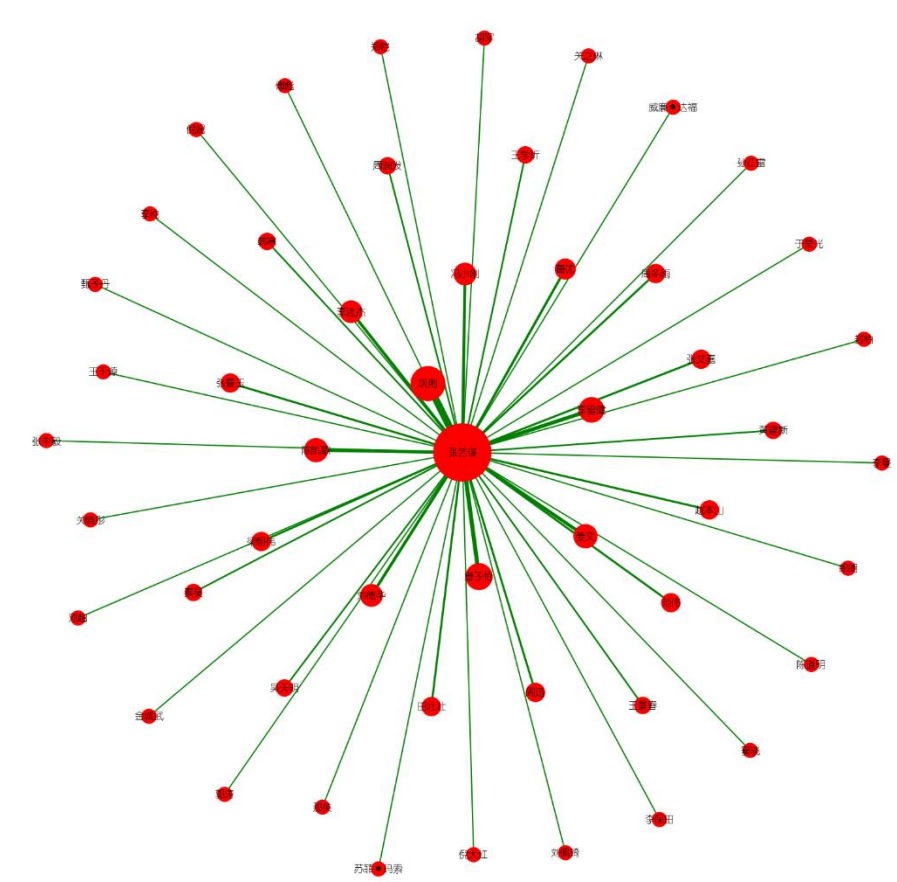
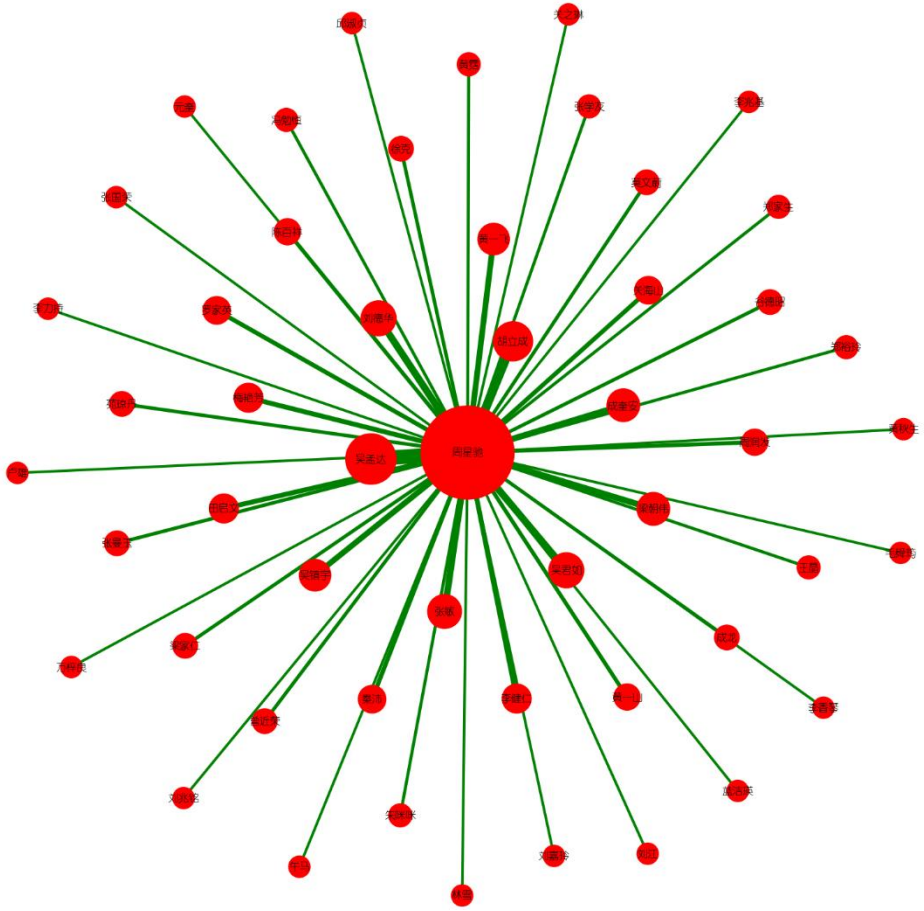
1. 从豆瓣获取某电影人与其他电影人的合作关系图
2. 从B站获取弹幕，并用词云展示
3. 从链家获取某区域的二手房信息（比如回龙观）

作业1

从豆瓣获取某电影人与其他电影人的合作关系图





作业1

整体需求：建立演员（比如张国立）与其他演员的合作关系图

- 整体功能封装成一个总函数，函数输入是演员姓名，合作关系图片的名称：演员姓名.png
- 总函数里调用不同的功能函数：
  - 函数1：get_person_by_name(person_name)搜索指定演员信息
    - ✓ 输入：演员姓名；输出：演员id
  - 函数2： get_movies_by_person(person_id)搜索演员出演电影列表
    - ✓ 输入：演员id；输出：所有出演电影的id列表
  - 函数3： get_movie_info(movie_id) 得到各电影的演员列表
    - ✓ 输入：电影信息字典；输出：演员姓名列表
  - 统计指定演员与其他演员的合作次数，形成统计字典，供绘图包所调用。

作业1

查找演员网址：https://movie.douban.com/celebrities/search?search_text=张国立


```python
import requests
from bs4 import BeautifulSoup

def get_person_by_name(search_name):
    headers = {
        'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 '
                      '(KHTML, like Gecko) Chrome/85.0.4183.83 Safari/537.36'}
    url = "https://movie.douban.com/celebrities/search"
    response = requests.get(url, params={'search_text':search_name}, headers=headers)

    soup = BeautifulSoup(response.text, "html.parser")
    item_list = soup.select("#content .result .content h3 a")
    if len(item_list) == 0:
        return None
    a = item_list[0]
    href = a['href']
    id = int(href.split('celebrity/', 1)[1][:-1])
    print(href, id)
    return id

if __name__ == '__main__':
    search_name = '梅艳芳'
    get_person_by_name(search_name)
```

注意：如果姓名输入错误，就可能找不到信息，所以，要考虑异常处理！

作业1

查找电影网址：https://movie.douban.com/celebrity/1015115/movies


```python
import requests
from bs4 import BeautifulSoup
import time
def get_movies_by_person(person_id):
    url_ora = f"https://movie.douban.com/celebrity/{person_id}/movies"
    headers = {
        'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) '
                      'AppleWebKit/537.36 (KHTML, like Gecko) Chrome/85.0.4183.83 Safari/537.36'}
    url,movies_info = url_ora, []
    while True:
        response = requests.get(url, headers=headers)
        print(url)
        soup = BeautifulSoup(response.text, 'html.parser')
        movie_list = soup.select(".grid_view ul li a.nbg")
        for movie in movie_list:
            href = movie['href']
            id = int(href.split('/subject/')[1][:-1])
            print(href, id)
            movies_info.append(id)
        next_btn_list = soup.select(".paginator .next a")
        if len(next_btn_list) == 0:
            break
        url = url_ora + next_btn_list[0]['href']
        time.sleep(1.0)
    return movies_info

if __name__ == '__main__':
    get_movies_by_person(1047976)
```

注意：分页爬虫！（通过找“后页”超链接来实现）

作业1

查找电影的演员信息：https://movie.douban.com/subject/3901418/celebrities


```python
def get_movie_info(movie_id):
    url_ora = f'https://movie.douban.com/subject/{movie_id}/celebrities'
    headers = {
        'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 '
                      '(KHTML, like Gecko) Chrome/85.0.4183.83 Safari/537.36'}
    url = url_ora
    print(url)
    persons = []
    time.sleep(1.0)
    response = requests.get(url, headers=headers)
    soup = BeautifulSoup(response.text, 'html.parser')
    person_header_list = soup.select(".article .list-wrapper h2")
    for person_header in person_header_list:
        if "演员" in person_header.text.strip():
            person_list = person_header.parent.select("ul.celebrities-list  li>a")
            for person in person_list:
                info = person['title'].split(' ', 1)[0].strip()
                if info:
                    print(info)
                    persons.append(info)
            break
    return persons
if __name__ == '__main__':
    persons = get_movie_info(27163278)
    for person in persons:
        print(person)
```

注意：只要演员，不要导演等其他人物！

作业1

几个函数组合在一起，形成了main()函数

```python
def get_person_list(search_name):
    person_info = get_person_by_name(search_name)
    print(person_info)

    if person_info is None:
        print("用户名称不存在！")
        return None

    movies_list = get_movies_by_person(person_info)

    person_list = []
    for movie in movies_list:
        person_list += get_movie_info(movie)

    return person_list
```

```python
def main():
    person_name = input("请输入演员名称:>>").strip()
    person_list = get_person_list(person_name)
    if not person_list:
        exit()
    num_dict = {}
    for person in person_list:
        if person in num_dict:
            num_dict[person] += 1
        else:
            num_dict[person] = 1
    draw_graph.main(person_name, num_dict)

import draw_graph
if __name__ == '__main__':
    main()
```

作业2

从B站获取弹幕，并用词云展示

作业2

整体需求：从B站获取弹幕，并用词云展示

- 整体功能封装成一个总函数，函数输入是B站视频的ID，输出是ID对应的词云图片
- 总函数里调用不同的功能函数：
  - 函数1：get_dmlist_by_movie(movie_id)，根据movie_id，得到弹幕列表
    - ✓ 输入： movie_id；输出：所有弹幕组成的list
  - 函数2： get_keyword_by_jieba(dmlist)，搜索演员出演电影列表
    - ✓ 输入：所有弹幕组成的list ；输出：调用jieba，得到的keyword （以list 形式）
  - 函数3： get_cloud_by_keyword(movie_id, kw_list)，生成词云的图片
    - ✓ 输入：图片名称、keyword_list；输出：{movie_id}.png

作业2

根据movie_id，得到弹幕列表：

https://api.bilibili.com/x/web-interface/view/detail?bvid=**BV1ha41197jq**

```
headers = {'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) '
                         'AppleWebKit/537.36 (KHTML, like Gecko) '
                         'Chrome/116.0.0.0 Safari/537.36 Edg/116.0.1938.69'}

def get_dm_list_by_url(dm_url_list):
    time.sleep(1.0)
    dm_list = []
    for cid in dm_url_list:
        url = f"https://api.bilibili.com/x/v1/dm/list.so?oid={cid}"
        response = requests.get(url)
        response.encoding = 'utf8'
        soup = BeautifulSoup(response.text, "xml")
        for dm in soup.select("d"):
            dm_list.append(dm.text.strip())
        time.sleep(1.0)
    print(dm_list)
```

https://api.bilibili.com/x/v1/dm/list.so?oid=1256186972

```
def get_dm_list_by_movie_id(movie_id):
    url = f"https://api.bilibili.com/x/web-interface/view/detail?bvid={movie_id}"
    response = requests.get(url=url, headers=headers)
    data = json.loads(response.text)
    dm_url_list = []
    pages = data["data"]["View"]['pages']
    for page in pages:
        dm_url_list.append(page['cid'])
    print(dm_url_list)
    return get_dm_list_by_url(dm_url_list)

if __name__ == "__main__":
    get_dm_list_by_movie_id("BV1Z8411q7gy")
```

作业2

根据dm_list，调用jieba，得到关键词：

```
import jieba.analyse
import jieba
def get_keyword_by_dm(dm_list):
    txt = ''.join(dm_list)
    kw_list = jieba.analyse.extract_tags(txt, topK=100)
#     kw_list = jieba.lcut(txt)
#     kw_list = jieba.cut_for_search(txt)
    print(kw_list)
    return kw_list
```

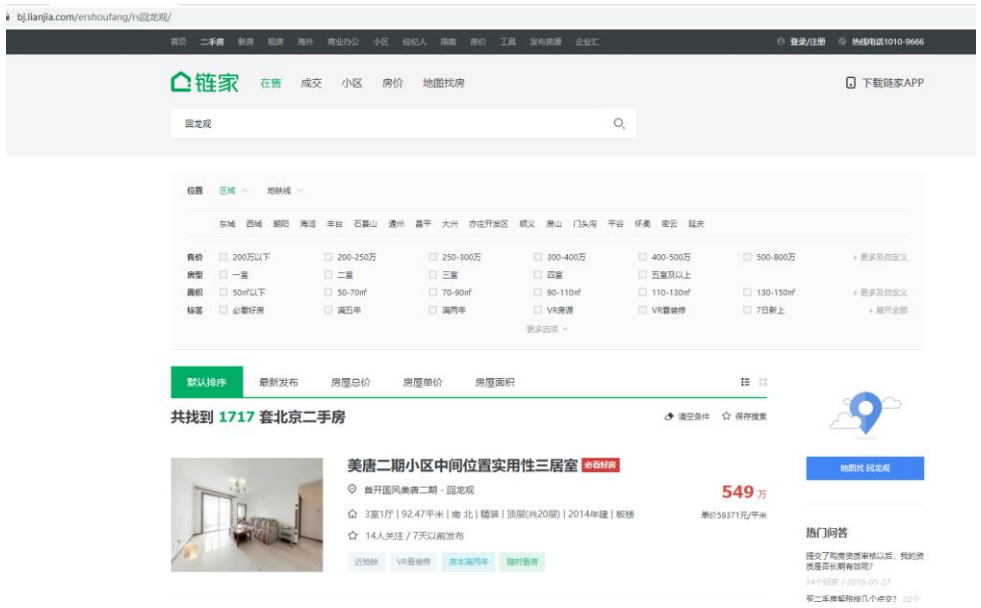
作业2

调用wordcloud，生成词云图片

```python
import wordcloud
def gen_word_cloud(kw_list, png_name):
    string = ' '.join(kw_list)
    print(string)
    wc = wordcloud.WordCloud(
        width=1000,
        height=700,
        background_color='white',
        font_path='msyh.ttc',  # 字体为微软雅黑，Windows 系统使用这个字体
        scale=15,
        contour_width=5,  # 轮廓的宽度
        contour_color='red',  # 轮廓的颜色
    )
    wc.generate(string)
    wc.to_file(png_name)
```

作业3

从链家获取某区域的二手房信息（比如回龙观）



小区名称 户型 面积 总价 单价 楼层位置 建筑时间 建筑类型 发布时间 关注人数

https://bj.lianjia.com/ershoufang/rs回龙观/

作业3

selenium的使用（以chrome为例）

```python
import requests
import time
from bs4 import  BeautifulSoup
from selenium import webdriver
brower = webdriver.Chrome()
url = "https://bj.lianjia.com/ershoufang/rs回龙观/"
while True:
    time.sleep(1.0)
    brower.get(url)
    soup = BeautifulSoup(brower.page_source, "html.parser")

    # 获取所有的房子信息
    sell_list = soup.select("ul.sellListContent li div.info")
    for sell in sell_list:
        print(sell)

    # 获取下一页
    page_list = soup.select("div.house-lst-page-box a")
    for page in page_list:
        print(page.text, page['href'])
        if page.text.strip() == "下一页":
            url = "https://bj.lianjia.com" + page['href']
            break
    else:
        break
```