Initialize: f empty associative array

for each token in the data stream
   Let i be the corresponding item of the token
   If i is in f then
      f[i] = f[i] + 1     (increment frequency count for f[i])
   else if |f| < k – 1 then     (if array is not full; k-1 is size of array)
      f[i] = 1               (insert item into array and set its frequency to 1)
   else
      for each i in I do f[i] = f[i] – 1
      if f[i] = 0 then remove i from f

for each item i in f do
   add i to the result set
data collection in the web - hacker style

Arend Hintze
the internet
THE INTERNET OF THINGS

www.comsoc.org/blog
We want to implant this RFID tag in you.

That violates my rights!

We want to implant this RFID tag in you and it’s also a cellphone, digital camera, and MP3 player.

Wrong

Right

Cool!
URLs

HTTP://WWW.BIGWEBFAQ.COM/FOLDER/FILE.HTM
http:\www.google.com\index.html
browser takes this:
turns it into this:
however HTML is “just” a “text” file…

```html
<!--html code starts here -->
<html>
<head>
  <title>Video Library </title>
</head>
<body bgcolor="#DEDEDE">
  <div align="center">
    <h2>Media Streaming : Video Library Project</h2>
    <h3>If you are viewing videos from this collection
    for the first time, please download
    <a href="http://localhost/videofiles/StreamingMediaPlayer.msi">
      Streaming Media Player</a></h3>
    <br/>
    <a href="ums:\TCP=localhost:5119\videofiles\hrm.mpg">
      Introduction Human Resource Management</a>
  </div>
</body> </html>
<!--html code ends here -->
```

a text file written in HyperText Markup Language
HTML

http://www.w3schools.com/html/html5_form_attributes.asp
let’s checkout wikipedia pages
static vs. dynamic pages

• static pages are indeed HTML only

• content never changes, or is changed by changing the file

• a program (php, server) creates a HTML file on the fly

• the HTML contains scripts (javascript) or (flash, media player…)

• a dynamic website can take parameters from the URL
wait, there is more: JSON

• java script object notation

• allows you to transform an object into a JSON string

• allows you to transform a JSON string into a JS object

• each language (more or less) has it’s own wrapper and parser

tutorial: http://pymotw.com/2/json/
now what?

• we have HTML pages…
• we have JSON data…
• -> CURL allows you to get data as a string
curl example
but what is “data”? 

- A collection of objects (items) and their associated attributes

- an object specifies an entity to be described: customer, product, page, day, location…

- an attribute describes or characterizes the object: eye color, temperature, value, price, time…
## Attributes

<table>
<thead>
<tr>
<th>Tid</th>
<th>Refund</th>
<th>Marital Status</th>
<th>Taxable Income</th>
<th>Cheat</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Yes</td>
<td>Single</td>
<td>125K</td>
<td>No</td>
</tr>
<tr>
<td>2</td>
<td>No</td>
<td>Married</td>
<td>100K</td>
<td>No</td>
</tr>
<tr>
<td>3</td>
<td>No</td>
<td>Single</td>
<td>70K</td>
<td>No</td>
</tr>
<tr>
<td>4</td>
<td>Yes</td>
<td>Married</td>
<td>120K</td>
<td>No</td>
</tr>
<tr>
<td>5</td>
<td>No</td>
<td>Divorced</td>
<td>95K</td>
<td>Yes</td>
</tr>
<tr>
<td>6</td>
<td>No</td>
<td>Married</td>
<td>60K</td>
<td>No</td>
</tr>
<tr>
<td>7</td>
<td>Yes</td>
<td>Divorced</td>
<td>220K</td>
<td>No</td>
</tr>
<tr>
<td>8</td>
<td>No</td>
<td>Single</td>
<td>85K</td>
<td>Yes</td>
</tr>
<tr>
<td>9</td>
<td>No</td>
<td>Married</td>
<td>75K</td>
<td>No</td>
</tr>
<tr>
<td>10</td>
<td>No</td>
<td>Single</td>
<td>90K</td>
<td>Yes</td>
</tr>
</tbody>
</table>
turning “the web” into “the data”

• use provided API (Application Programming Interface)
  • websites provide dedicated interfaces (url schemes)
  • often there are many modules/libraries for various languages
  • usually require some kind of credentials
• scrape, crawl, spider websites and extract the data
  • automated programs load page after page (scrape)
  • follow up on links (crawl, spider)
  • search engines do this to get content
• get data directly from devices
  • check card readers, RFID, cell phones, cameras …
  • usually requires dedicated hardware/software solution, no free/open access
MapQuest has an API!

- Application Programming Interface
- the company wants you to use those
- not every API is free
- usually fast, reliable, and supported
- some APIs work on one platform only … iPhone SDK works only in objective C
- more about this next Monday…
crawler / spider

- each HTML page has links to other pages:
  `<a href="http://www.w3schools.com/">Visit W3Schools</a>`

- following up on these links allows to automatically gather data
crawler constraints

• do not load the same page twice

• branching factor determines if you come to an end or have exponential growth

• load new pages either breadth first or depth first?

• when to stop?
anatomy of a web crawler

**Initialize**: append seed URLs to Queue

**Dequeue**: remove a URL from Queue

**Terminate?**

- yes: **Terminated**
- no:
  - **Fetch**: retrieve web page associated with URL
  - **Parse**: extract URLs from retrieved web page
  - **Enqueue**: append extracted URLs to queue

**depth first or breadth first?**
scrapers are different

http://forums.maschinen-mensch.com/viewforum.php?f=2&sid=60ba49c3faeb7470a2b32ab0e753716a
was that allowed?
Robot Exclusion Protocol!

• each site can have a robot.txt file which specifies if crawling is permitted

• legally binding?

• even if the data is “public”, keep in mind that you are using someone else's resources… !

• allows and disallows crawling
# robots.txt for http://archive.lib.msu.edu

User-agent: *
Disallow: /DMC/Elizabeth
Disallow: /DMC/Grant
Disallow: /DMC/Scann
Disallow: /TIC/ITS
Disallow: /TIC/grnma
Disallow: /TIC/strn
Disallow: /TIC/turfn
Disallow: /LDC
Disallow: /vendors

# Uncomment the following after Google re-indexes with redirect pages:
# Disallow: /DMC/Turf
# Disallow: /DMC/USGA
META tags on a webpage also tell a crawler what not to do

- Meta tags are placed between `<head>` ... `</head>` tags in HTML

```html
<html>
<head>
<title>...</title>
<META NAME="ROBOTS" CONTENT="NOINDEX, NOFOLLOW">
</head>
```

- `<META NAME="ROBOTS" CONTENT="NOFOLLOW">`
  - To not follow links on this page
- `<META NAME="ROBOTS" CONTENT="NOINDEX">`
  - To not appear in Google’s index
- `<META NAME="GOOGLEBOT" CONTENT="NOARCHIVE">`
  - To not archive copy in search results

Wget

- A freely available GNU utility for web crawling

- Supports both HTTP and FTP
  - Can recursively traverse the structure of HTML documents and FTP directory trees
  - Can specify wildcards to match certain types of files
  - Can restrict the maximum depth of the directory traversed

- Available with both command line argument and graphical user interface

- Included in most Unix and Linux systems
Wget example

- **wget** `http://www.cse.msu.edu`
  - Retrieve the index.html file from www.cse.msu.edu

- **wget** `-t 30 http://www.cse.msu.edu`
  - Retry 30 times if access fails

- **wget** `-r http://www.cse.msu.edu`
  - Recursively retrieve files under the hierarchy structure of www.cse.msu.edu (default: recurse up to 4 levels)

- **wget** `-o log.txt http://www.cse.msu.edu`
  - Direct output messages to log.txt file

For more examples: