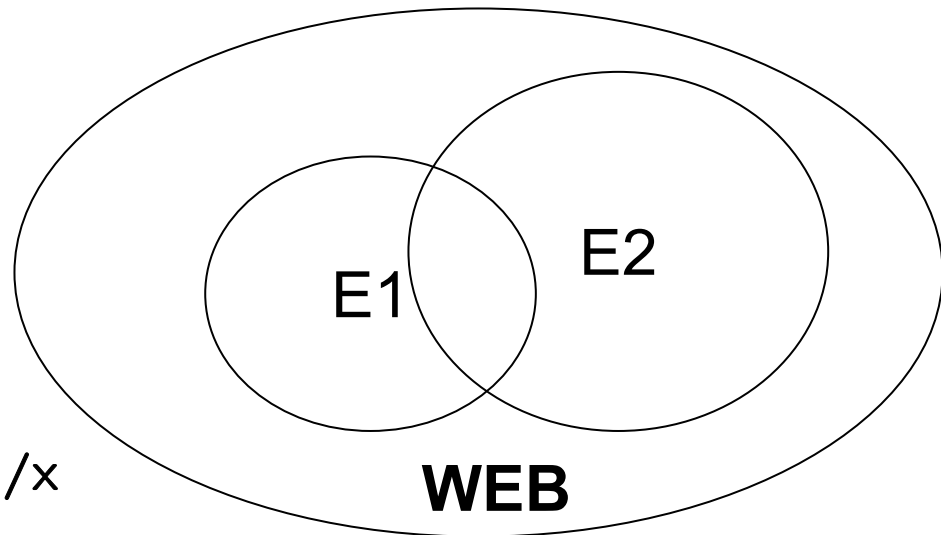


www size: estimation

- Capture – Recapture technique
 - Assumes engines get independent random subsets of the Web

E2 contains $x\%$ of E1.
Assume, E2 contains $x\%$
of the Web as well

Knowing size of E2
compute size of the Web
Size of the Web = $100 * E2 / x$



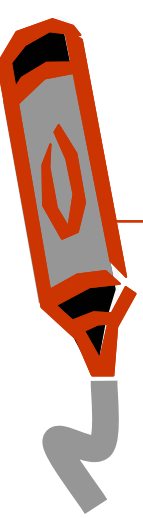
Bharat & Broder. 200 M (Nov 97), 275 M (Mar 98)
Lawrence & Giles. 320 M (Dec 97)



Random Searches

- Choose random searches extracted from a local log [Lawr97] or build “random searches” [Note02]
 - Use only queries with small results sets.
 - Count normalized URLs in result sets.
 - Use ratio statistics
- Advantage:
 - Might be a good reflection of the human perception of coverage
- 575 & 1050 queries from the NEC RI employee logs
- 6 Engines in '98, 11 in '99
- Implementation:
 - Restricted to queries with ≤ 600 results in total
 - Counted URLs from each engine after verifying query match
 - Computed size ratio & overlap for individual queries
 - Estimated index size ratio & overlap by averaging over all queries
- Issues
 - Samples are correlated with source of log
 - Duplicates
 - Technical statistical problems (must have non-zero results, ratio average)

Queries from Lawrence and Giles study



- 1. adaptive access control
- 2. neighborhood preservation topographic
- 3. hamiltonian structures
- 4. right linear grammar
- 5. pulse width modulation neural
- 6. unbalanced prior probabilities
- 7. ranked assignment method
- 8. internet explorer favourites importing
- 9. karvel thornber
- 10. zili liu
- 11. softmax activation function
- 12. bose multidimensional system theory
- 13. gamma mlp
- 14. dvi2pdf
- 15. john oliensis
- 16. rieke spikes exploring neural
- 17. video watermarking
- 18. counterpropagation network
- 19. fat shattering dimension
- 20. abelson amorphous computing



Random IP addresses

- Generate random IP addresses
 - Find, if possible, a web server at the given address
 - Collect all pages from server
 - Advantages : Clean statistics, independent of any crawling strategy
- HTTP requests to random IP addresses
 - Ignored: empty or authorization required or excluded
 - [Lawr99] Estimated 2.8 million IP addresses running crawlable web servers (16 million total) from observing 2500 servers.
 - OCLC using IP sampling found 8.7 M hosts in 2001
 - Netcraft [Netc02] accessed 37.2 million hosts in July 2002
- [Lawr99] exhaustively crawled 2500 servers and extrapolated
 - Estimated size of the web to be 800 million
 - Estimated use of metadata descriptors:
 - Meta tags (keywords, description) in 34% of home pages, Dublin core metadata in 0.3%
- Issues
 - Virtual hosting
 - Server might not accept <http://102.93.22.15>,
 - No guarantee all pages are linked to root page
 - Power law for # pages/host generates bias



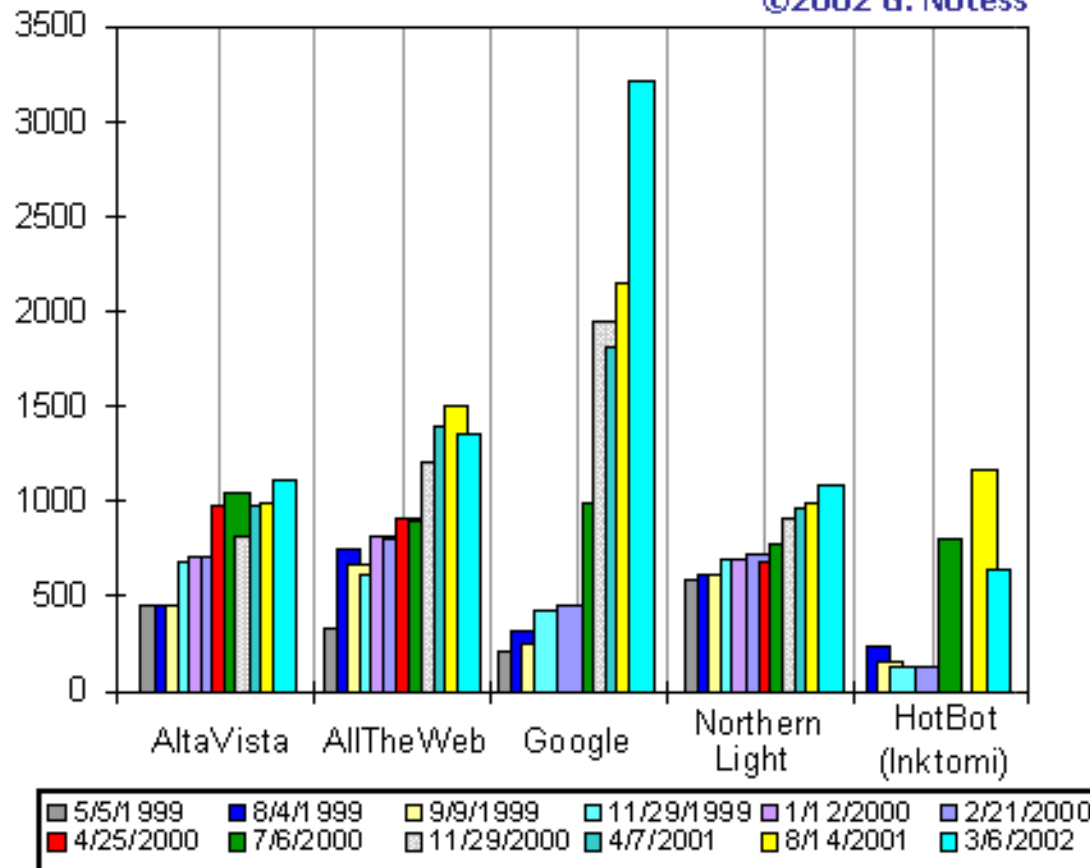
Random walks

- View the Web as a directed graph from a given list of seeds.
- Build a random walk on this graph
 - Includes various “jump” rules back to visited sites
 - Converges to a stationary distribution
 - Time to convergence not really known
 - Sample from stationary distribution of walk
 - Use the “strong query” method to check coverage by SE
 - “Statistically clean” method at least in theory!
 - Could work even for infinite web (assuming convergence) under certain metrics.
- Issues
 - List of seeds is a problem.
 - Practical approximation might not be valid
 - Non-uniform distribution, subject to link spamming
 - Still has all the problems associated with “strong queries”

www measurements (2002)

Top 5: Size Changes 5/99-3/2002 Results from same 8 searches

©2002 G. Notess



Source: <http://www.searchengineshowdown.com/stats/change.shtml>



www sampling: conclusions

- No sampling solution is perfect.
- Lots of new ideas ...
-but the problem is getting harder
- Quantitative studies are fascinating and a good research problem



Questions about the web graph

- How big is the graph?
 - How many links on a page (outdegree)?
 - How many links to a page (indegree)?
- Can one browse from any web page to any other?
How many clicks?
- Can we pick a random page on the web?
 - (Search engine measurement.)
- Can we exploit the structure of the web graph for searching and mining?
- What does the web graph reveal about social processes which result in its creation and dynamics?
- How different is browsing from a “random walk”?

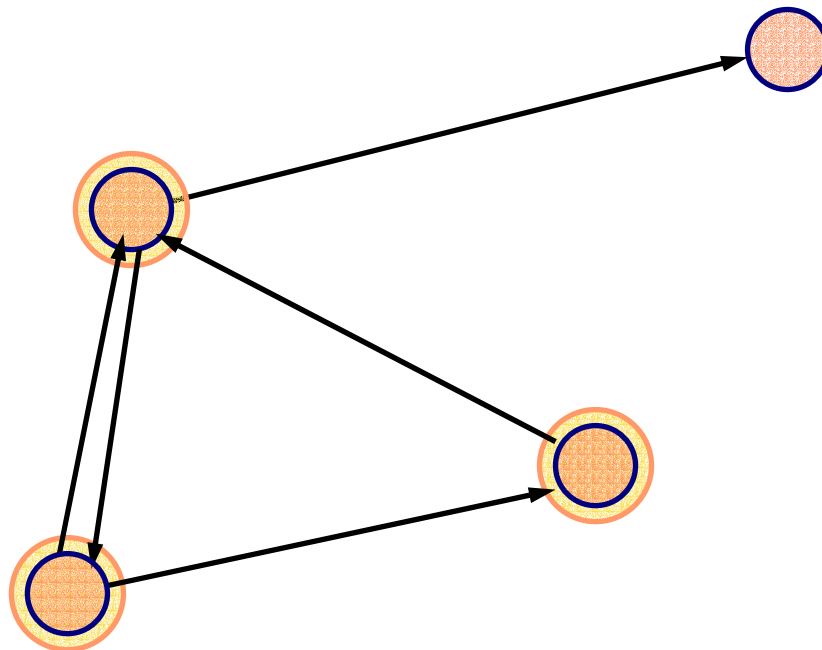


the web graph

- Exploit structure for Web algorithms
 - Crawl strategies
 - Search
 - Mining communities
- Classification/organization
- Web anthropology
 - Prediction, discovery of structures
 - Sociological understanding

algorithms

- Weakly connected components (WCC)
- Strongly connected components (SCC)
- Breadth-first search (BFS)
- Diameter

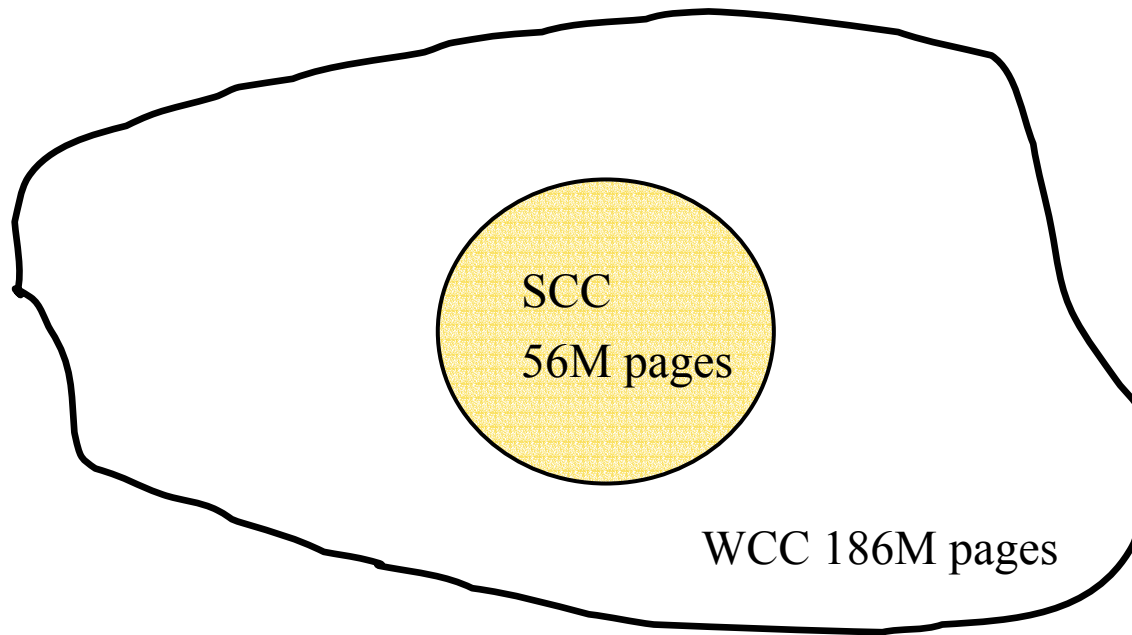
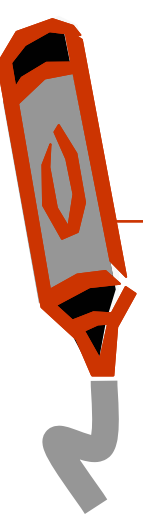




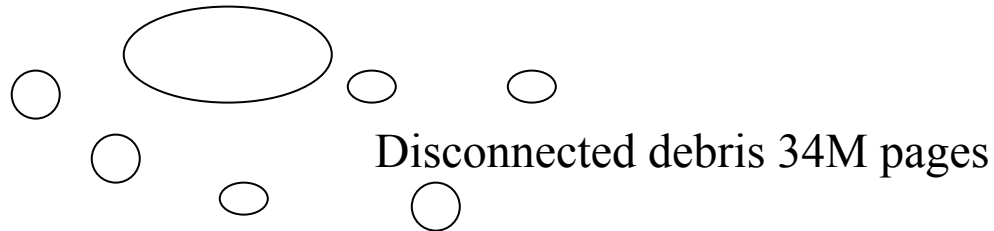
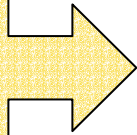
scale

- Typical diameter algorithm:
 - number of steps \sim pages \times links.
 - For 500 million pages, 5 billion links, even at a *very* optimistic $0.15\mu\text{s}/\text{step}$, we need ~ 4 billion seconds.
Hopeless.
- Will estimate diameter/distance metrics.
- On the other hand, can handle tasks linear in the links (5 billion) at $\sim 1 \mu\text{s}/\text{step}$.
 - E.g., breadth-first search
- First eliminate duplicate pages/mirrors.
- Linear-time implementations for WCC and SCC.

Tentative picture



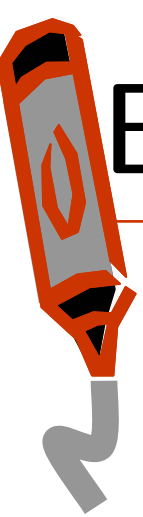
Where did
this come
from?





Breadth-first search (BFS)

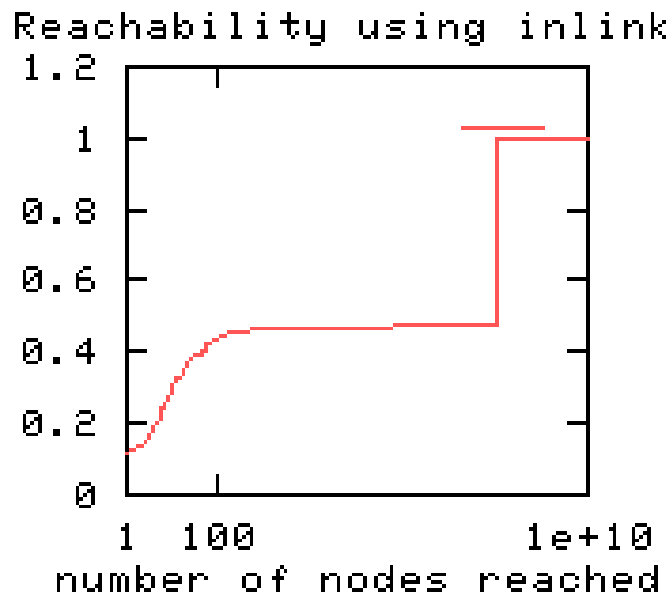
- Start at a page p
 - get its neighbors;
 - their neighbors, etc.
- Get profile of the number of pages reached by crawling out of p , as a function of distance d
- Can do this following links forwards as well as backwards
- Experiment
 - Start at 1000+ random pages
 - For each start page, build BFS (reachability vs. distance) profiles going forwards, and backwards



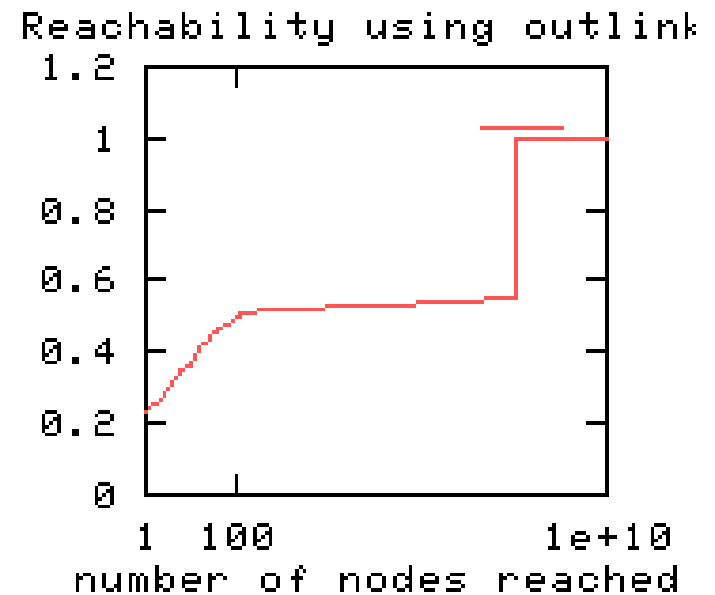
Breadth-first search (BFS)

How many pages are reachable from a random page?

frac. of starting node:



frac. of starting node:

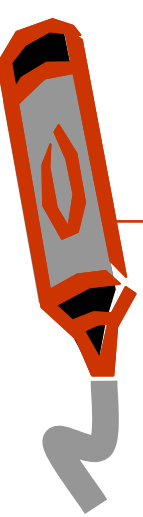




Interpreting BFS expts

- Need another $100 - 56 = 44\text{M}$ pages reachable from SCC
 - gives us 100M pages reachable from SCC
- Likewise, need another $\sim 44\text{M}$ pages reachable from SCC going backwards
- These together don't account for all 186M pages in giant WCC.

Distance measurements

- 
- For random pages $p1, p2$:
 $\Pr[p1 \text{ reachable from } p2] \sim \beta$
 - Maximum directed distance between 2 SCC nodes: > 28
 - Maximum directed distance between 2 nodes, given there is a path: > 900
 - Average directed distance between 2 SCC nodes: ~ 16
 - Average undirected distance: ~ 7

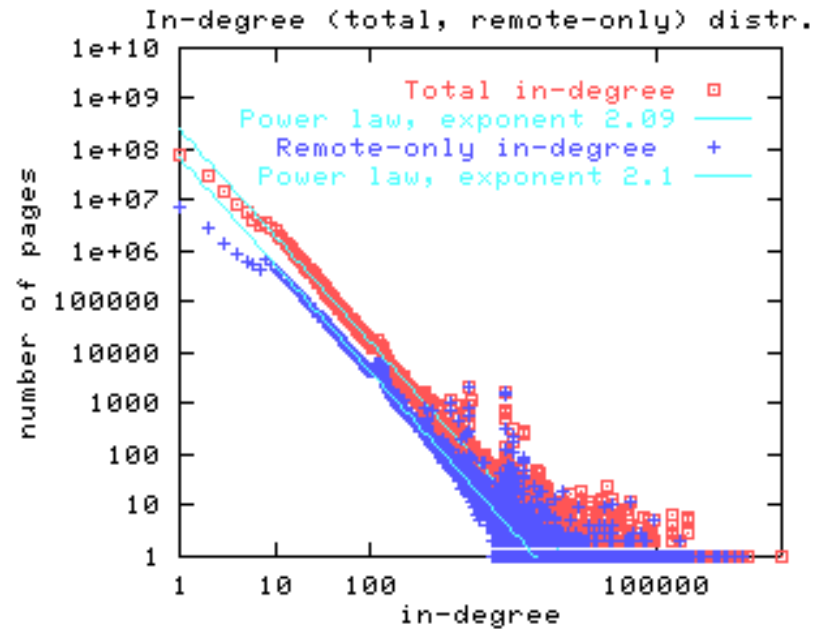


Bibliometric Laws

- Lotka's Law of Scientific Productivity
 - Number of authors making n contributions is proportional to $1/n^2$
- Bradford's Law of Scattering
 - Journals in a field are in groups of size k, kn, kn^2 contributing equal numbers of (useful) articles.
(*Roughly*. For every good journal there are n mediocre journals with articles that are $1/n$ th as useful)
- Zipf's Law (*Sociological Models*, [Zipf49])
 - $Freq(t)$ proportional to $1/rank(t)^a$
where a is close to 1
 - Also by Yule (*Linguistic Vocabulary*, [Yule44]) and by Pareto (*Economic Theory*, [Pa1897])
- Power laws on the Web
 - Inverse polynomial distributions:
 $Pr[k] \sim c/k^\alpha$ for a constant c . $\Leftrightarrow \log Pr[k] \sim c - \alpha \log k$
 - Thus plotting $\log Pr[k]$ against $\log k$ should give a straight line (of negative slope).

In-degree distribution

Probability that
a random page has
 k other pages
pointing to it is
 $\sim k^{-2.1}$ (Power law)



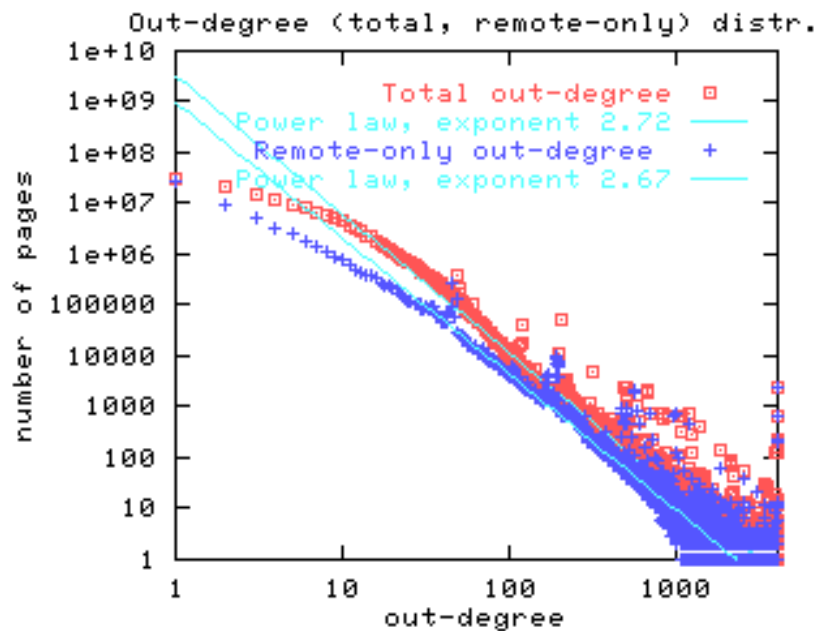
Slope = -2.1

Out-degree distribution

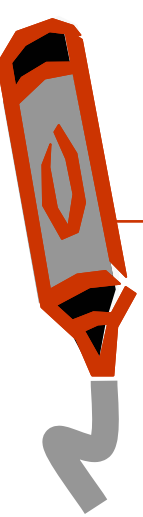


Probability that
a random page points
to k other pages is

$$\sim k^{-2.7}$$



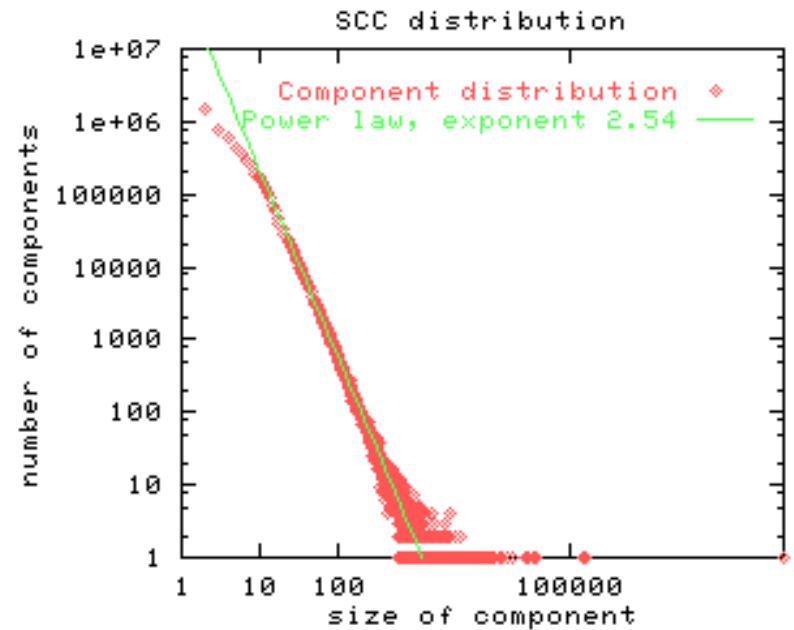
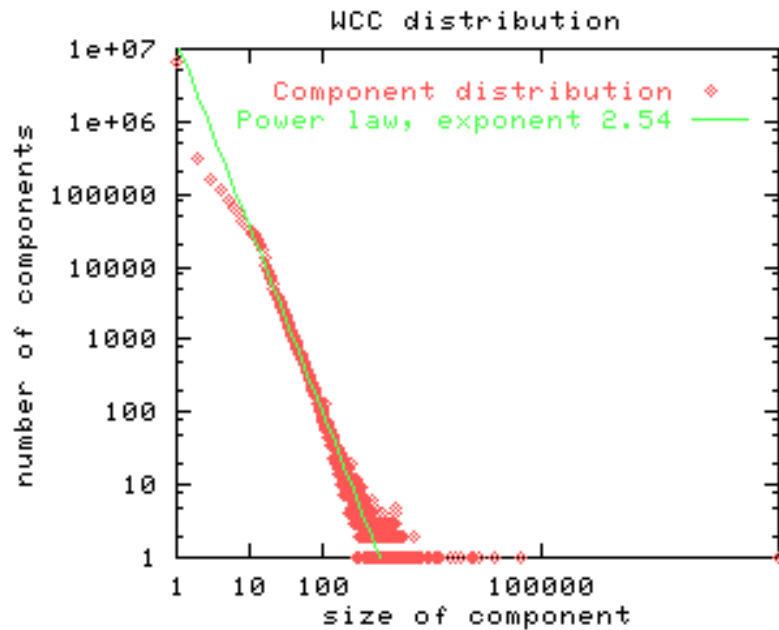
Slope = -2.7



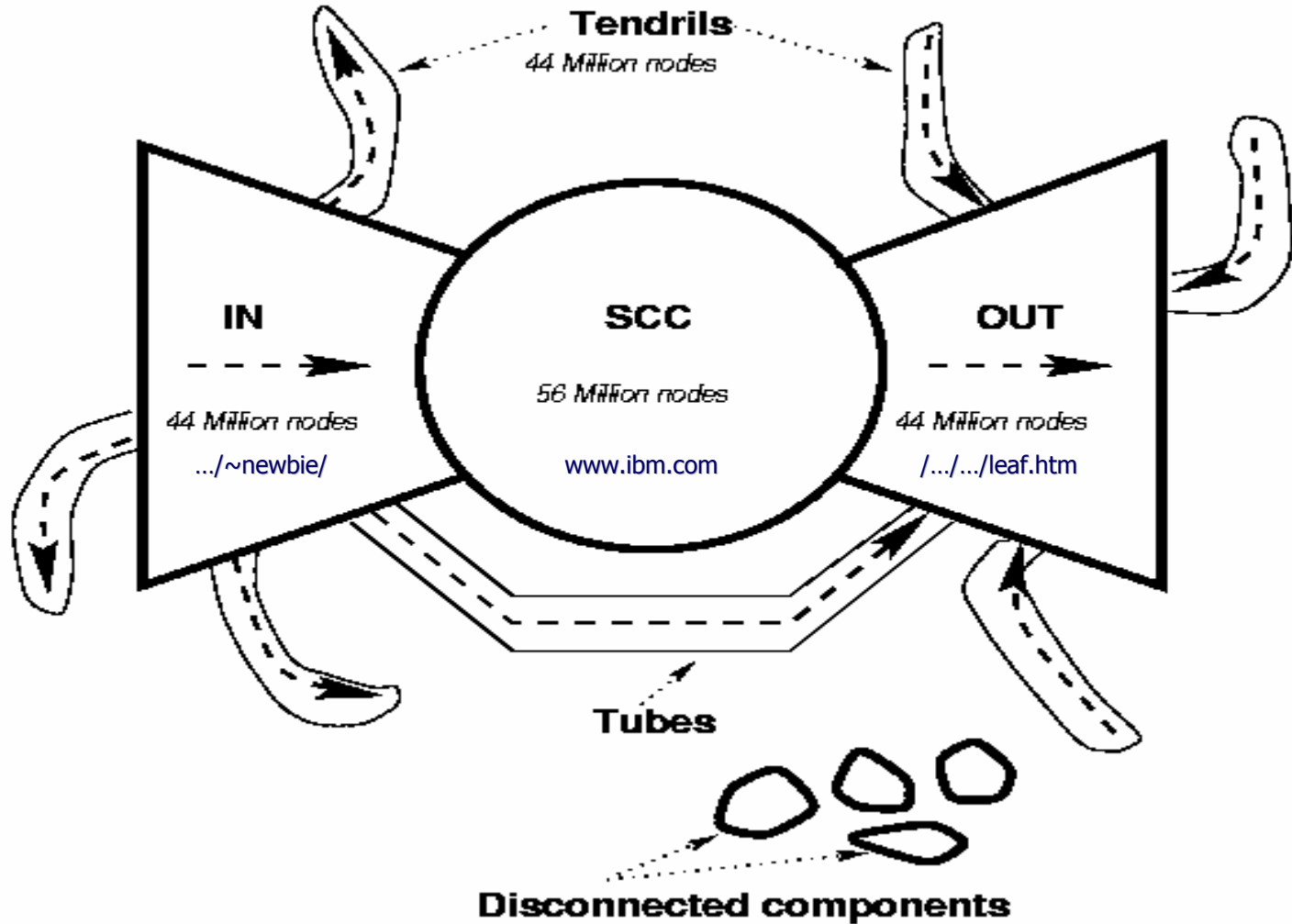
Connected components

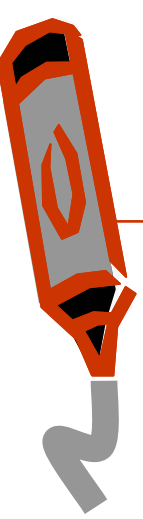
Largest WCC = 186M, SCC = 56M

Connected component sizes:



Web anatomy





www IR

- world wide web
- google, page rank
- markov chains
- HITS link analysis
- behavior-based web search
- crawling, indexing the web
- duplicates, mirrors and spam
- www infrastructure
- www size
- cache, hardware, systems