Project Title: Web Crawler

Web crawler is a computer program that is 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. Here you will create the most basic form of a web crawler, which is finding all the web links referred to by an HTML page recursively.

Basic Requirement:

Given the website link (given as command line argument), recursively find out all the unique referenced web links either with "href" or with "src" attribute used with different tags. Recursion means, from the first input link given, parsing its HTML will give you say 10 links. You then need to visit these 10 links individually and then from each of these get more links. Then, write all the links found in a file as per specification given below. NOTE: RECURSION SHOULD APPLY ONLY TO INTERNAL LINKS I.E. LINK SHOULD BELONG TO SAME DOMAIN. But when listing the files, you should list external links as well, even though you will not crawl them.

- **Example script usage:** "python3 web-crawler.py -u http://www.iitb.ac.in -t 5 -o <output-file-name>
    - -u: for the URL, If not given then must print an error on the command line.
    - -t: for the threshold of recursiveness, must be greater than 0, give an error for an invalid threshold
    - -o: For an output file, If not provided then by default print on the command line.

File output:

At a given level of recursion i.e depth, specify how many files were found in total. Then segregate them into counts of different types along with listing the individual links. Note I have shown only recursion level 2, but you have to print below for all depths upto as specified by the -t option.

Example:

At recursion level 2
Total files found: 27
Html: 20
http://www.iitb.ac.in/contact.html
http://www.iitb.ac.in/about.html
…..
Css: 5
http://www.iitb.ac.in/style/st.css
…
Jpg: 1

http://www.iitb.ac.in/images/iitb.jpg
Js: 1
http://www.iitb.ac.in/javascript/anim.js

Customization:

Here are some sample customization, but really you should come up with your own. Marks will be given per YOUR creativity.

1. You can extend the above by adding more detail at each recursion level like total size as well as individual size of the files.
2. You can also group links at a given recursion level as internal or external, further based on specific domain.

Process:

You are welcome to browse for code on the web and use it. In fact you will easily find python code that does web crawling, including with recursion. Your job is to understand the code and then customize it as per requirement (this is what we do in real life as well, rarely do we write stuff from scratch :-)

Marks Distribution (20 marks): These marks will be awarded via a viva, where you demonstrate the project and TAs will ask questions.

- Functionality and correctness of the code (recursion-2, count-4 and segregation per type-4): 10 marks
- Customization: 4 marks
- Viva: 4 marks
- Code quality, organization, and comments, latex based report: 2 marks

Note: All submitted code in a given project will be checked for plagiarism and if caught, your case will be forwarded to DDAC.

Upload: Please upload all relevant files, including report as a zip file on BodhiTree before the deadline. NO CHANGES TO THE FILES WILL BE PERMITTED POST THIS, INCLUDING DURING EVALUATION. During evaluation, TAs will download the zip from Bodhitree, setit up on their machine and then interact with you for evaluation. They will use their own input/testcases during this time.

Explanation for recursion: Output here is only specifying the links being crawled, you have to work more on top to do segregation of link types and count.

- **Definition (i)th Iteration:** Finding out all the links referred by the links found in (i-1)th iteration with 1st iteration being finding out all links referred by the URL passed in as command line argument.
- **Scenario:**
    - Assume we have a link http://www.qwe.com/index.html and we want to run our crawler on it.
    - Also assume:
        - http://www.qwe.com/index.html has references to the below links in its HTML page either in "src" or "href" attribute:
            - http://www.qwe.com/static/js/myjs.js , http://www.qwe.com/help.html , http://www.qwe.com/static/mycss.css, http://www.qwe.com/login
        - Assume http://www.qwe.com/help.html has referred to
            - http://www.qwe.com/contact.html and http://www.qwe.com/license.html
        - Assume http://www.qwe.com/login has referred to
            - http://www.qwe.com/forgot_password.php
        - Assume http://www.qwe.com/forgot_password.php has referred to
            - http://mail.qwe.com/reset_mail.php
        - Assume http://mail.qwe.com/reset_mail.php, http://www.qwe.com/contact.html, http://www.qwe.com/license.html http://www.qwe.com/static/js/myjs.js, and http://www.qwe.com/static/mycss.css does not have any other web links referred in it.
- **No threshold of recursion specified, i.e. -t option is not specified**
    - Then the web-crawler.py should output all the links recursively referred, that is
        - [http://www.qwe.com/static/js/myjs.js , http://www.qwe.com/help.html , http://www.qwe.com/static/mycss.css, http://www.qwe.com/login, http://www.qwe.com/contact.html, http://www.qwe.com/license.html, http://www.qwe.com/forgot_password.php, http://mail.qwe.com/reset_mail.php] - the format of printing doesn't matter.
- **Threshold with -t 2**
    - iteration=1: 1st Iteration of crawling: It will find all the links referred to in the index.html file, which is:
        - http://www.qwe.com/static/js/myjs.js , http://www.qwe.com/help.html , http://www.qwe.com/static/mycss.css, http://www.qwe.com/login
    - iteration=2: 2n iteration of crawling will crawl all the links referred by the above links we found that is:
        - No links referred by http://www.qwe.com/static/js/myjs.js and http://www.qwe.com/static/mycss.css
        - http://www.qwe.com/contact.html and http://www.qwe.com/license.html referred by http://www.qwe.com/help.html

        - http://www.qwe.com/forgot_password.php referred by http://www.qwe.com/login
    - Thus the overall output across two levels shall be:
        - [http://www.qwe.com/static/js/myjs.js , http://www.qwe.com/help.html , http://www.qwe.com/static/mycss.css, http://www.qwe.com/login, http://www.qwe.com/contact.html, http://www.qwe.com/license.html, http://www.qwe.com/forgot_password.php]
        - **NOTICE** that we don't print "http://mail.qwe.com/reset_mail.php" in the output as it will come in iteration 3 and the iteration threshold is 2.