Student ID: b0928024

Name:莊靜修

```
import requests
import re
from bs4 import BeautifulSoup

URL = "https://movies.yahoo.com.tw/movieinfo_main/"


class MovieCrawler(object):
    def __init__(self):
        self.movies = []


    def get_movies(self, page_url):
        page = 0
        for i in range(1, 15059):
            if i % 100 == 0: print(f'running {i}')
            url = URL + str(i)
            try:
                resp = requests.get(url)
            except:
                resp = None


            if resp and resp.status_code == 200:
                soup = BeautifulSoup(resp.text, 'html.parser')
                m = soup.find("div", class_ = "movie_intro_info_r")
                if m is None:
                    continue

                intro = soup.select_one("div.gray_infobox_inner span")

                temp = {}

                temp["doc_id"] = page
                page += 1

                cname = m.select_one("h1")
                temp["cname"] = cname.text.replace('\n', '').replace(' ', '')

                ename = m.select_one("h3")
                temp["ename"] = ename.text
                label = m.select('div.level_name a')
                ll = []
                for l in label:
                    t = l.text
                    t = t.replace('\n', '').replace(' ', '')
                    ll.append(t)
                temp["label"] = ll

                released_date = m.select_one('div').find_next_sibling('span')

                temp["released_date"] = released_date.text[5: ]

                intro_text = intro.text
                if intro_text and len(intro_text) > 0:
                    intro_text = intro_text.replace('\n', '').replace(' ', '')
                temp["intro"] = intro_text


                links = m.select("a")
                lin = []
                for l in links:
                    lin.append(l["href"])
                temp["links"] = lin


            self.movies.append(temp)

        return self.movies

crawler = MovieCrawler()
movies = crawler.get_movies(URL)


print(len(movies))
print(*movies, sep="\n")
```

```
running 100
running 200
running 300
running 400
running 500
running 600
running 700
running 800
running 900
running 1000
running 1100
running 1200
running 1300
running 1400
running 1500
running 1600
running 1700
running 1800
running 1900
running 2000
running 2100
running 2200
running 2300
running 2400
running 2500
running 2600
running 2700
running 2800
running 2900
running 3000
running 3100
running 3200
running 3300
running 3400
running 3500
running 3600
running 3700
running 3800
running 3900
running 4000
running 4100
running 4200
running 4300
running 4400
running 4500
running 4600
running 4700
running 4800
running 4900
running 5000
running 5100
running 5200
running 5300
running 5400
running 5500
running 5600
running 5700
running 5800
running 5900
running 6000
```

```
edges = []
for m in movies:
    for l in m["links"]:
        edges.append((m["doc_id"], l))
edges
```

```
[(0, 'https://movies.yahoo.com.tw/moviegenre_result.html?genre_id=5'),
 (0, 'https://movies.yahoo.com.tw/moviegenre_result.html?genre_id=6'),
 (0, 'https://movies.yahoo.com.tw/moviegenre_result.html?genre_id=15'),
 (0,

'https://movies.yahoo.com.tw/name_main/%E5%BC%B7%E5%B0%BC%E6%88%B4%E6%99%AE
johnny-depp-739'),
 (0,

'https://movies.yahoo.com.tw/name_main/%E6%BD%98%E5%A6%AE%E6%B4%9B%E6%99%AE
penelope-cruz-956'),
 (0, 'http://www.getsomeblow.com/index2.html'),
 (1, 'https://movies.yahoo.com.tw/moviegenre_result.html?genre_id=1'),
 (1, 'https://movies.yahoo.com.tw/moviegenre_result.html?genre_id=5'),
 (1, 'https://movies.yahoo.com.tw/moviegenre_result.html?genre_id=6'),
 (1, 'https://movies.yahoo.com.tw/moviegenre_result.html?genre_id=8'),
 (1,
  'https://movies.yahoo.com.tw/name_main/%E9%A6%AE%E8%BF%AA%E7%B4%A2-vin-
diesel-1217'),
 (1,

'https://movies.yahoo.com.tw/name_main/%E8%9C%9C%E9%9B%AA%E5%85%92%E7%BE%85
michelle-rodriguez-940'),
 (1,

'https://movies.yahoo.com.tw/name_main/%E4%BF%9D%E7%BE%85%E6%B2%83%E5%85%8B
paul-walker-1295'),
 (1, 'http://www.thefastandthefurious.com/'),
 (2, 'https://movies.yahoo.com.tw/moviegenre_result.html?genre_id=1'),
 (2, 'https://movies.yahoo.com.tw/moviegenre_result.html?genre_id=6'),
 (2, 'https://movies.yahoo.com.tw/moviegenre_result.html?genre_id=8'),
 (2, 'https://movies.yahoo.com.tw/moviegenre_result.html?genre_id=13'),
 (3, 'https://movies.yahoo.com.tw/moviegenre_result.html?genre_id=1'),
 (3, 'https://movies.yahoo.com.tw/moviegenre_result.html?genre_id=2'),
 (3, 'https://movies.yahoo.com.tw/moviegenre_result.html?genre_id=9'),
 (3,

'https://movies.yahoo.com.tw/name_main/%E5%B8%8C%E6%96%AF%E8%90%8A%E5%82%91
heath-ledger-97'),
 (3,

'https://movies.yahoo.com.tw/name_main/%E4%BF%9D%E7%BE%85%E8%B2%9D%E7%89%B9
paul-bettany-793'),
 (3, 'http://www.aknightstale.com/'),
 (4, 'https://movies.yahoo.com.tw/moviegenre_result.html?genre_id=9'),
 (4,

'https://movies.yahoo.com.tw/name_main/%E7%91%9E%E7%B5%B2%E8%96%87%E6%96%AF
reese-witherspoon-1050'),
 (4,

'https://movies.yahoo.com.tw/name_main/%E8%8E%8E%E7%91%AA%E5%B8%83%E8%90%8A
selma-blair-120'),
 (4, 'http://www.mgm.com/legallyblonde/'),
 (5, 'https://movies.yahoo.com.tw/moviegenre_result.html?genre_id=2'),
 (5, 'https://movies.yahoo.com.tw/moviegenre_result.html?genre_id=9'),
 (5, 'http://www.ratracemovie.com/'),
 (6, 'https://movies.yahoo.com.tw/moviegenre_result.html?genre_id=5'),
 (6, 'https://movies.yahoo.com.tw/moviegenre_result.html?genre_id=6'),
 (6, 'https://movies.yahoo.com.tw/moviegenre_result.html?genre_id=8'),
```

```
import networkx as nx

G = nx.DiGraph()
for e in edges:
    G.add_edge(e[0], e[1])

pagerank_list = nx.pagerank(G, alpha = 1)


for m in movies:
    tmp = {}
    if (m["doc_id"] in pagerank_list):
        m["pagerank"] = pagerank_list[m["doc_id"]]
    else:
        m["pagerank"] = 0

movies
```

```
[{'doc_id': 0,
  'cname': '一世狂野',
  'ename': 'Blow',
  'label': ['劇情', '犯罪', '歷史/傳記'],
  'released_date': '2001-10-12',
  'intro': '喬治戎格一生都在追求所謂的美國夢，也就是享受美好富裕的生活，但是他卻不願像
他父親那樣一輩子都只是個出賣勞力的建築工人。於是他搬到陽光明媚的加州，靠著販賣大麻賺錢，起
初，他販毒只是為了享受自由自在的生活，但是當他野心越來越大，他的勢力也日益坐大之際，卻在此
時被捕入獄。他在牢裡認識一個能言善道，自稱熟識哥倫比亞販毒集團的牢友狄亞哥，他出獄後果真把
當時勢力最大的毒梟艾斯科巴介紹給喬治認識，艾斯科巴計畫將古柯鹼大量引進美國的迪斯可舞廳，希
望能引領一股吸毒狂歡的風潮。除了毒品供應商之外，狄亞哥也介紹了一個美艷又狂野的女人瑪莎給喬
治，他們瘋狂相愛，之後馬莎還替他生下一個可愛的女兒克莉絲汀娜，也是喬治一生的最愛。喬治很快
就靠著販毒發大財，他還得買一棟大房子專門存放每天賺進來的大把鈔票，但是日進斗金卻整天提心吊
膽的生活卻讓喬治開始省思，到底他要繼續過著揮霍富裕的生活，還是為了自己心愛的女兒應該轉性投
資正當的事業？可是這時聯邦調查局的探員，也開始盯上毒源禍首的喬治……',
  'links': ['https://movies.yahoo.com.tw/moviegenre_result.html?
genre_id=5',
   'https://movies.yahoo.com.tw/moviegenre_result.html?genre_id=6',
   'https://movies.yahoo.com.tw/moviegenre_result.html?genre_id=15',

'https://movies.yahoo.com.tw/name_main/%E5%BC%B7%E5%B0%BC%E6%88%B4%E6%99%AE
johnny-depp-739',

'https://movies.yahoo.com.tw/name_main/%E6%BD%98%E5%A6%AE%E6%B4%9B%E6%99%AE
penelope-cruz-956',
   'http://www.getsomeblow.com/index2.html'],
  'pagerank': 3.035844391863196e-05},
 {'doc_id': 1,
  'cname': '玩命關頭',
   'cname': '玩命關頭',
   'ename': 'The Fast and the Furious',
   'label': ['動作', '劇情', '犯罪', '懸疑/驚悚'],
   'released_date': '2001-10-13',
   'intro': '唐米尼杜洛托是洛城街頭賽車界的老大哥，他身邊有一群忠心耿耿的手下，他白天忙
著組裝高性能跑車，晚上則是開著他的愛車，動輒以一次一萬美元的賭注和別人較車。布萊恩也渴望接
受極速的挑戰，他對自己的駕駛技術很有信心，但是在旁觀者的眼中他只是一個菜鳥，他開了一輛超炫
的跑車想和唐老大一較高下，也希望得到他的青睞，當比賽結束，布萊恩輸得一塌塗地之後，警方接獲
風聲前來取締，布萊恩在無意間從一名心狠手辣的幫派份子強尼手中救了唐老大一命，於是他就被納入
唐老大的權力核心，唐老大的妹妹蜜雅也對布萊恩產生好感，但是他們都不知道布萊恩其實是一名臥底
警探。布萊恩滲入賽車圈的目的是調查一連串的卡車搶案，嫌犯都是開著跑車的蒙面人，警方和聯邦調
查局希望能儘早逮到搶匪，以免卡車司機採取激烈的手段對這些搶匪進行報復行動，其中最有嫌疑的就
是唐老大和強尼。正當唐老大和強尼形成水火不相容的情勢。布萊恩和唐老大兄妹的關係卻越來越深，
他不但和唐老大結為好友，更忍不住對蜜雅產生好感，但是他也同時承受來自警方和ＦＢＩ的壓力，必
須儘快查出誰才是真正的搶匪，他在天人交戰之際，在法律和友情之間，必須做出困難的決定。',
   'links': ['https://movies.yahoo.com.tw/moviegenre_result.html?
genre_id=1',
    'https://movies.yahoo.com.tw/moviegenre_result.html?genre_id=5',
    'https://movies.yahoo.com.tw/moviegenre_result.html?genre_id=6',
    'https://movies.yahoo.com.tw/moviegenre_result.html?genre_id=8',
    'https://movies.yahoo.com.tw/name_main/%E9%A6%AE%E8%BF%AA%E7%B4%A2-vin-
diesel-1217',

 'https://movies.yahoo.com.tw/name_main/%E8%9C%9C%E9%9B%AA%E5%85%92%E7%BE%85
michelle-rodriguez-940',

 'https://movies.yahoo.com.tw/name_main/%E4%BF%9D%E7%BE%85%E6%B2%83%E5%85%8B
paul-walker-1295',
    'http://www.thefastandthefurious.com/'],
   'pagerank': 3.035844391863196e-05},
  {'doc_id': 2,
   'cname': '戰雷密佈'
```

```
import jieba
import jieba.posseg as pseg

def tokenize_words(contents):
    #Remove punctuation
    punc = '''，。...！！；（）()-[]{};:：'"\, <>./?@#$%^&*_~？——……【】\n「」　'''

    for s in contents:
        if s in punc:
            contents = contents.replace(s, "")

    words = jieba.cut(contents)
    return words
```

```
from nltk.tokenize import word_tokenize
import nltk
from nltk.corpus import stopwords
nltk.download('stopwords')

without_sw = []
for m in movies:
    text_tokens = tokenize_words(m["cname"] + m["ename"] + m["intro"])

    without_sw.append([
        word for word in text_tokens if not word in stopwords.words('chinese')])
print(without_sw)
```

```
    [nltk_data] Downloading package stopwords to /Users/hsiu/nltk_data...
    [nltk_data]   Package stopwords is already up-to-date!
    Building prefix dict from the default dictionary ...
    Loading model from cache /var/folders/8q/s5rn_lld0qj8zg63cy8t3xtw0000gn/T/j
    Loading model cost 0.169 seconds.
    Prefix dict has been built succesfully.
    IOPub data rate exceeded.
    The notebook server will temporarily stop sending output
    to the client in order to avoid crashing it.
    To change this limit, set the config variable
    `--NotebookApp.iopub_data_rate_limit`.

    Current values:
    NotebookApp.iopub_data_rate_limit=1000000.0 (bytes/sec)
    NotebookApp.rate_limit_window=3.0 (secs)
```

```
import collections
#inverted_index
index = collections.defaultdict(list)
for i, words in enumerate(without_sw):
    for w in set(words):
        index[w].append(i)
index
```

```
    defaultdict(list,
                {'生下': [0,
                  419,
                  439,
                  665,
                  770,
                  845,
                  869,
                  1015,
                  1059,
                  1077,
                  1218,
                  1306,
                  1355,
                  1378,
                  1390,
                  1477,
                  1537,
                  1624,
                  1766,
                  1850,
                  2131,
                  2153,
                  2303,
                  2403,
                  2483,
                  2577,
                  2581,
                  2588,
                  2643,
                  2687,
                  2743,
                  2767,
                  2768,
                  2788,
                  2804,
                  2915,
                  2921,
                  2928,
                  2943,
                  3063,
                  3114,
                  3280,
                  3403,
                  3411,
                  3786,
                  3790,
                  3925,
                  4188,
                  4238,
                  4264,
                  4286,
                  4299,
                  4340,
                  4345,
                  4362,
                  4397,
                  4504,
                  4594,
                  5229
```

```
import re

def query(keyword):
    print(f'您的搜尋結果 (Sorting by PageRank Value):共{len(index[keyword])}筆，符合
    if len(index[keyword]) != 0:
        record = []
        for i in index[keyword]:
            record.append(movies[i])
        record = sorted(record, key = lambda x: x["pagerank"], reverse = True)
        for r in record:
            print(f'{r["doc_id"]}({r["pagerank"]}): {r["cname"]} {r["ename"]} {r

        real = pred = 0
        for movie in movies:
            if (re.search(rf"{keyword}", movie["cname"] + movie["ename"] + movie
                real += 1
        for i in index[keyword]:
            if (re.search(rf"{keyword}", movies[i]["cname"] + movies[i]["ename"]
                pred += 1

        print(f'您的搜尋結果 (Sorting by PageRank Value):共{len(index[keyword])}筆
        print(f"\nPrecision = {pred / len(index[keyword]) * 100}%")
        print(f"Recall = {pred / real * 100}%")
```

```
query("可愛")
```

```
您的搜尋結果 (Sorting by PageRank Value):共422筆，符合 "可愛" - - - 共 indexing
0(3.035844391863196e-05): 一世狂野 Blow 喬治戎格一生都在追求所謂的美國夢，也就是享受

12(3.035844391863196e-05): 美國派2 American Pie 2 1999年暑假，全球影迷都愛上【美

104(3.035844391863196e-05): 甜姐不辣 The Sweetest Thing 克莉絲汀娜（卡麥隆迪亞茲

137(3.035844391863196e-05): 驚婚計 Birthday Girl 約翰（班查普林飾）是個溫柔、孤獨

204(3.035844391863196e-05): 一家之鼠2 Stuart Little 2 在第一集的故事中，一個人類

213(3.035844391863196e-05): 當女人真好 About Adam 一個普通的都柏林夜晚，歐露西（凱

233(3.035844391863196e-05): YaYa私密日記 The Divine Secrets of Ya-Ya sisterh

385(3.035844391863196e-05): 真情假愛 Intolerable Cruelty 法律永遠是伸張正義的舞

399(3.035844391863196e-05): 遠離天堂 Far from Heaven 1957年的秋天，凱西(茱莉安摩

598(3.035844391863196e-05): 麻雀變王妃 The Prince & Me 在美國威斯康辛州的大學校園

609(3.035844391863196e-05): 加菲貓 Garfield 老姜帶著加菲貓去看獸醫，獸醫甜心麗莎（

660(3.035844391863196e-05): 送信到哥本哈根 I Am David 這是一趟充滿希望的人生之旅，
```

```
660(3.035844391863196e-05): 送信到哥本哈根 I Am David 這是一趟充滿希望的人生之旅…
666(3.035844391863196e-05): 再見了，可魯 Quill 「他已經明白今後要與這個人一起生活下
669(3.035844391863196e-05): 超完美嬌妻 The Stepford Wives 瓊安艾柏哈以為她已經達
695(3.035844391863196e-05): 手札情緣 The Notebook 「初戀，永遠讓人刻骨銘心」——這
706(3.035844391863196e-05): BJ單身日記:男人禍水 Bridget Jones: The Edge of Rea
762(3.035844391863196e-05): 五歲的心願 Oseam 【五歲的心願】（OSEAM）電影改編自韓國
783(3.035844391863196e-05): 沙仙活地魔 Five Children and it 【沙仙--活地魔】改編
827(3.035844391863196e-05): 山村猶有讀書聲 To Be and to Have 這部片長105分鐘的法
840(3.035844391863196e-05): 現在，很想見你. Be With You 美麗動人的澪（竹內結子飾）
869(3.035844391863196e-05): 企鵝寶貝 The Emperor's Journey 一段關於勇氣與冒險的旅
880(3.035844391863196e-05): 無米樂 Let it be 75歲的崑濱伯每天早晨第一件事，就是三
908(3.035844391863196e-05): 純真11歲 Innocent Voices 【永遠11歲的我們】一部教人笑
925(3.035844391863196e-05): 永不遺忘的美麗 Yesterday 她叫昨天…一個身患重病的母親她
939(3.035844391863196e-05): 紅孩兒：決戰火燄山 Fire Ball 台灣史上第一部數位動畫短片
945(3.035844391863196e-05): 四眼天雞 Chicken Little 尺寸大小不重要，迪士尼個子最小
947(3.035844391863196e-05): 我的小牛與總統 The Cow and the President 小男孩盧卡
974(3.035844391863196e-05): 狗狗心事 All About My Dog 11段狗狗心事：1.狗狗的一生
983(3.035844391863196e-05): 狗仔隊 Paparazzi 對聲勢看漲的動作巨星阿波（柯爾豪瑟飾）
988(3.035844391863196e-05): 胡桃鉗 The Nutcracker and The Mouse King 俄國沙皇
```

```python
import json

mov = movies.copy()


with open('hw2.json', 'w', encoding = 'utf-8') as f:
    f.write(json.dumps(mov, indent = 4))
```