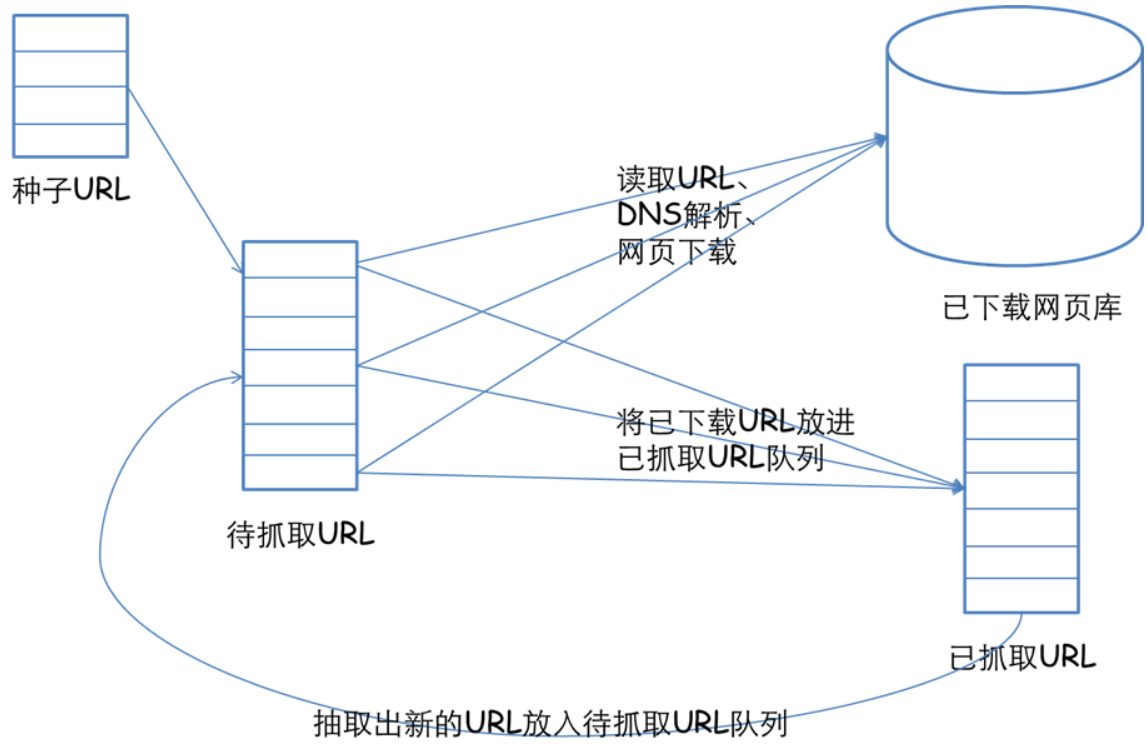
1551265 张伯阳

网络爬虫的基本原理

网络爬虫的基本结构

一个通用的网络爬虫的框架如图所示：



网络爬虫的工作流程

网络爬虫的基本工作流程如下：

1.首先选取一部分精心挑选的种子URL；

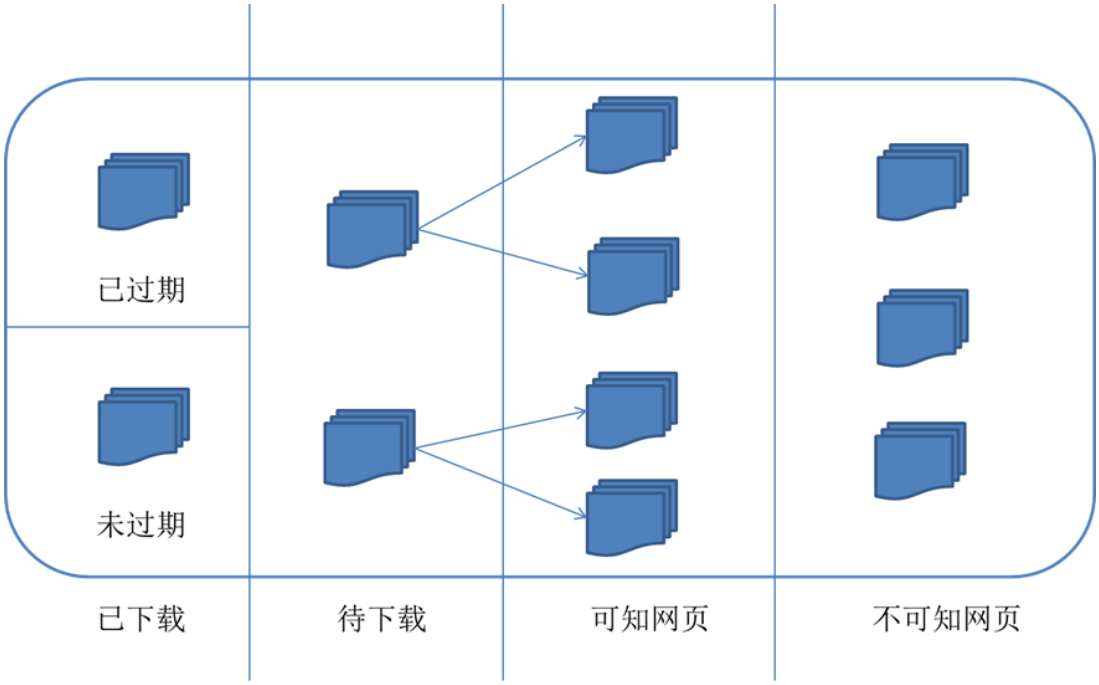
2.将这些URL放入待抓取URL队列；

3.从待抓取URL队列中取出待抓取在URL，解析DNS，并且得到主机的ip，并将URL对应的网页下载下来，存储进已下载网页库中。此外，将这些URL放进已抓取URL队列。

4.分析已抓取URL队列中的URL，分析其中的其他URL，并且将URL放入待抓取URL队列，从而进入下一个循环。

从爬虫角度看待互联网页的分类分别有几种?区分的标准是什么?

对应的，可以将互联网的所有页面分为五个部分：



1.已下载未过期网页

2.已下载已过期网页：抓取到的网页实际上是互联网内容的一个镜像与备份，互联网是动态变化的，一部分互联网上的内容已经发生了变化，这时，这部分抓取到的网页就已经过期了。

3.待下载网页：也就是待抓取URL队列中的那些页面

4.可知网页：还没有抓取下来，也没有在待抓取URL队列中，但是可以通过对已抓取页面或者待抓取URL对应页面进行分析获取到的URL，认为是可知网页。

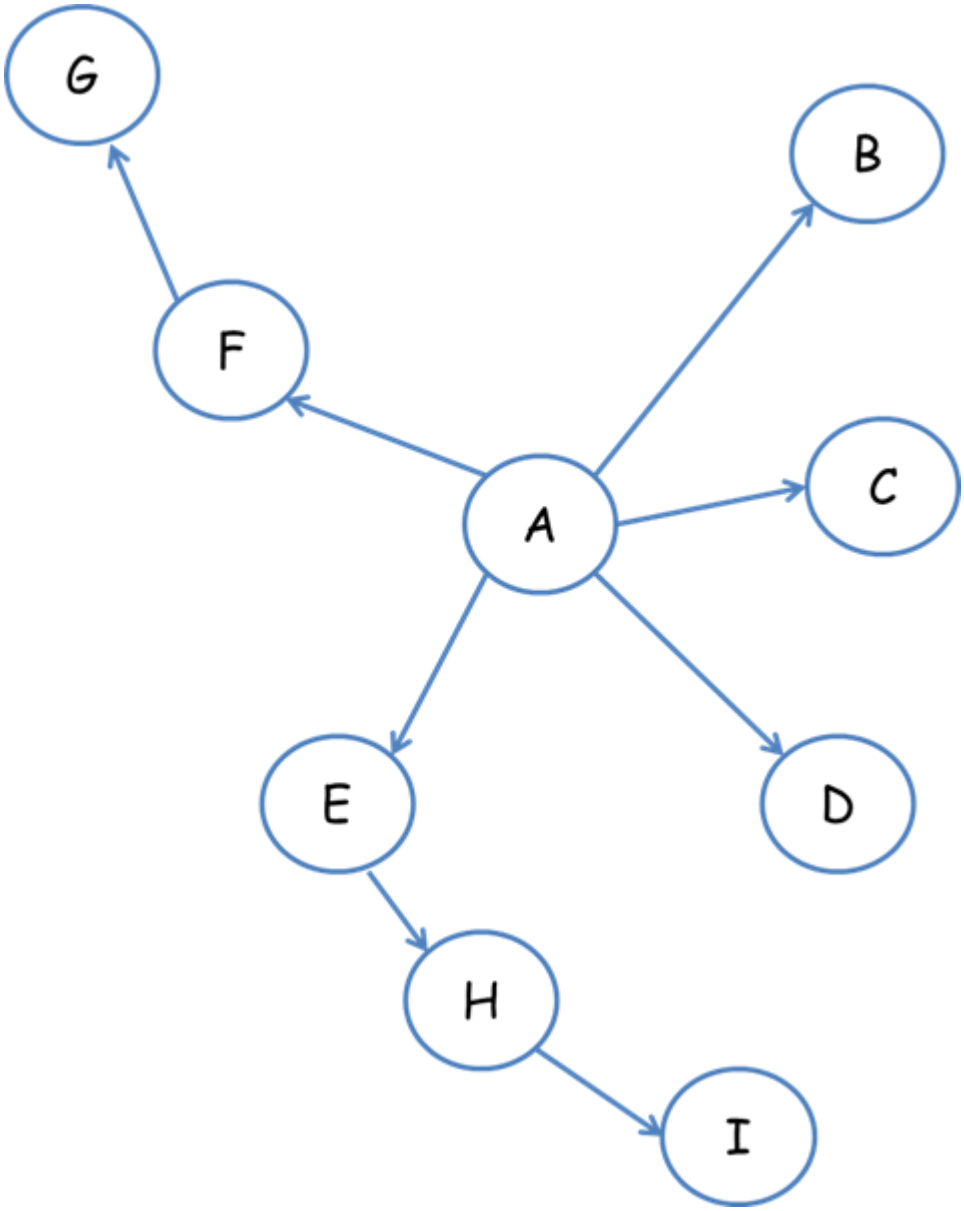
5.还有一部分网页，爬虫是无法直接抓取下载的。称为不可知网页。

常见的网络爬虫抓取策略有哪些?

在爬虫系统中，待抓取URL队列是很重要的一部分。待抓取URL队列中的URL以什么样的顺序排列也是一个很重要的问题，因为这涉及到先抓取那个页面，后抓取哪个页面。而决定这些URL排列顺序的方法，叫做抓取策略。下面重点介绍几种常见的抓取策略：

1.深度优先遍历策略

深度优先遍历策略是指网络爬虫会从起始页开始，一个链接一个链接跟踪下去，处理完这条线路之后再转入下一个起始页，继续跟踪链接。我们以下面的图为例：



遍历的路径：A-F-G E-H-I B C D

2.宽度优先遍历策略

宽度优先遍历策略的基本思路是，将新下载网页中发现的链接直接插入待抓取URL队列的末尾。也就是指网络爬虫会先抓取起始网页中链接的所有网页，然后再选择其中的一个链接网页，继续抓取在此网页中链接的所有网页。还是以上面的图为例：

遍历路径：A-B-C-D-E-F G H I

3.反向链接数策略

反向链接数是指一个网页被其他网页链接指向的数量。反向链接数表示的是一个网页的内容受到其他人的推荐的程度。因此，很多时候搜索引擎的抓取系统会使用这个指标来评价网页的重要程度，从而决定不同网页的抓取先后顺序。

在真实的网络环境中，由于广告链接、作弊链接的存在，反向链接数不能完全等他我那个也的重要程度。因此，搜索引擎往往考虑一些可靠的反向链接数。

4.Partial PageRank策略

Partial PageRank算法借鉴了PageRank算法的思想：对于已经下载的网页，连同待抓取URL队列中的URL，形成网页集合，计算每个页面的PageRank值，计算完之后，将待

抓取URL队列中的URL按照PageRank值的大小排列，并按照该顺序抓取页面。

如果每次抓取一个页面，就重新计算PageRank值，一种折中方案是：每抓取K个页面后，重新计算一次PageRank值。但是这种情况还会有一个问题：对于已经下载下来的页面中分析出的链接，也就是我们之前提到的未知网页那一部分，暂时是没有PageRank值的。为了解决这个问题，会给这些页面一个临时的PageRank值：将这个网页所有入链传递进来的PageRank值进行汇总，这样就形成了该未知页面的PageRank值，从而参与排序。下面举例说明：

5.OPIC策略策略

该算法实际上也是对页面进行一个重要性打分。在算法开始前，给所有页面一个相同的初始现金（cash）。当下载了某个页面P之后，将P的现金分摊给所有从P中分析出的链接，并且将P的现金清空。对于待抓取URL队列中的所有页面按照现金数进行排序。

6.大站优先策略

对于待抓取 URL 队列中的所有网页，根据所属的网站进行分类。对于待下载页面数多的网站，优先下载。这个策略也因此叫做大站优先策略。

写一个 demo 程序，要求可以输入特定主题关键字后进行简单的抓取，简述实现的基本原理

使用 python 在中国天气网进行上海之后 7 天最高温度和最低温度的抓取

程序主要由三个功能函数构成:

1: get_content 作用为获取网页中的 html 代码

2: get_data 作用为获取要抓取的网页数据

3: write_data 作用为写入 csv 文件

为了简便起见直接将要抓取的关键字写死为之后 7 天的最高和最低温度

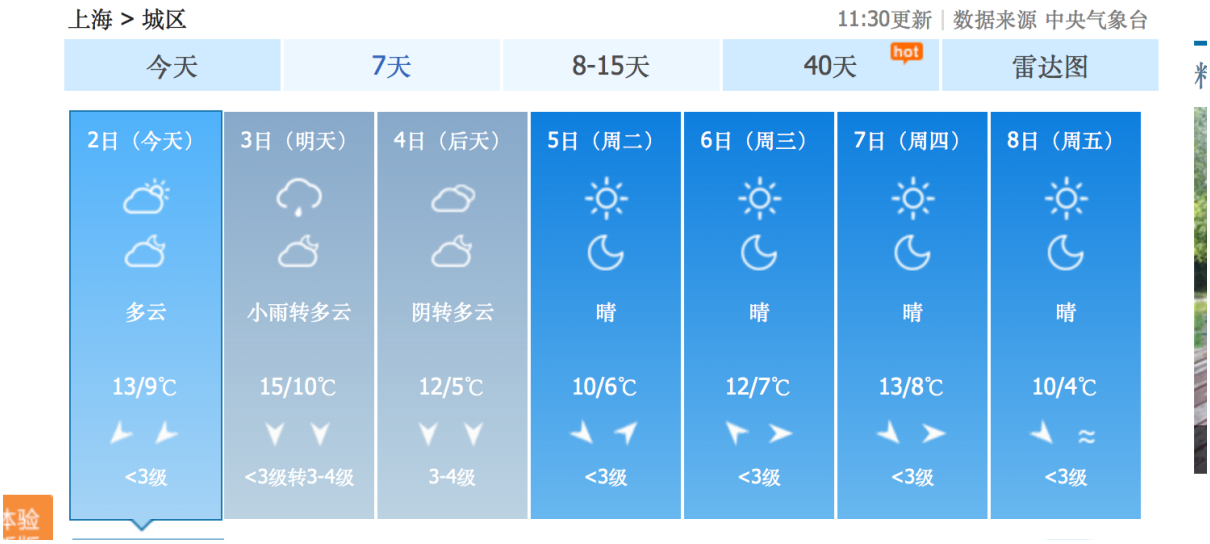
代码见附录

执行结果:

```
zby-MBP:desktop zby$ python weather.py
zby-MBP:desktop zby$
```

CSV 文件内容：

A	B	C	D
2		13	9
3		15	10
4		12	5
5		10	6
6		12	7
7		13	8
8		10	4


上海 > 城区
11:30更新 | 数据来源 中央气象台
今天
7天
8-15天
40天
hot
雷达图
2日（今天）
多云
13/9℃
<3级
3日（明天）
小雨转多云
15/10℃
<3级转3-4级
4日（后天）
阴转多云
12/5℃
3-4级
5日（周二）
晴
10/6℃
<3级
6日（周三）
晴
12/7℃
<3级
7日（周四）
晴
13/8℃
<3级
8日（周五）
晴
10/4℃
<3级

抓取结果正确

附录:

python 代码:

```
# coding : UTF-8
import requests
import csv
import random
import time
import socket
import http.client
# import urllib.request
from bs4 import BeautifulSoup
def get_content(url , data = None):
    header={

'Accept':'text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*
/*;q=0.8',
                'Accept-Encoding':'gzip, deflate',
                'Accept-Language':'zh-CN,zh;q=0.9,en;q=0.8',
                'Cache-Control':'max-age=0',
                'Connection':'keep-alive',
                'Cookie':'vjuids=1578751c0.16015f3abe5.0.5d08901870e8d;
userNewsPort0=1; Hm_lvt_080dabacb001ad3dc8b9b9049b36d43b=1512196779;
Hm_lpvt_080dabacb001ad3dc8b9b9049b36d43b=1512196779;
vjlast=1512196779.1512196779.30; f_city=%E4%B8%8A%E6%B5%B7%7C101020100%7C',
                'Host':'www.weather.com.cn',
                'Upgrade-Insecure-Requests':'1',
                'User-Agent':'Mozilla/5.0 (Macintosh; Intel Mac OS X 10_13_1)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/62.0.3202.94 Safari/537.36'
        }
        timeout = random.choice(range(80, 180))
        while True:
            try:
                rep = requests.get(url,headers = header,timeout = timeout)
                rep.encoding = 'utf-8'
                break
            except socket.timeout as e:
                print( '3:', e)
                time.sleep(random.choice(range(8,15)))

            except socket.error as e:
                print( '4:', e)
                time.sleep(random.choice(range(20, 60)))

            except http.client.BadStatusLine as e:
                print( '5:', e)
                time.sleep(random.choice(range(30, 80)))
```

```
            except http.client.IncompleteRead as e:
                print( '6:', e)
                time.sleep(random.choice(range(5, 15)))

        return rep.text
        # return html_text
    def get_data(html_text):
        final = []
        bs = BeautifulSoup(html_text, "html.parser") # 创建 BeautifulSoup 对象
        body = bs.body # 获取 body 部分
        data = body.find('div', {'id': '7d'}) # 找到 id 为 7d 的 div
        ul = data.find('ul') # 获取 ul 部分
        li = ul.find_all('li') # 获取所有的 li

        for day in li: # 对每个 li 标签中的内容进行遍历
            temp = []
            date = day.find('h1').string # 找到日期
            temp.append(date) # 添加到 temp 中
            inf = day.find_all('p') # 找到 li 中的所有 p 标签
            temp.append(inf[0].string,) # 第一个 p 标签中的内容（天气状况）加到 temp
中
            if inf[1].find('span') is None:
                temperature_highest = None # 天气预报可能没有当天的最高气温（到了傍
晚，就是这样），需要加个判断语句,来输出最低气温
            else:
                temperature_highest = inf[1].find('span').string # 找到最高温
                temperature_highest = temperature_highest.replace('℃', '') # 到了晚上网站
会变，最高温度后面也有个℃
            temperature_lowest = inf[1].find('i').string # 找到最低温
            temperature_lowest = temperature_lowest.replace('℃', '') # 最低温度后面有
个℃，去掉这个符号
            temp.append(temperature_highest) # 将最高温添加到 temp 中
            temp.append(temperature_lowest) #将最低温添加到 temp 中
            final.append(temp) #将 temp 加到 final 中

        return final

    def write_data(data, name):
        file_name = name
        with open(file_name, 'a', errors='ignore', newline='') as f:
            f_csv = csv.writer(f)
            f_csv.writerows(data)

    if __name__ == '__main__':
        url ='http://www.weather.com.cn/weather/101020100.shtml'
```

```
html = get_content(url)
result = get_data(html)
write_data(result, 'weather.csv')
```