CS 4250 – Assignment #3
Maximum Points: 100 pts.

Bronco ID: |__|__|__|__|__|__|__|__|__|

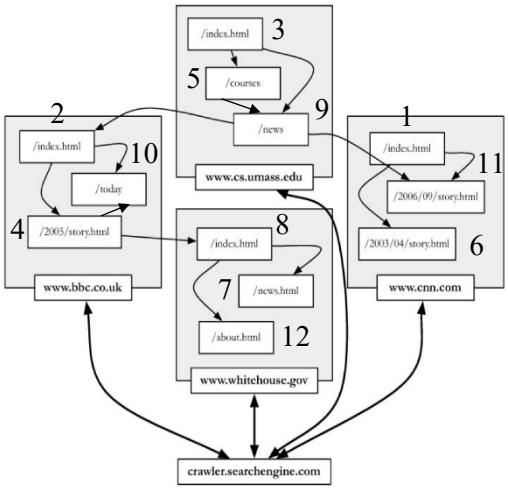
Last Name: _____

First Name: _____

Note 1: Your submission header must have the format as shown in the above-enclosed rounded rectangle.
Note 2: Homework is to be done individually. You may discuss the homework problems with your fellow students, but you are NOT allowed to copy – either in part or in whole – anyone else's answers.
Note 3: Your deliverable should be a .pdf file submitted through Gradescope until the deadline. Do not forget to assign a page to each of your answers when making a submission. In addition, source code (.py files) should be added to an online repository (e.g., github) to be downloaded and executed later.
Note 4: All submitted materials must be legible. Figures/diagrams must have good quality.
Note 5: Please use and check the Canvas discussion for further instructions, questions, answers, and hints. The bold words/sentences provide information for a complete or accurate answer.

1. [18 points]. Based on the pseudocode and navigation diagram below, answer the following questions:

```
procedure crawlerThread (frontier)
 while not frontier.done() do
   url <– frontier.nextURL()
   html <– retrieveURL(url)
   for each not visited url in parse (html) do
    frontier.addURL(url)
   end for
 end while
end procedure
```



a. [6 points]. What should be the sequence of page visits if the seed URL is only page 3? Use the notation 1-6-2…8.

3-5-9-2-11-4-10-8-7-12

b. [3 points]. In the hypothetical case that those 4 sites and 12 pages comprise the entire Web, which page(s) should be located on the deep web? Explain your answer for full marks.

1 and 6, since there are no links pointing to those two pages.

c. [6 points]. Now, considering that there is an arrow flowing from 3 to 1. What should be the sequence of page visits if the seed URL is only pages 3? Use the notation 1-6-2…8.

3-1-5-9-6-11-2-4-10-8-7-12

d. [3 points]. In the hypothetical case that those 4 sites and 12 pages comprise the entire Web, which page(s) should be located on the deep web when this new arrow (3→1) is added? Explain your answer for full marks.

None, since all 12 pages have links pointing got them.

2. [28 points]. Based on the HTML below, find and print the requested content by using BeautifulSoup.

```
<html>
<head>
    <title>My first web page</title>
</head>
<body>
<h1>My first web page</h1>
<h2>What this is tutorial</h2>
<p>A simple page put together using HTML. <em>I said a simple page.</em>.</p>
<ul>
  <li>To learn HTML</li>
  <li>
    To show off
    <ol>
      <li>To my boss</li>
      <li>To my friends</li>
      <li>To my cat</li>
      <li>To the little talking duck in my brain</li>
    </ol>
  </li>
  <li>Because I have fallen in love with my computer and want to give her some HTML loving.</li>
</ul>
<h2>Where to find the tutorial</h2>
<p><a href="http://www.aaa.com"><img src=http://www.aaa.com/badge1.gif></a></p>
<h3>Some random table</h3>
<table>
  <tr class="tutorial1">
    <td>Row 1, cell 1</td>
    <td>Row 1, cell 2<img src=http://www.bbb.com/badge2.gif></td>
    <td>Row 1, cell 3</td>
  </tr>
  <tr class="tutorial2">
    <td>Row 2, cell 1</td>
    <td>Row 2, cell 2</td>
    <td>Row 2, cell 3<img src=http://www.ccc.com/badge3.gif></td>
  </tr>
</table>
</body>
</html>
```

a. [4 points]. The title of the HTML page. Use the HTML tags to do this search.

```
print(bs.title.text)
print(bs.find('title').get_text())
```

b. [4 points]. The second list item element "li" below "To show off"? Use the HTML tags to do this search. The output should be "To my friends".

```
print(bs.find('ul').find('ol').find_all('li')[1].get_text())
```

c. [4 points]. All cells of Row 2. Use the HTML tags to do this search.

```
print(bs.find('tr', {'class': 'tutorial2'}).find_all('td'))
```

d. [4 points]. All h2 headings that includes the word "tutorial". Use the HTML tags to do this search.

```
print(bs.find_all('h2', text=re.compile('.*tutorial.*')))
```

e. [4 points]. All text that includes the "HTML" word. Use the HTML text to do this search.

```
print(bs.find_all(text=re.compile('.*HTML.*')))
```

f. [4 points]. All cells' data from the first row of the table. Use the HTML tags to do this search.

```
print([tag.get_text() for tag in bs.find('tr', {'class': 'tutorial1'}).find_all('td')])
```

g. [4 points]. All images from the table. Use the HTML tags to do this search.

```
print(bs.find('table').find_all('img'))
```

3. [24 points]. Provide two different matches for each of the regex(s) below.

a. [3 points]. b?a?c

bc, c.

b. [3 points]. c*a.

cca, a.

c. [3 points]. ac+a

aca, acca.

d. [3 points]. [A-Z]a{2,3}

Baa, Caaa.

e. [3 points]. c(a|b)c

cac, cbc.

f. [3 points]. \d{2,2}\/\d{2,2}\/\d{2,4}

12/05/23, 12/05/2023

g. [3 points]. \d(ab)*\d

11, 1abab2

h. [3 points]. ^cb+a$

cbba, cba

4. [15 points]. Write a Python program (crawler.py) that will crawl the CS website until the Permanent Faculty (they are 16 in total) page is found. The target URL is https://www.cpp.edu/sci/computer-science/faculty-and-staff/permanent-faculty.shtml.

Requirements:

1) To write this program, strictly follow the pseudocode shown below. Your `frontier` must include at the beginning only the single URL https://www.cpp.edu/sci/computer-science/ (CS home page) and from this page, search through all linked pages until the target page is found. Links might appear with full or relative addresses, and your crawler needs to consider this.
2) Stop criteria: when the crawler finds the "Permanent Faculty" heading on the page body.
3) Use the Python libraries `urllib` and `BeautifulSoup`, and `PyMongo`.
4) Use the MongoDB collection `pages` to persist pages' data.

```
procedure crawlerThread (frontier)
```

```
while not frontier.done() do
   url <– frontier.nextURL()
   html <– retrieveURL(url)
   storePage(url, html)
   if target_page (html)
      clear_frontier()
   else
     for each not visited url in parse (html) do
         frontier.addURL(url)
     end for
 end while
end procedure
```

Solution in Canvas

5. [15 points]. By using the data persisted in the previous question, write a Python program parser.py that will read the CS faculty information, parse faculty members *name*, *title*, *office*, *email*, and *website*, and persist this data in MongoDB. If you were not able to finish question 4 properly, you are allowed to include the "Permanent Faculty" page HTML data directly in MongoDB, so that you can try to complete question 5. Otherwise, use the target page URL to find the Permanent Faculty page in the database.

   Requirements:

   1) Use the Python libraries `BeautifulSoup` and `PyMongo`.
   2) Use the MongoDB collection `professors` to persist professors' data.

   Solution in Canvas

Important Note: Answers to all questions should be written clearly, concisely, and unmistakably delineated. You may resubmit multiple times until the deadline (the last submission will be considered).

NO LATE ASSIGNMENTS WILL BE ACCEPTED. ALWAYS SUBMIT WHATEVER YOU HAVE COMPLETED FOR PARTIAL CREDIT BEFORE THE DEADLINE!