实验二实验报告

数据分析及实验　刘沛　PB22061259

任务概述

DBLP（主页：https://dblp.uni-trier.de）是计算机领域学术研究的一个英文文献集成数据库系统，在学术界有
很好的声誉。 用户可以在搜索栏输入关键词（如论文名称、作者名、会议名称）以获取相关文献的元数据（如
标题、作者、发表日期等）。

任务列表

1. （15%）进入 DBLP 主页，通过搜索功能，打开罗列 KDD 2023 所有会议文献的页面。
Q. 读取整个页面的 html 内容并解码为文本串（可使用 urllib.request 的相应方法），将其以 UTF-8 编 码格式写入 page.txt 文件，留待后续处理

实验原码如下

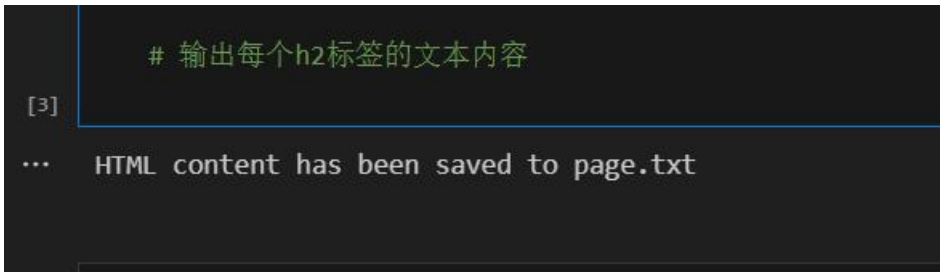
```
import urllib.request
import re

researchers_list = []

paper_list = []   #全局变量

def get_html_content(url):
    # 创建一个请求对象
    req = urllib.request.Request(url)

    # 发送请求并获取响应
    try:
        with urllib.request.urlopen(req) as response:
            # 读取响应内容
            html_bytes = response.read()
            # 根据响应的编码将字节解码为字符串
            html_content = html_bytes.decode(response.info().get_param('charset', 'utf-8'))
            return html_content
    except urllib.error.HTTPError as e:
        print(f"HTTP error occurred: {e.code}")
    except urllib.error.URLError as e:
```

```
        print(f"URL error occurred: {e.reason}")
    except Exception as e:
        print(f"An error occurred: {e}")
    return None
```

```
# 获取网页的 HTML 内容
url = 'https://dblp.org/db/conf/kdd/kdd2023.html'
html_content = get_html_content(url)
```

```
# 如果成功获取到 HTML 内容，将其写入到 page.txt 文件中
if html_content:
    with open('page.txt', 'w', encoding='utf-8') as file:
        file.write(html_content)
    print("HTML content has been saved to page.txt")
else:
    print("Failed to retrieve the HTML content.")
```

代码运行输出的结果如下图所示：



2. （15%）本页面展示了 KDD 2023 会议文献在不同 Track 下的论文收录情况。
Q. 打开 page.txt 文件，观察 Track 名称、论文标题等关键元素的组成规律。从这个文本串中提取各 Track 的名称并输出（可利用字符串类型的 split()和 strip()方法）。

代码实现：

```
def get_html_track(filename):

    pattern = r'<h2 id=".*?">(.*?)</h2>'   #设置模式串

    # 打开文件并读取内容
    with open(filename, 'r', encoding='utf-8') as file:

        html_content = file.read()

    matched_phrases = re.findall(pattern,html_content)   #在全篇里面去匹配模式串符合的
```

```
    for phrase in matched_phrases:
        print(phrase)                                    #输出打印
```

```
filename='page.txt'
get_html_track(filename)
```

输出结果如下：

```
Research Track Full Papers
Applied Data Track Full Papers
Hands On Tutorials
Lecture Style Tutorials
Workshop Summaries
```

3. （25%）可以看到，"Research Track Full Papers" 和 "Applied Data Track Full Papers" 中的论文占据了绝大多数，为更好地跟进数据挖掘领域学术前沿，现欲收集这两个 Track 下的论文信息。Q. 基于上述结果，输出这两个 Track 各自包含的论文数量。

实验代码：

```
def task3(filename):
    with open(filename, 'r', encoding='utf-8') as file:

        html_content = file.read()

    tracks =  html_content.split("<h2 id=")    #按 track 来划分

    num_flag=[]
    pattern = r'<span itemprop="pagination">(\d+)-(\d+)</span>'    #找到数字，页面的数字
```

```
    for index,track in enumerate(tracks):   #只需要前面两个 track
        if index == 1 or index  ==2 :
            flag=0
            match_trackname = re.search('>(.*?)</h2>',track)
            if match_trackname:
                trackname = match_trackname.group(1).strip()
            else :
                print("no found track")
            papers_list = []
#            print(track)
            papers = track.split(r'<cite class="data tts-content" itemprop="headline">')
            for dex,paper in enumerate(papers):
                if dex != 0:            #第一段不含所需要的论文信息
                    #title = re.findall(r'<span class="title"
itemprop="name">(.*?)</span>',paper)
                    flag+=1             #计算跑到的论文数量
                    names =re.findall(r'<span itemprop="name" title=".*?">(.*?)</span>',paper)
                    title = re.findall(r'<span class="title" itemprop="name">(.*?)</span>',paper)
```

```
                    match_page=re.search(pattern,paper)
                    if match_page:
```

```
                    startpage = match_page.group(1)
                    endpage = match_page.group(2)
```

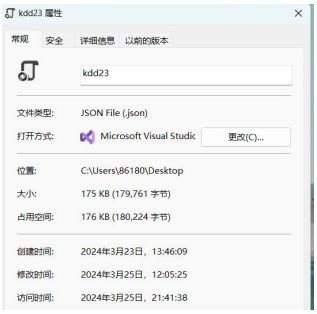
```
                papers_dict =
{"authors":names,"title":title[0],"startpage":startpage,"endpage":endpage}

                papers_list.append(papers_dict)
            num_flag.append(flag)
            paper_list.append({'track': trackname, 'papers':papers_list })
```

```
    for dict_item in paper_list:
       print(dict_item)
    for flag in num_flag:
        print(flag)
filename='page.txt'
task3(filename)
import json
file_path = 'C:\\Users\\86180\\Desktop\\kdd23.json'
with open(file_path, 'w') as f:
    json.dump(paper_list, f,indent = 2)
    print (1)
```

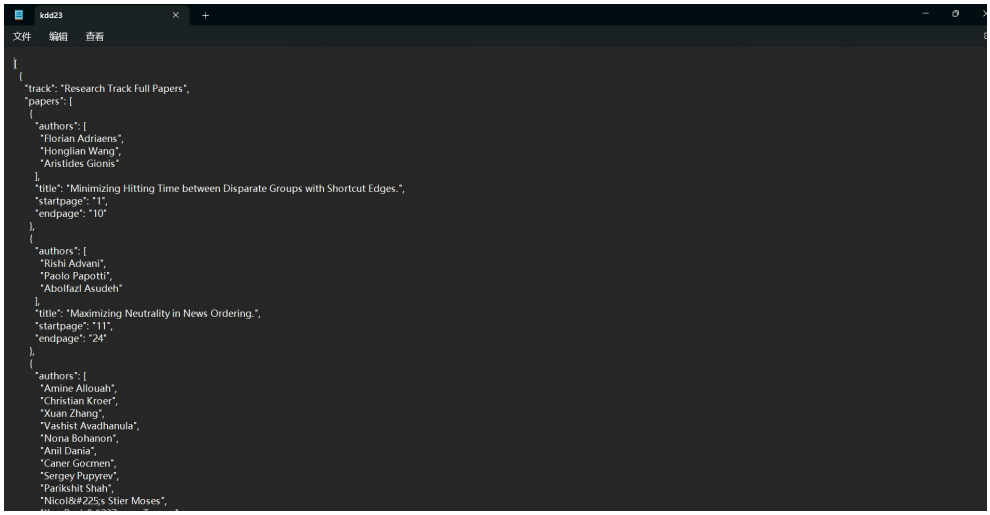
实验结果：

```
{'track': 'Research Track Full Papers', 'papers': [{'authors': ['Florian Adriaens', 'Honglian Wang', 'Aristides Gionis'], 'title': 'Minimizing Hitting
{'track': 'Applied Data Track Full Papers', 'papers': [{'authors': ['Abhinav Anand', 'Surender Kumar', 'Nandeesh Kumar', 'Samir Shah'], 'title': 'CADEN
313
183
```

我将我的.json 文件存在我的桌面上



在记事本中打开 KDD2023 的结果如下图所示


kdd23
文件 编辑 查看
[
{
"track": "Research Track Full Papers",
"papers": [
{
"authors": [
"Florian Adriaens",
"Honglian Wang",
"Aristides Gionis"
],
"title": "Minimizing Hitting Time between Disparate Groups with Shortcut Edges.",
"startpage": "1",
"endpage": "10"
},
{
"authors": [
"Rishi Advani",
"Paolo Papotti",
"Abolfazl Asudeh"
],
"title": "Maximizing Neutrality in News Ordering.",
"startpage": "11",
"endpage": "24"
},
{
"authors": [
"Amine Allouah",
"Christian Kroer",
"Xuan Zhang",
"Vashist Avadhanula",
"Nona Bohanon",
"Anil Dania",
"Caner Gocmen",
"Sergey Pupyrev",
"Parikshit Shah",
"Nicolás Stier Moses",

4. （35%）在论文作者条目中，作者姓名可超链接到其过往发表的论文列表页面，如第一篇论文第一作者的过往发表论文信息页面如下图所示。Q. 现要求基于之前爬取的页面文本，分别针对这两个 Track 前 10 篇论文的所有相关作者，仿照上述步骤爬取他们的以下信息：（1）该研究者的学术标识符 orcID；（2）该研究者从 2020 年至今发表的所有论文信息（包含作者 authors、标题 title、收录信息 publishInfo 和年份 year）。相应存储格式为：
请将最终结果转化为 json 对象，并以 2 字符缩进的方式写入 researchers.json 文件中。

实验代码如下：

```
def task4(filename):
    with open(filename, 'r', encoding='utf-8') as file:
        html_content = file.read()

    tracks =  html_content.split("<h2 id=")    #按 track 来划分
    for index,track in enumerate(tracks):   #只需要前面两个 track
            if index == 1 or index  ==2 :
                papers = track.split(r'<cite class="data tts-content" itemprop="headline">')
                for dex,paper in enumerate(papers):
                    if dex>=1 and  dex<=10:
                       links = re.findall(r'<a href="(.*?)" itemprop="url">',paper )
                      # for link in links:
                       #    print(link)
                       for link in links:
                           researcher_list = []
                           tmp= get_html_content(link)
                           if tmp :
                               #print("yes")
                              # content = tmp.split(r' <header id="the2010s" class="hide-head
h2">')
                               # 用于存储找到的 ORCID
                               orcid_pattern = r'\d{4}-\d{4}-\d{4}-\d{3}'
                               # 分割文章为行
                               lines = tmp.split('\n')

                               # 遍历每一行
                               researcher = re.findall(r'<title>dblp:
(.*?)</title>',tmp)   #researcher
                               for line in lines:
                                   # 检查是否包含图片标签
                                   if '<img alt=""
src="https://dblp.org/img/orcid.dark.16x16.png" class="icon">' in line:
                                       # 在该行中查找 ORCID 数字
                                       orcid_match = re.search(orcid_pattern, line)
                                       if orcid_match:
                                           orcID = orcid_match.group()            #orcID
```

```
                            articles = tmp.split(r'<cite class="data tts-content"
itemprop="headline">')
                            for n,article in enumerate(articles):
                                if n!=0:
                                    title = re.findall(r'<span class="title"
itemprop="name">(.*?)</span>',article)
                                    names =re.findall(r'<span itemprop="name"
title=".*?">(.*?)</span>',article)
```

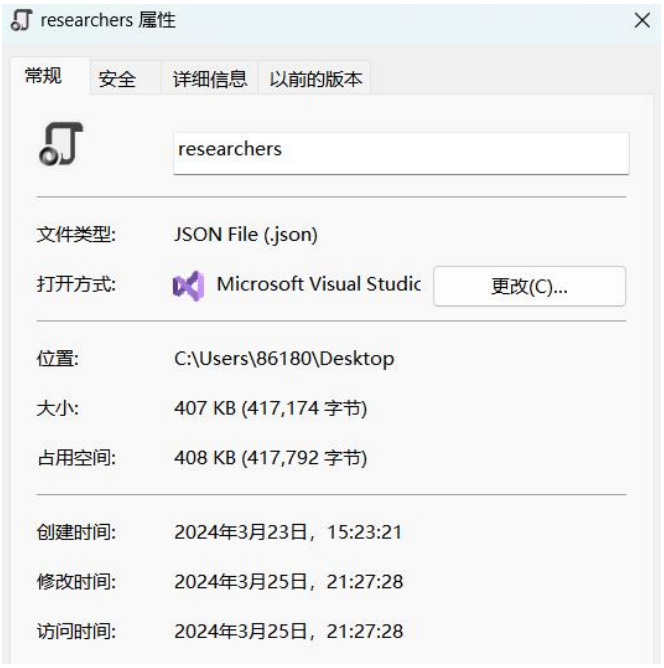
```
                                    if re.findall(r'<span
itemprop="name">(.*?)</span>',article) :
                                        Info1 = re.findall(r'<span
itemprop="name">(.*?)</span>',article)
                                        pubulishInfo = Info1[0]+' '
                                    if  re.findall(r'<span
itemprop="datePublished">(.*?)</span>',article):
                                        Info2  = re.findall(r'<span
itemprop="datePublished">(.*?)</span>',article)
                                        year = Info2[0]
                                        year_int = int(year)
                                        if year_int<2020:
                                            break
                                        pubulishInfo = pubulishInfo + Info2[0] + ':'
                                    if  re.findall(r'<span
itemprop="pagination">(.*?)</span>',article):
                                        Info3 = re.findall(r'<span
itemprop="pagination">(.*?)</span>',article)
                                        pubulishInfo =pubulishInfo+' '+ Info3[0]
                                       # print(Info3)
```

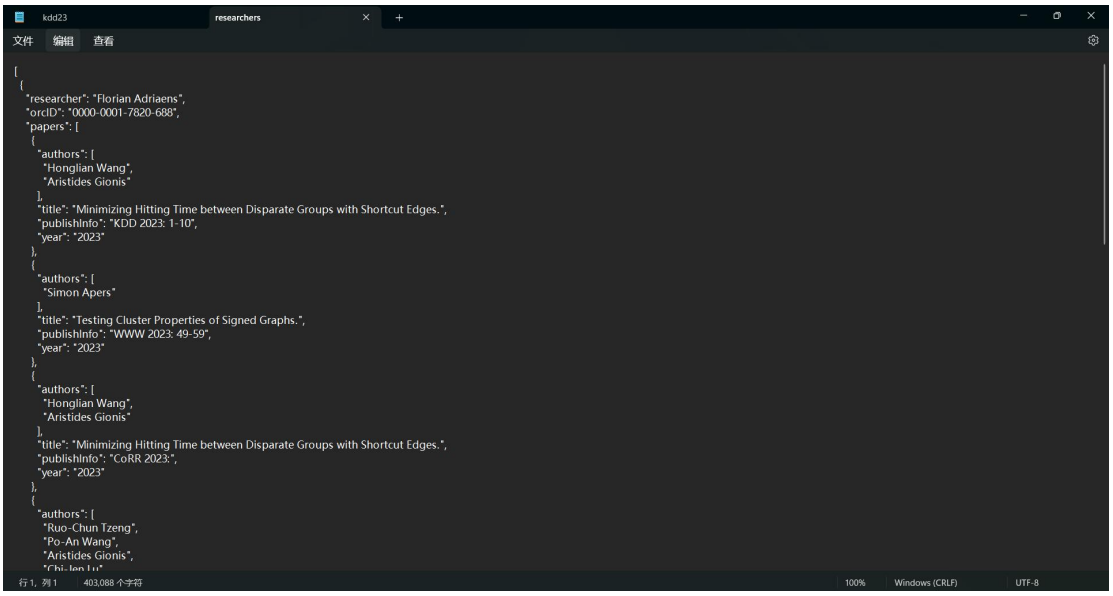
```
                                    papers_dict =
{"authors":names,"title":title[0],"publishInfo":pubulishInfo,"year":year}


                                    researcher_list.append(papers_dict)
                                researchers_list.append({"researcher":
researcher[0],"orcID":orcID,"papers":researcher_list})
                               # print({"researcher":
researcher[0],"orcID":orcID,"papers":researcher_list})
    for dict_item in researchers_list :
        print(dict_item)
```

```
import json
filename='page.txt'
task4(filename)
```

```
file_path2 = 'C:\\Users\\86180\\Desktop\\researchers.json'
with open(file_path2, 'w') as f:
    json.dump(researchers_list, f,indent = 2)
```

实验结果如下图所示：





总结

全部任务都顺利完成，过程艰难但不失乐趣，收获颇丰。