**SI 507**

**Homework 7:** Web Scraping & Crawling

**Homework Objectives**

* Become competent with website structure and crawling multiple pages
* Understanding user agent concept and how to utilize it in the request

**Supporting Material**

* Beautifulsoup Documentation  
  <https://www.crummy.com/software/BeautifulSoup/bs4/doc>
* Initial page of the UMSI directory website  
  <https://www.si.umich.edu/directory?field_person_firstname_value=&field_person_lastname_value=&rid=All>
* Additional pages of the directory are sequentially numbered through the “page” paramter in the URL. The second page in the directory is:  
  <https://www.si.umich.edu/directory?field_person_firstname_value=&field_person_lastname_value=&rid=All&page=1>  
  Third page has “page=2” as the last parameter, and so on.

**Starter Files**

We have provided you with the following files:

* [hw7\_part1.py](https://www.dropbox.com/s/1xfsshj7y4lcb0y/hw7_part1.py?dl=0)
* [hw7\_part2.py](https://www.dropbox.com/s/5izqythuik6a3ic/hw7_part2.py?dl=0)

Please use these python files as a template to add your code. You can chose to use additional functions or not, depending on how you would like to structure your code.

**Homework task in brief:**

Your job in this homework is to crawl the UMSI directory website and extract some basic information about each person in the directory. You will then use this information to find out the number of current UMSI PhD students.

**Part 1 (15 points): Crawling the UMSI member (faculty, phD, staff) website with cache**

Write a function called get\_umsi\_data. The function should access each page of the directory, get the HTML associated with it, and extract data from it. The extracted data should be used to create a dictionary named umsi\_people. The keys of this dictionary should be email addresses of each person in the UMSI directory, and the value for each key should be a dictionary containing the person’s name and title.

Your dictionary should end up looking something like this (with a lot more entries, obviously!):

{

“meabbadi@umich.edu”: {“name”: Mohamed Abbadi”, “title”: “PhD student”},

“eadar@umich.edu”: {“name”: “Eytan Adar”, “title”: “Associate Professor of Electrical Engineering and Computer Science, College of Engineering and Associate Professor of Information, School of Information”}

}

Then write out the dictionary to a file directory\_dict.json.

Note:

* To get data from the UMSI site, you must have the user-agent headers parameter in your request, so that you will not get blocked by the site. You should use the following user-agent, which has been set up by our system administrators to allow crawling: requests.get(<url you want>, headers={'User-Agent': 'SI\_CLASS'})
* **Please crawl one single page first and make sure that your code works before crawling the rest of the pages**
* **Please implement caching** to prevent overwhelming crawling of the UMSI website. (The overwhelming crawling might shut down the website and then you cannot check the crawling program. **The deadline will not be extended because of the shut down problem.** Please see every query as a critical one.)
* Start as early as possible to prevent the above situation.
* The directory\_dict.json is not the cache! As always, your cache should use URLs as keys and the html content of the whole web page as the value for each key

Sample Command:

$ python3 hw7\_part1.py

The output of this command should be a newly created directory\_dict.json file.

**Part 2 (5 points): Analyze the data from part 1**

Using the directory\_dict.json you generated in part 1, this program should search the data you crawled to find the number of PhD students in the directory.

Sample Command:

$ python3 hw7\_part2.py

Sample output:

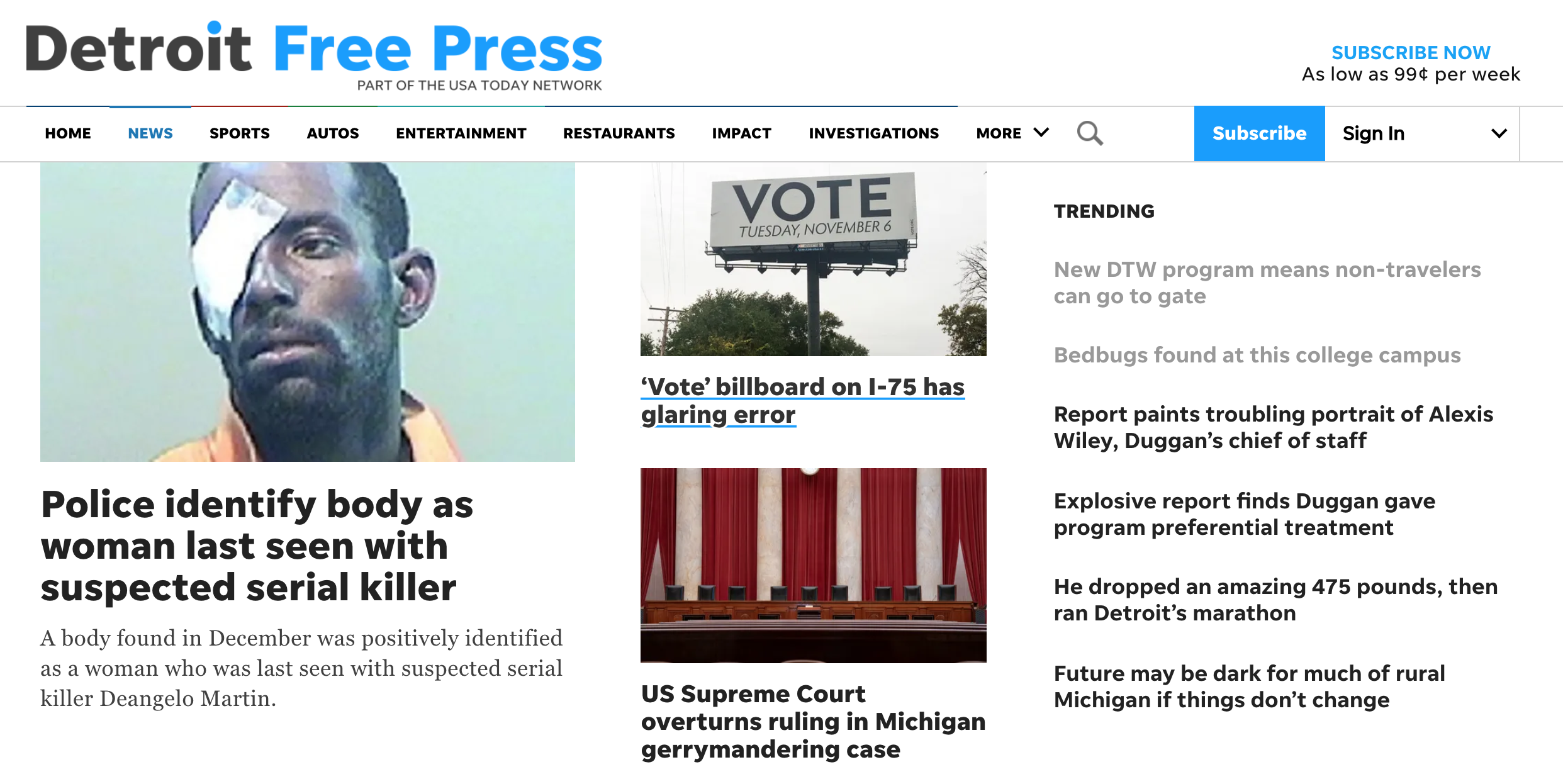
The number of PhD students: 123

Note:

The number in the sample output is not correct. Your program should return the correct number. (:

**Extra Credit 1 (2 points) Crawling headlines, authors, and time from Detroit Free Press**

Get the headlines, authors and times from the “Trending” section (top right) of the news page in Detroit Free Press (<https://www.freep.com/news/>):



Create a json file called freep.json that contains a list of dictionaries, each dictionary containing the article’s headline, authors/byline, and publication time.

Sample Command:

$ python3 hw7\_ec1.py

Sample content of the output json file:

[{“title”: “Sign up for new DTW program, and you won't need a ticket to pass security at Detroit Metro”, “byline”: “Eric D. Lawrence”, “publication time”: “1:08 p.m. ET Oct. 21, 2019”},

{“title”: “Bedbugs found at Mott Community College's Flint campus”, “byline”: “Associated Press”, “publication\_time”: “1:45 p.m. ET Oct. 21, 2019”}]

(with more entries, obviously)

Notes:

* The publication time is provided as a string that is a little more complicated than just the date and time as shown above. It usually has the following form “ Published 1:08 p.m. ET Oct. 21, 2019 | “ with both leading and trailing spaces, and the pipe symbol after the date. So, you’ll need to clean this up a bit before saving it.
* If the article is brand new, the time may only say “ Published 58 minutes ago” or something like that. It’s ok to save this as the content of your publication\_time key.
* The html of the page is messy, so it will take some hunting to get the data you need, but this is how most real website are. Unfortunately.

**What to turn in on Canvas**

* hw7\_part1.py
* hw7\_part2.py
* Any additional extra credit files