Assignment 1

CS 532: Introduction to Web Science

Spring 2018

Hrishikesh Gadkari

Finished on January 28, 2018

1

Question

```
1.  Demonstrate that you know how to use "curl" well enough to
correctly POST data to a form.  Show that the HTML response that
is returned is "correct".  That is, the server should take the
arguments you POSTed and build a response accordingly.  Save the
HTML response to a file and then view that file in a browser and
take a screen shot.
```

Answer

To Post data to a form using curl I used the following command:

```
curl -i -d "fname=hrishi&lname=gadkari" -X POST http://www.cs.
    odu.edu/~anwala/files/temp/namesEcho.php > output.html
```

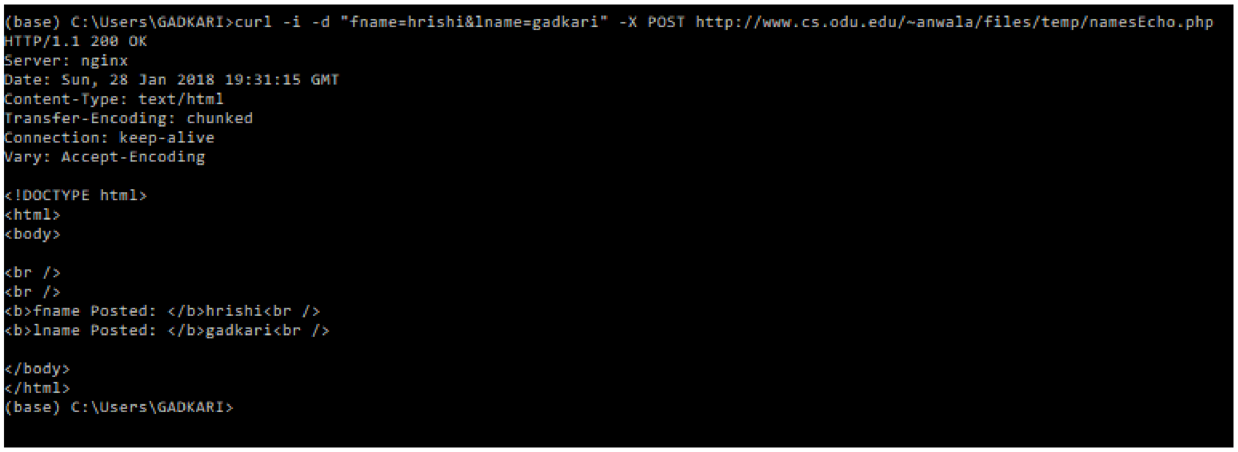
Listing 1: Curl command to post data

To approach this problem I referred the `https://gist.github.com/subfuzion/08c5d85437d5d4f00e58` url.

The use of each of the command options mentioned in Listing 1 is as follows:

- -i: To include HTTP headers in the response
- -d: Send specified data in POST request which is followed by the data(parameters) to be sent to the form
- -X: The request method to be used. In this case we have used POST method followed by the php page where data is to be posted

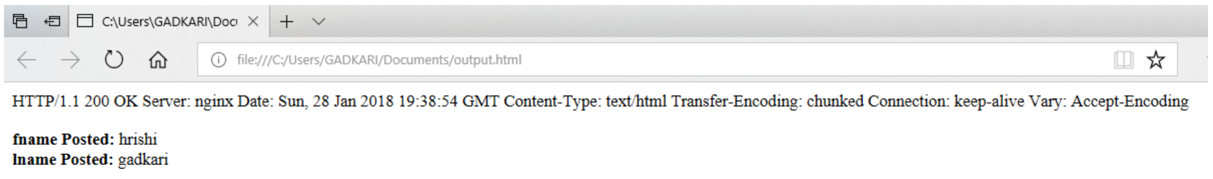
The output in the terminal looks as displayed below:



```
(base) C:\Users\GADKARI>curl -i -d "fname=hrishi&lname=gadkari" -X POST http://www.cs.odu.edu/~anwala/files/temp/namesEcho.php
HTTP/1.1 200 OK
Server: nginx
Date: Sun, 28 Jan 2018 19:31:15 GMT
Content-Type: text/html
Transfer-Encoding: chunked
Connection: keep-alive
Vary: Accept-Encoding

<!DOCTYPE html>
<html>
<body>

<br />
<br />
<b>fname Posted: </b>hrishi<br />
<b>lname Posted: </b>gadkari<br />

</body>
</html>
(base) C:\Users\GADKARI>
```

Figure 1: Response rendered in terminal

The image below displays the view of the output in the browser after saving it in output.html file.



HTTP/1.1 200 OK Server: nginx Date: Sun, 28 Jan 2018 19:38:54 GMT Content-Type: text/html Transfer-Encoding: chunked Connection: keep-alive Vary: Accept-Encoding

fname Posted: hrishi
lname Posted: gadkari

Figure 2: Response rendered in browser

2

Question

```
2.  Write a Python program that:
  1. takes as a command line argument a web page
  2. extracts all the links from the page
  3. lists all the links that result in PDF files, and prints out
     the bytes for each of the links.  (note: be sure to follow
     all the redirects until the link terminates with a "200 OK".)
  4. show that the program works on 3 different URIs, one of which
     needs to be:
     http://www.cs.odu.edu/~mln/teaching/cs532-s17/test/pdfs.html
```

Answer

To solve the above problem I wrote the following script using Python3:

```
#!/usr/bin/env python3

from bs4 import BeautifulSoup
import requests
import sys
import urllib
from urllib.parse import urlparse, urljoin

#Part 1 : Read in command line arguments
if len (sys.argv) == 2 :
        weblink = sys.argv[1]

else :
        print ("Usage:python crawl.py url")
        sys.exit (1)

# Request data using get
try :
        urllib.request.urlopen(weblink)
        responsedurl = requests.get(weblink)

#get http response
#if multiple-redirection exists get new URI
        if responsedurl.history :
                print ("Request was redirected to new link")
                print ("New Link: " + responseurl.url)
                response = requests.get(responseurl.url)
                data = response.text

        else :
                response = requests.get(weblink)
                data = response.text

except :
        print ("Enter a valid url")
        sys.exit (1)

#Extract all links from the webpage using BeautifulSoup
soup = BeautifulSoup(data, 'lxml')
print ("Extracting all links from webpage:\n")
for link in soup.findAll('a', href=True):
        webb = link.get('href')
        print (webb)

#Extract all pdf links from the webpage if present
print ("\nExtracting pdf links from webpage if present:\n")
for pdflinks in soup.findAll('a', href=True):
        web = pdflinks.get('href')

#check if any relative links present
        if bool(urlparse(web).netloc) == False :
                web = urljoin(weblink, web)


        try:
                urllib.request.urlopen(web)
                req = requests.get(web)
                resp = requests.head(web)

#return pdf links with status_code 200
                if req.status_code == 200 and resp.headers.get('content-type') is not None :

                        if "application/pdf" in resp.headers['content-type'] :
                                print ("First Link: " + web)
                                print ("Final Link: " + web)
                                print("Bytes: " + resp.headers['content-length'] + '\n')

```

```
#return pdf links for the final URI if any redirects
                if req.history:
                        final = req.url
                        finalresp = requests.head(final)
                        if finalresp.headers.get('content-type')
                            is not None :
                                if "application/pdf" in
                                    finalresp.headers['content-
                                    type'] :
                                        print ("Request was
                                            redirected")
                                        print ("First Link: " +
                                            web)
                                        print ("Final Link: " +
                                            final)
                                        print("Bytes: " +
                                            finalresp.headers['
                                            content-length'] + '\
                                            n')

        except:
                pass
```

Listing 2: Python script that searches for links that end in pdf files

The following libraries were used in my script:

- Beautiful Soup: To parse the HTML and XML documents and extract the data(third party library)
- sys: This module provides access to some variables used or maintained by the interpreter and to functions that interact strongly with the interpreter
- requests: To make HTTP requests
- urllib: This package collects several modules for working with URLs
- urlparse: Focuses on splitting a URL string into its components, or on combining URL components into a URL string
- urljoin: To append a url to the base url

My solution took an iterative approach doing one URI at a time and waiting for each response until moving onto the next URI found.

For the part 1 of the problem the program first checks whether the number of arguments are correct using the command line arguments and if correct, it will pass the first argument after the script name and considers only a properly formatted URI and performs a HTTP get request using the requests library. To check whether the URI is redirected to a new address, response.history [4] was used, after which get request is made to the new URI.

For the part 2 of the problem, that is to extract all links (absolute as well as relative) from the website, Beautiful Soup [2] as a third party library was used to find all the html a elements that contained href tags using the specified lxml parser.

For the part 3 of the problem, the program iterates through each of the URIs found on the page and request again each of those URIs to determine if the URI would point to pdf file. Also if the program encounters a relative link it appends the base url with the relative url. The flow behind this is that it first checks whether the link is absolute using urlparse(web).netloc [1]. If False it changes the link into absolute link using urljoin() [3]. For this I referred stackoverflow website [5]. After this using head and get requests, the content-type, content-length and status code are obtained for each of the URIs and checks whether it returns status-code [1] as 200 or if the URI has redirects [4] and also if content-type [1] is PDF, thus giving the PDF link for the Final URI.

The script is ran with the command as shown below:

```
python crawl.py URI
```

The URIs I used for this problem were:

- `http://www.cs.odu.edu/~mln/teaching/cs532-s17/test/pdfs.html`
- `http://www.cs.odu.edu/~mweigle/`
- `http://www.cs.odu.edu/~yaohang/`(the output displays a relative link which is converted to absolute and eventually giving a pdf link)

The output for each of the URIs is displayed below:

```
Extracting all links from webpage:

http://twitter.com/webscidl
```

```
http://www.dlib.org/dlib/november15/vandesompel/11vandesompel.html
http://arxiv.org/abs/1508.02315
http://arxiv.org/abs/1508.02315
http://www.cs.odu.edu/~mln/pubs/ht-2015/hypertext-2015-temporal-violations.pdf
http://www.cs.odu.edu/~mln/pubs/tpdl-2015/tpdl-2015-annotations.pdf
http://arxiv.org/pdf/1512.06195
http://www.cs.odu.edu/~mln/pubs/tpdl-2015/tpdl-2015-off-topic.pdf
http://www.cs.odu.edu/~mln/pubs/tpdl-2015/tpdl-2015-stories.pdf
http://www.cs.odu.edu/~mln/pubs/tpdl-2015/tpdl-2015-profiling.pdf
http://dx.doi.org/10.1007/s00799-015-0150-6
http://www.cs.odu.edu/~mln/pubs/jcdl-2014/jcdl-2014-brunelle-damage.pdf
http://arxiv.org/abs/1506.06279
http://dx.doi.org/10.1007/s00799-015-0155-1
http://bit.ly/1ZDatNK
http://www.cs.odu.edu/~mln/pubs/jcdl-2015/jcdl-2015-mink.pdf
http://www.cs.odu.edu/~mln/pubs/jcdl-2015/jcdl-2015-arabic-sites.pdf
http://www.cs.odu.edu/~mln/pubs/jcdl-2015/jcdl-2015-dictionary.pdf
http://bit.ly/jcdl-pdf
http://dx.doi.org/10.1007/s00799-015-0140-8

Extracting pdf links from webpage if present:

First Link: http://www.cs.odu.edu/~mln/pubs/ht-2015/hypertext-2015-temporal-violations.pdf
Final Link: http://www.cs.odu.edu/~mln/pubs/ht-2015/hypertext-2015-temporal-violations.pdf
Bytes: 2184076

First Link: http://www.cs.odu.edu/~mln/pubs/tpdl-2015/tpdl-2015-annotations.pdf
Final Link: http://www.cs.odu.edu/~mln/pubs/tpdl-2015/tpdl-2015-annotations.pdf
Bytes: 622981

Request was redirected
First Link: http://arxiv.org/pdf/1512.06195
Final Link: https://arxiv.org/pdf/1512.06195.pdf
```

```
Bytes: 1748961

First Link: http://www.cs.odu.edu/~mln/pubs/tpdl-2015/tpdl-2015-off-topic.pdf
Final Link: http://www.cs.odu.edu/~mln/pubs/tpdl-2015/tpdl-2015-off-topic.pdf
Bytes: 4308768

First Link: http://www.cs.odu.edu/~mln/pubs/tpdl-2015/tpdl-2015-stories.pdf
Final Link: http://www.cs.odu.edu/~mln/pubs/tpdl-2015/tpdl-2015-stories.pdf
Bytes: 1274604

First Link: http://www.cs.odu.edu/~mln/pubs/tpdl-2015/tpdl-2015-profiling.pdf
Final Link: http://www.cs.odu.edu/~mln/pubs/tpdl-2015/tpdl-2015-profiling.pdf
Bytes: 639001

First Link: http://www.cs.odu.edu/~mln/pubs/jcdl-2014/jcdl-2014-brunelle-damage.pdf
Final Link: http://www.cs.odu.edu/~mln/pubs/jcdl-2014/jcdl-2014-brunelle-damage.pdf
Bytes: 2205546

Request was redirected
First Link: http://bit.ly/1ZDatNK
Final Link: http://www.cs.odu.edu/~mln/pubs/jcdl-2015/jcdl-2015-temporal-intention.pdf
Bytes: 720476

First Link: http://www.cs.odu.edu/~mln/pubs/jcdl-2015/jcdl-2015-mink.pdf
Final Link: http://www.cs.odu.edu/~mln/pubs/jcdl-2015/jcdl-2015-mink.pdf
Bytes: 1254605

First Link: http://www.cs.odu.edu/~mln/pubs/jcdl-2015/jcdl-2015-arabic-sites.pdf
Final Link: http://www.cs.odu.edu/~mln/pubs/jcdl-2015/jcdl-2015-arabic-sites.pdf
Bytes: 709420

```

```
First Link: http://www.cs.odu.edu/~mln/pubs/jcdl-2015/jcdl-2015-
    dictionary.pdf
Final Link: http://www.cs.odu.edu/~mln/pubs/jcdl-2015/jcdl-2015-
    dictionary.pdf
Bytes: 2350603
```

Listing 3: Output from `http://www.cs.odu.edu/~mln/teaching/cs532-s17/test/pdfs.html`

```
Extracting all links from webpage:

http://www.cs.odu.edu
http://www.odu.edu
http://www.cs.odu.edu/~mweigle/Main/Home?action=login
http://www.cs.odu.edu/~mweigle/Main/Home
http://www.cs.odu.edu/~mweigle/Main/Research
http://www.cs.odu.edu/~mweigle/Main/PubsByYear
http://www.cs.odu.edu/~mweigle/Main/Students
http://www.cs.odu.edu/~mweigle/files/CV.pdf
http://www.cs.odu.edu/~mweigle/Resources/WorkingWithMe
http://www.cs.odu.edu/~mweigle/Resources/ResearchMethods
http://www.cs.odu.edu/~mweigle/Resources/InfoVis
http://www.cs.odu.edu/~mweigle/Main/Teaching
http://www.cs.odu.edu/~mweigle/Main/Sched
http://www.cs.odu.edu/~mweigle/Main/Personal
http://www.cs.odu.edu/~mweigle/CS725-S18/Home
https://graduate.cs.odu.edu/
http://www.cs.odu.edu/~yaohang
https://securegrants.neh.gov/publicquery/main.aspx?f=1&gn=HAA
    -256368-17
https://www.neh.gov/divisions/odh/grant-news/announcing-new
    -2017-odh-grant-awards
http://ws-dl.cs.odu.edu
http://ws-dl.blogspot.com
http://ws-dl.blogspot.com/2018/01/2018-01-08-introducing-
    reconstructive.html
http://ws-dl.blogspot.com/2018/01/2018-01-07-review-of-ws-dls
    -2017.html
http://ws-dl.blogspot.com/2018/01/2018-01-06-two-wsdl-classes-
    offered-for.html
http://ws-dl.blogspot.com/2018/01/2018-01-02-link-to-web-
    archives-not.html
http://ws-dl.blogspot.com/2017/12/2017-12-31-digital-blackness-
    in-archive.html
http://www.cs.odu.edu/~mweigle/Research/InfoVis-Gallery
```

```
30 https://arxiv.org/abs/1712.03140
31 http://www.cs.odu.edu/~mweigle/Main/Home?action=bibentry&bibfile
       =Main.bibtex&bibref=aturban-arxiv17
32 https://arxiv.org/abs/1708.05790
33 http://www.cs.odu.edu/~mweigle/Main/Home?action=bibentry&bibfile
       =Main.bibtex&bibref=mccoy-arxiv17
34 http://dx.doi.org/10.1145/3041656
35 http://www.cs.odu.edu/~mweigle/papers/alkwai-tois17-preprint.pdf
36 http://www.cs.odu.edu/~mweigle/Main/Home?action=bibentry&bibfile
       =Main.bibtex&bibref=alkwai-tois17
37 http://www.cs.odu.edu/~mweigle/papers/alam-jcdl17.pdf
38 http://www.cs.odu.edu/~mweigle/Main/Home?action=bibentry&bibfile
       =Main.bibtex&bibref=alam-jcdl17
39 http://www.cs.odu.edu/~anwala/files/publications/NwalaJCDL_LMP.
       pdf
40 http://www.cs.odu.edu/~mweigle/Main/Home?action=bibentry&bibfile
       =Main.bibtex&bibref=nwala-jcdl17
41 http://www.cs.odu.edu/~mweigle/papers/alnoamany-websci17.pdf
42 http://www.cs.odu.edu/~mweigle/Main/Home?action=bibentry&bibfile
       =Main.bibtex&bibref=alnoamany-websci17
43 http://www.cs.odu.edu/~mweigle/papers/brunelle-jcdl17.pdf
44 http://www.cs.odu.edu/~mweigle/Main/Home?action=bibentry&bibfile
       =Main.bibtex&bibref=brunelle-jcdl17
45 http://dx.doi.org/10.1109/JCDL.2017.7991619
46 http://www.cs.odu.edu/~mkelly/papers/2017_jcdl_wail.pdf
47 http://www.cs.odu.edu/~mweigle/Main/Home?action=bibentry&bibfile
       =Main.bibtex&bibref=berlin-jcdl17
48 http://dx.doi.org/10.1109/JCDL.2017.7991601
49 http://www.cs.odu.edu/~mkelly/papers/2017_jcdl_countingMementos.
       pdf
50 http://www.cs.odu.edu/~mweigle/Main/Home?action=bibentry&bibfile
       =Main.bibtex&bibref=kelly-jcdl17
51 http://arxiv.org/abs/1705.06218
52 http://www.cs.odu.edu/~mweigle/Main/Home?action=bibentry&bibfile
       =Main.bibtex&bibref=alnoamany-arxiv17
53 http://dx.doi.org/10.1109/JCDL.2017.7991601
54 http://www.cs.odu.edu/~mkelly/papers/2017_jcdl_countingMementos.
       pdf
55 http://www.cs.odu.edu/~mweigle/Main/Home?action=bibentry&bibfile
       =Main.bibtex&bibref=kelly-jcdl17
56 http://www.cs.odu.edu/~mln/pubs/jcdl-2015/jcdl-2015-mink.pdf
57 http://www.cs.odu.edu/~mweigle/Main/Home?action=bibentry&bibfile
       =Main.bibtex&bibref=jordan-jcdl15
58 http://www.cs.odu.edu/~mln/pubs/jcdl-2015/jcdl-2015-arabic-sites
       .pdf
```

```
59 http://www.cs.odu.edu/~mweigle/Main/Home?action=bibentry&bibfile
     =Main.bibtex&bibref=alkwai-jcdl15
60 http://dx.doi.org/10.1109/MASS.2014.91
61 http://www.cs.odu.edu/~mweigle/papers/mohrehkesh-misenet14.pdf
62 http://www.cs.odu.edu/~mweigle/Main/Home?action=bibentry&bibfile
     =Main.bibtex&bibref=mohrehkesh-misenet14
63 http://dx.doi.org/10.1109/JCDL.2014.6970187
64 http://www.cs.odu.edu/~mln/pubs/jcdl-2014/jcdl-2014-brunelle-
     damage.pdf
65 http://www.cs.odu.edu/~mweigle/Main/Home?action=bibentry&bibfile
     =Main.bibtex&bibref=brunelle-jcdl14
66 http://dx.doi.org/10.1145/2509338.2509340
67 http://www.cs.odu.edu/~mweigle/papers/olariu-misenet13.pdf
68 http://www.cs.odu.edu/~mweigle/Main/Home?action=bibentry&bibfile
     =Main.bibtex&bibref=olariu-misenet13
69 http://dx.doi.org/10.1007/978-3-642-40501-3_35
70 http://www.cs.odu.edu/~mln/pubs/tpdl-2013/paper_149.pdf
71 http://arxiv.org/abs/1309.4016
72 http://www.cs.odu.edu/~mweigle/Main/Home?action=bibentry&bibfile
     =Main.bibtex&bibref=alnoamany-tpdl13
73 http://dx.doi.org/10.1109/SOLI.2009.5203967
74 http://www.cs.odu.edu/~mweigle/papers/yan-soli09.pdf
75 http://www.cs.odu.edu/~mweigle/Main/Home?action=bibentry&bibfile
     =Main.bibtex&bibref=yan-soli09
76 https://securegrants.neh.gov/publicquery/main.aspx?f=1&gn=HAA
     -256368-17
77 https://mellon.org/grants/grants-database/grants/old-dominion-
     university/11600663/
78 http://www.nsf.gov/awardsearch/showAward?AWD_ID=1526700
79 http://www.imls.gov/grants/awarded/lg-71-15-0077-15
80 http://www.cs.odu.edu/~mweigle/files/CV.pdf
81 http://www.cs.odu.edu/~acmw
82 http://www.ncwit.org/alliances/aa
83 http://www.cs.odu.edu/
84 http://www.odu.edu/
85 http://www.clemson.edu/ces/departments/computing/
86 http://www.clemson.edu
87 http://www.cs.unc.edu
88 http://www.unc.edu
89 http://www.ulm.edu/cba/computerscience/index.html
90 http://www.ulm.edu/honors
91 http://www.ulm.edu
92 http://www.cs.odu.edu/~mweigle/Main/Home?action=print
93 http://www.cs.odu.edu/~mweigle/Site/Search
94 http://www.cs.odu.edu/~mweigle/Main/Home?action=login
```

```

Extracting pdf links from webpage if present:

First Link: http://www.cs.odu.edu/~mweigle/files/CV.pdf
Final Link: http://www.cs.odu.edu/~mweigle/files/CV.pdf
Bytes: 101583

First Link: http://www.cs.odu.edu/~mweigle/papers/alkwai-tois17-
    preprint.pdf
Final Link: http://www.cs.odu.edu/~mweigle/papers/alkwai-tois17-
    preprint.pdf
Bytes: 1430568

First Link: http://www.cs.odu.edu/~mweigle/papers/alam-jcdl17.
    pdf
Final Link: http://www.cs.odu.edu/~mweigle/papers/alam-jcdl17.
    pdf
Bytes: 1600140

First Link: http://www.cs.odu.edu/~anwala/files/publications/
    NwalaJCDL_LMP.pdf
Final Link: http://www.cs.odu.edu/~anwala/files/publications/
    NwalaJCDL_LMP.pdf
Bytes: 17623699

First Link: http://www.cs.odu.edu/~mweigle/papers/alnoamany-
    websci17.pdf
Final Link: http://www.cs.odu.edu/~mweigle/papers/alnoamany-
    websci17.pdf
Bytes: 6962016

First Link: http://www.cs.odu.edu/~mweigle/papers/brunelle-
    jcdl17.pdf
Final Link: http://www.cs.odu.edu/~mweigle/papers/brunelle-
    jcdl17.pdf
Bytes: 1276346

First Link: http://www.cs.odu.edu/~mkelly/papers/2017_jcdl_wail.
    pdf
Final Link: http://www.cs.odu.edu/~mkelly/papers/2017_jcdl_wail.
    pdf
Bytes: 412476

First Link: http://www.cs.odu.edu/~mkelly/papers/2017
    _jcdl_countingMementos.pdf
```

```
Final Link: http://www.cs.odu.edu/~mkelly/papers/2017_jcdl_countingMementos.pdf
Bytes: 274265

First Link: http://www.cs.odu.edu/~mkelly/papers/2017_jcdl_countingMementos.pdf
Final Link: http://www.cs.odu.edu/~mkelly/papers/2017_jcdl_countingMementos.pdf
Bytes: 274265

First Link: http://www.cs.odu.edu/~mln/pubs/jcdl-2015/jcdl-2015-mink.pdf
Final Link: http://www.cs.odu.edu/~mln/pubs/jcdl-2015/jcdl-2015-mink.pdf
Bytes: 1254605

First Link: http://www.cs.odu.edu/~mln/pubs/jcdl-2015/jcdl-2015-arabic-sites.pdf
Final Link: http://www.cs.odu.edu/~mln/pubs/jcdl-2015/jcdl-2015-arabic-sites.pdf
Bytes: 709420

First Link: http://www.cs.odu.edu/~mweigle/papers/mohrehkesh-misenet14.pdf
Final Link: http://www.cs.odu.edu/~mweigle/papers/mohrehkesh-misenet14.pdf
Bytes: 1231147

First Link: http://www.cs.odu.edu/~mln/pubs/jcdl-2014/jcdl-2014-brunelle-damage.pdf
Final Link: http://www.cs.odu.edu/~mln/pubs/jcdl-2014/jcdl-2014-brunelle-damage.pdf
Bytes: 2205546

First Link: http://www.cs.odu.edu/~mweigle/papers/olariu-misenet13.pdf
Final Link: http://www.cs.odu.edu/~mweigle/papers/olariu-misenet13.pdf
Bytes: 405542

First Link: http://www.cs.odu.edu/~mln/pubs/tpdl-2013/paper_149.pdf
Final Link: http://www.cs.odu.edu/~mln/pubs/tpdl-2013/paper_149.pdf
Bytes: 813692
```

```

First Link: http://www.cs.odu.edu/~mweigle/papers/yan-soli09.pdf
Final Link: http://www.cs.odu.edu/~mweigle/papers/yan-soli09.pdf
Bytes: 278184

First Link: http://www.cs.odu.edu/~mweigle/files/CV.pdf
Final Link: http://www.cs.odu.edu/~mweigle/files/CV.pdf
Bytes: 101583
```

Listing 4: Output from `http://www.cs.odu.edu/~mweigle/`

```
Extracting all links from webpage:

http://www.cs.odu.edu/~yaohang
teaching.html
projects.html
Tools.html
education.html
pubs.html
people.html
contact.html
YaohangLiCV2010.pdf
Services.html
software.html
links.html
http://www.cs.odu.edu/
http://www.odu.edu/
mailto:yaohang@cs.odu.edu
http://www.cs.odu.edu/
http://www.odu.edu/
http://www.shodor.org
http://www.cs.appstate.edu/nc-hpc/
http://www.sura.org
postdoctoral_researcher_position.htm
phdstudent_positions.htm
http://www.orau.org
http://www.nsf.gov/awardsearch/showAward.do?AwardNumber=0845702
http://www.fsu.edu
http://www.scut.edu.cn
http://www.ornl.gov
http://fellowships.ncsa.uiuc.edu/summer/past.html
http://www.ncsa.uiuc.edu/
http://www.ornl.gov
http://www.ncat.edu/

Extracting pdf links from webpage if present:
```

```

First Link: http://www.cs.odu.edu/~yaohang/YaohangLiCV2010.pdf
Final Link: http://www.cs.odu.edu/~yaohang/YaohangLiCV2010.pdf
Bytes: 39166
```

Listing 5: Output from `http://www.cs.odu.edu/~yaohang/`

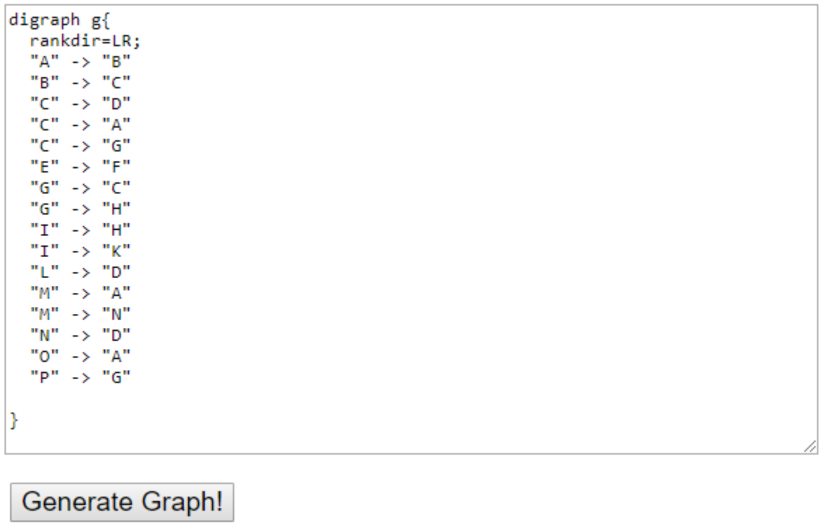
3

Question

```
3.  Consider the "bow-tie" graph in the Broder et al. paper (fig 9):
    http://www9.org/w9cdrom/160/160.html

    Now consider the following graph:

    A --> B
    B --> C
    C --> D
    C --> A
    C --> G
    E --> F
    G --> C
    G --> H
    I --> H
    I --> K
    L --> D
    M --> A
    M --> N
    N --> D
    O --> A
    P --> G

    For the above graph, give the values for:

    IN:
    SCC:
    OUT:
    Tendrils:
    Tubes:
    Disconnected:
```

Answer

A graph was generated for the above values using webgraphviz [6]. The following snippets show how it got generated:



```
digraph g{
  rankdir=LR;
  "A" -> "B"
  "B" -> "C"
  "C" -> "D"
  "C" -> "A"
  "C" -> "G"
  "E" -> "F"
  "G" -> "C"
  "G" -> "H"
  "I" -> "H"
  "I" -> "K"
  "L" -> "D"
  "M" -> "A"
  "M" -> "N"
  "N" -> "D"
  "O" -> "A"
  "P" -> "G"

}
```

Figure 3: Graph generation with WebGraphviz

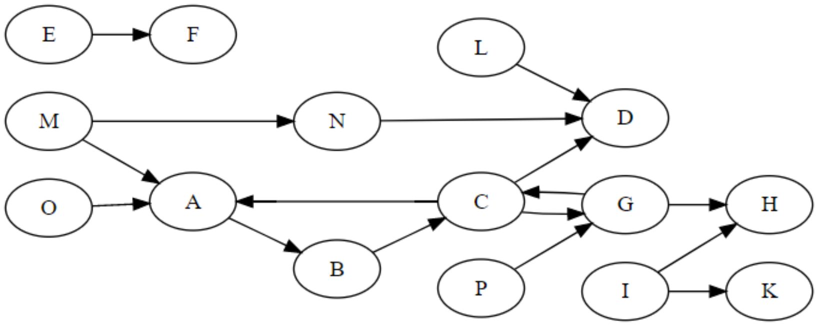


Figure 4: Graph generated with WebGraphviz

As per the figure 9 in Broder et al. paper and also from the reference to `https://www.harding.edu/fmccown/classes/comp475-s13/web-structure-homework.pdf`, it made my approach clear to this problem that I should start from SCC. From my observation the values could be as follows:

SCC: A, B, C, G

Since SCC is at the center of the graph, it will contain values which have directed links amongst each other and also have in-links or out-links to other nodes outside of SCC.

IN: M, O, P

Values which have no in-links but can reach the SCC values are considered as IN values.

OUT: D, H

Values which have in-links from SCC values but have no out-links are considered as OUT values.

Tendrils: I, K, L

Values which have out-links to OUT values and have in-links from IN values but do not refer to SCC values at any point are considered as Tendrils.

Tubes: N

Value which is in the middle of IN and OUT value and acts as a direct path from IN to OUT but do not refer to SCC at any point is considered as a Tube.

Disconnected: E, F

These values do not connect to anything in the graph, but only amongst them.

References

[1] "Urllib.requests Documentation." Developer Interface Requests 2.18.4 documentation, Web. 28 Jan, 2018. https://docs.python.org/3.0/library/urllib.request.html.

[2] Richardson, Leonard. "Beautiful Soup Documentation." Beautiful Soup Documentation - Beautiful Soup 4.4.0 Documentation. N.p., n.d. Web. 28 Jan. 2018. https://www.crummy.com/software/BeautifulSoup/bs4/doc/.

[3] "Urllib.parse Documentation." 21.8. urllib.parse Parse URLs into components Python 3.6.4 documentation, Web. 28 Jan. 2018. https://docs.python.org/3/library/urllib.parse.html.

[4] Martijn Pieters. "How to Check Redirects using Python?" Stack Overflow. N.p., n.d. Web. 28 Jan. 2018. https://stackoverflow.com/questions/20475552/python-requests-library-redirect-new-url.

[5] Lalinsk, Luk. "How Can I Check If a URL Is Absolute Using Python?" Stack Overflow. N.p., n.d. Web. 28 Jan. 2018. https://stackoverflow.com/questions/8357098/how-can-i-check-if-a-url-is-absolute-using-python.

[6] "Webgraphviz." Webgraphviz. N.p., n.d. Web. 28 Jan. 2018. http://www.webgraphviz.com/.