Assignment 1 – Part I (count for 50% credit for assignment 1):
(Part II will be posted in the following weeks)

This is an individual assignment. If you get help from others you must write their names down on your submission and explain how they helped you. If you use external resources you must mention them explicitly. You may use third party libraries but you need to cite them, too.

Date posted: Friday January 22, 2016

Date Due: Monday February 1, 2016

Description:

Goal: Implementing your own web crawler. Performing focused crawling

Description:

Task 1: Crawling the documents:

A. Start with the following seed URL: http://en.wikipedia.org/wiki/Sustainable_energy; a Wikipedia article about green energy.
B. Your crawler has to respect the politeness policy by using a delay of at least one second between your HTTP requests.
C. Follow the links with the prefix http://en.wikipedia.org/wiki that lead to articles only (avoid administrative links containing :). Non-English articles and external links must not be followed.
D. Crawl to depth 5. The seed page is the first URL in your frontier and thus counts for depth 1.
E. Stop once you've crawled 1000 unique URLs. Keep a list of these URLs in a text file. Also, keep the downloaded documents (raw html, in text format) with their respective URL for future tasks (transformation and indexing)

Task 2: Focused Crawling:

Your crawler should be able to consume two arguments: a URL and a keyword to be matched against text, anchor text, or text within a URL. Starting with the same seed in Task 1, crawl to depth 5 at most, using the keyword "solar". You should return at most 1000 URLS for each of the following:

A. Breadth first crawling
B. Depth first crawling*
C. In a few sentences compare and explain the approaches above. Briefly compare the results obtained in A & B in this task in terms of the total number of URLs crawled, and the top 5 URLs (topical content).

What to hand in?

1- Your source code for solving this assignment.
2- A readme text file explaining in detail how to setup, compile, and run your program.
3- THREE text files each containing 1000 URLs at most (one file for Task 1-E, and two files for Task 2- A & B).
4- A text file with your explanation for Task 2-C.

*: Even though DFS is not particularly introduced in class, and not in the paper of Baeza-Yates et al on "Crawling the country: better strategies than breadth-first for web page ordering", it is a well-known and widely used graph traversal algorithm. It would be interesting to test it and compare with BFS.