CS 431 - Homework 1

Joshua Hall

Fall 2025

Question 1

SCC: A, B, C, D, G

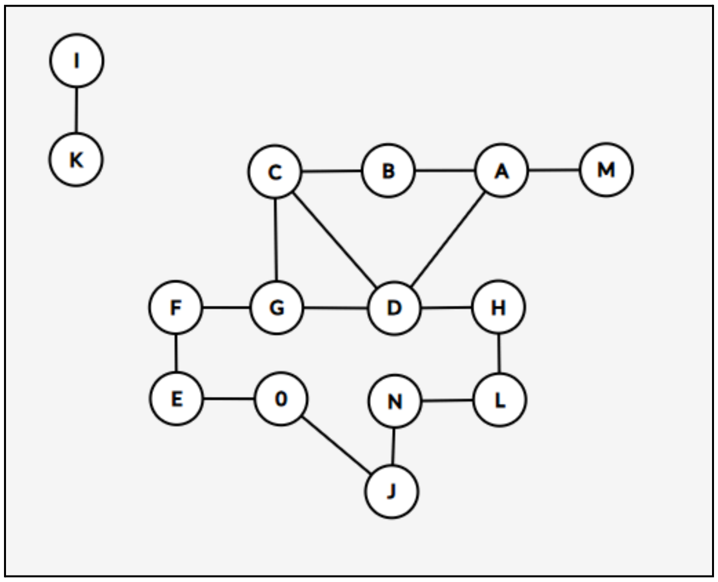
IN: E, F, M

OUT: H, L

Tendrils: J, N, O. O is reachable from IN (E → O). J, N, O can also reach OUT (… → N → L)

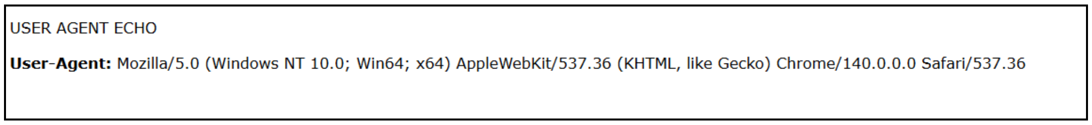
Tubes: O, J, N. Path from IN to OUT: E → O → J → N → L, bypassing the SCC

Disconnected: I, K



Question 2

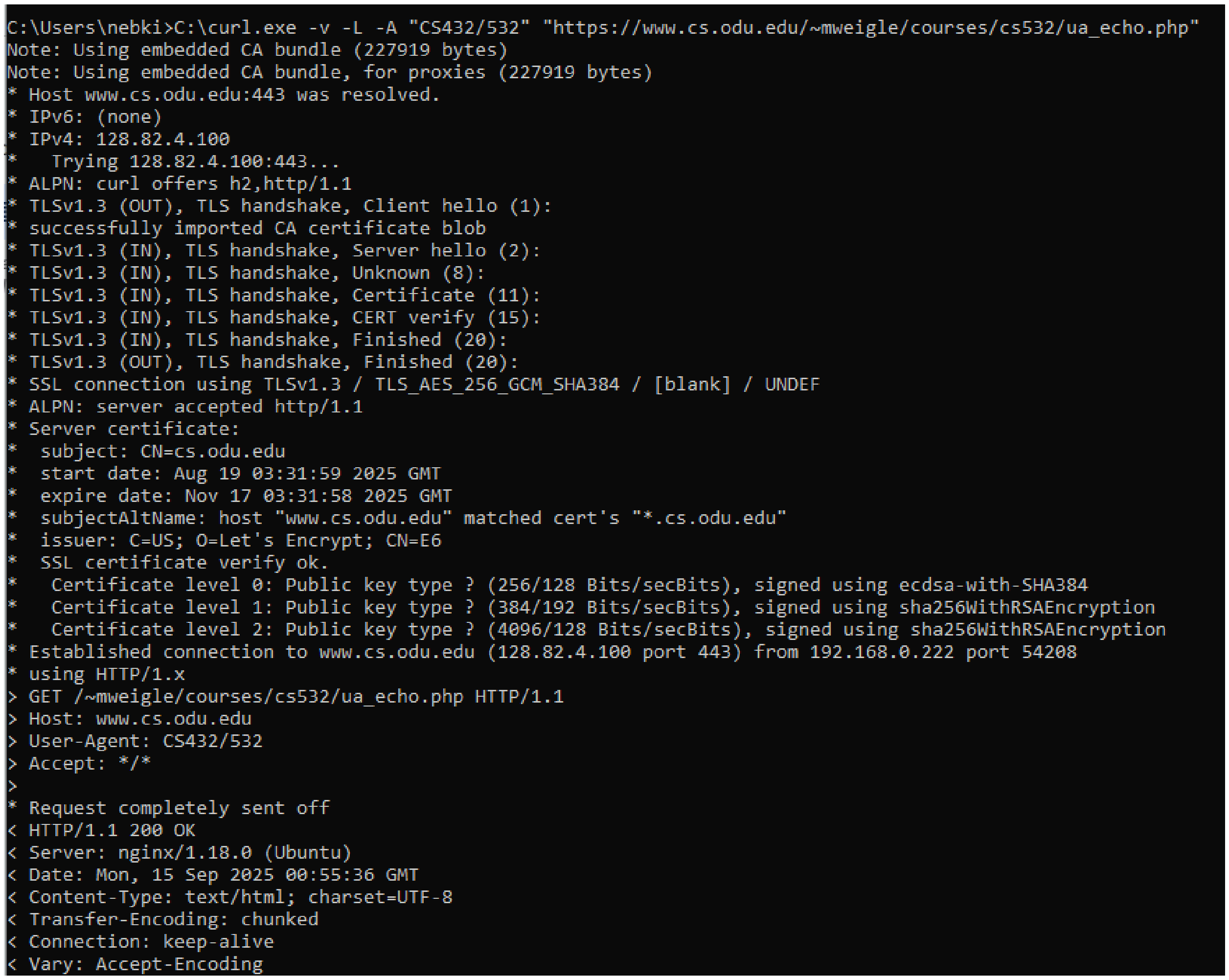
Q2.a



USER AGENT ECHO

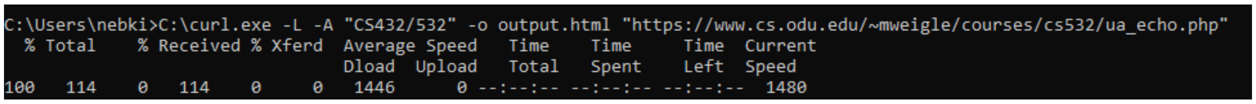
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/140.0.0.0 Safari/537.36

Q2.b



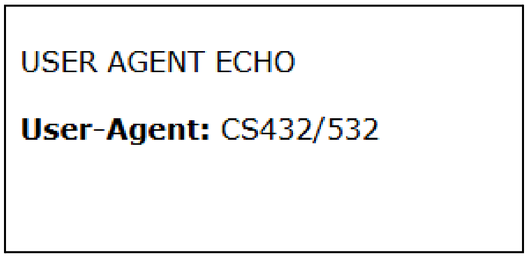
```
C:\Users\nebki>C:\curl.exe -v -L -A "CS432/532" "https://www.cs.odu.edu/~mweigle/courses/cs532/ua_echo.php"
Note: Using embedded CA bundle (227919 bytes)
Note: Using embedded CA bundle, for proxies (227919 bytes)
* Host www.cs.odu.edu:443 was resolved.
* IPv6: (none)
* IPv4: 128.82.4.100
*   Trying 128.82.4.100:443...
* ALPN: curl offers h2,http/1.1
* TLSv1.3 (OUT), TLS handshake, Client hello (1):
* successfully imported CA certificate blob
* TLSv1.3 (IN), TLS handshake, Server hello (2):
* TLSv1.3 (IN), TLS handshake, Unknown (8):
* TLSv1.3 (IN), TLS handshake, Certificate (11):
* TLSv1.3 (IN), TLS handshake, CERT verify (15):
* TLSv1.3 (IN), TLS handshake, Finished (20):
* TLSv1.3 (OUT), TLS handshake, Finished (20):
* SSL connection using TLSv1.3 / TLS_AES_256_GCM_SHA384 / [blank] / UNDEF
* ALPN: server accepted http/1.1
* Server certificate:
*  subject: CN=cs.odu.edu
*  start date: Aug 19 03:31:59 2025 GMT
*  expire date: Nov 17 03:31:58 2025 GMT
*  subjectAltName: host "www.cs.odu.edu" matched cert's "*.cs.odu.edu"
*  issuer: C=US; O=Let's Encrypt; CN=E6
*  SSL certificate verify ok.
*   Certificate level 0: Public key type ? (256/128 Bits/secBits), signed using ecdsa-with-SHA384
*   Certificate level 1: Public key type ? (384/192 Bits/secBits), signed using sha256WithRSAEncryption
*   Certificate level 2: Public key type ? (4096/128 Bits/secBits), signed using sha256WithRSAEncryption
* Established connection to www.cs.odu.edu (128.82.4.100 port 443) from 192.168.0.222 port 54208
* using HTTP/1.x
> GET /~mweigle/courses/cs532/ua_echo.php HTTP/1.1
> Host: www.cs.odu.edu
> User-Agent: CS432/532
> Accept: */*
>
* Request completely sent off
< HTTP/1.1 200 OK
< Server: nginx/1.18.0 (Ubuntu)
< Date: Mon, 15 Sep 2025 00:55:36 GMT
< Content-Type: text/html; charset=UTF-8
< Transfer-Encoding: chunked
< Connection: keep-alive
< Vary: Accept-Encoding
```

Q2.c



```
C:\Users\nebki>C:\curl.exe -L -A "CS432/532" -o output.html "https://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   114    0   114    0     0   1446      0 --:--:-- --:--:-- --:--:--  1480
```

Q2.d



USER AGENT ECHO

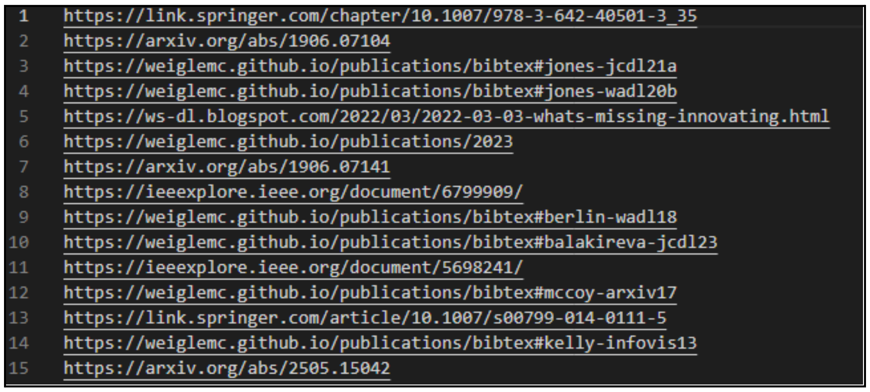
User-Agent: CS432/532

Question 3

To collect the required 500 URIs, I implemented a Python crawler (collect-webpages.py) using the requests and BeautifulSoup libraries.

1. The program begins with a seed URI provided via the command line.
2. It downloads the page, verifies that the response is of type text/html, and parses the HTML with BeautifulSoup to extract <a href> links.
3. For each link:
    - The program follows redirects automatically.
    - It checks the Content-Type header to ensure the resource is HTML (text/html).
    - It checks the Content-Length header (or falls back to the actual response size) to ensure the page is larger than 1000 bytes.
4. If the URI passes these checks and is unique, it is added to the results.
5. If the crawler runs out of links before reaching 500 unique URIs, it randomly picks an already-collected URI as a new seed and continues.
6. Every request uses a 5-second timeout to avoid hanging on unresponsive pages.
7. Once 500 unique URIs are collected, they are written to collected_uris.txt for use in later assignments.

My seed website was: https://weiglemc.github.io/. The following is a small sample of the output:



```
https://link.springer.com/chapter/10.1007/978-3-642-40501-3_35
https://arxiv.org/abs/1906.07104
https://weiglemc.github.io/publications/bibtex#jones-jcdl21a
https://weiglemc.github.io/publications/bibtex#jones-wadl20b
https://ws-dl.blogspot.com/2022/03/2022-03-03-whats-missing-innovating.html
https://weiglemc.github.io/publications/2023
https://arxiv.org/abs/1906.07141
https://ieeexplore.ieee.org/document/6799909/
https://weiglemc.github.io/publications/bibtex#berlin-wadl18
https://weiglemc.github.io/publications/bibtex#balakireva-jcdl23
https://ieeexplore.ieee.org/document/5698241/
https://weiglemc.github.io/publications/bibtex#mccoy-arxiv17
https://link.springer.com/article/10.1007/s00799-014-0111-5
https://weiglemc.github.io/publications/bibtex#kelly-infovis13
https://arxiv.org/abs/2505.15042
```