22 10 2015 **SEWN 2015 - Lab 3 MSc version (Assessed Coursework) - Web crawling**

In this lab you will build a simple web crawler, or robot, which will extract the links from the pages from a given series of web pages to calculate various statistics. This will help you to understand how search engines build an index of the web, and why some pages may never be found.

In order to do this assignment, there are two prerequisite tasks:

1. In the lecture slides 'Searching the Web', the architecture of a web crawler is described, and the basic algorithm crawlers use to traverse pages and extract links is described.

   We want you to write a program that takes a URL as input (either on the command line, at a prompt, or simply embedded in the program code), downloads the page at the given URL and then outputs a list of the links on the page (i.e. any urls in an `<a>` tag). You can use *any programming language* that you are familiar with to do this but preferably one which can be demonstrated within DCSIS. You can test your program by using:

   `http://www.dcs.bbk.ac.uk/~martin/sewn/ls3/testpage.html`

   Where your program should write the five links, e.g. `<a href="address of some page">link text</a>`, it contains to screen.

2. Find out about the use of `robots.txt` to specify which pages of a website should be crawled by robots (a web search for "robots.txt" or "Robots Exclusion Protocol" should lead to plenty of information).

We now require you to use (1) and (2) above as a basis to write a simple web crawler which will index the links from a series of approximately a dozen 'Visited' web pages to calculate various statistics. In order to do this, you have a choice of **one** of the following options (3), (4) or (5):

3. To download the pages of the web site from:
   `http://www.dcs.bbk.ac.uk/~martin/sewn/ls3/SEWN_2015_Labsheet3_archive.zip`

   And extend the program above to index each page and extract their links. However, in doing this, your program must obey the instructions contained in the `robots.txt` supplied at the root URL:
   `http://www.dcs.bbk.ac.uk/~martin/sewn/ls3/`.

   Please note that the location of `robots.txt` for this assignment is non-standard behaviour, i.e. if `http://www.foo/bar/webtech/` is supplied as the root URL you should find `www.foo/bar/webtech/robots.txt` and NOT `www.foo/robots.txt` as the robots protocol would usually dictate.

   For each page your crawler should:

   a. Log and follow all allowed links within the site (i.e. that are relative links or that begin with the root URL).

   b. Log links to pages that are disallowed by `robots.txt` and any links to pages outside of the root site (i.e. *record the URL* of such pages but *do not download the page*).

   NOTE: There are approximately a dozen pages to be visited and indexed. Remember to **log and read each page only once!**

OR

4. This option is similar to (3) above but requires the program from (1) to be extended to accept the root URL `http://www.dcs.bbk.ac.uk/~martin/sewn/ls3/` as a 'seed'. This seed can be a program parameter or can be hardcoded. Upon receipt of this 'seed', your program should dynamically crawl the pages of the web site, but again obey the `robots.txt` file described in (3) above.

   This option carries *bonus* points.

OR

5. As an alternative to the programming required of (3) and (4) above, you are free to download the pages of the web site and use any **demonstrable method of your choice** to parse the links from each page. However, you are still required to obey the `robots.txt` file described above.

   You may want to import the contents of the pages into Excel and use macros to derive the links per page, or dump the text files into a database and use regular expressions in SQL queries to extract links.

   For further information, see:

   - http://howtouseexcel.net/how-to-extract-a-url-from-a-hyperlink-on-excel
   - http://dev.mysql.com/doc/refman/5.1/en/regexp.html

6. With your outputs from (3), (4) or (5) above, produce a file called `crawl.txt` which lists the links for each Visited and a list of the links contained in it. The file should be in the format:

```
<Visited URL1>
      <Link URL1>
      <Link URL2>
<Visited URL2>
      <Link URL1>
... etc.
```

   We also request that you list the number of links to visited pages per `<Visited URL>`, numerically in a file called `results.txt`. The file should be in this format where `X` and `Y` are hypothetical numbers of links:

```
<Visited URL1>
      <No of links to Visited pages: X>
<Visited URL2>
      <No of links to Visited pages: Y>
... etc.
```

NOTES:

1. Any alternative (3), (4) or (5) used above index the Visited pages of the web site must include both relative and absolute links. Further information about relative and absolute links can be found here: http://webdesign.about.com/od/beginningtutorials/a/aa040502a.htm.

2. If using a programmatic solution for (3) or (4) above, your crawler should **only follow URLs beginning with the root URL** http://www.dcs.bbk.ac.uk/~martin/sewn/ls3/**.**

3. If using a programmatic solution for (3) or (4) above, you are not restricted to Java. You are free to use any current programming language, preferably one which can be run in the PC labs if this is possible.

DO NOT RUN YOUR CRAWLER ON ANY OTHER SITE!

TO FIND OUT WHY NOT, READ http://www.robotstxt.org/guidelines.html

What to hand in:
Submit a single .zip file to the **Lab 3: Web crawling** drop box in Moodle containing:

1. An A4 page describing how your solution works (`.doc(x)` or `.pdf` format).
2. Any program code, along with any special instructions for compiling and running it (if there are several code files, please include the folder structure in your `.zip` file), and / or other files required. If you use (5) above, you may be required to demonstrate your solution in the labs.
3. The `crawl.txt` and `results.txt` files produced from the crawl in part (6).

Submission deadline: 12 11 2015.

Late assignments: No extensions are available as for this lab and any late submissions will be graded as per the guidelines of the relevant course being studied.