CECS 429/529 - Homework 2

1. From *Introduction to Information Retrieval*, do Exercise 1.2 in section 1.2.
2. From *Introduction to Information Retrieval*, do Exercise 1.3 in section 1.2.
3. Give an example of postings lists `a`, `b`, and `c` of sizes $x < y < z$ respectively, for which

   `(a AND b) AND c`

   will have a longer running time than

   `(b AND c) AND a`.

   Justify your answer.
4. Your project, when it is assigned, will index a directory of files, where each individual file is considered a document in the corpus. Your first homework assignment involved text file documents for each chapter in Moby Dick. Your project will involve thousands of articles downloaded from the National Parks Service website, `http://www.nps.gov`. Unfortunately those articles were downloaded as a single file. Your task in this assignment is to "split" this large file of articles so that each article exists in its own physical file on disk.

   The tool I used to download the articles placed the entire corpus into a single JSON ("**J**ava**S**cript **O**bject **N**otation") file. JSON is a data description format suitable for transmitting custom data types in a language-neutral way. An arbitrary object is expressed in JSON as a sequence of key/value pairs; the JSON data for a "student" type object might look like this:

   ```
   {
     "id" :  "00123456",
     "lastName" :  "Terrell",
     "firstName" :  "Neal",
     "majors" :  ["Computer Science", "Mathematics"]
   }
   ```

   The curly braces denote the start and end of a single object. Square brackets denote a value that is a list/array. The data above define an object with four fields, one of which is a list of strings. Simple! For more details, visit `http://json.org`.

   Perform the following tasks:

   (a) Download the file `all-nps-sites.json` from BeachBoard's Project section. This file contains 36,803 articles from the NPS website scraped as JSON data.

   (b) Open the file in a sophisticated IDE like Visual Studio or text editor like Sublime. If you're lucky, your editor will be able to format the JSON to make it look nice, and you'll see something like this:

   ```
   {
     "documents" :  [
       {
         "title" :  "The title of the article",
         "body" :  "The body of the article",
         "url" :  "https://www.nps.gov/........."
       },
       ...
     ]
   }
   ```

   (c) The file contains a single object with a single field called "`documents`". The value in that field is a list of objects, where each object represents one article that was scraped from the website. Each article object has three fields: "title", "body", and "url". We will use those fields later.

   (d) For this assignment, you need to create a short program in whatever language you prefer. Your program will open the JSON file of documents, iterate through the `documents` list and, for each

entry in that list, create a new JSON file whose contents are the JSON data for that entry. Name the files `article1.json`, `article2.json`, etc.

(e) **Do not try and parse the JSON file manually**. Find a JSON-parsing library for your chosen language and download it. Figure out how to use it in your program. Most libraries should be easy to use: they will involve a function call to load the main JSON file into some kind of "JSON" object, then will allow you to iterate through each entry in the "`documents`" field. For each entry in that field, you will create a new JSON file using the library and add the title, body, and url fields from the document entry to the new file. Some suggestions after brief searching:

i. Java: Google GSON

ii. C#: Json.NET

iii. C++: Boost property_tree