CS104 PROJECT REPORT :
Python Web Crawler

Vishal Bysani
22B1061

14 June 2023

Introduction

I am Vishal Bysani. I have made a web crawler using Python for the CS104 project.
A web crawler, crawler or web spider, is a computer program that's used to search and automatically index website content and other information over the internet. These programs, or bots, are most commonly used to create entries for a search engine index.

Libraries

The libraries and dependencies used in this project are:

- **argparse** : This library helps in parsing the command line arguments[1]
- **BeautifulSoup** : Beautiful Soup is a Python library for getting data out of HTML, XML, and other markup languages.
- **requests** : Python requests is a library for making HTTP requests. It provides an easy-to-use interface that makes working with HTTP very simple, which means it simplifies the process of sending and receiving data from websites by providing a uniform interface for both GET and POST methods.
- **Numpy and Matplotlib** : I used matplotlib to generate the plots of the files distribution at various levels recursed by the web crawler[2]
- **warnings** : I used this library to not display unnecessary warnings.

Usage

The web crawler is run using the command:
`python3 Web_Crawler.py -u <site-name> -t <threshold> -o <output-filename> -s <user-input>`

- -u: for the URL. This is a compulsory argument and must print an error if not provided
- -t: for the threshold of recursiveness. It must be greater than zero. If not provided, the web crawler will recurse till the end. In case of invalid (negative values), it must print an error
- -o: For an output file. If not provided then by default print on the command line.
- -s: If the user gives 'Y' argument, then the web crawler computes and prints the files sizes as well. It only prints the links if anything else is given as argument.

Code Structure

Function of base code

Using the base code from [3], we can scrape all urls from the given site and prints them all. However, it doesn't crawl into these urls again. Figure 1 was taken from [3]

```
http://example.webscraping.com///places/default/user/register?_next=/places/default/index
http://example.webscraping.com///places/default/user/register?_next=/places/default/index/places/default/user/regi
ster
http://example.webscraping.com///places/default/user/register?_next=/places/default/index/places/default/user/regi
ster/places/default/user/register
http://example.webscraping.com///places/default/user/register?_next=/places/default/index/places/default/user/regi
ster/places/default/user/register/places/default/user/register
http://example.webscraping.com///places/default/user/register?_next=/places/default/index/places/default/user/regi
ster/places/default/user/register/places/default/user/register/places/default/user/register
http://example.webscraping.com///places/default/user/register?_next=/places/default/index/places/default/user/regi
ster/places/default/user/register/places/default/user/register/places/default/user/register/places/default/user/re
gister
http://example.webscraping.com///places/default/user/register?_next=/places/default/index/places/default/user/regi
ster/places/default/user/register/places/default/user/register/places/default/user/register/places/default/user/re
gister/places/default/user/register
http://example.webscraping.com///places/default/user/register?_next=/places/default/index/places/default/user/regi
ster/places/default/user/register/places/default/user/register/places/default/user/register/places/default/user/re
gister/places/default/user/register/places/default/user/register
http://example.webscraping.com///places/default/user/register?_next=/places/default/index/places/default/user/regi
ster/places/default/user/register/places/default/user/register/places/default/user/register/places/default/user/re
gister/places/default/user/register/places/default/user/register/places/default/user/register
```

Figure 1: Output of a basic Web crawler

Implementation of modified code

I have modified the code from [3] in the following manner to meet the requirements of the problem statement:

- The code first extracts all the links in the *href* and *src* tags using BeautifulSoup and then modifies them to make valid urls.
- The code recursively crawls through the url provided by the use upto a specified threshold or until the end if not specified.
- The code also segregates all the files at a particular recursion level into various categories like HTML, CSS, JS, JPG, PNG and Others.

Customization

- The code keeps printing the number of files and the url crawled by it to indicate the progress of the program.2

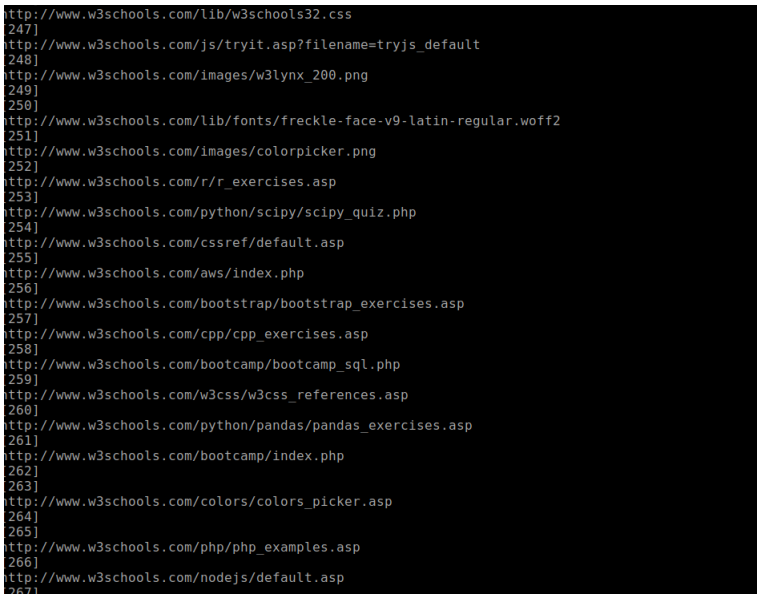

Figure 2: Indication of running of the program

- The modified code also identifies if a link is of internal or external domain, and selectively crawls only the internal ones. It separately counts the number of internal and external domain files at each level and prints them in the end.
- It also plots bar graphs showing the various categories of files in internal and external domains at each recursion level and saves this plot in Files_Distribution.png.

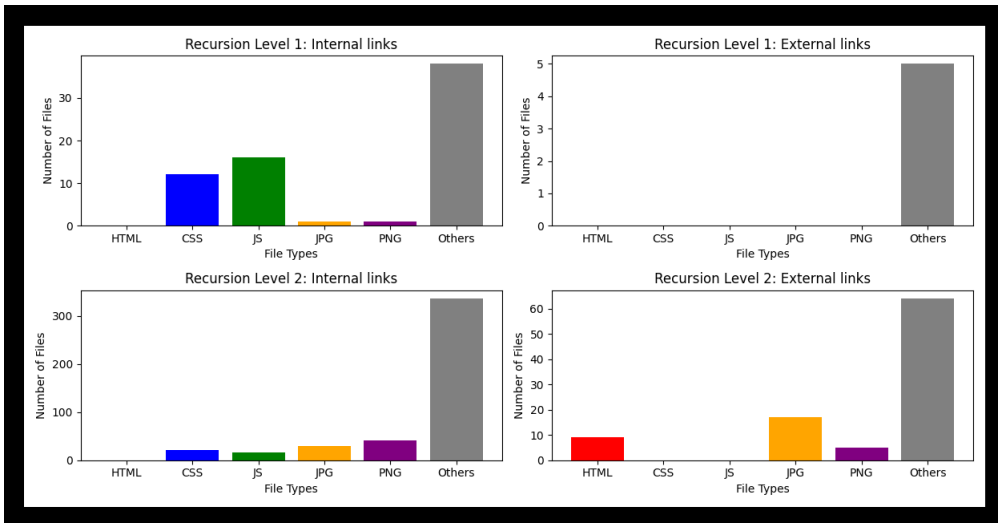

Figure 3: Bar Graph obtained from modified code

- The code also finds file sizes based upon the argument given by the user to the `s` tag.

Figure 4: Output redirected to the output file

As it can be seen from figures 3 and 4, the modified web crawler categorizes the files at various recursion levels. Further the web-crawler also finds the file sizes of all the files and prints it along with the links.

Description of the Submitted Folder

I have submitted all the files and sub-folder under the folder named `22B1061_project`. It includes the following:

- Web_Crawler.py: It contains the python code.
- report.pdf: It contains the report in the pdf format.
- report.tex: It contains the latex code for the pdf.
- report.bib: It contains all the references I used for this project
- BarGraphs.png: It shows the bar graphs it prints for various recursion levels.
- OutputFilePrintFormat.png: It shows the format in which it prints into the output file given by the user

- Screenshot.jpeg: It shows the output of code which I took as a base for my web crawler.

References

[1] URL: https://docs.python.org/3/library/argparse.html.

[2] URL: https://matplotlib.org/stable/gallery/subplots_axes_and_figures/subplots_demo.html.

[3] URL: https://www.geeksforgeeks.org/python-program-to-recursively-scrape-all-the-urls-of-the-website/.