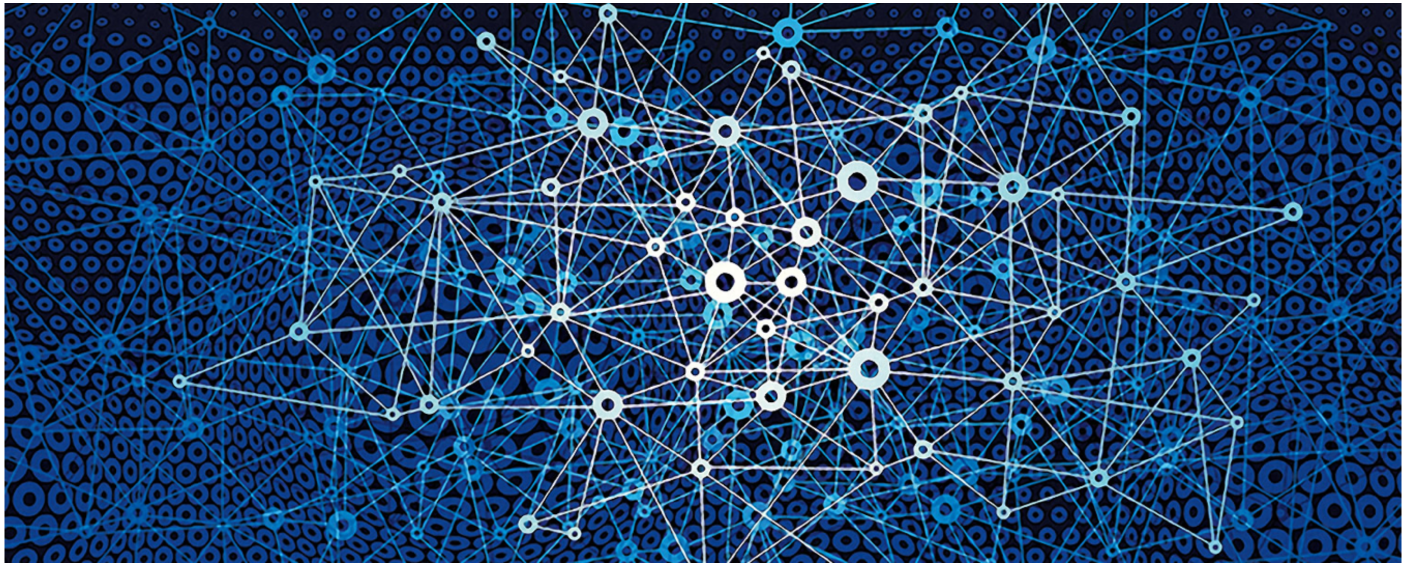
Search Engines for the Web



Tecnologie Internet
a.a. 2022/2023

Web search

Very similar to **information retrieval**, which is the part of computer science that studies the **retrieval of information** from a collection of **written documents.**

The retrieved documents aim at satisfying a **user information need** usually expressed in **natural language.**

With respect to information retrieval, Web search has specific properties:

- **Links** between Web pages can be exploited
- **Collecting,** storing, and **updating** documents is more difficult
- Usually, the **number of users** is very large
- **Spam** is a problem

Web search

Brief history:

- 1989: **Tim Berners-Lee** “invents” the World Wide Web

First Web search engines:

– **Archie:** Query **file names** by regular expressions

– **Architext/Excite:** Full text search, simple ranking (1993)

- Until 1998, web search meant information retrieval
- 1998: **Google** was founded

– Exploits **link structure** using the **PageRank** algorithm

Heterogeneity of document types

Some file types a search engine should be able to process:

application/ms-excel (different versions)
application/mspowerpoint (different versions)
application/msword (different versions)
application/pdf (different versions)
application/postscript
application/x-dvi
application/x-tar
application/x-zip-compressed
text/html (different versions and encodings)
text/plain (different encodings)
text/rtf
application/xml
text/xml
...

Heterogeneity of queries

There are **four main types of queries:**

– **Informational queries:**
Find general information about some topic, e.g., “Web search”

– **Navigational queries:**
Find a specific website, e.g., “Facebook”

– **Transactional queries:**
Find websites providing some service,
e.g., “Adobe Reader download”

– **Connectivity queries:**
Find connected pages, e.g., “link:www.unipr.it”
(finds all pages that link to http://www.unipr.it)

Heterogeneity of queries

Several studies analyzed user behaviors:

– The **average length** of a query is **2.4 terms**

– About **half of all queries** consist of a **single term**

– About **half of the users** looked only at the **first 20 results**

– Less than 5% of users use advanced search features (e.g., Boolean operators)

– About **20%** of all queries contain a **geographic term**

– About **a third of the queries** from the same user were **repeated queries;** about 90% of the time the user would click on the same result

– **Term frequency distributions** conform to the **power law**

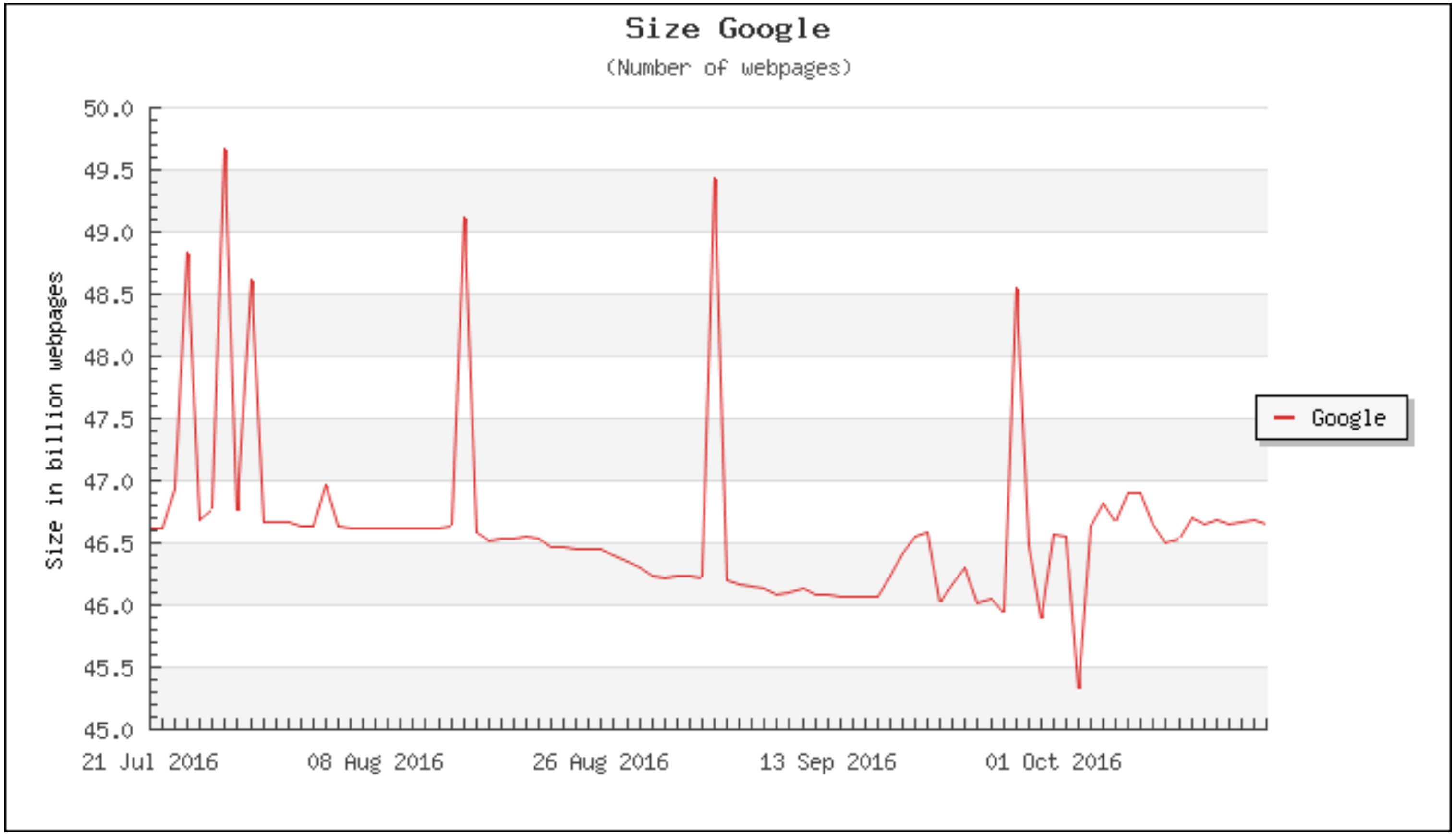
Search engines

Search engines have three main functionalities:

1. **Crawling:** Scour the Internet for content, looking over the code/content for each URL they find.
2. **Indexing:** Store and organize the content found during the crawling process. Once a page is in the index, it’s in the running to be displayed as a result to relevant queries.
3. **Ranking:** Provide the pieces of content that will best answer a searcher's query, which means that results are ordered by most relevant to least relevant.

Index size

How large is a typical search engine's index?



Web traffic and bandwidth

The index must be regularly updated:
- New Web pages
- Deleted Web pages
- Modified Web pages

How much data must be transferred for doing this?

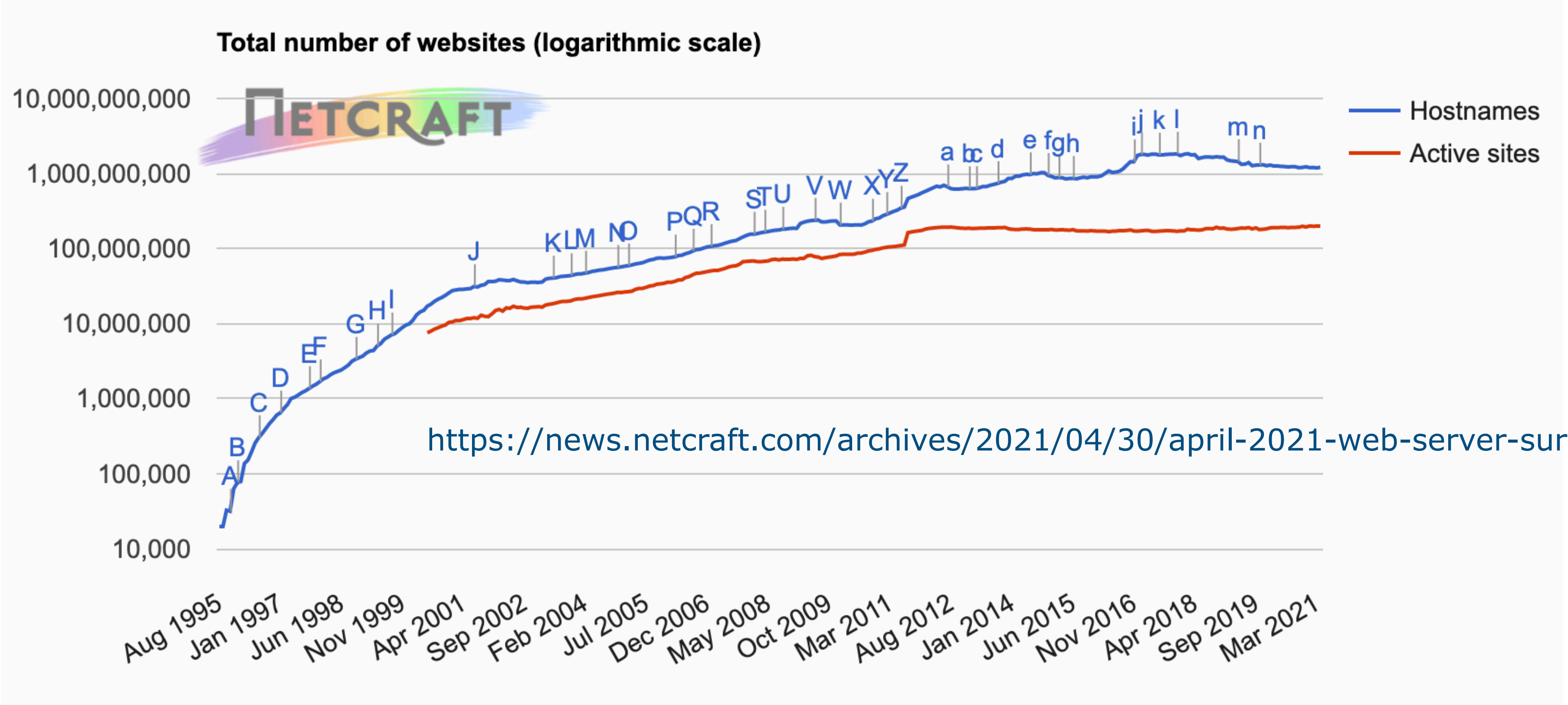
Some recent numbers from netcompetition.org:
- Within the US part of the Internet, Google transfers around **60 petabytes per month:** 60,000,000,000 megabytes!

Now you know why **Web search is expensive…**

https://httparchive.org/

Scalability

The Web grows fast (exponentially?)...



https://news.netcraft.com/archives/2021/04/30/april-2021-web-server-survey.html

A Web search engine must **scale well** to keep up

Scalability

Of course, these estimates only cover the so-called **"surface web,"** i.e., the part of the Web that can be accessed automatically by current Web crawlers

– Even today's best Web crawlers cannot find pages without inlinks or all pages that have been generated dynamically…

The term **"Deep Web"** refers to all web pages that currently are not indexed by any Web search engine

There are different estimates on the Deep Web's size

– **The Deep Web is 15–500x as large as the surface Web**

Scalability

Some types of “deep resources”:

– Dynamic content that cannot be accessed automatically, e.g. pages that are generated dynamically after filling out Web forms

– Unlinked or private content

– “Scripted” content, which requires code execution (e.g., Java, JavaScript, or Flash)

– “Strange” file formats not handled by current search engines

The “**Dark Web**” is a subset of the Deep Web. To reach the Dark Web, it is necessary to join specific overlay networks that allows applications to send messages to each other pseudonymously and securely (“darknets”), such as Tor, I2P and Freenet.

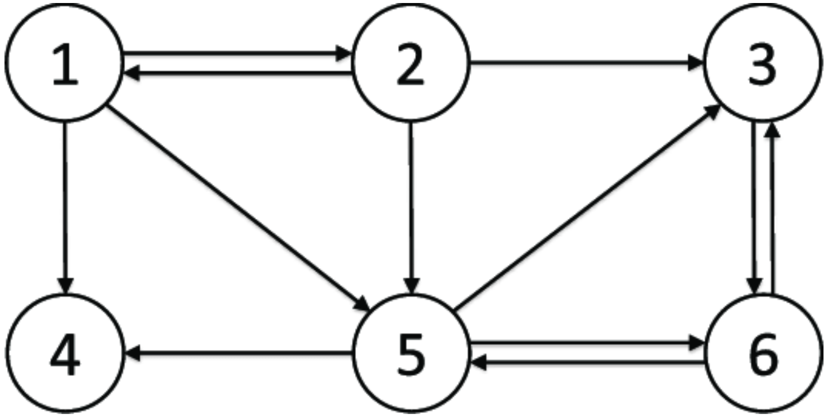
The Web graph

We can view the static Web consisting of static HTML pages together with the hyperlinks between them as a directed graph

- Each Web page is a node
- Each hyperlink is a directed edge

The hyperlinks into a page are called **inlinks**

The hyperlinks out of a page are called **outlinks**



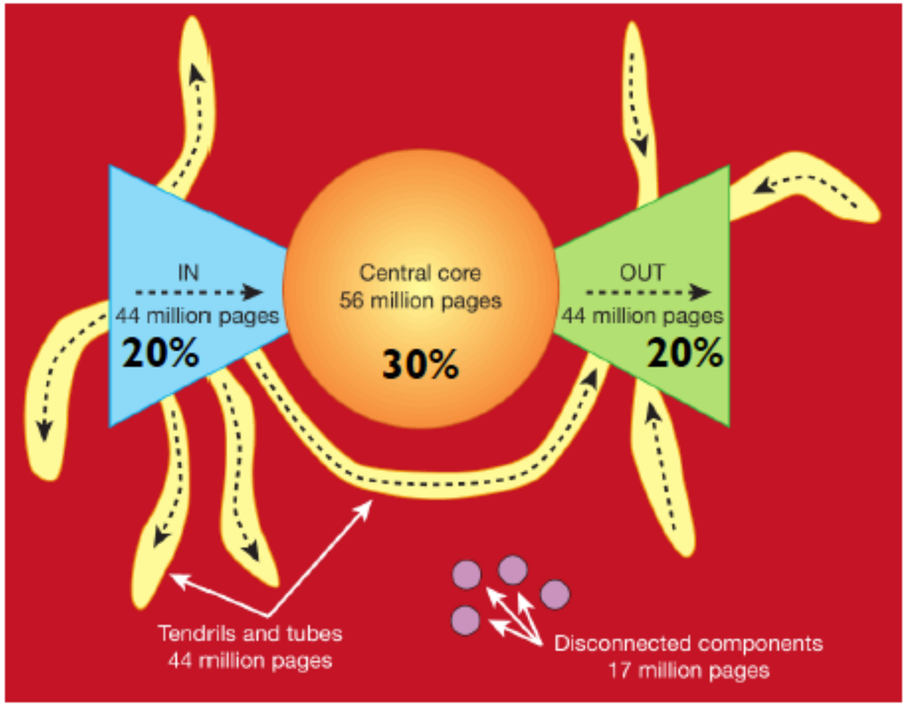
The Web graph

There is evidence that these links are not randomly distributed

The distribution of inlinks seems to follow a **power law**
– The total number of pages having exactly *k* inlinks is proportional to $k^{-2.1}$

Furthermore, several studies have suggested that the Web graph has a **bow tie shape**

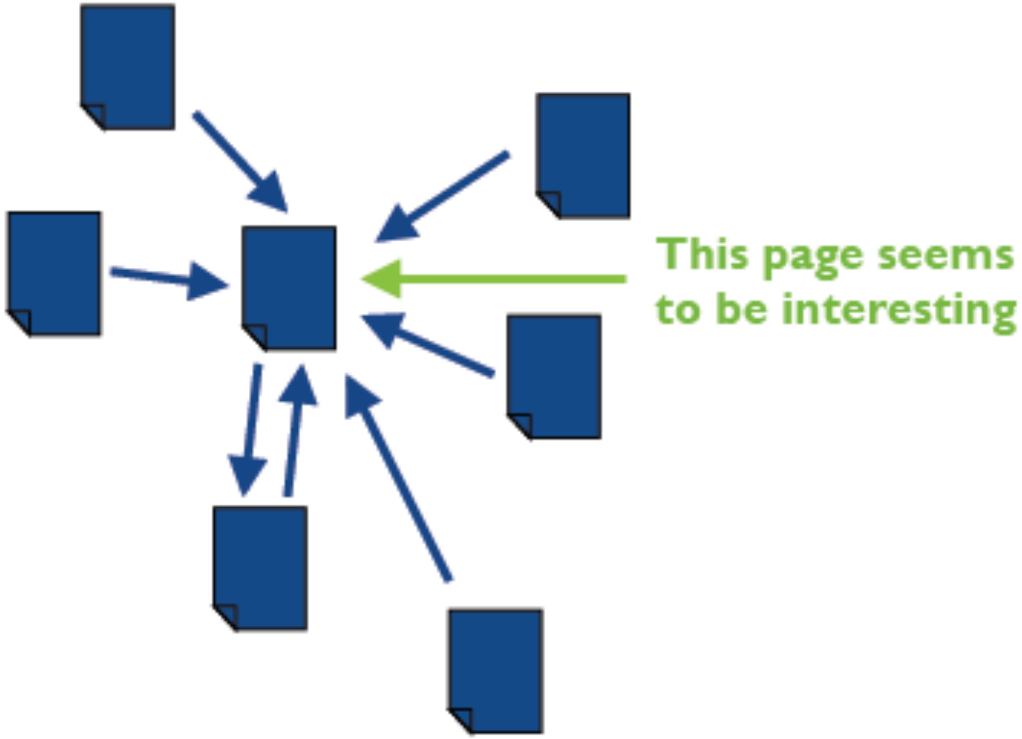
https://www.nature.com/articles/35012155

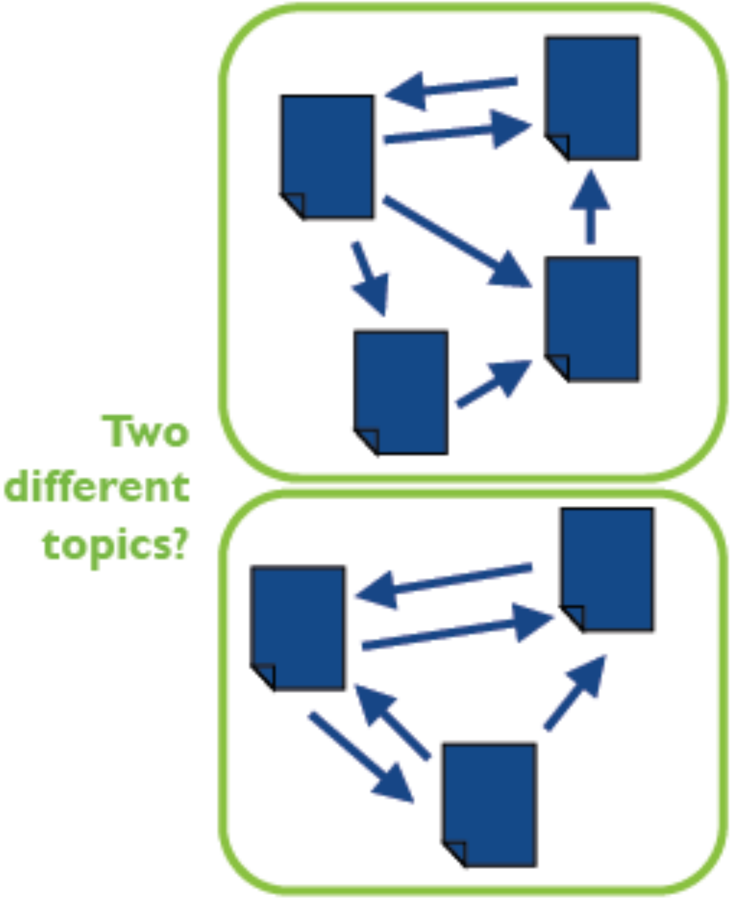


Note: The numbers given are as of 2000

The Web graph

Links are not created randomly





Web crawling

A basic crawler (aka robot, bot, spider) consists of:

– A **queue** of URIs to be visited
– A method to **retrieve** Web resources and process HTTP data
– A **page parser** to extract links from retrieved resources
– A **connection** to the search engine's **indexer**

The basic mode of operation:

Initialize the queue with URIs of known **seed pages**
Repeat forever:
1. Take URI from queue
2. Retrieve and parse page
3. Extract URIs from page
4. Add new URIs to queue
5. Send page to the indexer

Web crawling

The Web is large: **60 billion pages** (more or less…)

Let’s assume we want to **crawl each page once a year**

How many pages do we have to crawl **per second** then?

– 60,000,000,000 pages per year
– 5,000,000,000 pages per month
– 166,666,667 pages per day
– 6,944,444 pages per hour
– 115,740 pages per minute
– **1929 pages per second**

Well, it seems like we need a **highly scalable** crawler…

Web crawling

Apart from scalability, there are further issues

- How to detect **spam** pages?
- How to detect **duplicates** or pages already seen?
- How to avoid **spider traps?**
- We need many machines, how do we **distribute?**
- How to handle **latency** problems?
- How to limit the used **bandwidth?**
- How **deep** should we crawl sites?
- How to comply with the **site owner's** wishes?

Web crawling

Robot exclusion standard

– **Exclude some resources** from access by robots, and thus from indexing by search engines

– Put a file named **robots.txt** in your domain's top-level directory (e.g. http://en.wikipedia.org/robots.txt), which specifies what resources crawlers are allowed to access

– **Caution:** This "standard" is not a standard in the usual sense, it's purely advisory!

Examples:

– **Allow all robots to view all files:**

```
User-agent: *
Disallow:
```

– **Keep all robots out:**

```
User-agent:*
Disallow: /
```

Ranking based on *link analysis*

- Apply ideas from network analysis to the **Web graph…**
- **Links are recommendations**
- **Anchor texts** can be used as document descriptions

Assumption 1:
A hyperlink is signal of quality or popular interest

Assumption 2:
The anchor text of a link (or its surrounding text) describes the target page

Two highly popular algorithms:

- PageRank (Page *et al.*, 1998)
- HITS (Kleinberg, 1999)

PageRank

PageRank was invented by Larry Page at Stanford

The method for computing the PageRank and related stuff are patented!

US patent 6,285,999
“Method for node ranking in a linked database”

– Patent was assigned to Stanford University (not to Google)
– **Google has exclusive license rights**
– Stanford received **1.8 million shares in Google**
in exchange for use of the patent
– These shares were sold in 2005 for **336 million dollars**

PageRank

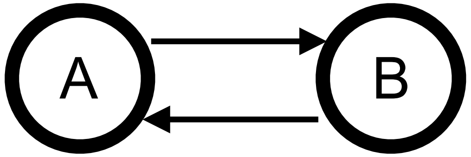
PageRank is a link analysis algorithm that produces a ranking of the web pages that does not depend on the queries. The algorithm is executed periodically by the **indexer**. The result is a **ranked index**.

We assume that page *p* is linked by pages $q_1..q_n$.
Let *d* be the probability that a "*random surfer*" gets bored of the page he is visiting and jumps to another page chosen at random (usually *d* = 0,85).
Let *C(p)* be the number of links from *p*.
The *Page Rank* of *p* is then:

$$PR(p) = (1-d) + d\,[PR(q_1)/C(q_1) + \ldots + PR(q_n)/C(q_n)]$$

User queries get a subset of the whole set of pages, ordered according to the Page Rank.

PageRank



Example 1

$d = 0.85$

$PR'(A) = (1-d) + d\ PR(B)/C(B) = (1-d) + d\ PR(B)$
$PR'(B) = (1-d) + d\ PR(B)/C(A) = (1-d) + d\ PR(A)$

1) guess $PR(A) = PR(B) = 1$

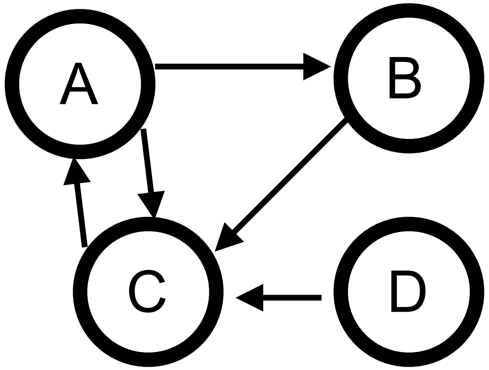
then $PR'(A) = PR'(B) = 1$ the guess was correct!

2) guess $PR(A) = PR(B) = 0$

then $PR'(A) = PR'(B) = 0.15$
then $PR'(A) = PR'(B) = 0.2775$
.. until $PR'(A) = PR'(B) = 1$

3) guess $PR(A) = PR(B) = 40$
then?

PageRank



Example 2

d = 0.85

PR'(A) = *(1-d) + d PR(*C*)/C(*C*)*
PR'(B) = *(1-d) + d PR(*A*)/C(*A*)*
PR'(C) = *(1-d) + d [PR(*A*)/C(*A*) + PR(*B*)/C(*B*) + PR(*D*)/C(*D*)]*
PR'(D) = *(1-d) + d 0 = 1-d*

Start with *PR(x) = 1* for each page *x*, then compute PR' until convergence.

PageRank

How to compute the PageRank for a Web graph containing 60 billion nodes?

– Use a highly scalable distributed algorithm
– Actually, this is one of Google’s secrets…

PageRank

A search engine myth:
"PageRank is the most important component of ranking"

The reality:

– There are several components that are at least as important: anchor text, phrases, proximity, …

– Google uses **hundreds of different features** for ranking

– There are rumors that PageRank in its original form (as presented here) has a negligible effect on ranking

– However, variants of PageRank are still an essential part of ranking

– Addressing **link spam** is difficult and crucial!

PageRank

A disadvantage of PageRank is that it computes only a single overall score for each web resource
– A web resource might be unimportant from a global view but highly important for a specific topic

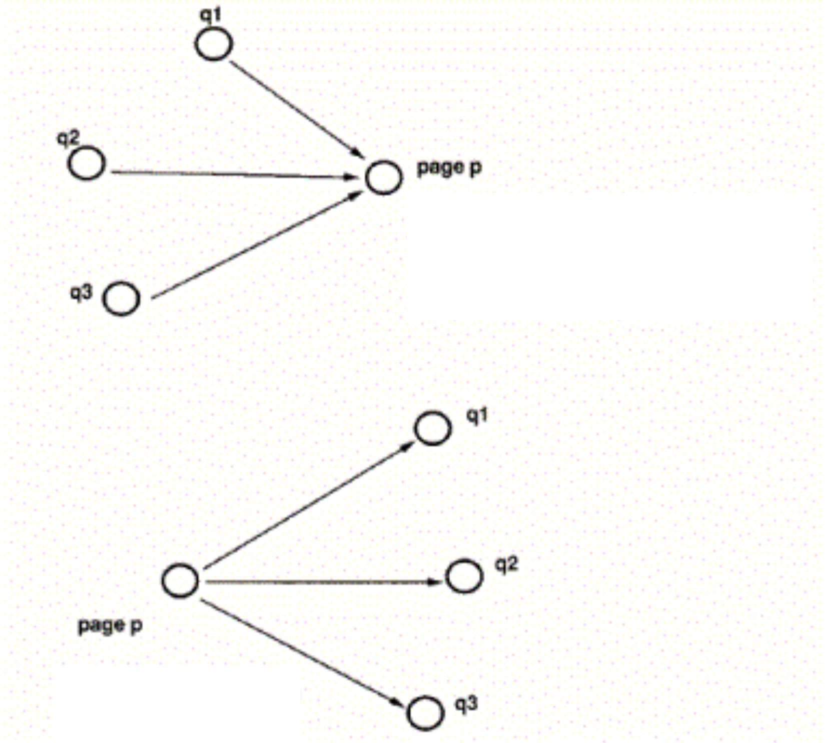
Topic-sensitive PageRank tries to address this issue:
– Define a set of popular **topics** (e.g. football, Windows, Obama)
– Use **classification** algorithms to assign each Web resource to one (or more) of these topics
– For each topic, compute a **topic-sensitive PageRank** by **limiting the random teleports** to pages of the current topic
– At query time, **detect the query’s topics** and **use the corresponding PageRank scores…**

HITS

Kleinberg identifies two types of web pages:

authority – a page that is an authoritative information source for the query

hub – a list of pointers to pages that are related to the topic of the query



There is a mutual reinforcement relation between the two types of web pages:
“Good hubs point to good authorities and vice versa”.

HITS

1) Sampling phase

The words of the query are used to build a **root set** of pages, using an engine that analyzes textual content.

Then the root set is expanded to a **base set**, by adding all the pages that link and are linked by the pages of the root set.

The base set should contain all the pages that best fit the query (the root set is not sufficient, in this sense).

HITS

2) Weight-propagation phase

An **authority weight *a_p*** and a **hub weight *h_p***, non-negative and initialized with value 1, are assigned to each page *p* of the base set.

Update rule:

- a_p is the sum of the hub weights of the pages that link *p*
- h_p is the sum of the authority weights of the pages linked by *p*
- normalization:
  - $a_p / \Sigma_i \sqrt{a_i}$
  - $h_p / \Sigma_i \sqrt{h_i}$

is iteratively applied until hub weight and authority weight converge.

In the end, two rankings are produced: the authority one and the hub one.

HITS

Sometimes HITS tends to generalize or deviate from a given topic, in particular when hubs cover different topics.

A possible solution is to compare the words of the query with the text that surrounds a link, to achieve a "weighted" version of the update rule.

Another possibility is to fragment a large hub in several smaller *hublets*, and to ignore those that are less related to the topic of the query.

HITS

As PageRank, **HITS has been patented:**

– US patent 6,112,202
– "Method and system for identifying authoritative information resources in an environment with content-based links between information resources"
– Inventor: Jon Kleinberg
– **Assignee: IBM**

HITS vs PageRank

PageRank can be precomputed, HITS has to be computed at query time

– HITS is very expensive

Different choices regarding the formal model

– HITS models hubs and authorities

– HITS uses a subset of the Web graph

– But: We could also apply PageRank to a subset and HITS on the whole Web graph…

References

- https://www2022.thewebconf.org/
- https://httparchive.org/
- Information Retrieval and Web Search Engines http://www.ifis.cs.tu-bs.de/ws-1415/irws
- A taxonomy of web search. A. Broder, ACM SIGIR Forum, 36(2), 2002.
- The Web and Social Networks. R. Kumar, P. Raghavan, S. Rajagopalan, A. Tomkins. IEEE Computer, November, 2002.
- Authoritative Sources in a Hyperlinked Environment. J. Kleinberg. ACM-SIAM Symposium on Discrete Algorithms, 1998.
- Graph Structure in the Web. A. Broder et al. Proc. 9th International World Wide Web Conference (WWW9), 2000.
- The Anatomy of a Large-Scale Hypertextual Web Search Engine. S. Brin, L. Page. Proc. 7th International World Wide Web Conference (WWW7), 1998.
- The Google Pagerank Algorithm and How It Works. Ian Rogers, http://www.cs.princeton.edu/~chazelle/courses/BIB/pagerank.htm
- Trawling the Web for Cyber Communities. R. Kumar et al. Proc. 8th International World Wide Web Conference (WWW8), 1999.
- Emergence of Scaling in Random Networks. A. Barabàsi, R. Albert. Science, 1999.