Extra Credit 4

Due: Monday, March 4, 2019 at 11:55 p.m. **Points:** 100

Please turn in your answers for the homework assignment on Canvas under Extra Credit 4 there.

A *web crawler* is a program that starts at a web site. It then follows the links on that page recursively. If you think of the links as branches of a tree, and the original web site as the root, the crawler follows the branches to the next "root" (i.e., web site) and then repeats this process some number of times (the *depth*).

Your job is to write a simple web crawler. We will do this in 2 steps — that makes it less overwhelming. Also, we will focus only on one type of link, that which begins with "http:".

1. Write a program that reads in a URL and the depth as an integer. The program should print the integer and the links on the page.

   All links will look like this: `href="`*`link's url`*`"` so you have to extract the 'link's url" part. You can do this in two ways. First, you can use the string method `find()` to look for "`href="`", and then for the closing "`"`", and extract the string between the two. Or, you can use the regular expression package. To do this, import "re" and then use the line:

   ```
               re.findall('href="(http://.*?)"', webpagetextstring)
   ```

   where *`webpagetextstring`* is the contents of the web page you are checking. This returns a list of links, for example

   ```
   [ 'http://nob.cs.ucdavis.edu/mhi289i/sub1/index.html',
   'http://nob.cs.ucdavis.edu/secure-exer/index.html',
   'http://nob.cs.ucdavis.edu/mhi289i/sub2/index.html',
   'http://nob.cs.ucdavis.edu/mhi289i/next.html' ]
   ```

   Print the links in the following form (your listing may differ):

   ```
   http://nob.cs.ucdavis.edu/mhi289i/index.html contains:
        http://nob.cs.ucdavis.edu/mhi289i/sub1/index.html
        http://nob.cs.ucdavis.edu/secure-exer/index.html
        http://nob.cs.ucdavis.edu/mhi289i/sub2/index.html
        http://nob.cs.ucdavis.edu/mhi289i/next.html
   ```

   or, if there are no links, print:

   ```
   http://nob.cs.ucdavis.edu/secure-exer/index.html contains no links
   ```

   Call this program "crawler1.py" when you submit it.

2. Now for the recursion. Recursively visit each link and print the links on the page, as above. Do this to the indicated depth. Be sure you don't repeat pages; once you print the links, do not print them again should you revisit the page.

   A good web site to test your program on is http://nob.cs.ucdavis.edu/mhi289i/index.edu.

   Print the links in the following form (your listing may differ):

   ```
   http://nob.cs.ucdavis.edu/mhi289i/index.html contains:
        http://nob.cs.ucdavis.edu/mhi289i/sub1/index.html
        http://nob.cs.ucdavis.edu/secure-exer/index.html
        http://nob.cs.ucdavis.edu/mhi289i/sub2/index.html
        http://nob.cs.ucdavis.edu/mhi289i/next.html
   http://nob.cs.ucdavis.edu/mhi289i/sub1/index.html contains:
        http://nob.cs.ucdavis.edu/mhi289i/sub2/index.html
        http://nob.cs.ucdavis.edu/mhi289i/index.html
   ```

Hint: Use a dictionary for this. The key would be the URL of the current web page and the value would be the list of links. Then, when you visit a web page, check that its URL is not in the dictionary. If it is, you already visited it and all its links, so simply return.

Call this program “crawler2.py” when you submit it.