

Assignment Project Exam Help

Add WeChat powcoder

Introduction to
Assignment Project Exam Help
Information Retrieval
<https://powcoder.com>

Add WeChat powcoder

Lecture 17: Crawling and web indexes

Assignment Project Exam Help

Today's lecture

- Crawling

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Assignment Project Exam Help

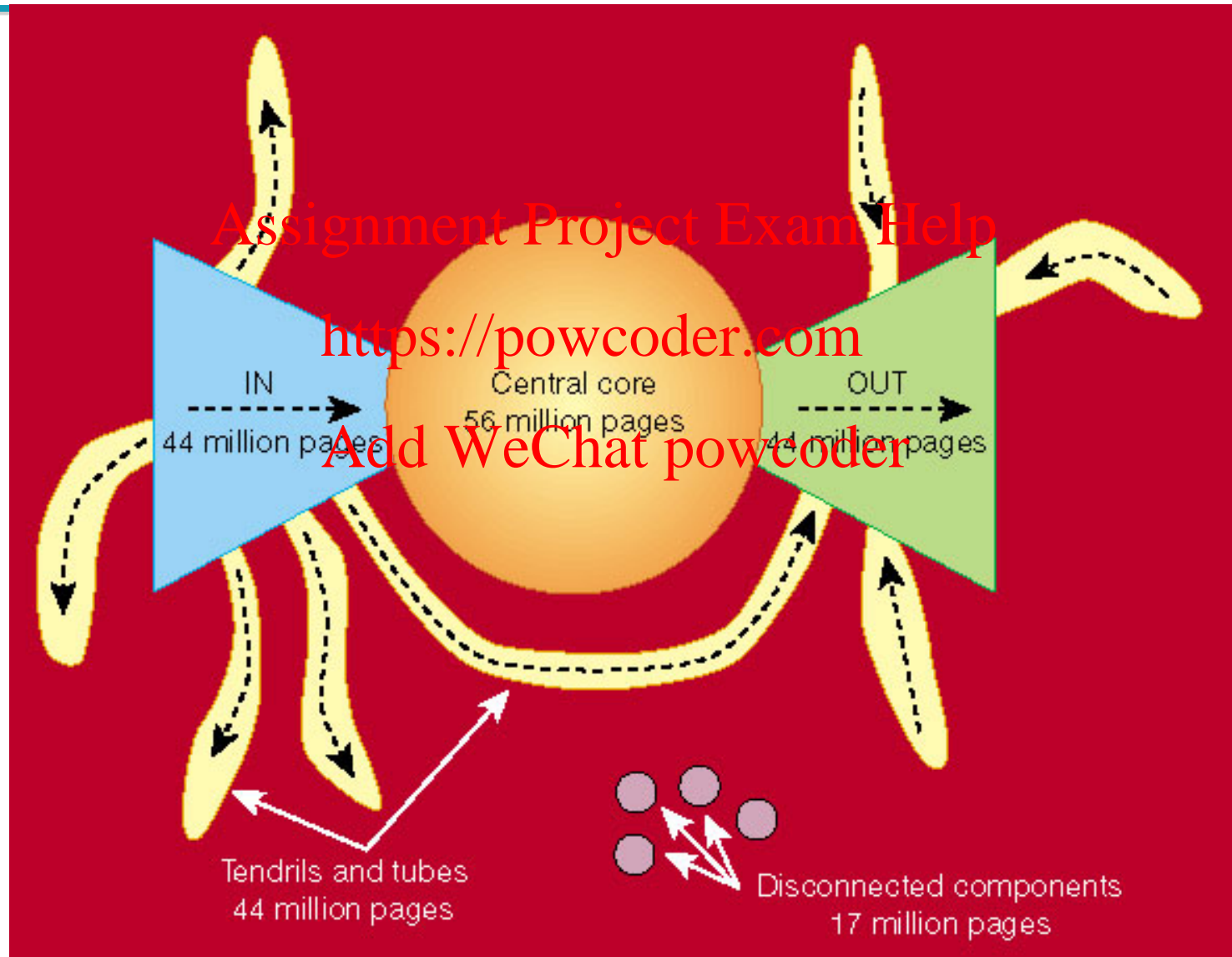
Basic crawler operation

- Begin with known “seed” URLs
- Fetch and parse them
 - Extract URLs they point to
 - Place the extracted URLs on a queue
- Fetch each URL on the queue and repeat

Assignment Project Exam Help

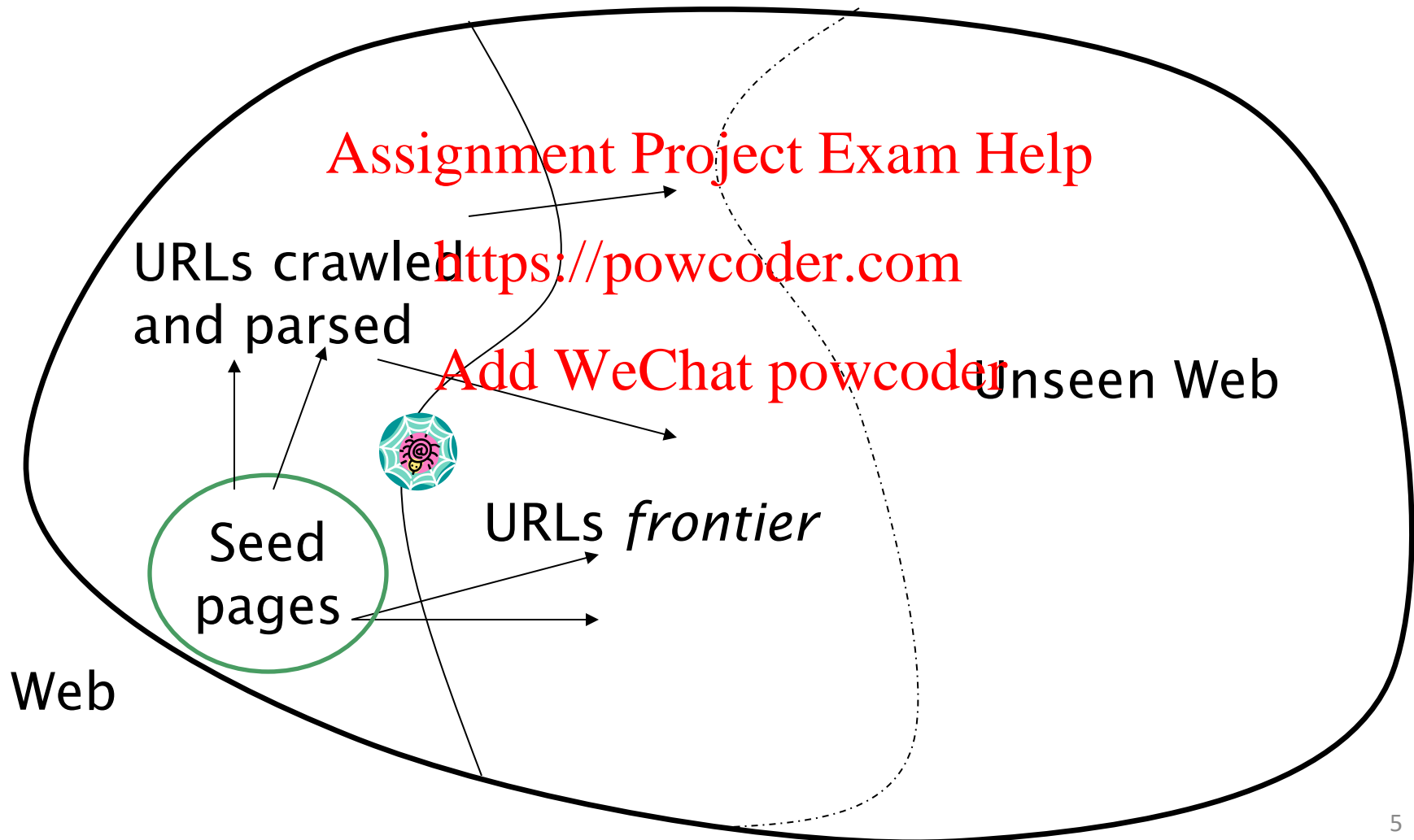
<http://www.nature.com/nature/journal/v405/n6783/pdf/405112a0.pdf>

Structure of the Web (circa 2000)



Assignment Project Exam Help

Crawling picture



Assignment Project Exam Help

Simple Crawler Thread

```
1  procedure CRAWLERTHREAD(frontier)
2    while not frontier.done() do
3      website ← frontier.nextSite()
4      url ← website.nextURL()
5      if website.permitsCrawl(url) then
6        text ← retrieveURL(url)
7        storeDocument(url, text)
8        for each url in parse(text) do
9          frontier.addURL(url)
10       end for
11     end if
12     frontier.releaseSite(website)
13   end while
14 end procedure
```

Assignment Project Exam Help

Simple picture – complications

- Web crawling isn't feasible with one machine
 - All of the above steps distributed
- Malicious pages
 - Spam pages
 - Spider traps – incl dynamically generated
- Even non-malicious pages pose challenges
 - Latency/bandwidth to remote servers vary
 - Webmasters' stipulations
 - How "deep" should you crawl a site's URL hierarchy?
 - Site mirrors and duplicate pages
- Politeness – don't hit a server too often

Assignment Project Exam Help

What any crawler must do

- Be Polite: Respect implicit and explicit politeness considerations
 - Only crawl allowed pages
 - <https://powcoder.com>
 - Respect *robots.txt* (more on this shortly)
- Be Robust: Be immune to spider traps and other malicious behavior from web servers

Assignment Project Exam Help

What any crawler should do

- Be capable of distributed operation: designed to run on multiple distributed machines
- Be scalable: designed to increase the crawl rate by adding more machines
- Performance/efficiency: permit full use of available processing and network resources

Assignment Project Exam Help

What any crawler should do

- Fetch pages of “higher quality” first
- Continuous operation: Continue fetching fresh copies of a previously fetched page
- Extensible: Adapt to new data formats, protocols

Assignment Project Exam Help

Freshness

Add WeChat powcoder

- HTTP protocol has a special request type called HEAD that makes it easy to check for page changes
 - returns information about page, not page itself

Assignment Project Exam Help

<https://powcoder.com>

Client request: HEAD /csinfo/people.html HTTP/1.1

Host: www.cs.umass.edu

HTTP/1.1 200 OK

Date: Thu, 03 Apr 2008 05:17:54 GMT

Server: Apache/2.0.52 (CentOS)

Last-Modified: Fri, 04 Jan 2008 15:28:39 GMT

Server response: ETag: "239c33-2576-2a2837c0"

Accept-Ranges: bytes

Content-Length: 9590

Connection: close

Content-Type: text/html; charset=ISO-8859-1

Assignment Project Exam Help

Freshness

- Web pages are constantly being added, deleted, and modified
- Web crawler must continually revisit pages it has already crawled to see if they have changed in order to maintain the *freshness* of the document collection
 - *stale* copies no longer reflect the real contents of the web pages

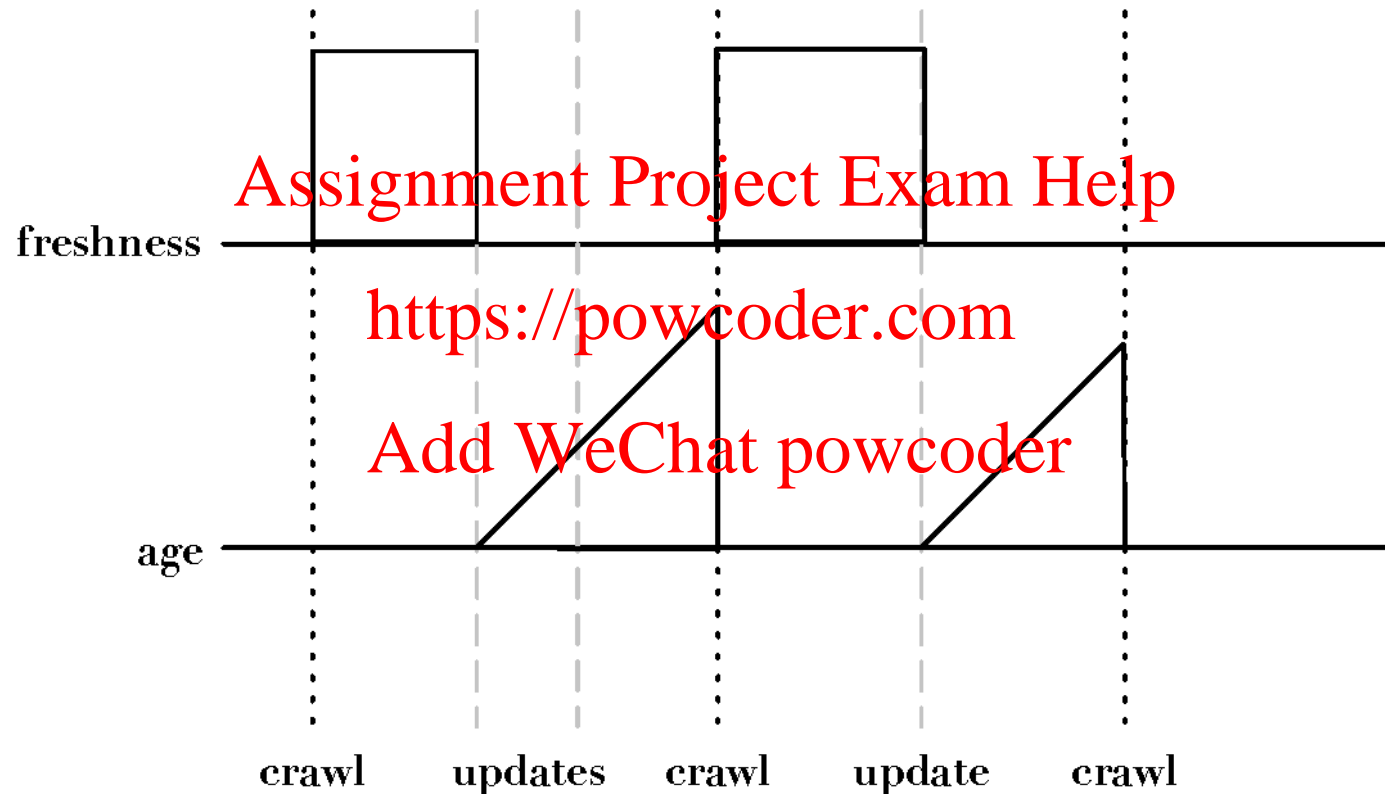
Assignment Project Exam Help

Freshness

- Not possible to constantly check all pages
 - must check important pages and pages that change frequently
- Freshness is the proportion of pages that are fresh
- Optimizing for this metric can lead to bad decisions, such as not crawling popular sites
- *Age* is a better metric

Assignment Project Exam Help

Freshness vs. Age



Assignment Project Exam Help

Age

Add WeChat powcoder

- Expected age of a page t days after it was last crawled:

Assignment Project Exam Help

$$\text{Age}(\lambda, t) = \int_0^t P(\text{page changed at time } x)(t - x)dx$$

<https://powcoder.com>

- Web page updates follow the Poisson distribution on average
 - time until the next update is governed by an exponential distribution

Add WeChat powcoder

$$\text{Age}(\lambda, t) = \int_0^t \lambda e^{-\lambda x}(t - x)dx$$

Assignment Project Exam Help

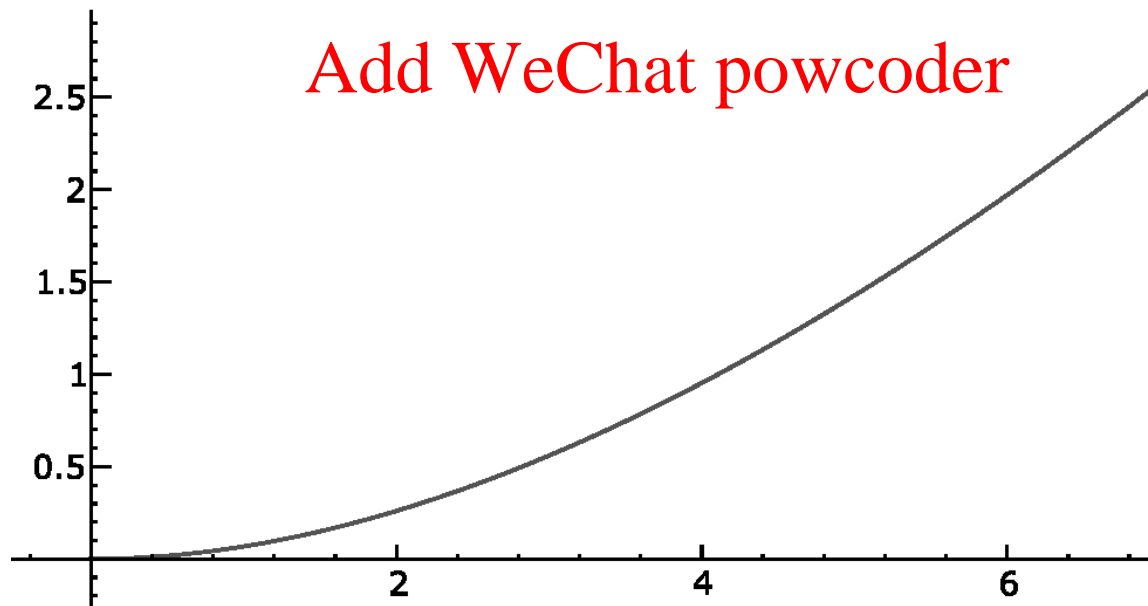
Age

Add WeChat powcoder

- The older a page gets, the more it costs not to crawl it
 - e.g., expected age with mean change frequency $\lambda = 1/7$ (one change per week)

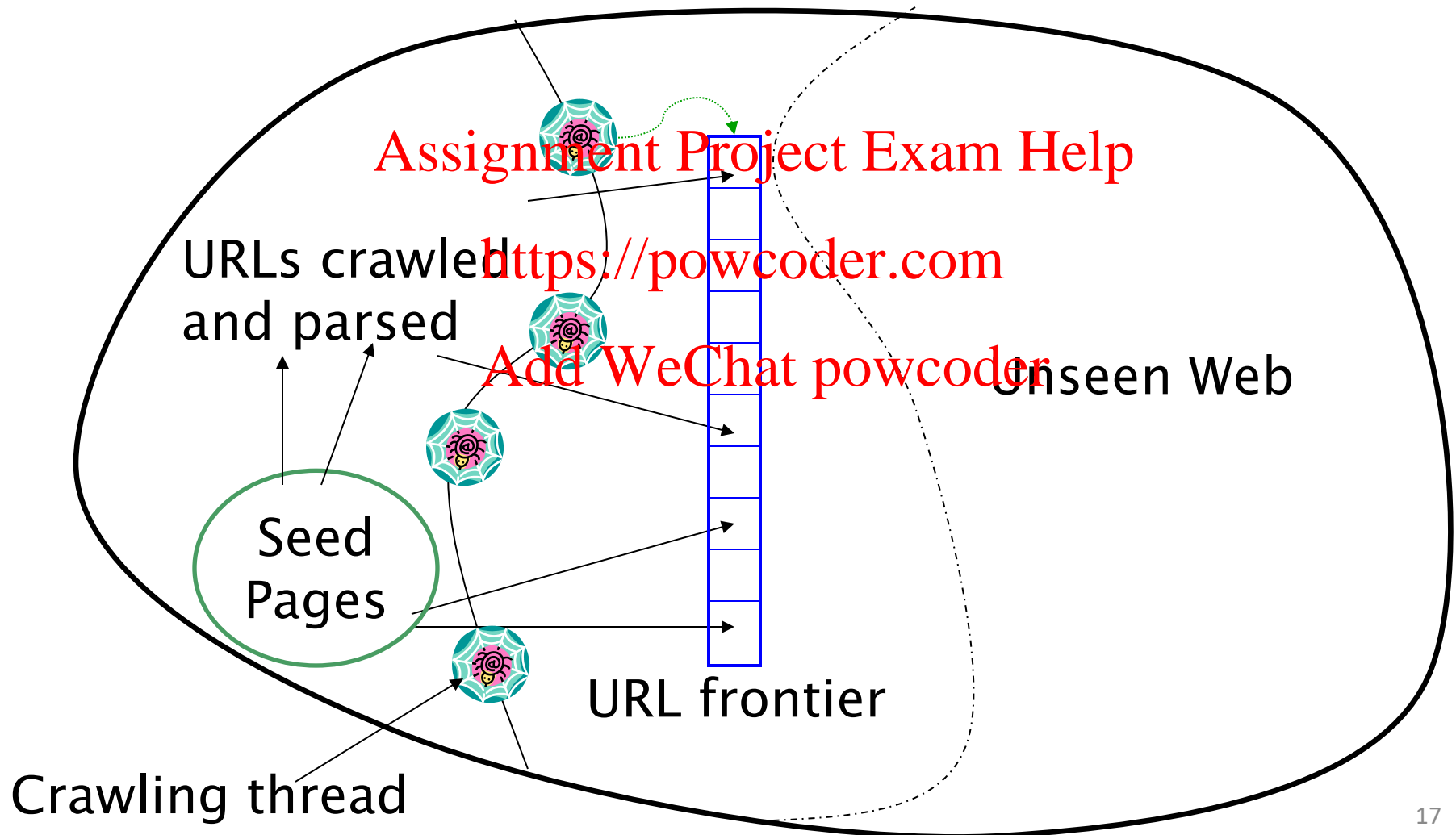
<https://powcoder.com>

Add WeChat powcoder



Assignment Project Exam Help

Updated crawling picture



Assignment Project Exam Help

Simple Crawler Thread

```
1  procedure CRAWLERTHREAD(frontier)
2    while not frontier.done() do
3      website ← frontier.nextSite()
4      url ← website.nextURL()
5      if website.permitsCrawl(url) then
6        text ← retrieveURL(url)
7        storeDocument(url, text)
8        for each url in parse(text) do
9          frontier.addURL(url)
10       end for
11     end if
12     frontier.releaseSite(website)
13   end while
14 end procedure
```

Assignment Project Exam Help

URL frontier

- Can include multiple pages from the same host
- Must avoid trying to fetch them all at the same time
- Must try to keep all crawling threads busy

Assignment Project Exam Help

Explicit and implicit politeness

- Explicit politeness: specifications from webmasters on what portions of site can be crawled
 - robots.txt
- Implicit politeness: even with no specification, avoid hitting any site too often

Assignment Project Exam Help

Robots.txt Add WeChat powcoder

- Protocol for giving spiders (“robots”) limited access to a website, originally from 1994
 - www.robotstxt.org/wc/norobots.html
- Website announces its request on what can(not) be crawled <https://powcoder.com> Add WeChat powcoder
 - For a URL, create a file `URL/robots.txt`
 - This file specifies access restrictions

Assignment Project Exam Help

Robots.txt example

- No robot should visit any URL starting with "/yoursite/temp/", except the robot called "searchengine".

<https://powcoder.com>

```
User-agent: *
```

```
Disallow: /yoursite/temp/
```

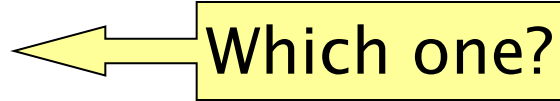
```
User-agent: searchengine
```

```
Disallow:
```

try <http://www.taobao.com/robots.txt>

Assignment Project Exam Help

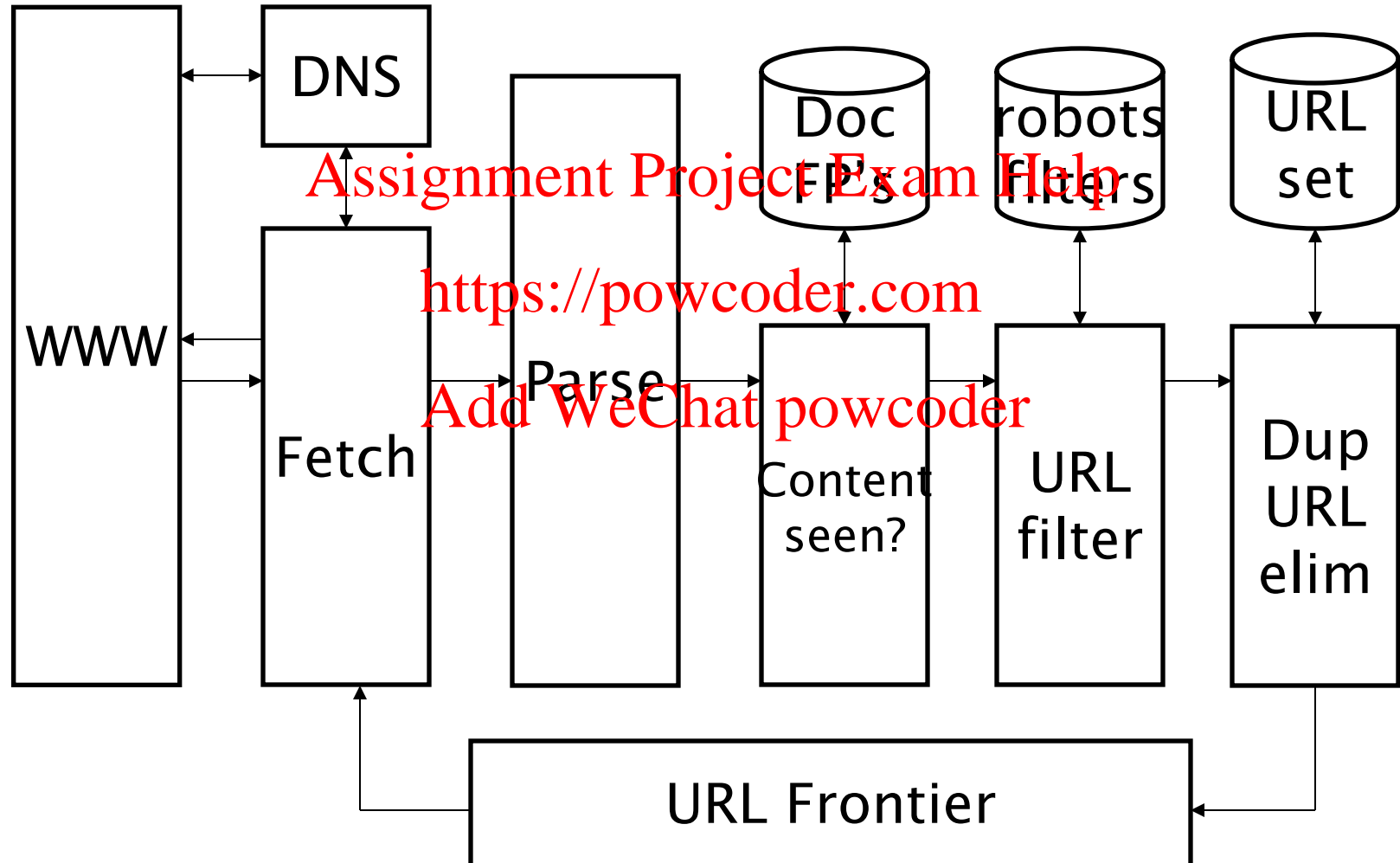
Processing steps in crawling

- Pick a URL from the frontier 
- Fetch the document at the URL
- Parse the URL
 - Extract links from it to other docs (URLs)
- Check if URL has content already seen
 - If not, add to indexes
- For each extracted URL
 - Ensure it passes certain URL filter tests
 - Check if it is already in the frontier (duplicate URL elimination)

E.g., only crawl .edu,
obey robots.txt, etc.

Assignment Project Exam Help

Basic crawl architecture



Assignment Project Exam Help

DNS (Domain Name Server)

- A lookup service on the internet
 - Given a URL, retrieve its IP address
 - Service provided by a distributed set of servers – thus, lookup latencies can be high (even seconds)
- Common OS implementations of DNS lookup are *blocking*: only one outstanding request at a time
- Solutions
 - DNS caching
 - Batch DNS resolver – collects requests and sends them out together

Assignment Project Exam Help

Parsing: URL normalization

- When a fetched document is parsed, some of the extracted links are *relative* URLs
- E.g., at http://en.wikipedia.org/wiki/Main_Page
we have a relative link to <https://powcoder.com>
/wiki/Wikipedia:General_disclaimer which is the same as the absolute URL
http://en.wikipedia.org/wiki/Wikipedia:General_disclaimer
- During parsing, must normalize (expand) such relative URLs

Assignment Project Exam Help

Content seen?

- Duplication is widespread on the web
- If the page just fetched is already in the index, do not further process it
- This is verified using document fingerprints or shingles

Assignment Project Exam Help

Removing Noise

- Many web pages contain text, links, and pictures that are not directly related to the main content of the page
- This additional material is mostly *noise* that could negatively affect the ranking of the page
- Techniques have been developed to detect the content blocks in a web page
 - Non-content material is either ignored or reduced in importance in the indexing process

Assignment Project Exam Help

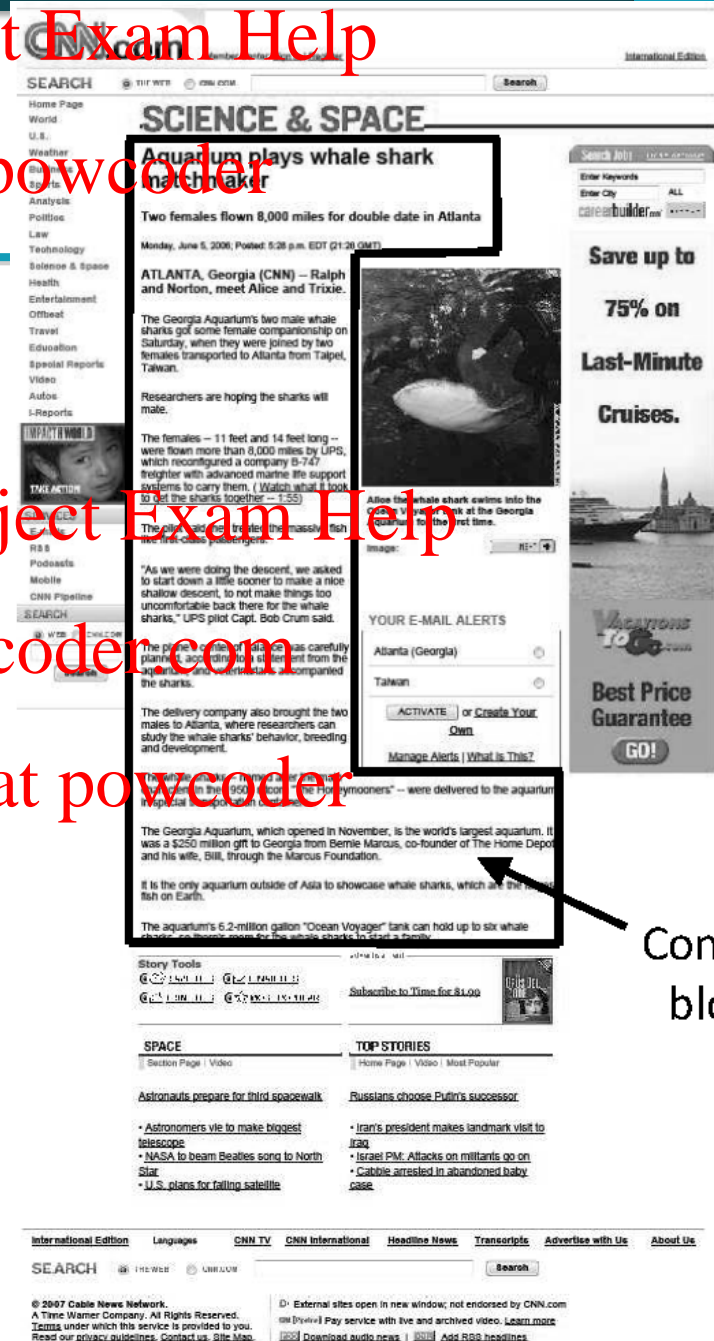
Add WeChat powcoder

Noise Example

Assignment Project Exam Help

https://powcoder.com

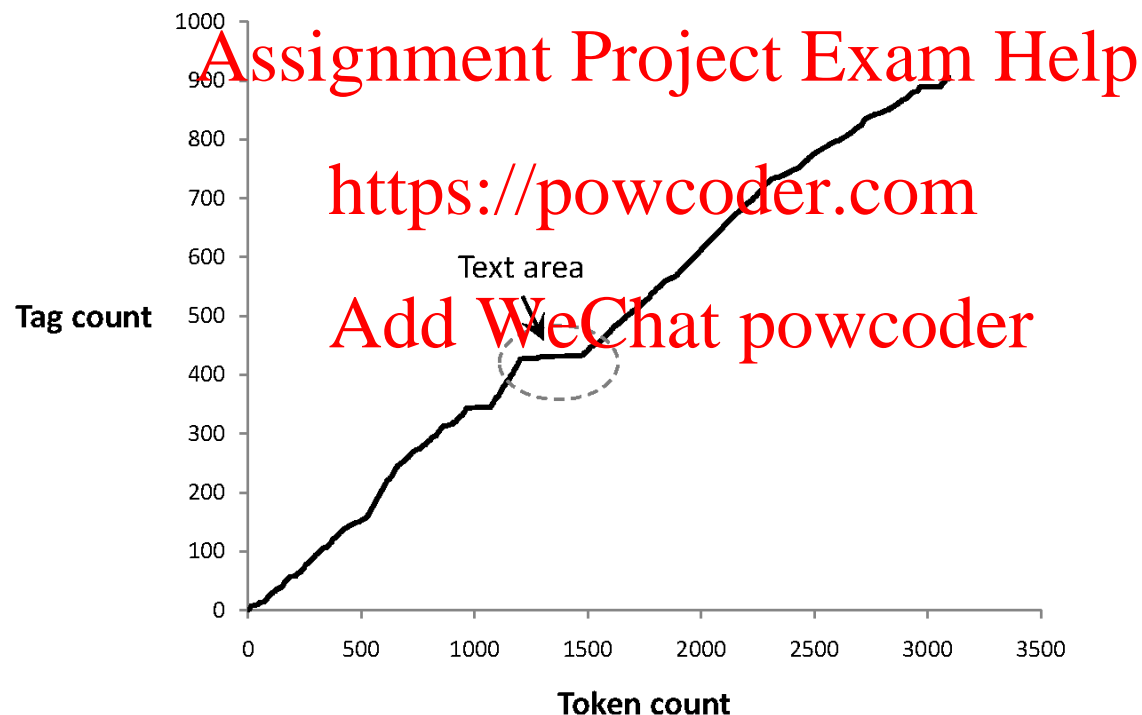
Add WeChat powcoder



Assignment Project Exam Help

Finding Content Blocks

- Cumulative distribution of tags in the example web page



- Main text content of the page corresponds to the “plateau” in the middle of the distribution

Assignment Project Exam Help

Finding Content Blocks

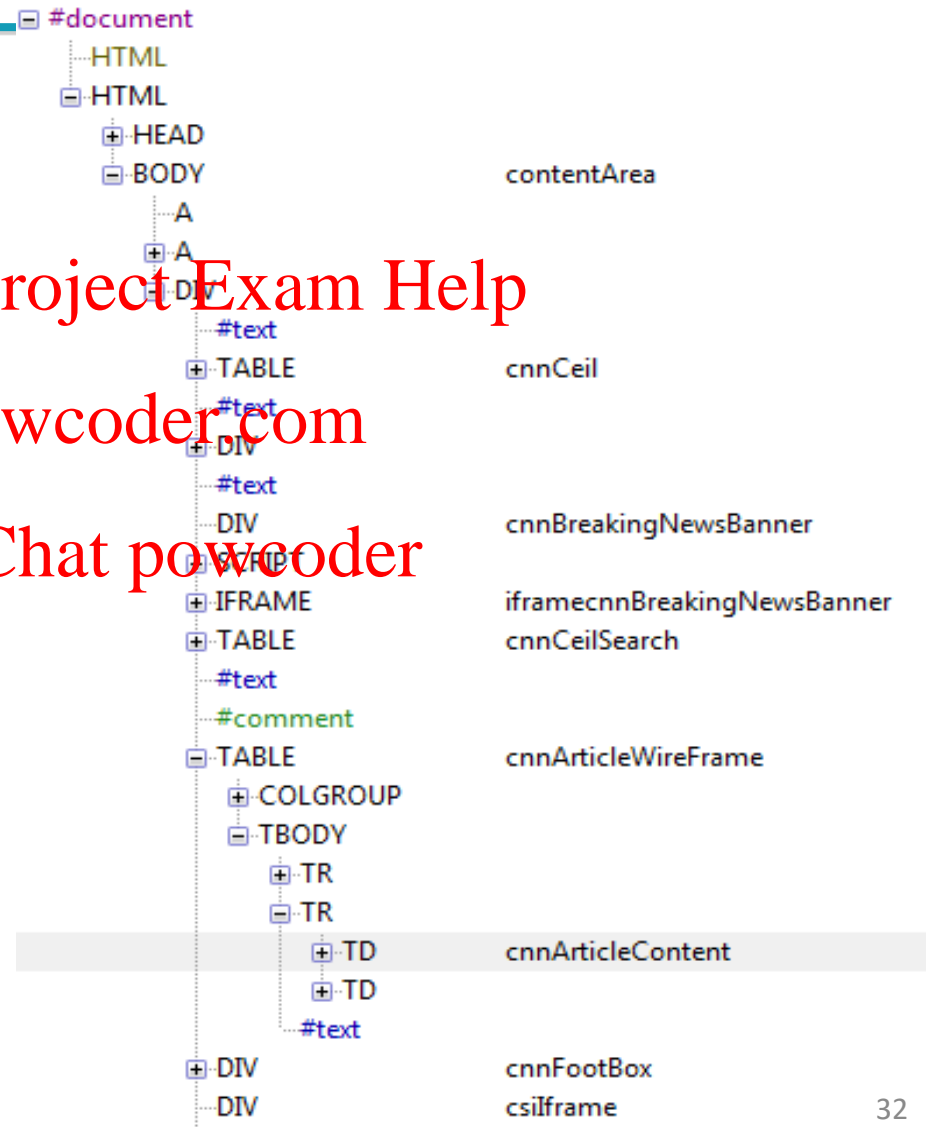
- Represent a web page as a sequence of bits, where $b_n = 1$ indicates that the n th token is a tag
- **Optimization problem** where we find values of i and j to maximize both the number of tags below i and above j and the number of non-tag tokens between i and j
- i.e., maximize

$$\sum_{n=0}^{i-1} b_n + \sum_{n=i}^j (1 - b_n) + \sum_{n=j+1}^{N-1} b_n$$

Assignment Project Exam Help

Finding Content Blocks

- Other approaches use DOM structure and visual (layout) features



Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Assignment Project Exam Help

Filters and robots.txt

- Filters – regular expressions for URL's to be crawled/not
- Once a robots.txt file is fetched from a site, need not fetch it repeatedly
 - Doing so burns bandwidth, hits web server
- Cache robots.txt files

Assignment Project Exam Help

Duplicate URL elimination

- For a non-continuous (one-shot) crawl, test to see if an extracted+filtered URL has already been passed to the frontier
- For a continuous crawl – see details of frontier implementation

Assignment Project Exam Help

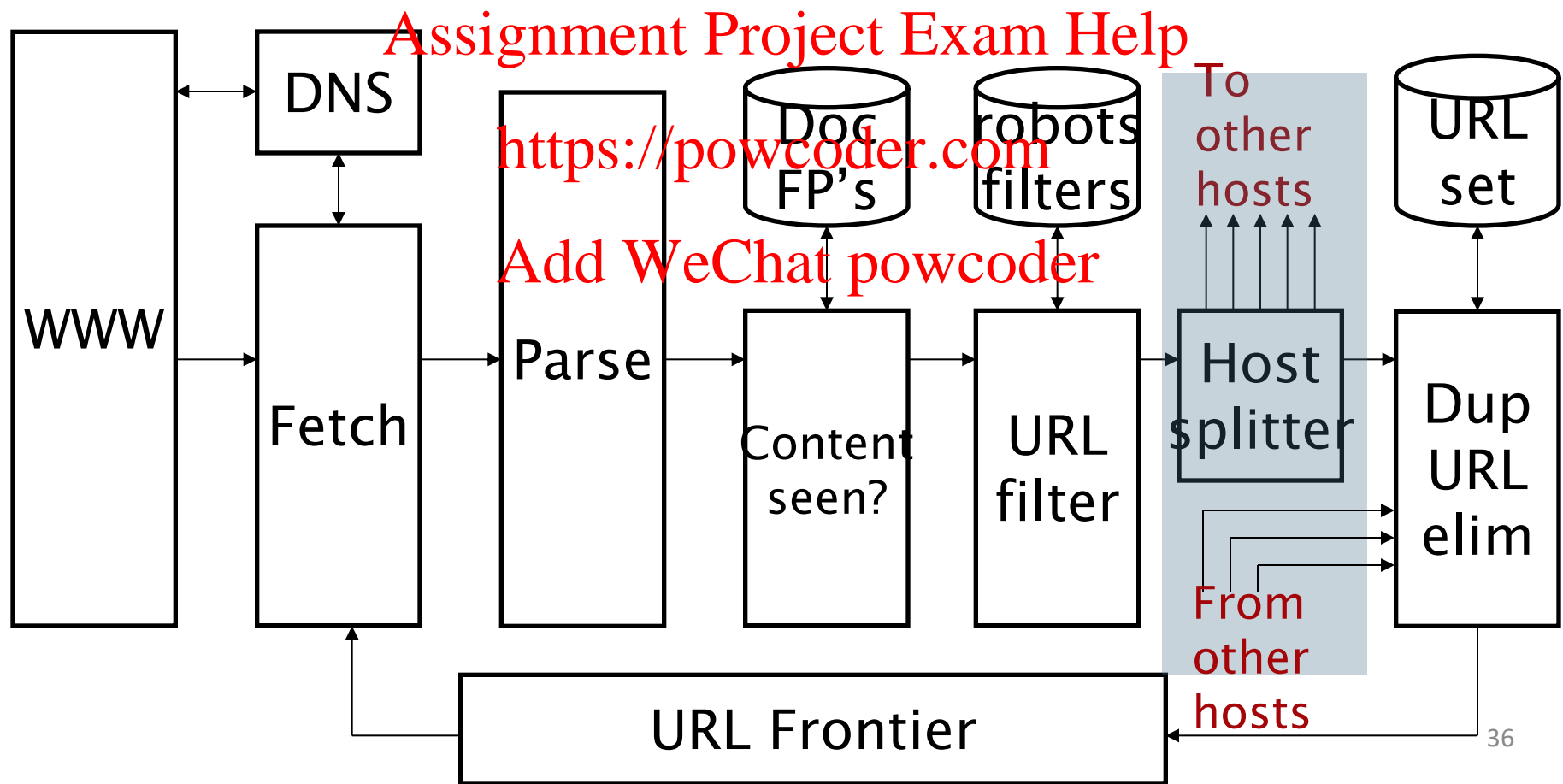
Distributing the crawler

- Run multiple crawl threads, under different processes – potentially at different nodes
 - Geographically distributed nodes
- Partition hosts being crawled into nodes
 - Hash used for partition
- How do these nodes communicate?

Assignment Project Exam Help

Communication between nodes

- The output of the URL filter at each node is sent to the Duplicate URL Eliminator at all nodes



Assignment Project Exam Help

URL frontier: two main considerations

- Politeness: do not hit a web server too frequently
- Freshness: crawl some pages more often than others
 - E.g., pages (such as News sites) whose content changes often

These goals may conflict each other.

(E.g., simple priority queue fails – many links out of a page go to its own site, creating a burst of accesses to that site.)

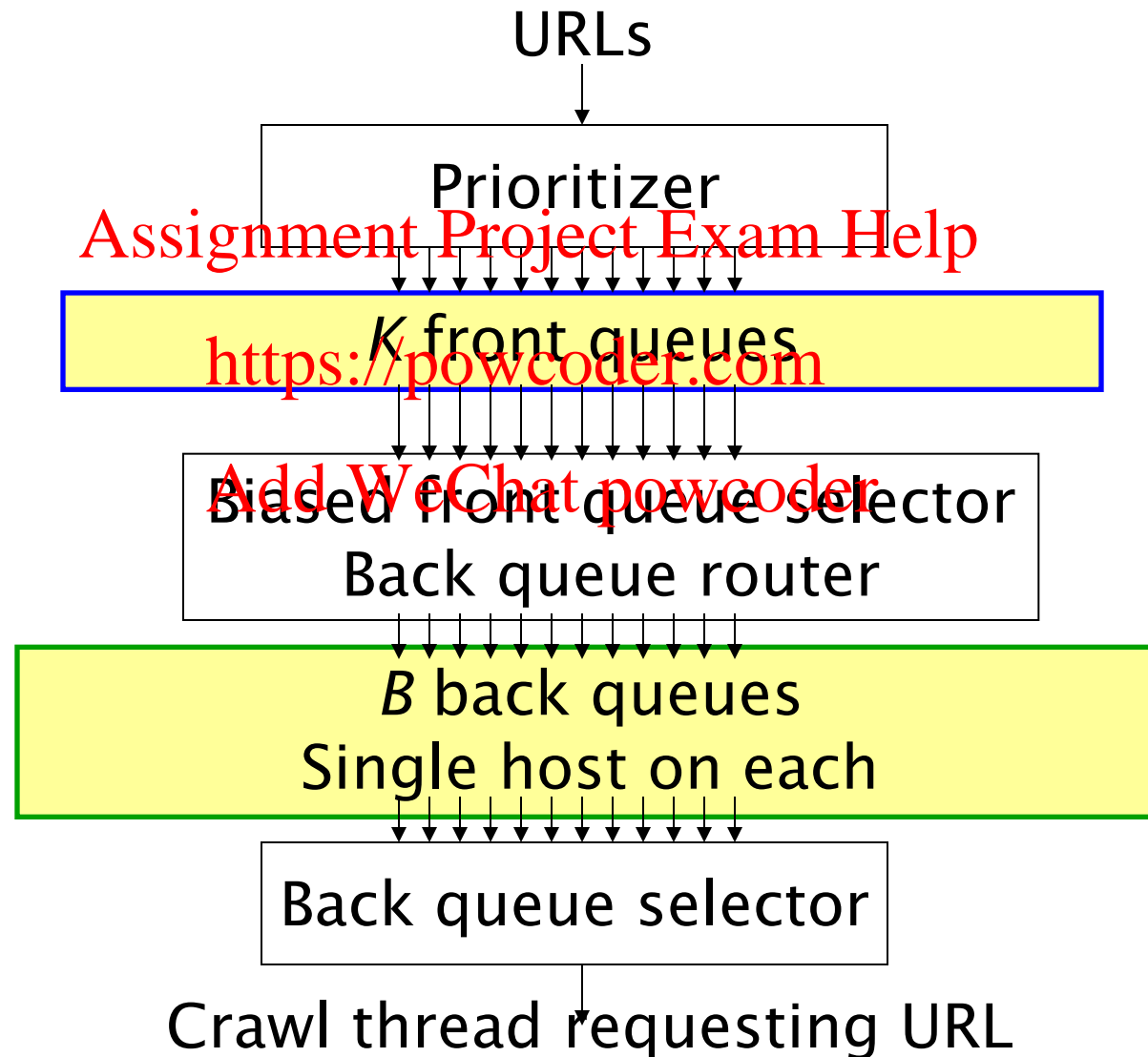
Assignment Project Exam Help

Politeness – challenges

- Even if we restrict only one thread to fetch from a host, can hit it repeatedly
- Common heuristic: insert time gap between successive requests to a host that is \gg time for most recent fetch from that host

Assignment Project Exam Help

URL frontier: Mercator scheme



Assignment Project Exam Help

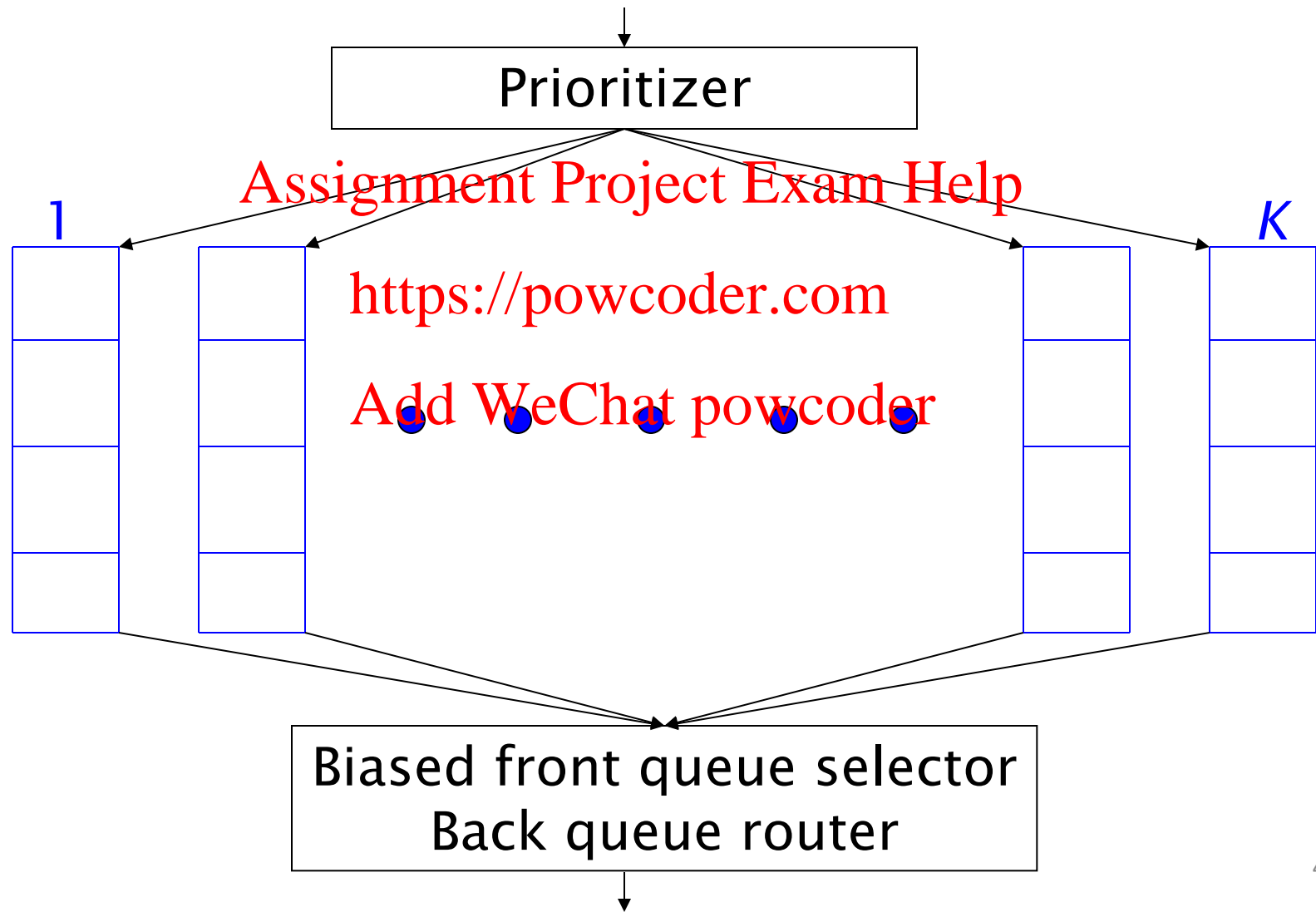
Mercator URL frontier

- URLs flow in from the top into the frontier
- **Front queues** manage prioritization
- **Back queues** enforce politeness
- Each queue is FIFO

<https://powcoder.com>
Add WeChat powcoder

Assignment Project Exam Help

Front queues



Assignment Project Exam Help

Front queues

- Prioritizer assigns to URL an integer priority between 1 and K
 - Appends URL to corresponding queue
- Heuristics for assigning priority
 - Refresh rate sampled from previous crawls
 - Application-specific (e.g., “crawl news sites more often”)

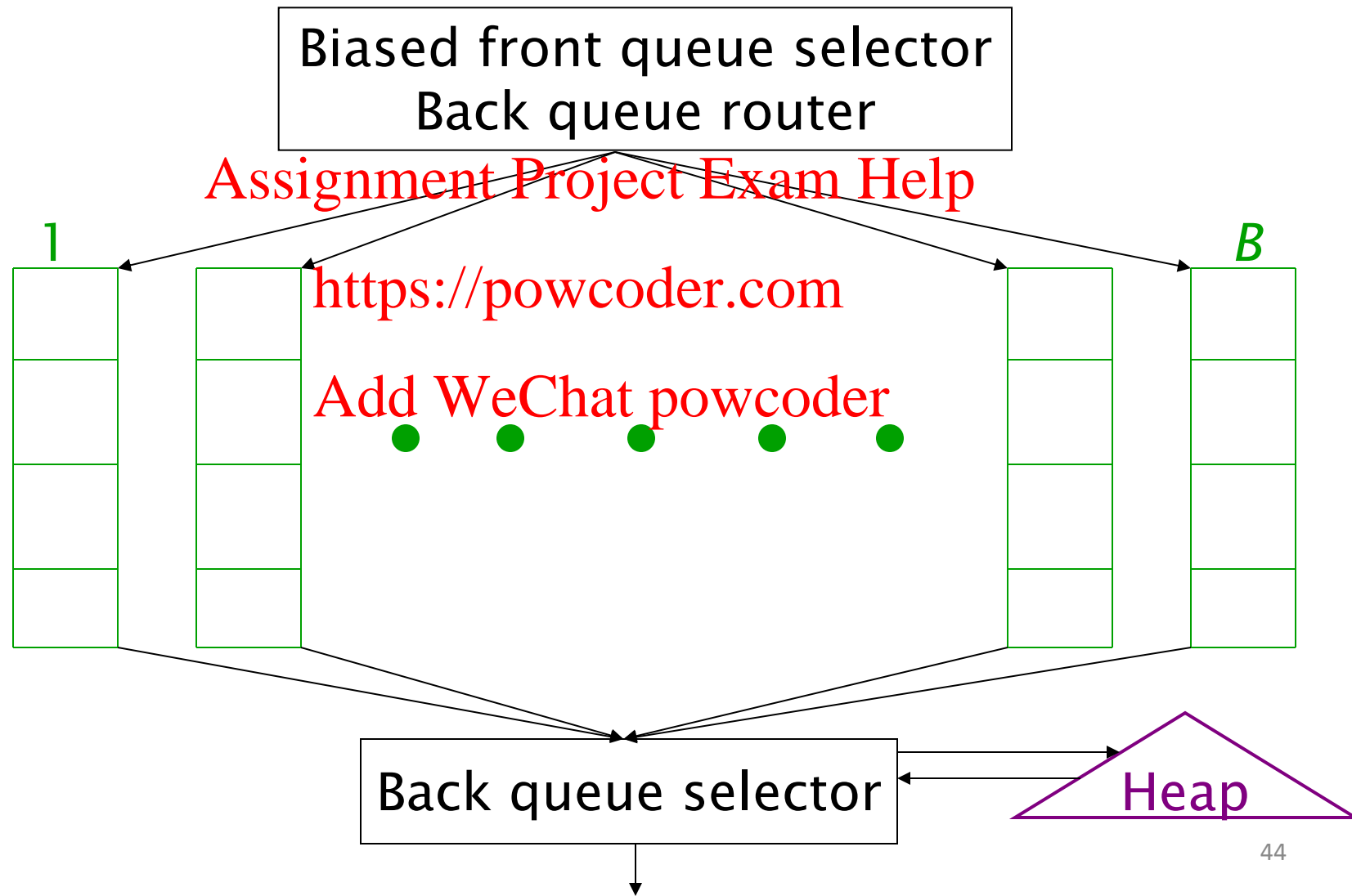
Assignment Project Exam Help

Biased front queue selector

- When a back queue requests a URL (in a sequence to be described): picks a **front queue** from which to pull a URL
- This choice can be round robin biased to queues of higher priority, or some more sophisticated variant
 - Can be randomized

Assignment Project Exam Help

Back queues



Assignment Project Exam Help

Back queue invariants

- Each back queue is kept non-empty while the crawl is in progress
- Each back queue only contains URLs from a single host
 - Maintain a table from hosts to back queues

Host name	Back queue
...	3
	1
	<i>B</i>

Assignment Project Exam Help

Back queue heap

- One entry for each back queue
- The entry is the earliest time t_e at which the host corresponding to the back queue can be hit again
- This earliest time is determined from
 - Last access to that host
 - Any time buffer heuristic we choose

Assignment Project Exam Help

Back queue processing

- A crawler thread seeking a URL to crawl:
- Extracts the root of the heap
- Fetches URL at head of corresponding back queue q (look up from table)
- Checks if queue q is now empty or so, pulls a URL v from front queues
 - If there's already a back queue for v 's host, append v to q and pull another URL from front queues, repeat
 - Else add v to q
- When q is non-empty, create heap entry for it

Assignment Project Exam Help

Number of back queues

- Keep all threads busy while respecting politeness
- Mercator recommendation: three times as many back queues as crawler threads

<https://powcoder.com>

Add WeChat powcoder

Assignment Project Exam Help

Resources

- IIR Chapter 20
- [Mercator: A scalable, extensible web crawler \(Heydon et al. 1999\)](#)
- [A standard for robot exclusion](#)

<https://powcoder.com>

Add WeChat powcoder