SD21063 TEAN JIN HE Data Wrangling Lab Report 5

June 16, 2023

1 DATA WRANGLING LAB REPORT 5

1.0.1 Name:Tean Jin He

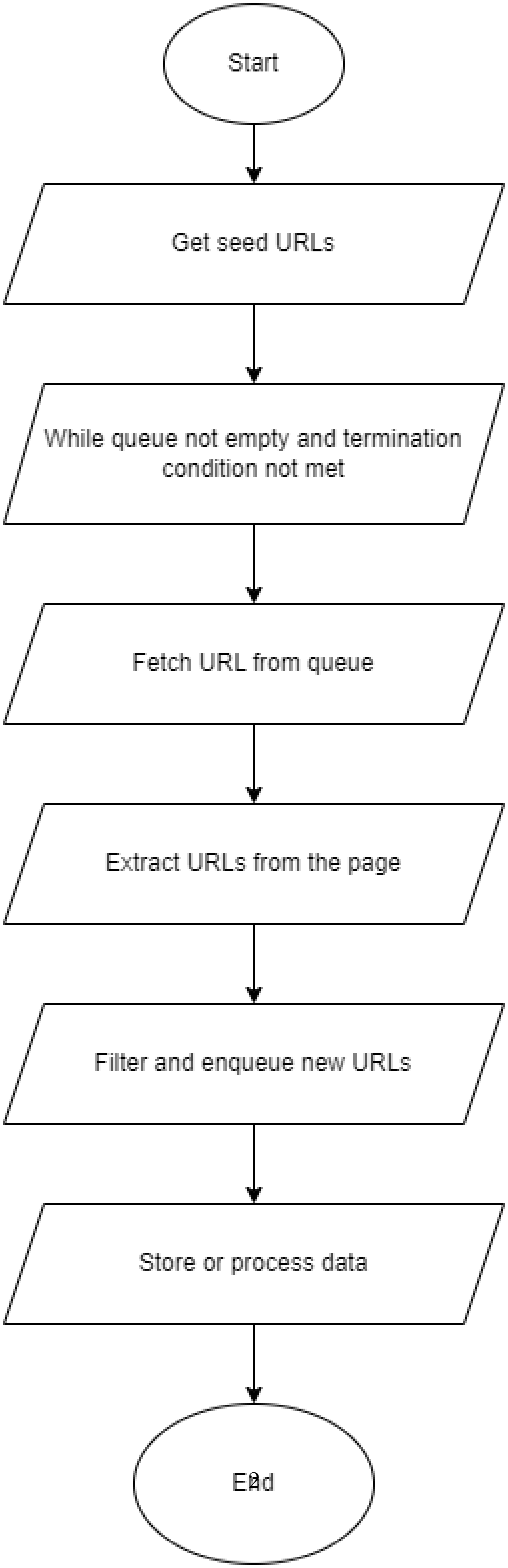
1.0.2 Matric ID : SD21063

1.0.3 Section: 02G

Question 1: Understanding Web Crawling Proses

[1]:
```
from IPython.display import Image
Image("web crawling workflow.png")
```

[1]:


Start
Get seed URLs
While queue not empty and termination condition not met
Fetch URL from queue
Extract URLs from the page
Filter and enqueue new URLs
Store or process data
End

1. Start with a seed list of URLs: The web crawling process begins by selecting one or multiple URLs as a starting point. These URLs are typically chosen based on specific criteria or domains of interest.

2. Visit the seed URL(s): The crawler accesses the first URL(s) from the seed list using an HTTP request. This request fetches the HTML content of the webpage.

3. Extract URLs from the visited page: The crawler parses the HTML content of the webpage and identifies all the URLs present on that page. These URLs can be found in anchor tags (<a> tags) or other relevant HTML elements.

4. Filter and enqueue new URLs: The crawler filters the extracted URLs based on specific criteria to determine which ones should be followed and crawled. Common filters include checking for domain restrictions, excluding certain file types (e.g., images, videos), or excluding URLs that have already been visited. The filtered URLs are then added to a queue for future crawling.

5. Fetch the next URL from the queue: The crawler dequeues the next URL from the queue and repeats steps 2 to 4 for this new URL. This process continues until the queue becomes empty or a predefined termination condition is met (e.g., a maximum number of pages to crawl).

6. Store or process the crawled data: As the crawler visits each URL, it may extract and store relevant data from the webpages. This data can include text, images, metadata, or other information of interest. The storage mechanism can vary depending on the specific requirements, such as storing the data in a database or writing it to files.

7. Repeat the crawling process: After processing a URL, the crawler moves on to the next URL in the queue and continues the crawling process until all URLs in the queue have been visited or the termination condition is met.

8. Respect website policies and etiquette: To ensure ethical crawling, the crawler must respect website policies such as robots.txt files, which specify rules for web crawlers. The crawler should also follow good etiquette by spacing out requests, avoiding excessive load on websites, and adhering to any other guidelines set by the website owners.

Question 2: Web Crawling Challenge! Creating a simple Web Crawler

[2]:

```python
import requests
from bs4 import BeautifulSoup
from xlwt import *

# Create a new workbook and table
workbook = Workbook(encoding='utf-8')
table = workbook.add_sheet('LOL Champions Statistic Table')

# Write headers to the table
table.write(0, 0, 'Champions')
table.write(0, 1, 'HP')
table.write(0, 2, 'HP+')
table.write(0, 3, 'HP5')
table.write(0, 4, 'HP5+')
table.write(0, 5, 'MP')
table.write(0, 6, 'MP+')
table.write(0, 7, 'MP5')
table.write(0, 8, 'MP5+')
table.write(0, 9, 'AD')
table.write(0, 10, 'AD+')
table.write(0, 11, 'AS')
table.write(0, 12, 'AS+')
table.write(0, 13, 'AR')
table.write(0, 14, 'AR+')
table.write(0, 15, 'MR')
table.write(0, 16, 'MR+')
table.write(0, 17, 'MS')
table.write(0, 18, 'Range')

# Make a request to the URL
url = "https://leagueoflegends.fandom.com/wiki/List_of_champions/Base_statistics"
headers = {
    'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/114.0.0.0 Safari/537.36'
}
response = requests.get(url, headers=headers)

# Parse the HTML content
soup = BeautifulSoup(response.content, 'html.parser')

# Find the table rows
rows = soup.find_all('tr')

line = 1

# Iterate over the rows and extract data
for row in rows[1:]:
    cells = row.find_all('td')
    if len(cells) >= 10:
        Champions = cells[0].text.strip()
        HP = cells[1].text.strip()
        HPplus = cells[2].text.strip()
        HP5 = cells[3].text.strip()
        HP5plus = cells[4].text.strip()
        MP = cells[5].text.strip()
        MPplus = cells[6].text.strip()
        MP5 = cells[7].text.strip()
        MP5plus = cells[8].text.strip()
        AD = cells[9].text.strip()
        ADplus = cells[10].text.strip()
        AS = cells[11].text.strip()
        ASplus = cells[12].text.strip()
        AR = cells[13].text.strip()
        ARplus = cells[14].text.strip()
        MR = cells[15].text.strip()
        MRplus = cells[16].text.strip()
        MS = cells[17].text.strip()
        Range = cells[18].text.strip()


        # Write data to the table
        table.write(line, 0, Champions)
        table.write(line, 1, HP)
        table.write(line, 2, HPplus)
        table.write(line, 3, HP5)
        table.write(line, 4, HP5plus)
        table.write(line, 5, MP)
        table.write(line, 6, MPplus)
        table.write(line, 7, MP5)
        table.write(line, 8, MP5plus)
        table.write(line, 9, AD)
        table.write(line, 10, ADplus)
        table.write(line, 11, AS)
        table.write(line, 12, ASplus)
        table.write(line, 13, AR)
        table.write(line, 14, ARplus)
        table.write(line, 15, MR)
        table.write(line, 16, MRplus)
        table.write(line, 17, MS)
        table.write(line, 18, Range)



        line += 1

# Save the workbook
workbook.save('C:\\Users\\user\\OneDrive\\Desktop\\Sem4 slide\\data␣
 ↪wrangling\\LOL_champion_tables.xls')
```