Web Search

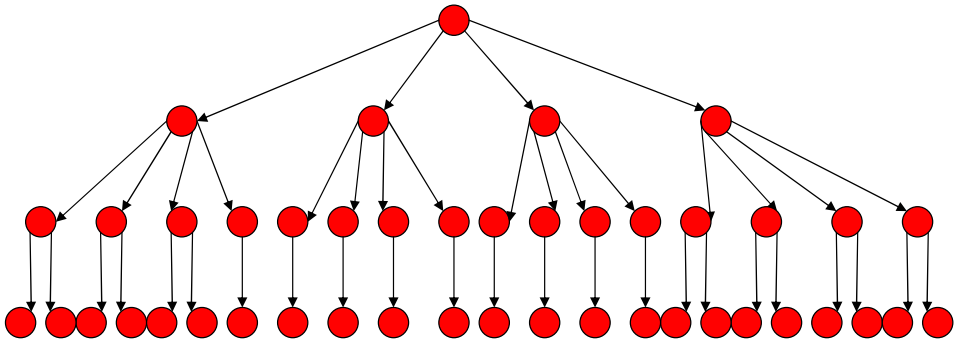
Spidering (Crawling)

Spiders (Robots/Bots/Crawlers)

- Start with a comprehensive set of root URL's from which to start the search.
- Follow all links on these pages recursively to find additional pages.
- Store all **novel** found pages in a repository to be indexed later on (as an inverted index).
- May allow users to directly submit pages to be indexed (and crawled from).

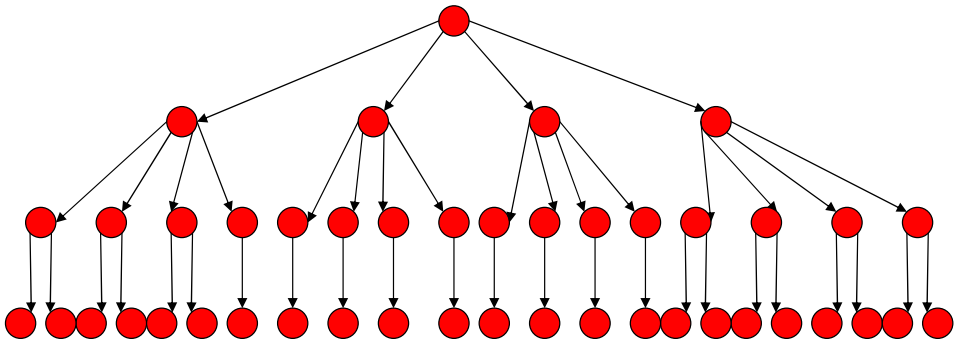
Search Strategies

Breadth-first Search



Search Strategies (cont)

Depth-first Search



Search Strategy Trade-Off's

- Breadth-first explores uniformly outward from the root page but requires memory of all nodes on the previous level (exponential in depth). Standard spidering method.
- Depth-first requires memory of only depth times branching-factor (linear in depth) but gets "lost" pursuing a single thread.
- Both strategies implementable using a queue of links (URL's).

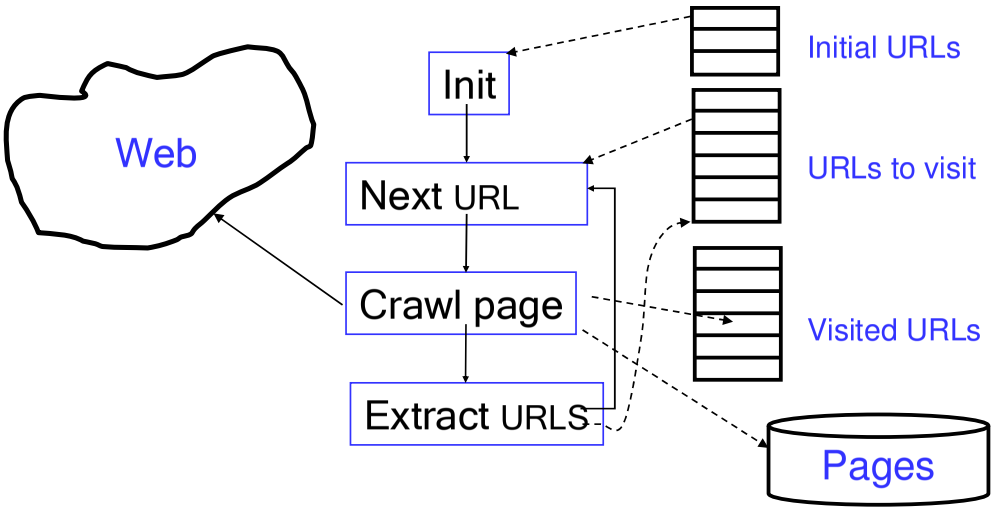
Avoiding Page Duplication

- Must detect when revisiting a page that has already been spidered (web is a graph not a tree).
- Must efficiently store visited pages to allow rapid recognition test.
  - Tree indexing (e.g. trie)
  - Hash table
- Index page using URL as a key.
  - Must canonicalize URL's (e.g. delete ending "/")
  - Not detect duplicated or mirrored pages.
- Index page using textual content as a key.
  - Requires first downloading page.

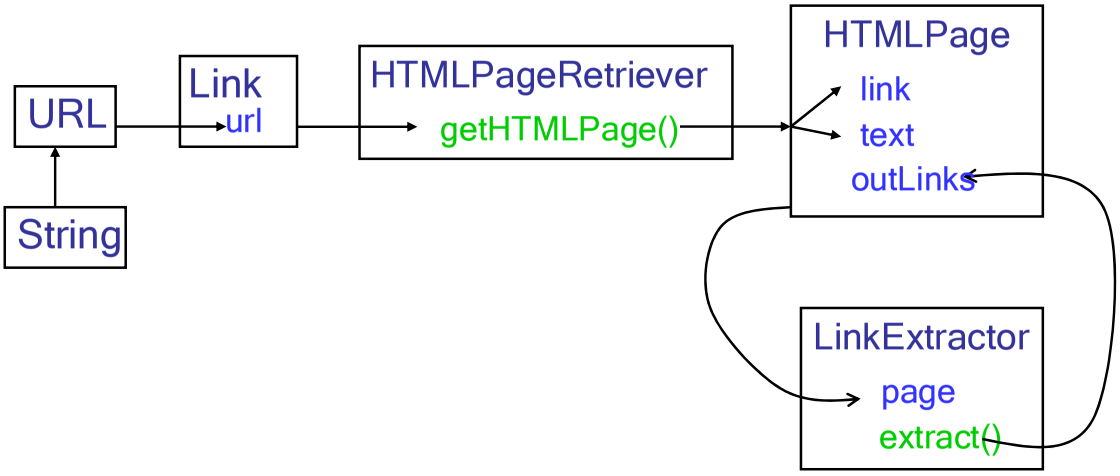
Spidering Algorithm

Initialize queue (Q) with initial set of known URL's.
Until Q empty or page or time limit exhausted:
- Pop URL, L, from front of Q.
- If L is not to an HTML page (.gif, .jpeg, .ps, .pdf, .ppt…)
  - continue loop.
- If already visited L, continue loop.
- Download page, P, for L.
- If cannot download P (e.g. 404 error, robot excluded)
  - continue loop.
- Store P.
- Parse P to obtain list of new links N.
- Append N to the end of Q.

Crawl Scheme



Spider Classes



Queueing Strategy

- How new links added to the queue determines search strategy.
- FIFO (append to end of Q) gives breadth-first search.
- LIFO (add to front of Q) gives depth-first search.
- Heuristically ordering the Q gives a “focused crawler” that directs its search towards “interesting” pages.

Restricting Spidering

- Restrict spider to a particular site.
  - Remove links to other sites from Q.
- Restrict spider to a particular directory.
  - Remove links not in the specified directory.
- Obey page-owner restrictions (robot exclusion).

Link Extraction

- Must find all links in a page and extract URLs.
  - <a href="http://www.cs.utexas.edu/users/mooney/ir-course">
  - <frame src="site-index.html">
- Must complete relative URL's using current page URL:
  - <a href="proj3"> to http://www.cs.utexas.edu/users/mooney/ircourse/proj3
  - <a href="../cs343/syllabus.html"> to http://www.cs.utexas.edu/users/mooney/cs343/syllabus.html

URL Syntax

- A URL has the following syntax:
  - http://www.ibm.com/support/us/index.html
  - Access method: http
  - Domain name: www.ibm.com
  - Page name: www.ibm.com/support/us/index.html
- Domains that have internet addresses: hosts
  - www.ibm.com = 129.42.19.99
  - resolve the server host name to an Internet address (IP)
    - Use Domain Name Server (DNS)
    - DNS is a distributed database of name-to-IP mappings maintained at a set of known servers
  - Not every host runs a web server

URL Syntax

- A URL has the following syntax:
  - <scheme>://<authority><path>?<query>#<fragment>
- An *authority* has the syntax:
  - <host>:<port number>
- A *query* passes variable values from an HTML form and has the syntax:
  - <variable>=<value>&<variable>=<value>…
- A *fragment* is also called a *reference* or a *ref* and is a pointer within the document to a point specified by an anchor tag of the form:
  - <A NAME="<fragment>">

Link Canonicalization

- Equivalent variations of ending directory normalized by removing ending slash.
  - http://www.cs.utexas.edu/users/mooney/
  - http://www.cs.utexas.edu/users/mooney
- Internal page fragments (ref's) removed:
  - http://www.cs.utexas.edu/users/mooney/welcome.html#courses
  - http://www.cs.utexas.edu/users/mooney/welcome.html

Robot Exclusion

- Web sites and pages can specify that robots should not crawl/index certain areas.
- Two components:
  - Robots Exclusion Protocol: Site wide specification of excluded directories.
  - Robots META Tag: Individual document tag to exclude indexing or following links.

Robots Exclusion Protocol

- Site administrator puts a "robots.txt" file at the root of the host's web directory.
  - http://www.ebay.com/robots.txt
  - http://www.cnn.com/robots.txt
- File is a list of excluded directories for a given robot (user-agent).
  - Exclude all robots from the entire site:

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


Robot Exclusion Protocol Examples

- Exclude specific directories:

```
User-agent: *
Disallow: /tmp/
Disallow: /cgi-bin/
Disallow: /users/paranoid/
```

- Exclude a specific robot:

```
User-agent: GoogleBot
Disallow: /
```

- Allow a specific robot:

```
User-agent: GoogleBot
Disallow:
```

Robot Exclusion Protocol Details

- Only use blank lines to separate different User-agent disallowed directories.
- One directory per “Disallow” line.

Robots META Tag

- Include META tag in HEAD section of a specific HTML document.
  - <meta name=“robots” content=“none”>
- Content value is a pair of values for two aspects:
  - index | noindex: Allow/disallow indexing of this page.
  - follow | nofollow: Allow/disallow following links on this page.

Robots META Tag (cont)

- Special values:
  - all = index,follow
  - none = noindex,nofollow
- Examples:

```
<meta name="robots" content="noindex,follow">
<meta name="robots" content="index,nofollow">
<meta name="robots" content="none">
```


Robot Exclusion Issues

- META tag is newer and less well-adopted than "robots.txt".
- Standards are conventions to be followed by "good robots."
- Companies have been prosecuted for "disobeying" these conventions and "trespassing" on private cyberspace.
- "Good robots" also try not to "hammer" individual sites with lots of rapid requests.
  - "Denial of service" attack.

Multi-Threaded Spidering

- Bottleneck is network delay in downloading individual pages.
- Best to have multiple threads running in parallel each requesting a page from a different host.
- Distribute URL's to threads to guarantee equitable distribution of requests across different hosts to maximize through-put and avoid overloading any single server.
- Early Google spider had multiple co-ordinated crawlers with about 300 threads each, together able to download over 100 pages per second.

Directed/Focused Spidering

- Sort queue to explore more "interesting" pages first.
- Two styles of focus:
  - Topic-Directed
  - Link-Directed
  - Others

Topic-Directed Spidering

- Assume desired topic description or sample pages of interest are given.
- Sort queue of links by the similarity (e.g. cosine metric) of their source pages and/or anchor text to this topic description.
- Preferentially explores pages related to a specific topic.

Important Metrics

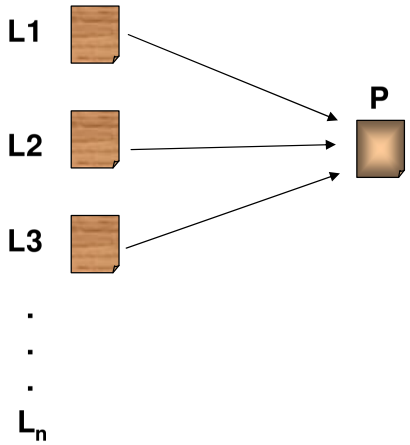
- Similarity
- Backlink Count
- Page Rank
- HITS
- Foward Link Count

Similarity

- Queries are used to determine the importance of pages.
- An IR model is used to compute the similarity between a given query Q and a document P (web page).

Backlink Count

- The importance of a web page P is defined by the number of links that point to the page.



PageRank

- The importance of a page P is given by the equation:
  - $IR(P) = (1-d) + d (IR(T1)/c1 + ... + IR(Tn)/cn)$

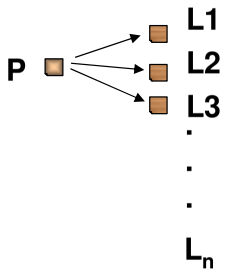
  d – dump factor (generally between 0.1 e 0.9)
  T_i – page that point to P
  c_i – number of links in T_i
- Page Rank computes the probability of a page being accessed.

HITS

- Uses values of hub and authority to define the importance of a page P.
  - **hub** of a page "P": given as a function of the values of authority of the pages it points to.
  - **Autority** of a page "P": given as a function of the values of hub of the pages that point to P.
- A good hub is a page that points to good authorities and a good authority is a page that is pointed to by good hubs.

Forward Link Count

- The importance of a page P is determined by the number of existent links in the page.
- All links have weight 1, but different weights could be used as a function of the importance of the page.



Link-Directed Spidering (Summary)

- Monitor links and keep track of in-degree and out-degree of each page encountered.
- Sort queue to prefer popular pages with many in-coming links (*authorities*).
- Sort queue to prefer summary pages with many out-going links (*hubs*).

Keeping Spidered Pages Up to Date

- Web is very dynamic: many new pages, updated pages, deleted pages, etc.
- Periodically check spidered pages for updates and deletions:
  - Just look at header info (e.g. META tags on last update) to determine if page has changed, only reload entire page if needed.
- Track how often each page is updated and preferentially return to pages which are historically more dynamic.
- Preferentially update pages that are accessed more often to optimize freshness of more popular pages.

Types of Scheduling

- **Offline:** Priority queue to crawl is sorted periodically offline.
- **Online:** Priority queue is sorted all the time periodically offline.

Design Challenges

- Define periodicity of updating (Freshness) X Find new url's.
- Use of maximum bandwidth without overloading visited sites.
- Identify duplicates and near duplicates (e.g. Mirror).
- Crawl "good" pages.

Practical Problems

- Overload of DNS
- Repeated access error
- Link extraction could generate false URLs or infinite links
- Crawl of dynamic pages
- Canonicalization of URLs
- Speed differences among servers could harm speed efficience
- ... (There are many other problems)

Overload of DNS

- Spiders generate a high number of DNS requests
- DNS servers are potential bottlenecks to the spider
  - Solution: keep a cache with previously solved DNSs

Repeated access error (false attacks)

- Spider could create a false attack to a web server:
  - Use of different names for a server
  - Many servers in one same place

Infinite Links

- Link extraction problems could generate errors that lead to infinite links that are validated by the web server
  - www.aa.bb.com/musica
  - www.aa.bb.com/musica/musica

Dynamic Pages

- Some sites generate an infinite number of valid pages.
  - Example: a site that returns a HTML page with the day of the week for any date, where the date enters in the URL

Anchor Text Indexing

- Extract anchor text (between <a> and) of each link followed.
- Anchor text is usually descriptive of the document to which it points to.
- Add anchor text to the content of the destination page to provide additional relevant keyword indices.
- Used by Google:
  - Evil Empire
  - IBM

Anchor Text Indexing (cont)

- Helps when descriptive text in destination page is embedded in image logos rather than in accessible text.
- Many times anchor text is not useful:
  - "click here"
- Increases content more for popular pages with many in-coming links, increasing recall of these pages.
- May even give higher weights to tokens from anchor text.