

## HW 1 - Web Science Intro

Swathi Venkatesh

09/19/2021

### Q1

*Consider the "bow-tie" structure of the web in the Broder et al. paper "Graph Structure in the Web" that was described in Module 1. Now consider the following links:*

*Draw the resulting directed graph (either sketch on paper or use another tool) showing how the nodes are connected to each other and include an image in your report. This does not need to fit into the bow-tie type diagram, but should look more similar to the graph on slide 24 from Module-01 Web-Science-Architecture.*

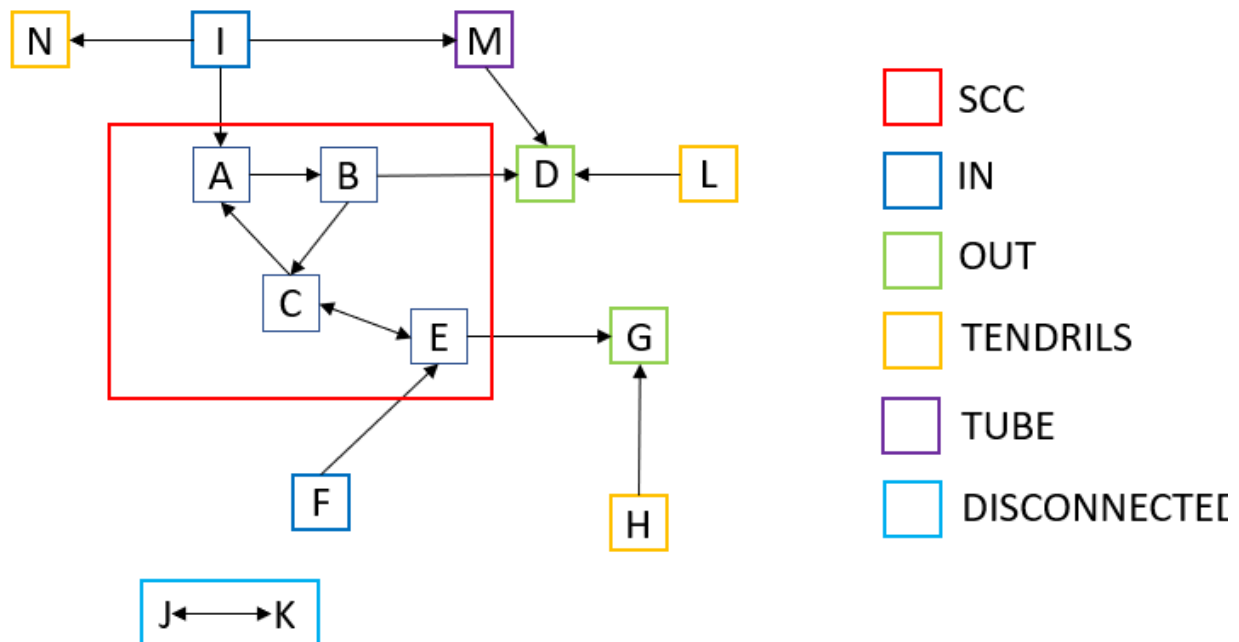
A -- B  
B -- C  
B -- D  
C -- A  
C -- E  
E -- C  
E -- G  
F -- E  
H -- G  
I -- A  
I -- M  
I -- N  
J -- K  
L -- D  
M -- D

*For the graph, list the nodes (in alphabetical order) that are each of the following categories:*

- SCC:
- IN:
- OUT:
- Tendrils:
  - o indicate if the tendril is reachable from IN or can reach OUT
- Tubes:
  - o explain how the nodes serve as tubes
- Disconnected:

## Answer

Figure 1 shows the Bow-tie structure of the given link.



**Figure 1:** Bow-tie structure of the given link

**SCC:** It stands for strongly connected network. In the Figure 1 the nodes A, B, C, E are connected to one another. They belong to one SCC.

**IN:** In the Figure 1 nodes I and F belong to IN category because they can reach the SCC but can't be reached by it.

**OUT:** In the Figure 1 nodes D and G belong to OUT category. They are nodes that can be reached from SCC but they don't link back to it.

**Tendrils:** In the Figure 1 the nodes N, L, H and M belong to tendrils. The tendril comes from the IN group (node I and M) and into the outgroup (node D and G). These nodes cannot reach the SCC and also cannot be reached by the SCC.

**Tubes:** In the Figure 1 the node M belongs to tube as it connects IN node to OUT node by temporarily routing the SCC. Node M has inlink from the node I (IN group) and has outlink to node D (OUT group) without crossing SCC.

**Disconnected:** In the Figure 1 the nodes J and K are disconnected as they have no inlink from other nodes and no outlink to other nodes.

## Discussion

The tools such as Microsoft Power Point was used to create the Figure 1.

## Q2

Demonstrate that you know how to use curl and are familiar with the available options.

URI to request: `http://www.cs.odu.edu/~mweigle/courses/cs532/ua_echo.php`

a) First, load the URI directly in your browser and take a screenshot. The resulting webpage should show the "User-Agent" HTTP request header that your web browser sends to the web server.

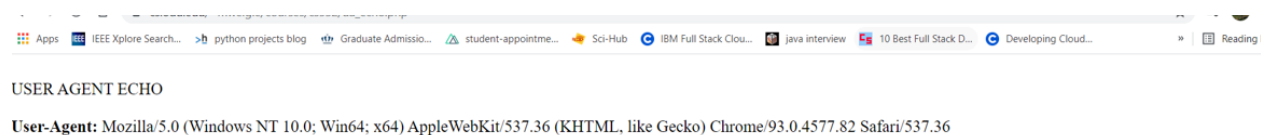
b) In a single curl command, request the URI, show the HTTP response headers, follow any redirects, and change the User-Agent HTTP request field to "CS432/532". Show command you used and the result of your execution on the command line. (Either take a screenshot of your terminal or copy/paste into a code segment.)

c) In a single curl command, request the URI, follow any redirects, change the User-Agent HTTP request field to "CS432/532", and save the HTML output to a file. Show the command you used and the result of your execution on the command line. View the HTML output file that was produced by curl in a web browser and take a screenshot.

Explain the results you get for each of these steps.

## Answer

a) The Figure 2 shows the User-Agent HTTP header that the browser sends to the web server. Figure 2 shows the URI opened in the browser.



**Figure 2:** URI opened in the browser

b) The curl command used here is:

```
curl -v -L -H "User-Agent: CS432/532" http://www.cs.odu.edu/~mweigle/courses/cs532/ua_echo.php
```

The `-v` stands for verbose. This option is used to get more data on the input. In the curl command above all the details about the user agent are displayed because of `-v` option.

`-L`: this option is used when the request made has been moved to another location. In the output we can see that the web link has 301 status that means redirection has taken place and requested page is moved to different location.

`-H`: the header content is changed using the `-H` option. In the above command the user agent is set to CS432/532 because of the `-H` option.

After executing the above command the User-Agent is displayed as CS 432/532. The screenshot provides the result of execution of the above command.

```

sol.cs.odu.edu - PuTTY
cs_svenk001@sol:~$ curl -v -L -H "User-Agent: CS432/532" http://www.cs.odu.edu/~mweigle/courses/cs532/ua_echo.php
* Trying 128.82.4.100:80...
* TCP_NODELAY set
* Connected to www.cs.odu.edu (128.82.4.100) port 80 (#0)
> GET /~mweigle/courses/cs532/ua_echo.php HTTP/1.1
> Host: www.cs.odu.edu
> Accept: */*
> User-Agent: CS432/532
>
* Mark bundle as not supporting multiuse
< HTTP/1.1 301 Moved Permanently
< Server: nginx/1.18.0 (Ubuntu)
< Date: Fri, 17 Sep 2021 22:20:18 GMT
< Content-Type: text/html
< Content-Length: 178
< Connection: keep-alive
< Location: https://www.cs.odu.edu/~mweigle/courses/cs532/ua_echo.php
<
* Ignoring the response-body
* Connection #0 to host www.cs.odu.edu left intact
* Issue another request to this URL: 'https://www.cs.odu.edu/~mweigle/courses/cs532/ua_echo.php'
* Trying 128.82.4.100:443...
* TCP_NODELAY set
* Connected to www.cs.odu.edu (128.82.4.100) port 443 (#1)
* ALPN, offering h2
* ALPN, offering http/1.1
* successfully set certificate verify locations:
* CAfile: /etc/ssl/certs/ca-certificates.crt
* Capath: /etc/ssl/certs
* TLSv1.3 (OUT), TLS handshake, Client hello (1):
* TLSv1.3 (IN), TLS handshake, Newsession Ticket (4):
* TLSv1.3 (IN), TLS handshake, Newsession Ticket (4):
* old SSL session ID is stale, removing
* Mark bundle as not supporting multiuse
< HTTP/1.1 200 OK
  
```

**Figure 3:** Code executed using curl command

c) The curl command used here is:

`curl -o "one.html" -v -H "User-Agent:CS432/532" http://www.cs.odu.edu/~mweigle/courses/cs532/ua_echo.php`

`-o`: this option is used to save the curl output to the file.

`-L`: this option is used when the request made has been moved to another location. In the output we can see that the web link has 301 status that means redirection has taken place and requested

page is moved to different location.

-H: the header content is changed using the -H option. In the above command the user agent is set to CS432/532 because of the -H option.

Figure 4 shows the command used to change User-Agent to CS432/532 and Figure 5 shows the output in web browser when User-Agent is changed to CS432/532.

```

: Transfer-Encoding: chunked
: Connection: keep-alive
: Vary: Accept-Encoding
: Vary: Accept-Encoding
:
<!DOCTYPE html>
<html>
<body>

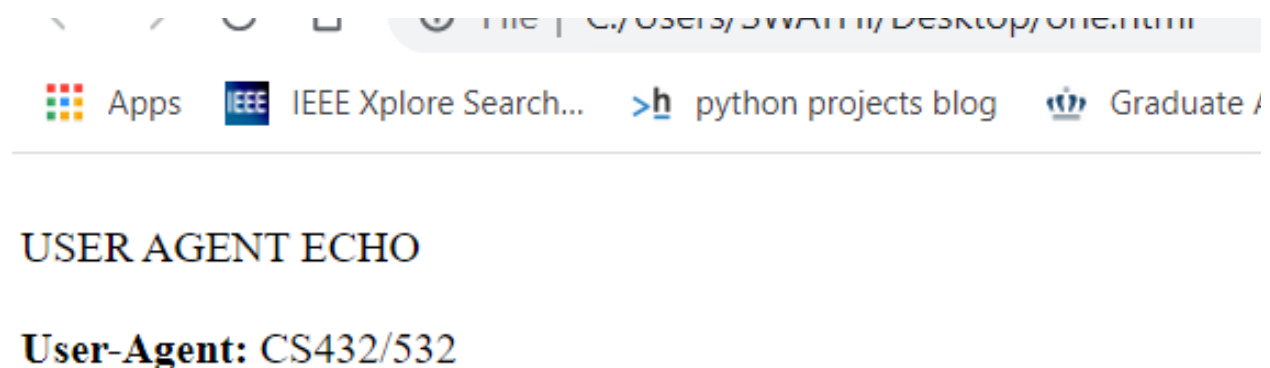
<br/>USER AGENT ECHO
<br/><br/>
<b>User-Agent:</b> CS432/532<br/>

</body>
</html>
* Connection #1 to host www.cs.odu.edu left intact

C:\Users\SWATHI\Desktop>curl -o "one.html" -v -H "User-Agent: CS432/532" http://www.cs.odu.edu/~mweigle/courses/cs532/ua_echo.php
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total     Spent    Left     Speed
100  178    100    178     0     0  44500      0 --:--:-- --:--:-- --:--:-- 44500
* TCP_NODELAY set
* Connected to www.cs.odu.edu (128.82.4.100) port 80 (#0)
* GET /~mweigle/courses/cs532/ua_echo.php HTTP/1.1
* Host: www.cs.odu.edu
* Accept: */*

```

**Figure 4:** User agent output in command line



USER AGENT ECHO

**User-Agent: CS432/532**

**Figure 5:** User agent output in web browser

## Discussion

The curl documentation and slides were referred to work on it.

## Q3

Write a Python program to find links to PDFs in a webpage.

Your program must do the following:

take the URI of a webpage as a command-line argument extract all the links from the page for each link, request the URI and use the Content-Type HTTP response header to determine if the link references a PDF file for all links that reference a PDF file, print the original URI (found in the source of the original HTML), the final URI (after any redirects), and the number of bytes in the PDF file. (Hint: Content-Length HTTP response header) Here is a snippet of the expected operation:

Show that the program works on 3 different URIs, one of which must be <https://www.cs.odu.edu/~mweigle/courses/cs532/pdfs.html>, which contains 8 links to PDFs.

Many faculty members have a list of their publications in PDF form on their webpages. You can discover ODU CS faculty webpages is through the Research page on the CS homepage. Click on a faculty member's name and that will take you to their ODU directory page. Most of us have another link on that page that goes to our homepages where you can then find a list of publications that will often link to PDFs.

Also, there are a set of pages linked on our CS 432/532 Syllabus that say "pdf available". If you follow some of those links, you'll likely find a page that links to at least one PDF.

You will likely want to use the BeautifulSoup Python library for this question. On the ODU-CS Linux machines, you may need to run `pip3 install beautifulsoup4` before you can use BeautifulSoup, but you don't need root privileges to do this.

## Answer

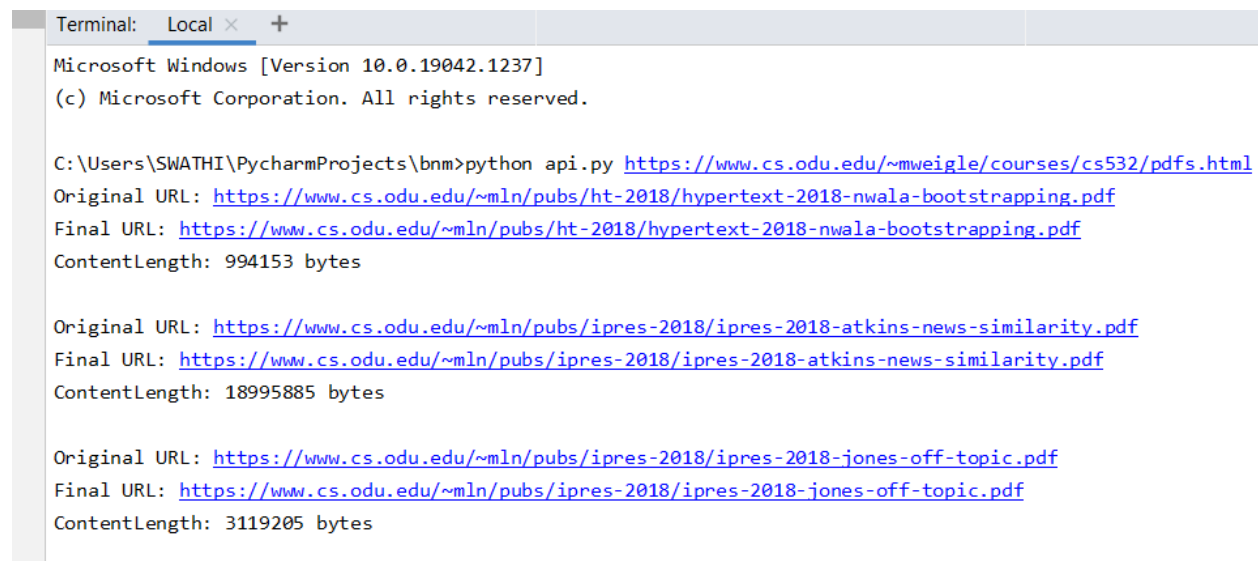
```
1 import sys
2 from bs4 import BeautifulSoup
3 import requests
4 if __name__ == '__main__':
5     u = sys.argv[1]
6     pt = 'application/pdf'
7     if u != '':
8         file = requests.get(u)
9         # getting html file from response
10        html = file.text
11
12        for link in BeautifulSoup(html, 'html.parser').find_all('a', attrs
                               ={'href': True}):
```

```
13     file = requests.get(link['href'])
14     o = file.url
15     i = 0
16     for url in file.history:
17         i+= 1
18         if i > 100:
19             break
20         continue
21         # fetch final url after checking if it is pdf link or not
22     f = file.url
23     # examine if content type is pdf link
24     if file.headers['Content-Type'] == pt:
25         cl = file.headers['Content-Length']
26         print(f'Original URL: {o}\n'f'Final URL: {f}\n'f'
              ContentLength: {cl} bytes\n')
27 else:print('')
```

**Listing 1: Python code**

```
1 C:\Users\SWATHI\PycharmProjects\bnm>python api.py https://www.cs.odu.
  edu/~mweigle/courses/cs532/pdfs.html
2 Original URL: https://www.cs.odu.edu/~mln/pubs/ht-2018/hypertext-2018-
  nwala-bootstrapping.pdf
3 Final URL: https://www.cs.odu.edu/~mln/pubs/ht-2018/hypertext-2018-
  nwala-bootstrapping.pdf
4 ContentLength: 994153 bytes
5
6 Original URL: https://www.cs.odu.edu/~mln/pubs/ipres-2018/ipres-2018-
  atkins-news-similarity.pdf
7 Final URL: https://www.cs.odu.edu/~mln/pubs/ipres-2018/ipres-2018-
  atkins-news-similarity.pdf
8 ContentLength: 18995885 bytes
9
10 Original URL: https://www.cs.odu.edu/~mln/pubs/ipres-2018/ipres-2018-
  jones-off-topic.pdf
11 Final URL: https://www.cs.odu.edu/~mln/pubs/ipres-2018/ipres-2018-jones
  -off-topic.pdf
12 ContentLength: 3119205 bytes
13
14 Original URL: https://www.cs.odu.edu/~mln/pubs/ipres-2018/ipres-2018-
  jones-archiveit.pdf
15 Final URL: https://www.cs.odu.edu/~mln/pubs/ipres-2018/ipres-2018-jones
  -archiveit.pdf
16 ContentLength: 2639215 bytes
17
18 Original URL: https://www.cs.odu.edu/~mln/pubs/jcdl-2018/jcdl-2018-
  nwala-scraping-serps-seeds.pdf
19 Final URL: https://www.cs.odu.edu/~mln/pubs/jcdl-2018/jcdl-2018-nwala-
```

```
scraping-serps-seeds.pdf
20 ContentLength: 2172494 bytes
21
22 Original URL: https://www.cs.odu.edu/~mln/pubs/jcdl-2018/jcdl-2018-
    kelly-private-public-web-archives.pdf
23 Final URL: https://www.cs.odu.edu/~mln/pubs/jcdl-2018/jcdl-2018-kelly-
    private-public-web-archives.pdf
24 ContentLength: 2553579 bytes
25
26 Original URL: https://www.cs.odu.edu/~mln/pubs/jcdl-2018/jcdl-2018-
    aturban-archivenow.pdf
27 Final URL: https://www.cs.odu.edu/~mln/pubs/jcdl-2018/jcdl-2018-aturban
    -archivenow.pdf
28 ContentLength: 3998654 bytes
29
30 Original URL: https://www.cs.odu.edu/~mln/pubs/jcdl-2018/jcdl-2018-alam
    -archive-banner.pdf
31 Final URL: https://www.cs.odu.edu/~mln/pubs/jcdl-2018/jcdl-2018-alam-
    archive-banner.pdf
32 ContentLength: 596000 bytes
```

**Listing 2:** Python code

```
Terminal: Local x +
Microsoft Windows [Version 10.0.19042.1237]
(c) Microsoft Corporation. All rights reserved.

C:\Users\SWATHI\PycharmProjects\bnm>python api.py https://www.cs.odu.edu/~mweigle/courses/cs532/pdfs.html
Original URL: https://www.cs.odu.edu/~mln/pubs/ht-2018/hypertext-2018-nwala-bootstrapping.pdf
Final URL: https://www.cs.odu.edu/~mln/pubs/ht-2018/hypertext-2018-nwala-bootstrapping.pdf
ContentLength: 994153 bytes

Original URL: https://www.cs.odu.edu/~mln/pubs/ipres-2018/ipres-2018-atkins-news-similarity.pdf
Final URL: https://www.cs.odu.edu/~mln/pubs/ipres-2018/ipres-2018-atkins-news-similarity.pdf
ContentLength: 18995885 bytes

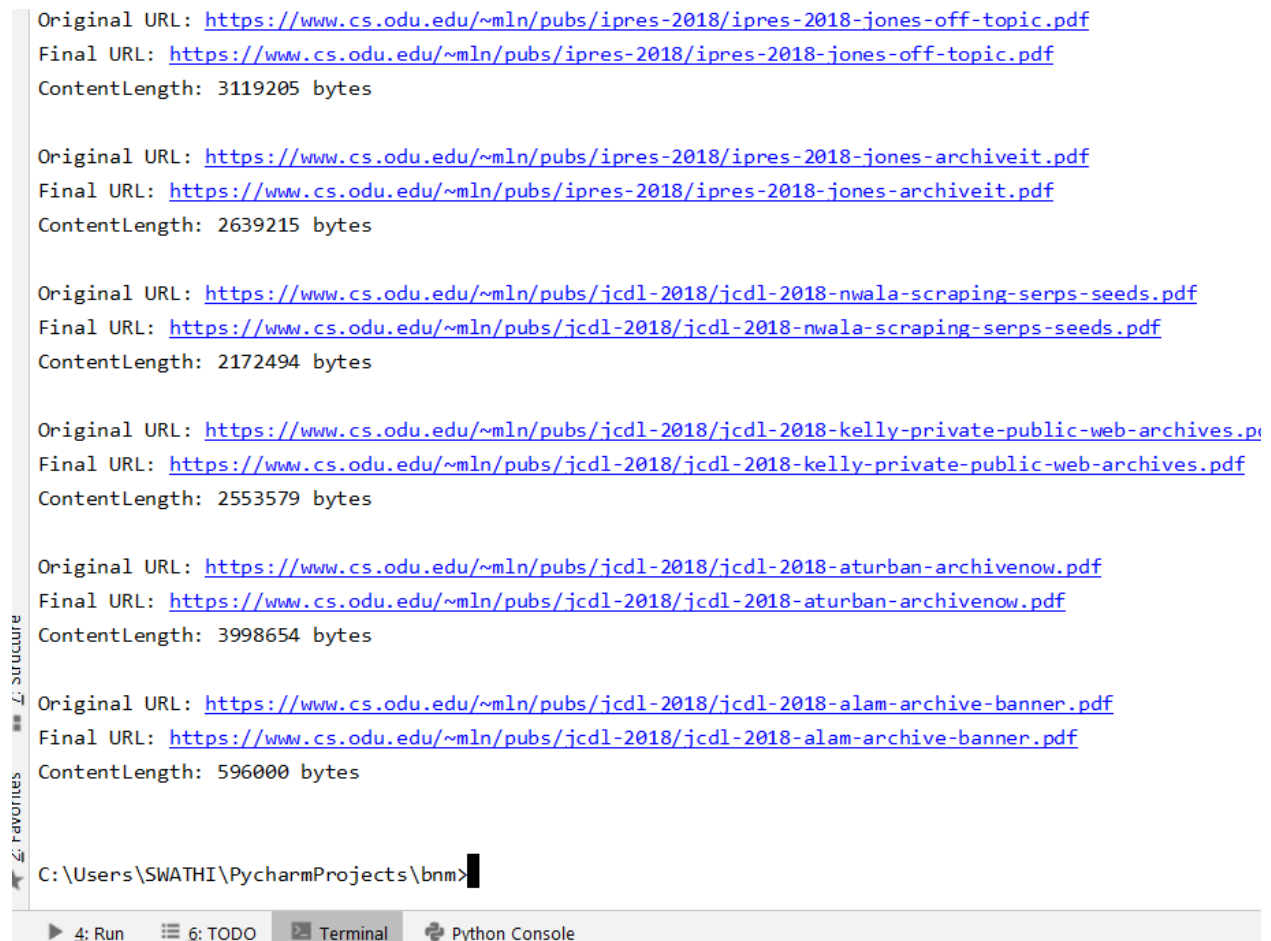
Original URL: https://www.cs.odu.edu/~mln/pubs/ipres-2018/ipres-2018-jones-off-topic.pdf
Final URL: https://www.cs.odu.edu/~mln/pubs/ipres-2018/ipres-2018-jones-off-topic.pdf
ContentLength: 3119205 bytes
```

**Figure 6:** Program output

## Discussion

The Pycharm 2019.2.3 is used to execute the code.





```
Original URL: https://www.cs.odu.edu/~mln/pubs/ipres-2018/ipres-2018-jones-off-topic.pdf
Final URL: https://www.cs.odu.edu/~mln/pubs/ipres-2018/ipres-2018-jones-off-topic.pdf
ContentLength: 3119205 bytes

Original URL: https://www.cs.odu.edu/~mln/pubs/ipres-2018/ipres-2018-jones-archiveit.pdf
Final URL: https://www.cs.odu.edu/~mln/pubs/ipres-2018/ipres-2018-jones-archiveit.pdf
ContentLength: 2639215 bytes

Original URL: https://www.cs.odu.edu/~mln/pubs/jcdl-2018/jcdl-2018-nwala-scraping-serps-seeds.pdf
Final URL: https://www.cs.odu.edu/~mln/pubs/jcdl-2018/jcdl-2018-nwala-scraping-serps-seeds.pdf
ContentLength: 2172494 bytes

Original URL: https://www.cs.odu.edu/~mln/pubs/jcdl-2018/jcdl-2018-kelly-private-public-web-archives.pdf
Final URL: https://www.cs.odu.edu/~mln/pubs/jcdl-2018/jcdl-2018-kelly-private-public-web-archives.pdf
ContentLength: 2553579 bytes

Original URL: https://www.cs.odu.edu/~mln/pubs/jcdl-2018/jcdl-2018-aturban-archivenow.pdf
Final URL: https://www.cs.odu.edu/~mln/pubs/jcdl-2018/jcdl-2018-aturban-archivenow.pdf
ContentLength: 3998654 bytes

Original URL: https://www.cs.odu.edu/~mln/pubs/jcdl-2018/jcdl-2018-alam-archive-banner.pdf
Final URL: https://www.cs.odu.edu/~mln/pubs/jcdl-2018/jcdl-2018-alam-archive-banner.pdf
ContentLength: 596000 bytes

C:\Users\SWATHI\PycharmProjects\bnm>
```

Figure 7: Program output

## References

- CodeGrepper, <https://www.codegrepper.com/code-examples/python/how-to-find-pdf-file-in-link-beautifulsoup>
- Use of BeautifulSoup, <https://www.dataquest.io/blog/web-scraping-python-using-beautiful-soup/>
- Github, <https://github.com>
- StackOverflow, <https://stackoverflow.com/questions/866946/how-can-i-see-the-request-headers-made-by-curl-when-sending-a-request-to-the-ser>
- Curl Usage, <https://mkyong.com/web/curl-display-request-headers-and-response-headers/>
- Curl documentation, <https://stackoverflow.com/questions/21226980/curl-output-to-file>