webcrawler.py

This script makes use of the python's beautiful soup module for crawling the websites. The file crawls the specified link and recursively crawls the URLs found on the main file and keeps on crawling all the links up to the specified depth (i.e 3).

After crawling each link, the beautiful soup helps to fetch the html data for each of the web page, and from there we can extract the <a href='url' …> tags and add all the distinct URLs from the current page to the list of URLs to be crawl further.

The script only crawls valid links, and the definition of valid is stated below; i.e it should start with /wiki/Main_Page or /wiki/

```
#below function checks the URL provided for the various filters
def _validLink(url):
   if (not url.startswith('/wiki/Main_Page')
      and ":" not in url
      and url.startswith('/wiki/')):
      return True
   else:
      return False
```

While crawling the page/URL, we can feed in the word phrase along with the code, and if we do so, then the crawler will only fetch the URLs, whose HTML data has that phrase inside it or not.

This is done until it iteratively (using BFS approach) finds all the required URLs, their count, and at which depth (at max 3) were they found. All this statistical data is written onto an external file for analysis purposes.

NOTE: For further detailed explanation about the code please check the python file, and look at the comments.