**中国矿业大学计算机学院**

**2018 级本科生课程报告**

课程名称 Python语言与安全程序设计

报告时间 2021.7. 8

学生姓名 田浪

学 号 08183006

专 业 信息安全

任课教师 朱长征

课程考查评分

|  |  |  |
| --- | --- | --- |
| 毕业要求 | 指标点 | 得分 |
| 3.设计/开发解决方案 | 3.2能够在信息安全系统中合理地组织、存储和处理数据，正确地设计算法并对算法分析和评价 |  |
| 5.使用现代工具 | 5.2能够在信息安全领域复杂工程问题的预测、模拟或解决过程中，开发、选择使用恰当的技术、软硬件及系统资源及相关工程研发工具，获得复杂工程问题的良好解决方案 |  |
| 总合 |  |  |

目录

[1 验证码识别 1](#_Toc77173798)

[1.1 设计思路 1](#_Toc77173799)

[1.2 流程图 1](#_Toc77173800)

[1.3 运行环境 1](#_Toc77173801)

[1.4 运行效果截图 2](#_Toc77173802)

[1.5 分析总结 3](#_Toc77173803)

[2分析firefox浏览记录 3](#_Toc77173804)

[2.1设计思路 3](#_Toc77173805)

[2.2 流程图 4](#_Toc77173806)

[2.3开发环境配置 4](#_Toc77173807)

[2.4运行效果截图 4](#_Toc77173808)

[2.5 分析总结 6](#_Toc77173809)

[3 破解wifi密码 6](#_Toc77173810)

[3.1 设计思路 6](#_Toc77173811)

[3.2 流程图 7](#_Toc77173812)

[3.3 开发环境配置 7](#_Toc77173813)

[3.4 运行效果截图 7](#_Toc77173814)

[3.5 分析总结 9](#_Toc77173815)

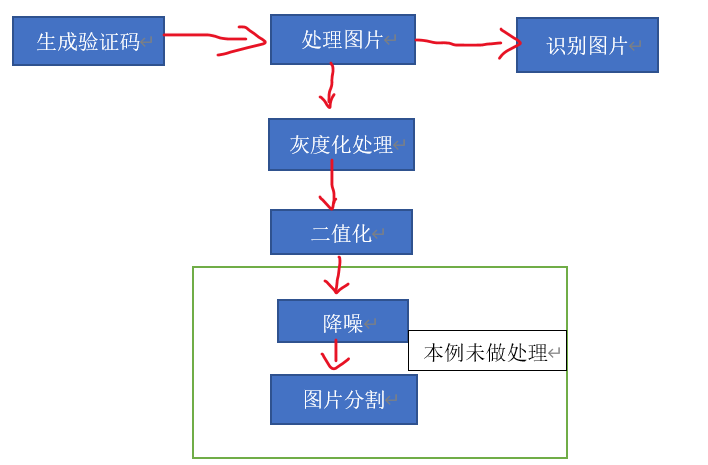
# 验证码识别

## 1.1 设计思路

由于主要处理的是验证码的问题，所以这里不设计爬虫程序。

1. 利用PIL库生成验证码
2. 调用pytesseract库处理图片

## 1.2 流程图



验证码识别流程图

## 1.3 运行环境

环境：

Anaconda 1.7.2+Spider+python3.8+tesseract

依赖库：

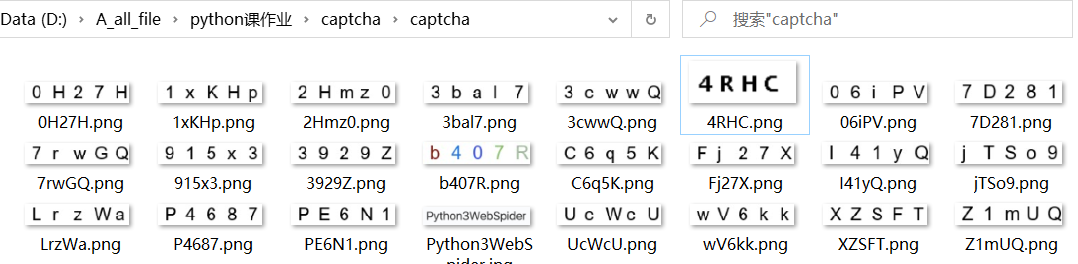
pillow+pytesseract+os+random

## 1.4 运行效果截图

生成验证码：

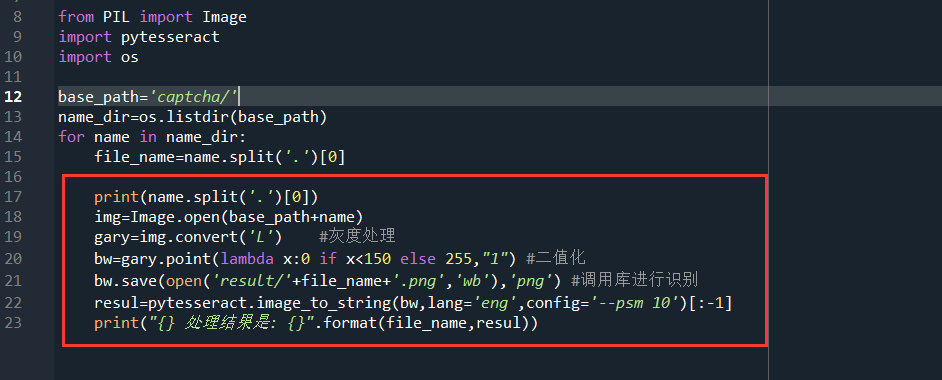


代码图



生成的验证码

运行结果



验证码处理代码



运行效果截图

## 1.5 分析总结

实现了pillow库生成简单验证码，但是并未做添加噪声的处理。

验证码识别部分做的简陋，没有考虑有点、线噪声的情况。

在最开始做的时候，调用captcha库生成含有噪声的验证码，然后使用pillow库进行降噪处理，最后使用pytsseract库进行识别，发现效果并不好，于是改成了无噪声的图片。

看了很多文章，发现可以利用机器学习或者搭建深度训练模型进行图片处理，效果会更好。

# 2分析firefox浏览记录

## 2.1 设计思路

1、找到Firefox浏览记录存储位置，在C:\Users\latpurple\AppData\Roaming\Mozilla\Firefox\Profiles\033vcuw5.default-release-1\places.sqlite中

2、将places.sqlite文件复制到代码所在的文件夹，这样就可以离线进行操作了

3、通过python的sqlite3库，连接这个数据库文件

4、对浏览记录进行预处理，最后进行可视化分析

## 2.2 流程图



Firefox浏览记录分析流程图

## 2.3开发环境配置

环境：

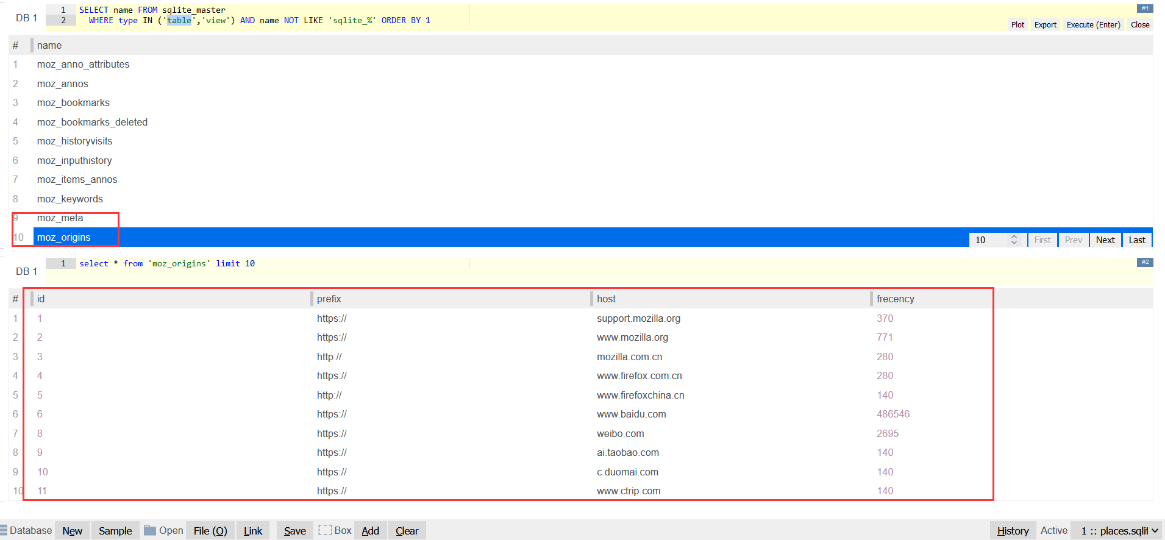
Anaconda 1.7.2+Spyder+python3.8+Firefox浏览器+sqlite manager插件

依赖库：

sqlite3+matplotlib

## 2.4运行效果截图

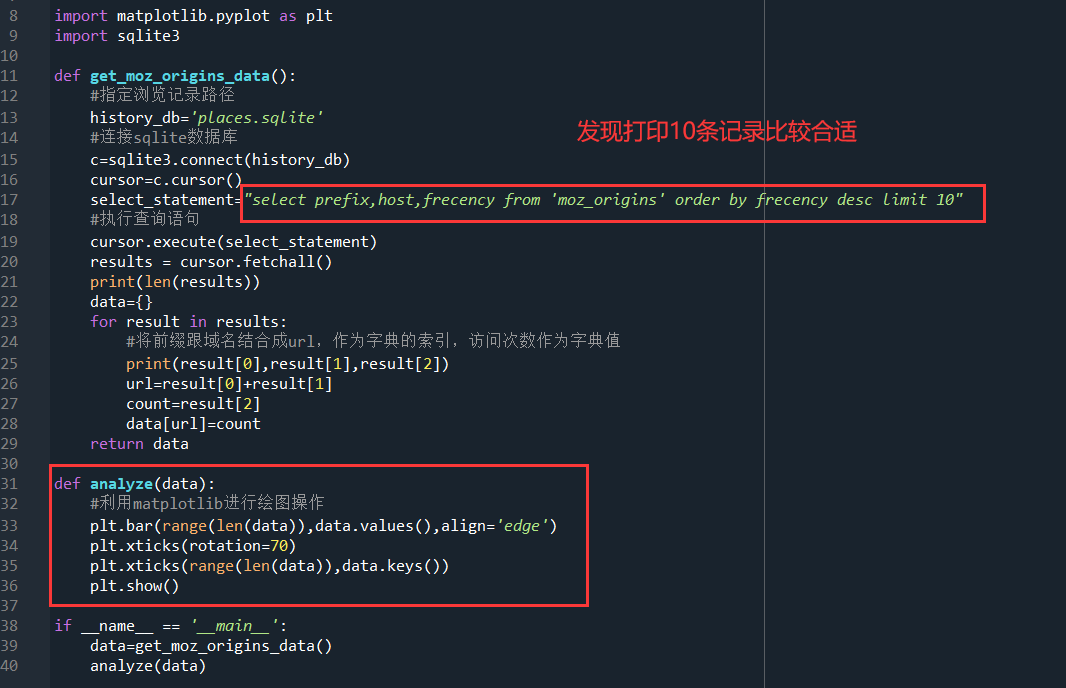
1、首先通过Firefox浏览器的sqlite manager插件打开places.sqlite文件，通过查询测试发现比较中要的数据在moz\_origins表中，如图：



places.sqlite视图

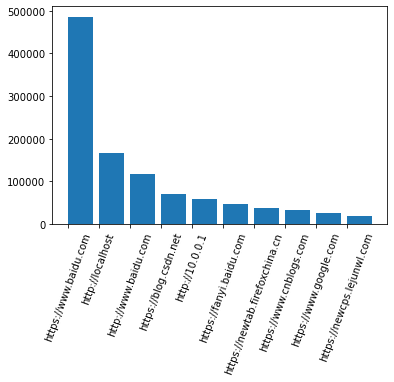
2、代码部分

执行查询，合并数据，然后绘图



代码图

3、可视化结果



可视化结果图

## 2.5 分析总结

通过查找资料，发现，Firefox浏览器跟Google浏览器在本地的浏览记录存储位置不太一样，而且其存储在sqlite数据库中的表的结构也不一样。

通过实际操作，熟悉python使用sqlite数据库，并且用matplotlib库进行制图，但是仅仅只对URL的访问次数做了可视化。

我没能实现与其他表相结合，做出更进一步分析，例如搜索记录，访问记录。

有些表中有数据，但是不太清楚有什么用，不仅如此，还有好几个空表。

# 3 破解wifi密码

## 3.1 设计思路

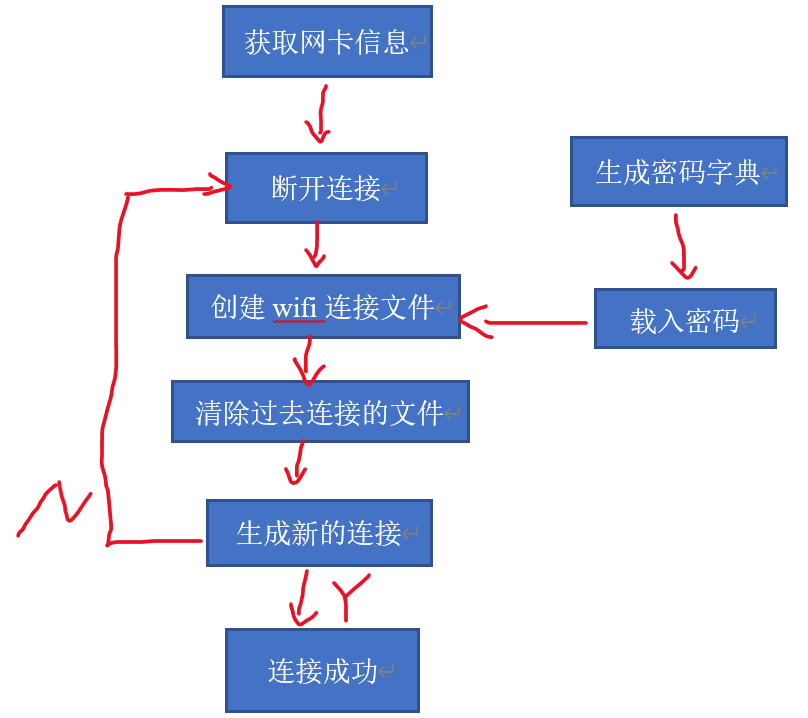
1、创建密码字典，破解wifi密码本质上是一种暴力破解

2、利用pywifi库，操控网卡去连接热点（我自己的热点名称是”qingdu”）

3、选择加密模式为WPA2PSK（目前手机热点所采用的加密方式），输入密码（从字典载入）

4、通过字典进行爆破，直到成功，或者字典跑完。

## 3.2 流程图



破解WiFi流程图

## 3.3 开发环境配置

环境：

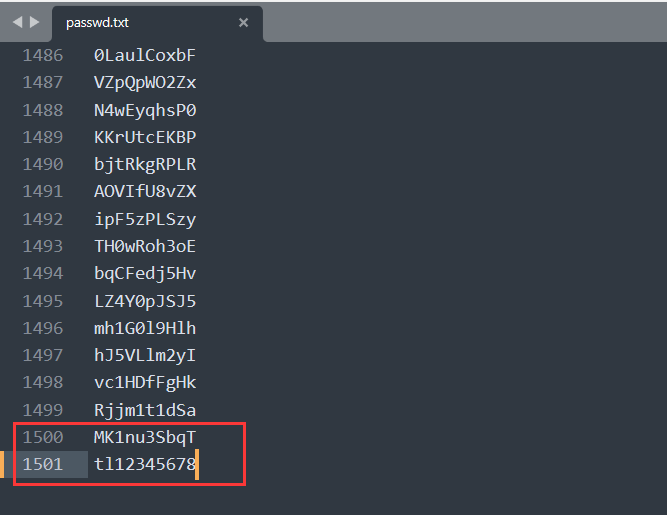
Anaconda 1.7.2+Spyder+python3.8+sublime text3

依赖库：

random+pywifi+time

## 3.4 运行效果截图

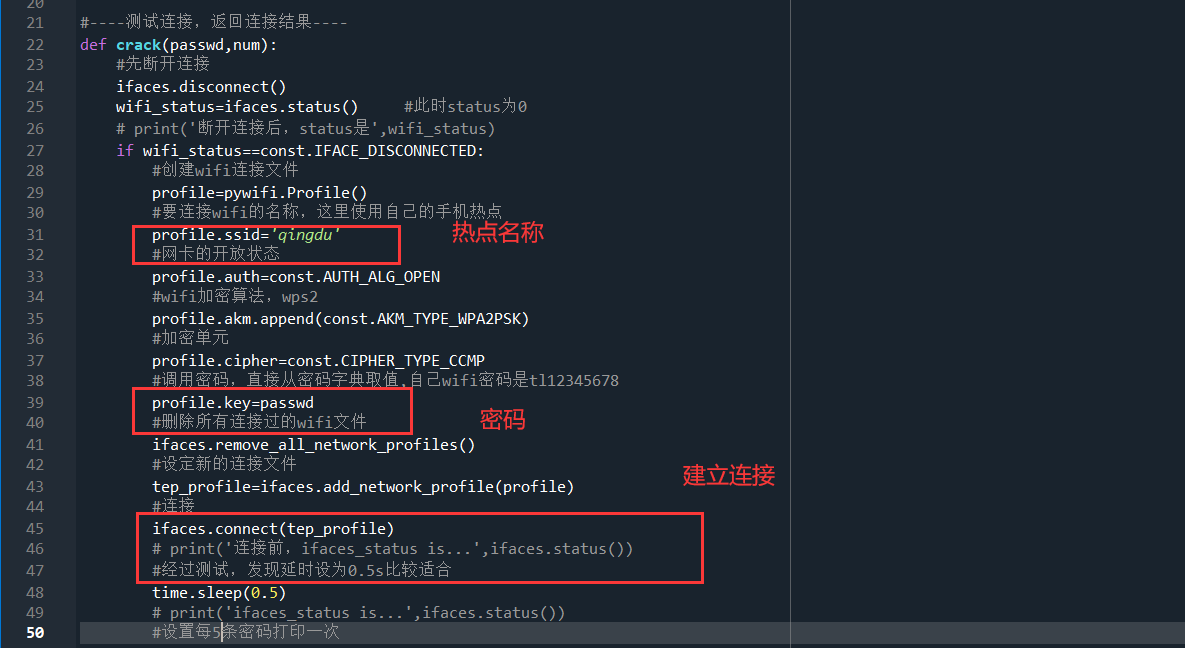
1、生成了3\*500的密码字典



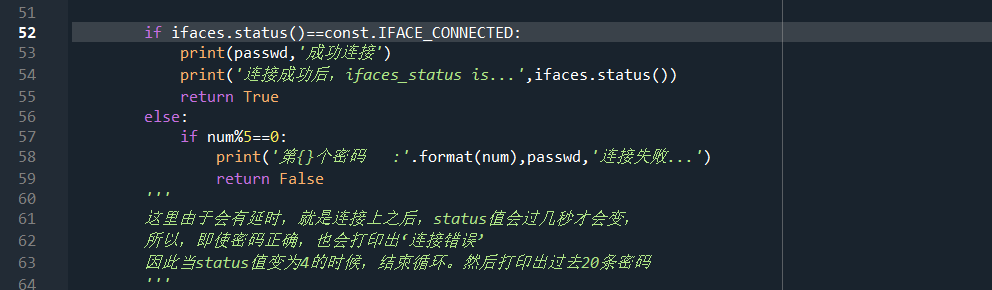
生成的字典图，最后一个为热点的密码

2、核心代码图

当网卡的status值等于4的时候，连接成功。如图：判断网卡状态值



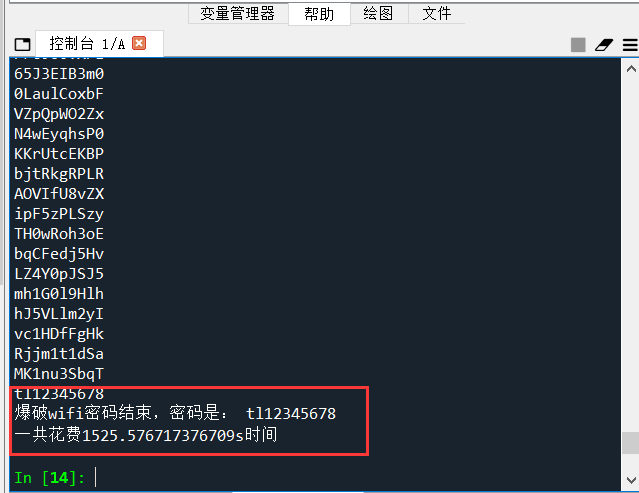
建立连接过程图



判断网卡状态值



const中的部分常量值



运行结果图

## 3.5 分析总结

程序可以实现手机WiFi热点的破解，但是是在知道加密方式的和密码的前提下，才能成功。

生成的随机密码字典，很难破解人为设计、有规律的密码。因此，更有效的方法是，设计更为强大的字典，可以通过社交工程，或者社工库中的密码泄露扩充字典。

可以设计对周围热点的自动破解。通过pywifi库操控网卡可以扫描出附近的WiFi热点，同样也可以识别WiFi热点的加密模式，再结合强大的密码字典，就能实现对周围WiFi热点的破解了。