

Student Name : Ahmad Sharif

Student ID : K436765

Exercise 2.1: Data acquisition - Building a better web crawler.

The web crawler code shown on Lecture 2 is poor in at least two respects:

1. It can crawl the same page multiple times, if a link on a later crawled page points to the already-crawled page.
2. It inserts all links from each page in order as pages to be crawled. If some page contains thousands of links, the crawling will crawl those first and may never get to the links from the next page, especially if the total number of pages are limited.

To fix this duplicate issue, at first before insert into the list, I check if the url already exist list data-structure.

```
import requests
import bs4

webpage_url = "https://www.sis.uta.fi/~tojape/"
webpage_html = requests.get(webpage_url)
webpage_parsed_html = bs4.BeautifulSoup(webpage_html.content, 'html.parser')

def getpageurls(webpage_parsed):
    pagelinkelements=webpage_parsed.find_all('a')
    pageurls = [];
    for pagelink in pagelinkelements:
        pageurl_isok=1
        try:
            pageurl = pagelink['href']
        except:
            pageurl_isok=0
        if pageurl_isok == 1:
            if (pageurl.find('.pdf') !=-1)|(pageurl.find('.ps')!=-1):
                pageurl_isok = 0
            if (pageurl.find('http') ==-1)|(pageurl.find('.fi')==-1):
                pageurl_isok = 0
            if pageurl_isok == 1 and pageurl not in pageurls: # Before Append we need to check
                pageurls.append(pageurl)
    return(pageurls)

mywebpage_urls = getpageurls(webpage_parsed_html)
print(mywebpage_urls)
```

```
In [2]: import requests
import bs4
import urllib.request
```

```
In [3]: webpage_url = "https://www.gutenberg.org/browse/scores/top#books-last30"
webpage_html = requests.get(webpage_url)
webpage_parsed_html = bs4.BeautifulSoup(webpage_html.content, 'html.parser')
```

## 2> a

```
In [6]: pageList = webpage_parsed_html.find("h2", {"id": "books-last30"})
ol = pageList.find_next_sibling("ol")
book_list = {}

def collect_all_download_link(count):
    for x in ol.findAll('li'):
        # print(x.a.text, '19')
        url = x.a['href']
        link = 'https://www.gutenberg.org/files/' + url.split('/')[2]
        if(len(book_list) < count):
            book_list[x.a.text] = {
                "book_name": x.a.text,
                "download_link": link,
            }

collect_all_download_link(20)
print(len(book_list))
```

20

## 2 > b Name And Link

```
In [7]: for key in book_list:
        print("Book Name", key)
```

Book Name A Room with a View by E. M. Forster (98953)  
Book Name Middlemarch by George Eliot (93808)  
Book Name Little Women; Or, Meg, Jo, Beth, and Amy by Louisa May Alcott (85289)  
Book Name The Enchanted April by Elizabeth Von Arnim (84452)  
Book Name The Blue Castle: a novel by L. M. Montgomery (81894)  
Book Name Cranford by Elizabeth Cleghorn Gaskell (79784)  
Book Name Pride and Prejudice by Jane Austen (57171)  
Book Name Frankenstein; Or, The Modern Prometheus by Mary Wollstonecraft Shelley (53537)  
Book Name The Yellow Wallpaper by Charlotte Perkins Gilman (27545)  
Book Name The Adventures of Sherlock Holmes by Arthur Conan Doyle (27406)  
Book Name Dracula by Bram Stoker (27090)  
Book Name Alice's Adventures in Wonderland by Lewis Carroll (26824)  
Book Name The Scarlet Letter by Nathaniel Hawthorne (24241)  
Book Name Moby Dick; Or, The Whale by Herman Melville (23678)  
Book Name The Picture of Dorian Gray by Oscar Wilde (22076)  
Book Name A Modest Proposal by Jonathan Swift (21431)  
Book Name A Tale of Two Cities by Charles Dickens (21015)  
Book Name The Strange Case of Dr. Jekyll and Mr. Hyde by Robert Louis Stevenson (18230)  
Book Name The Great Gatsby by F. Scott Fitzgerald (17693)  
Book Name A Doll's House : a play by Henrik Ibsen (17018)

In [ ]: