Information Retrieval
**Project 2 - Crawling**
**Due date: 2/8**

This assignment is to be done in groups of 1, 2 or 3. You can use text processing code that you or any classmate wrote for the previous assignment. You cannot use crawler code written by non-group-member classmates. Use code found over the Internet at your own peril -- it may not do exactly what the assignment requests. If you do end up using code you find on the Internet, you must disclose the origin of the code. **As stated in the course policy document, concealing the origin of a piece of code is plagiarism**. Use the Message Board for general questions whose answers can benefit you and everyone.

**Crawling library, Java:** http://code.google.com/p/crawler4j/
**Crawling library, Python:** https://github.com/Mondego/crawler4py

**Goal**: Using one of the libraries above, write a program to crawl the domain `ics.uci.edu` in order answer the questions below. Also **you should store the pages that you crawl, because they will be needed for the next project.**

## Specifications

1. **VERY IMPORTANT**: Set the name of your crawler's User Agent to "UCI WebCrawler *student_IDs* " with one one/two/three (group project) student IDs
2. **VERY IMPORTANT**: wait 300ms before sending page requests to the same subdomain.
3. You should use Java or Python to complete this homework, so that you can take advantage of crawler4j/crawler4py. Otherwise, you need to write your own crawler infrastructure (not recommended).
4. Start with this seed http://www.ics.uci.edu and crawl from there
5. Only the domain ics.uci.edu, and all of its subdomains (*anything*.ics.uci.edu), should be crawled; all other domains should be ignored.
6. We will verify the **execution** of your crawler in one or more of Prof. Lopes' web servers' logs. These servers are in any correctly-written crawler path. If we don't find log entries for your student ID, that means your crawler didn't perform as it should or you didn't set its name correctly; in the latter case we can't verify whether it ran successfully or not, so we'll assume it didn't.
7. **At points, the assignment may be underspecified. In those cases, make your own assumptions and document them**.

## Questions:

1. How much time did it take to crawl the entire domain?

2. How many unique pages did you find in the entire domain? (Uniqueness is established by the URL)

3. How many subdomains did you find? Submit the list of subdomains ordered alphabetically and the number of unique pages detected in each subdomain. The file should be called Subdomains.txt, and its content should be lines containing
   *URL*, *number*
   http://vision.ics.uci.edu, 10 (← not the actual number here)
   etc.

4. What is the longest page in terms of number of words? (HTML markup doesn't count as words)

5. What are the 500 most common words in this domain? (**Ignore English stop words**, which can be found, for example, here) Submit the list of common words ordered by frequency.

6. What are the 20 most common 2-grams? (again ignore English stop words) A 2-gram, in this case, is a sequence of 2 words that aren't stop words and that haven't had a stop word in between them. Submit the list of 20 2-grams ordered by frequency.

## Submitting Your Assignment

Your submission should be one single zip file submitted to EEE. This file should contain: (a) your source code; (b) your answers to ALL the questions in a file called "answers" (.txt, .doc or .pdf). If there is anything else you wish to communicate to the Reader, such as implementation assumptions made, this should be placed into an additional README.txt file within the zip file.

## Evaluation Criteria

Your assignment will be graded on the following three criteria.

1. Correctness
    a) Did you crawl the domain correctly? We will verify that in our servers' logs.
    b) Are your answers to the questions reasonable?
        i) Note that correct answers are not valid without evidence of correct crawling
        ii) Answers by different crawlers will vary due to a number of factors. "Correctness" of answers will be based on how reasonable they are.
2. Understanding
    a) There will be a face-2-face meeting with the Reader where you will be asked questions related to your crawler's implementation
    b) All members of the group are expected to demonstrate in-depth understanding of the crawler. In cases where differences of understanding are detected, the scores will reflect that.