MSCI 541 - HW2
Due: noon, Friday, Feb 17, 2012, in the 541 drop box.

Data
Same data as HW1.

Tasks
1. Perform an in memory inversion of the data such that you save a count of each word in each document. You do not need to store position information.
2. Save your inverted data to a file.
3. Write code that can read in your inverted file back into memory.
4. Also record for each document its length and save this information to a file and be able to read it back into memory.
5. Your code must retain an ability to know the DOCNO for each document.
6. Compute and report:
    a. How many words are in your vocabulary?
    b. Plot a histogram of the length of postings.
    c. Plot a histogram of the length of documents.

Suggestions
1. Consider using a stopword list to reduce the amount of memory required. Here is a decent set of stopwords: http://www.search-engines-book.com/data/stopwords
2. Consider using a stemmer to conflate terms to common roots. Code for Porter stemmer: http://tartarus.org/martin/PorterStemmer/
3. Downcase words, etc. These recall enhancing techniques don't hurt precision much but they also save space.
4. A hashtable (Dictionary in C#/VB.net) from term to postings should be what you need. I would make the postings a List (C#/VB) of a struct that holds an integer document ID and the count of the term in the document. For example:

```
struct Posting
{
        public int docID ;
        public int termCount ;
}
```

You'll need to save the integer doc ID and string DOCNO to a file for later purposes, but you don't need to hold them in memory for the inversion.

I was able to invert the LA Times documents using less than 800 MB of RAM in a couple of minutes with some C# code using the built in Dictionary and List classes. As such, I expect you to be able to invert all of the LA Times documents and hold all of the postings and a mapping from docID to docno in RAM.

What to Turn In

1. Print all source code.
2. Write a short report (approximately 2 pages) detailing:
    a. Methods. Please describe your implementation and data structure choices. Explain you have satisfied tasks 1-5.
    b. Results. Include your plots and the vocabulary size. Also consider reporting interesting bits such as: How long does it take to run? How much memory does it consume in RAM? On disk?
    c. A discussion of your results. Does the data make sense? How good are your data structure choices, etc. What worked, what didn't work, what you'd change in the future, etc.
    d. Any implementation issues encountered as part of completing this project. Was the implementation easy, hard, or just right?

Use of Existing Code Etc.

You may program in your language of choice. You must work alone on this assignment except that you may openly discuss design choices and the homework with anyone. You may use any code that you are legally allowed to use for non-commercial purposes excluding code written by classmates, but you **must carefully acknowledge all code that you have not written yourself. You may not use code written by any other 541/720 student.** Please see the course outline for general academic honesty guidelines.