Problem 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.

Solution:

Started with understanding curl from the secure shell by using curl --help command.

Got various options included in curl. Referred few sites in order to get in touch with curl.

My next step was to get the web forms which use POST method. It was hard to find them.

Later, I found the link in the google groups and started using it.

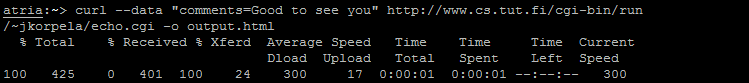
The easiest way to post data to a form with curl command is

Curl --data "comments=Good to see you" <http://www.cs.tut.fi/cgi-bin/run/~jkorpela/echo.cgi> -o output.html

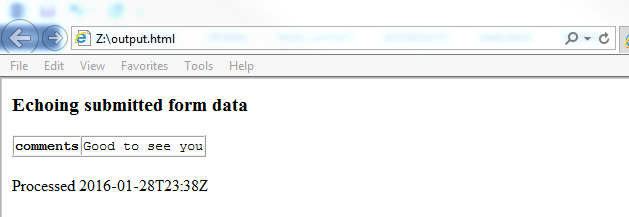


Screenshot of the webpage with the form to which I posted data is shown above.

The response I got after implementing the curl command is shown below



The curl command automatically generates a file in my Z drive with the name that I gave. I viewed that file in a browser and here is the response I got



Problem 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-s16/test/pdfs.html

The algorithm I followed in order to write this program is stated below.

1. First I tried to write the program and gather all the links from one particular link like [www.cs.odu.edu](http://www.cs.odu.edu).
2. For doing this I imported the required libraries.
3. Extracted the code using BeautifulSoup library.
4. Read the entire data from the above url using urllib2 and Beautifulsoup libraries.
5. Got all the data within the anchor tag using beautifulsoup function called soup. find\_all().
6. Wrote a sample function which checks for the url and returns false if it is not.
7. Now I ran a for loop where it stores the headers of each link and then also checks whether the content-type in the headers is pdf or not. If it is pdf then I am printing the url, content-length and the status code.
8. Similar process is carried on with 2 other links and the results are shown below.

**@misc**{website:UsingCurltoautomatehttpjobs,

title="Curl Preface",

howpublished = "http://curl.haxx.se/docs/httpscripting.html#Background"

}

**@misc**{website:ThestructureofWeb,

title="Graph Strducture in the Web",

author="Andrei Broder1,Ravi Kumar2,Farzin Maghoul1",

howpublished = "http://www9.org/w9cdrom/160/160.html"

}

**@misc**{website:BeautifulSoup,

title = "BeautifulSoup Documentation",

howpublished = "http://www.crummy.com/software/BeautifulSoup/bs4/doc/"

}

**@misc**{website:Urllib2,

title = "How to fetch Internet Resources Using urllib2",

howpublished = "https://docs.python.org/2/howto/urllib2.html"

}

http://stackoverflow.com/questions/2143674/how-do-i-get-a-content-type-of-a-file-in-python-with-url

Problem 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 --> J

I --> K

J --> D

L --> D

M --> A

M --> N

N --> D

O --> A

P --> G

For the above graph, give the values for:

IN:

SCC:

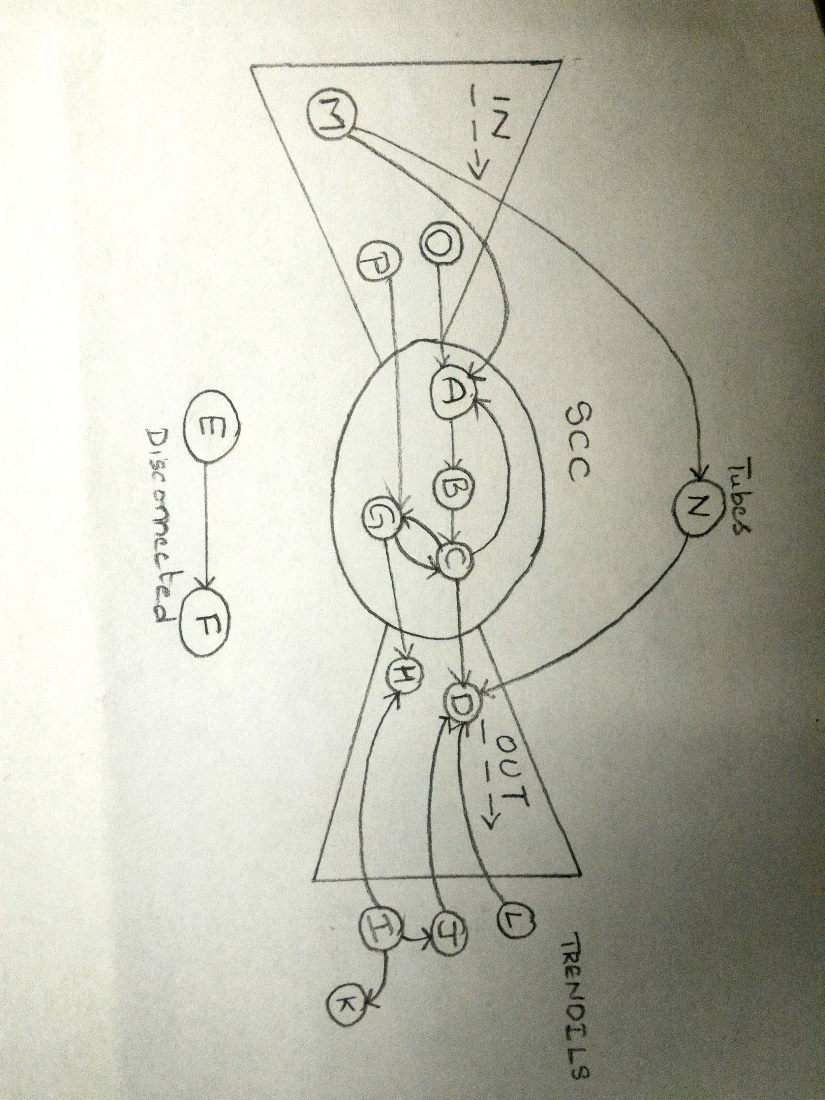
OUT:

Tendrils:

Tubes:

Disconnected:

Solution:



IN: M,O,P

SCC: A,B,C,G

OUT: D,H

Tendrils: I,J,K,L

Tubes: N

Disconnected: E,F

Model graph is drawn based on the bow-tie concept including all the node points stated above.

Explanations for each node have been provided below.

IN: O, M, P

According to the definition of IN, any node in IN can be connected to any other node in IN or can be connected to any node in SCC but not with any other nodes. M,O,P falls into this category they are connected to A,G nodes in SCC.

SCC: A, B, C, G

Strongly Connected Component consists of group of nodes which are connected to each other directly or indirectly and these nodes can be connected from IN and connected to OUT. A,B,C,G come under this category.

OUT: D,H

Nodes which exit from SCC and are not connected back to SCC come under OUT. D and H show these characteristics so they come under OUT.

Tendrils: I,J,K,L

The nodes which cannot be directly connected to the nodes in SCC. I,J,K,L come under Tendrils.

Tubes: N

Nodes which pass from IN to OUT without any interaction with SCC are called Tubes. N is a tube linking from M to D.

Disconnected: E,F

Nodes which are not connected to any part of the graph are called Disconnected nodes. E and F are connected to each other but not connected to any part of the graph.

References

[1] Andrew Broder, Ravi Kumar, Farzin Maghoul, Prabhakar Ragha-

van, Sridhar Rajagopalan, Raymie Stata, Andrew Tomkins, and Janet

Wiener. Graph structure in the web. In Ninth International World Wide

Web Conference, 2000 <http://www9.org/w9cdrom/160/160.html>.

<http://curl.haxx.se/docs/httpscripting.html#Background>

<http://curl.haxx.se/docs/manpage.html#URL>