Alex Richards
CS 146
Section #8
Homework #11 (BONUS ASSIGNMENT)

1 Design

The crawler, itself, is really simple; it takes in (as a minimum) a seed URL, parses that webpage for more links, appends them to the crawling queue and dequeues from that queue (parsing each link for more links) until either the queue is empty or a defined limit of page visits has been reached. Once that's been completed, all the visited links are written to an output file. Additionally, the user may define a limit and/or output file other than the default.

2 Sample Input/Output

When running WebCrawler with the following input:

```
python3 WebCrawler.py -u "https://en.wikipedia.org/wiki/Web_crawler" -l 20
```

the default output file ("visited.txt") has the following lines written to it:

```
https://en.wikipedia.org/wiki/web_crawler/wiki/Talk:Web_crawlerProposal_to_merge_Knowbot...
https://en.wikipedia.org/wiki/web_crawler/wiki/Web_search_engine
https://en.wikipedia.org/wiki/web_crawler/wiki/WebCrawler
https://en.wikipedia.org/wiki/web_crawler/wiki/Wikipedia:Merging
https://en.wikipedia.org/wiki/web_crawler/wiki/Web_indexing
https://en.wikipedia.org/wiki/web_crawler/wiki/Index_(search_engine)
https://en.wikipedia.org/wiki/web_crawler/wiki/File:WebCrawlerArchitecture.svg
https://en.wikipedia.org/wiki/web_crawler/wiki/Software_agent
https://en.wikipedia.org/wiki/web_crawler/wiki/Web_scraping
https://en.wikipedia.org/wiki/web_crawler/wiki/Arac_(video_game)
https://en.wikipedia.org/wiki/web_crawler/wiki/Web_content
https://en.wikipedia.org/wiki/web_crawler/wiki/World_Wide_Web
https://en.wikipedia.org/wiki/web_crawler/wiki/Website
https://en.wikipedia.org/wiki/web_crawler/wiki/Offline_reader
https://en.wikipedia.org/wiki/web_crawler/wiki/Hyperlink
https://en.wikipedia.org/wiki/web_crawler/wiki/Robots.txt
https://en.wikipedia.org/wiki/web_crawler/wiki/User_(computing)
https://en.wikipedia.org/wiki/web_crawler/wiki/HTML
https://en.wikipedia.org/wiki/web_crawler/wiki/Internet_bot
https://en.wikipedia.org/wiki/web_crawler/wiki/Knowbot
```

3 Challenges & Potential Improvements

When testing WebCrawler, BeautifulSoup was able to successfully retrieve the anchor tags from most of the test seed URLs. However, I found that anchor tags on Wikipedia were not so easily captured. After a bit of searching, I found a stackoverflow post in which a user suggested using regular expressions to match the characteristic form of Wikipedia's internal links. Unfortunately, I'm unable to find that post again to cite it, but in the process of searching for it I found a simpler solution (cited belwo)...just adding 'href=True' as an additional parameter to BeautifulSoup's 'findAll' function.

4 Resources Used

David's solution from:
https://stackoverflow.com/questions/499345/regular-expression-to-extract-url-from-an-html-link