Homework 6

In this assignment, you will make a basic search engine for Project Gutenberg.

This assignment is due on October 14 at 9AM.

Your program will have three phases.

PHASE 1

In the first phase, you will load your search engine with the catalog of books through which it will search. You will do this by reading a file that we provide: "`catalog.txt`". (As we explain below, we provide two such files -- one containing a list of the top 10 book on Project Gutenberg, and the second containing links to some simple files for testing purposes.)

`catalog.txt` will include on separate lines, input of the following form: Book title,URL

For example, `catalog.txt` could contain the following:

```
Pride and Prejudice,http://www.gutenberg.org/cache/epub/
1342/pg1342.txt
The Yellow Wallpaper,http://www.gutenberg.org/cache/epub/
1952/pg1952.txt
```

Your program should create a dictionary `Books` with the book titles as a key. For each key, you will assign a value which is itself a 2-item list containing (a) a unique sequential number that you create and assign, starting with 0 and (b) the URL of the book.

For the above example, the contents of the dictionary `Books` would thus be

```
{ 'Pride and Prejudice' : [0, 'http://www.gutenberg.org/
cache/epub/1342/pg1342.txt'], 'The Yellow Wallpaper' : [1,
'http://www.gutenberg.org/cache/epub/1952/pg1952.txt'] }
```

At the same time, you will also make a list `Titles` containing the titles of the books in book number order, e.g.,

```
['Pride and Prejudice', 'The Yellow Wallpaper']
```

We will give you two `catalog.txt` files -- one containing the top 10 titles on Project Gutenberg and the other one containing links to several short text files for debugging purposes.

PHASE 2

In the second phase, you will populate your search engine with the contents of the books in the catalog. You will use the the Python library called `urllib` to open the URL of the books in the catalog. (We will talk through the use of `urllib` in lab this week.) You will read the text file found at each URL (converting upper-case letters to lower-case letters and discarding punctuation, as in HW 5) and count the words. However, unlike in HW5, we will count the words using a dictionary data structure rather than a binary search tree.

To count the words, create a dictionary `Words` with words as an index. The value associated with each word will itself be a list (for clarity, let's call this list `WordCounts`) counting the frequency of that word in each book in the catalog, using the book numbers as an index to `WordCounts`. You will read in the contents of each of the books linked to in the catalog and for every word that you encounter, you will increment the corresponding entry in `WordCounts` (using the book number as an index into the list `WordCounts`).

For example, if "`declared`" occurs 15 times in Pride and Prejudice but only once in The Yellow Wallpaper, while "`daughters`" occurs 50 times in Pride and Prejudice but does not occur in The Yellow Wallpaper, then:

The corresponding entries in the dictionary would be

```
Words['declared'] = [15, 1]
Words['daughters'] = [50, 0]
```

PHASE 3

In the third phase of your program, your program will perform a search engine function based on input from the user. It will run a loop asking for search words until it receives the word "`<terminate>`". It should report books sorted in the order of how many times a word appears in the book. (If a word appears an equal number of times in more than one book, it can return those books in any order.)

In addition, please implement the following debug commands:

"`<catalog>`" - prints out the contents of the dictionary `Books` in an easy to

read format
“<titles>” - prints out the contents of the list `Titles` in an easy to read format

Here is a sample session (notice how the program figures out whether it is appropriate to use the word "times" or "time"):

```
Search term?  declared
1.  The word declared appears 15 times in Pride and
Prejudice (link:  http://www.gutenberg.org/cache/epub/1342/
pg1342.txt)
2.  The word declared appears 1 time in The Yellow
Wallpaper (link:  http://www.gutenberg.org/cache/epub/1952/
pg1952.txt)

Search term?  daughters
1.  The word daughters appears 50 times in Pride and
Prejudice (link:  http://www.gutenberg.org/cache/epub/1342/
pg1342.txt)

Search term?  groucho
The word groucho does not appear in any books in the
library

Search term?  <catalog>
'Pride and Prejudice' : [0, 'http://www.gutenberg.org/
cache/epub/1342/pg1342.txt']
'The Yellow Wallpaper' : [1, 'http://www.gutenberg.org/
cache/epub/1952/pg1952.txt']

Search term?  <titles>
Pride and Prejudice
The Yellow Wallpaper

Search term?  <terminate>
```

TESTING AND SUBMISSION

Please extensively and carefully test your assignment.

In your write up, please discuss these issues: What are the boundary cases? What happens if catalog.txt comes in an unexpected format? What happens if a link is broken or the contents come in an unexpected format? etc.

When your program is complete, put it in a script named `hw6.<lastname>.py`. Write a document (acceptable formats include pdf and text files) explaining how you tested your program (what test cases and strategies did you use) and the answer to the above question as `hw6-test.<lastname>.txt` or `hw6-test.<lastname>.pdf`. Upload these files using the file upload tool available at `https://www.ischool.berkeley.edu/uploader/?s=i206` Login with your ISchool userid and password and follow the directions.

EXTRA CREDIT

For extra credit, write an additional program to automatically generate the top 10 books `catalog.txt` from this web page (top 100 EBooks yesterday): `<a href="http://www.gutenberg.org/browse/scores/top">http://www.gutenberg.org/browse/scores/top</a>`

hw6