北京航空航天大学

2018 - 2019 学年 第二学期

《大学计算机基础（理科）》
大作业实验报告

班 级_____27 班__ 学 号 __18377470_____

姓 名____庄嘉恒__成 绩 _____

2019 年 6 月

作业 1　简易的文本编辑器

一、实验目的

1.GUI 页面设计，tkinter 的使用

2.数据可视化处理，matplotlib 库的使用和 wordcloud 库的使用

3.对字符串的各项操作，复习字符串，字典，列表等数据类型的使用

4.通过正则表达式匹配字符串

5.time 库的使用，复习数字的基本转换

6.文件的读写操作

7.PyInstaller 库的使用，cmd 命令行的操作

二、实验任务

描述本题目需要完成的内容和要求。

1. 基础要求

（1）导入英文文章

（2）将内容拆解为单词，删去周围空白字符和标点，转换为小写

（3）统计文章中出现全部英文单词的总数量

（4）统计文章中每个英文单词的词频

（5）采用停用词表，绘制词频最高六个单词的柱状图

（6）将上述功能包装为 GUI 软件

2. 可选项

（1）查找替换功能

（2）清空文本的删除功能

（3）高亮显示用户指定文本

（4）添加菜单栏

（5）增加对话框和消息框

（6）插入超链接

3. 补充项

（1）高亮指定的单词

（2）生成词云图

（3）保存当前文本

（4）撤销和恢复操作

（5）统计操作时间功能

（6）文件导出功能

（7）将文件打包为小程序

三、总体设计方案

介绍作业完成的功能，实现方案（如程序分为几部分，每部分采用什么方法、解决什么问题）。

程序主要分为六个部分。

第一部分为文本编辑部分。主要功能包括导入导出文本，清空文本，撤销恢复操作，将文本转换为标准模式。这部分主要采用的方法有 tkinter.filedialog 库中的 askopenfilename() 和 asksavefilename()函数,和 tkinter 中 Text 函数的各项功能。

第二部分为统计和数据可视化部分。主要功能宝包括字符和词数统计，词频统计，以及绘制柱状图和云图。字符统计通过字符串的 len()实现；词数统计通过 split 将字符串转换成列表，统计列表长度；柱状图通过 matplotlib.pyplot 中的 bar 函数绘制；云图通过 wordcloud 库绘制。

第三部分为扩展功能，即可选项部分。主要功能包括高亮与取消高亮，标记、查找、修改单词。高亮通过 Text 中的 tag 函数实现，标记通过 search 配合 tag 的高亮方法实现，查找通过正则表达式进行匹配达到最准确的索引，修改通过字符串的 replace 配合空格和换行符实现单词的替换。

第四部分为超链接。主要功能包括链接到 126,163 邮箱，并模拟真实的点击网络链接的效果。采用创建 tag、定义回调函数、和将 tag 与事件绑定的方法实现。

第五部分为时间管理。可以得到用户持续操作的时间。主要通过 time 库和数据进制的转换实现。

第六部分为整体的 GUI 设计。包括 Button、Label、Text、Canvas、Menu、Scrollrbar 等基本组件，用 place 方法进行主要布局，部分结合 pack 方法。

四、设计思路和关键语法

注意：这是本报告的重点！

详细分析实验的实现方法，介绍你的设计用到了哪些函数或者涉及哪些知识点；详述每个模块的关键语法及其实现。并给出关键代码。

I. 文本编辑

```
def getText(txt):   #文本初步处理函数
    txt=txt.lower()
    for ch in '!"#$%()*+,-./:;<=>?@[\\]^_`{|}~\' - ’ “ ” 0123456789':
        txt=txt.replace(ch,"")
    return txt
```

1. 导入文本

```
def originaltext():
    global text   #让此处的 text 被其他地方调用
    fd=tkinter.filedialog.askopenfilename()
    try:
        txt=open(fd,"r").read()
    except UnicodeDecodeError:   #使其适用于不同编码类型的文本
        txt=open(fd,"r",encoding="utf-8").read()
    textbox.insert(tk.INSERT,txt)
    text=getText(txt)    #读入文件后进行相应的处理
```

2. 文件保存

```
def save():
    global text
    txt=textbox.get(1.0,tk.END)
    text=getText(txt)
```

3. 撤销和恢复操作

```
def key_callback(event):      #定义键盘回调函数，绑定键盘操作
    textbox.edit_separator()#插入一个分隔符到存放到操作记录的栈中，用于表示完成一次完整操作(每次撤销一个操作)
def undo_callback():      #撤回
    textbox.edit_undo()
def redo_callback():      #恢复
    textbox.edit_redo()
```

4. 清空文本

```
def clear():
    ans=tkinter.messagebox.askokcancel("Hint","Are you sure to clear?")
    if ans:
        textbox.delete(1.0,tk.END)
```

5. 标准文本

```
def modifiedtext(text):
    textbox.delete(1.0,tk.END)
    textbox.insert(tk.INSERT,text)   #INSERT 表示光标位置加入
    tkinter.messagebox.showinfo("Hint","Successfully saved!")
```

6. 导出文本

```
def onSave():
    filename = tkinter.filedialog.asksaveasfilename()
    if filename:
      alltext=textbox.get(1.0,tk.END)
      open(filename, 'w').write(alltext)     #写文件模式
```

II. 统计功能

1.字符统计

```
def strlong():   #统计字符数
    printnum1=str(len(textbox.get(1.0,tk.END)))   #包括换行符
    #字数统计标签
    labelNum1=tk.Label(top,
                       text="字符统计: "+printnum1,
                       justify=tk.RIGHT,
                       width=80)
    labelNum1.place(x=10,y=550,width=125,height=20)
```

2. 词数统计

```
#统计单词数
    def wordlong():
```

```
        caltxt=textbox.get(1.0,tk.END)
        words2=caltxt.split()
        rlt=len(words2)
        #字数统计标签
        labelNum2=tk.Label(top,
                           text="词数统计: "+str(rlt),
                           justify=tk.RIGHT,
                           width=80)
        labelNum2.place(x=10,y=575,width=125,height=20)
```

3. 词频统计

```
def frequency(text):   #词频统计函数
    words=text.split()
    counts={}
    for word in words:
        counts[word]=counts.get(word,0)+1#如果这个单词在字典中，则这个单词的键对应的值加一，如果不在，则将这个单词的值初始化为1
    items=list(counts.items())
items.sort(key=lambda x:x[1],reverse=True)#按照值的大小倒序(从大到小)排列，
    for i in range(len(items)):
        word,count=items[i]
        print("{0:<15}{1:>5}".format(word,count))
    print("英文单词的总数为:",len(words))
    #在 GUI 中输出
    cipin=tk.Tk()
    cipin.title("词频统计结果")
    cipin.geometry("400x300")
    textbox1 = tk.Text(cipin)
    textbox1.place(x=0,y=0,width=400,height=300)
    ybar=tk.Scrollbar(cipin,orient=tk.VERTICAL)
```

```
    ybar.config(command=textbox1.yview)
    textbox1.config(yscrollcommand=ybar.set)
    ybar.pack(side=tk.RIGHT,fill=tk.Y)    #fill 充满 y 轴
    #在 IPython 环境中输出
    for i in range(len(items)):
        word,count=items[i]
        ch1="{0:<15}{1:>5}".format(word,count)+"\n"
        textbox1.insert(tk.INSERT,ch1)
    ch2="英文单词的总数为:"+str(len(words))
    textbox1.insert(tk.INSERT,ch2)
```

4. 绘柱状图

```
def draw(text,stopwords):
    words=text.split()
    counts={}
    for word in words:
        if word not in stopwords:
            counts[word]=counts.get(word,0)+1
    items=list(counts.items())
    items.sort(key=lambda x:x[1],reverse=True)
    drawords=[];nums=[]
    for i in range(6):
        word,count=items[i]
        drawords.append(word);nums.append(count)
        print("{0:<15}{1:>5}".format(word,count))
    color = tkinter.simpledialog.askstring('Hint','Please enter the graphic color')
    plt.rcParams['font.sans-serif']=['SimHei'] #用来正常显示中文标签
    plt.rcParams['axes.unicode_minus']=False #用来正常显示负号
    ind = np.arange(1,7,1)
    width = 0.75
```

```
    plt.bar(ind, nums, width=width, color=color, label=u'关键词数目')
    plt.ylabel(u'nums')
    plt.xlabel(u'words')
    plt.title(u'词频统计表')
    plt.xticks(ind,drawords)
    plt.legend()
    plt.savefig(r"bar.jpg")     #保存图像要在 show 之前完成，防止空白
    plt.show()
```

5. 绘制云图

```
def drawcloud(text):
    w=wordcloud.WordCloud(font_path="msyh.ttc",
                          max_words=20,
                          width=1000,height=700,
                          background_color="white")
#生成词云的函数，可以设置的参数有背景色，宽度高度，最大次数，字体，图形的形状等
    w.generate(text)
    w.to_file("wordcloud.jpg")
    tkinter.messagebox.showinfo("Hint","Successfully generated!")
```

III. 文本显示

1. 高亮文本

```
def highlight():   #添加 tag 函数，设置黄底红字的高亮模式
    textbox.tag_add("highlight",tk.SEL_FIRST,tk.SEL_LAST)
    textbox.tag_config("highlight",background="yellow",foreground="red")
```

2. 取消高亮

```
def cancellight():   #用重新打印的办法解决 tag 无法移除
    reprint=textbox.get("1.0",tk.END)
    textbox.delete("1.0",tk.END)
    textbox.insert(tk.INSERT,reprint)
```

3. 标记单词

```
def wordlight(pos,speword,specolor):
    op='+'+str(len(speword))+'c'
    textbox.tag_add("highlight_word",pos,pos+op)
textbox.tag_config("highlight_word",background=specolor,foreground="black"
)
def searchword():
    start="1.0"
    speword = tkinter.simpledialog.askstring('Hint','Please enter the word')
    specolor = tkinter.simpledialog.askstring('Hint','Please enter the color')
    while True:
        pos=textbox.search(speword,start,stopindex=tk.END)
        if not pos:   #返回起始位
            break
        #print("位置是:",getindex(textbox,pos))
        wordlight(pos,speword,specolor)
        start=pos+'+1c'
#search 函数只能返回第一个满足条件的字符串，欲找到所有满足条件的字符串，增设一层循环，当找到满足条件的词后从下一个字符再次开始寻找，直到遍历完所有的文本后用 break 退出循环。
```

4. 查找单词

```
def searchword0():
    speword = tkinter.simpledialog.askstring('Hint','Please enter the word')
    it=re.finditer("[\s]*"+speword+"[^a-z]",text,re.I)
#通过正则表达式，匹配空格(或什么都没有)+单词+空格(或\n)的结构，[\s]*表示匹配到空格一次或零次，[^a-z]表示匹配到不是 a-z 中的字符
    #GUI 前期绘制
    find=tk.Tk()
    find.title("查找结果")
```

```
    find.geometry("400x300")
    textbox2 = tk.Text(find)
    textbox2.place(x=0,y=0,width=400,height=300)
    ybar=tk.Scrollbar(find,orient=tk.VERTICAL)
    ybar.config(command=textbox2.yview)
    textbox2.config(yscrollcommand=ybar.set)
    ybar.pack(side=tk.RIGHT,fill=tk.Y)   #fill 充满 y 轴
    i=0
    for match in it: #得到迭代器里的东西需要循环
        i+=1
        a=match.start()    #对 match 对象进行操作，获取 match 对象里的信息
        b=match.end()
        ch=" 第 {} 个 {} 的 位 置 为 :{} 字 符 ——{} 字 符 ".format(i,speword,a+1,b-1)+"\n"    #单词的索引值要扣除两边的特殊符号
        textbox2.insert(tk.INSERT,ch)
        print(" 第 {} 个 {} 的 位 置 为 :{} 字 符 ——{} 字 符 ".format(i,speword,a+1,b-1))
```

5. 修改单词

```
def change(text):
    text=" "+text[:-1]+" "
    old = tkinter.simpledialog.askstring('Hint','Please enter the old word')
    new = tkinter.simpledialog.askstring('Hint','Please enter the new word')
    #text=re.sub("[\s]*"+old+"[^a-z]"," "+new+" ",text,re.I)
    old=" "+old+" ";new=" "+new+" "
    text=text.replace(old,new)#通过补空格的方式整体替换，最后在去两端空格
    text=text.strip()
    textbox.delete(1.0,tk.END)
    textbox.insert(tk.INSERT,text)
```

IV. 发送邮件

```
def click163(event):    定义鼠标点击事件的回调函数
    webbrowser.open("http://mail.163.com/")
def link163():    #设置显示模式
    textbox.insert("1.0","我使用 163 邮件服务器\n")
    textbox.tag_add("link163","1.3","1.6")
    textbox.tag_config("link163",foreground="blue",underline=True)
def click126(event):
    webbrowser.open("http://mail.126.com/")
def link126():
    textbox.insert("1.0","我使用 126 邮件服务器\n")
    textbox.tag_add("link126","1.3","1.6")
    textbox.tag_config("link126",foreground="blue",underline=True)
#定义鼠标移入移除事件的回调函数
def show_hand_cursor(event):
    textbox.config(cursor="arrow")
def show_arrow_cursor(event):
    textbox.config(cursor="xterm")
#链接效果 163
textbox.tag_bind("link163","<Enter>",show_hand_cursor)
textbox.tag_bind("link163","<Button-1>",click163)
textbox.tag_bind("link163","<Leave>",show_arrow_cursor)
#链接效果 126
textbox.tag_bind("link126","<Enter>",show_hand_cursor)
textbox.tag_bind("link126","<Button-1>",click126)
textbox.tag_bind("link126","<Leave>",show_arrow_cursor)
```

V. 时间管理

```
#计时功能
start=time.perf_counter()
def caltime():
```

```
    end=time.perf_counter()
    dtime=float(end-start)
    if dtime>=3600:      #进制的转换
        q1=dtime//3600;r1=dtime%3600
        q2=int(r1//60);r2=int(r1%60)
        printtime=str(q1)+"(h):"+str(q2)+"(m):"+str(r2)+"(s)"
    elif dtime>=60:
        q1=int(dtime//60);r1=int(dtime%60)
        printtime=str(q1)+"(m):"+str(r1)+"(s)"
    else:
        printtime=str(round(dtime,2))+"(s)"
    labelTime=tk.Label(top,
                       text="时间统计: "+printtime,
                       justify=tk.RIGHT,
                       width=80)
    labelTime.place(x=300,y=550,width=200,height=20)
```

VI. GUI 页面设计

```
top = tk.Tk()    #Toplevel
top.geometry("625x600")
top.resizable(False,False)    #可以用 pack() 布局滚动条的基础
top.title("简易的文本编辑器")
#创建文本框
textbox = tk.Text(top,undo=True,maxundo=20)    #最大撤销次数
textbox.place(x=5,y=50,width=600,height=500)
#插入一个分隔符到存放操作记录的栈中，用于表示已经完成一次完整的操作
textbox.edit_separator()
#链接效果 163
textbox.tag_bind("link163","<Enter>",show_hand_cursor)
textbox.tag_bind("link163","<Button-1>",click163)
```

```
textbox.tag_bind("link163","<Leave>",show_arrow_cursor)
#链接效果 126
textbox.tag_bind("link126","<Enter>",show_hand_cursor)
textbox.tag_bind("link126","<Button-1>",click126)
textbox.tag_bind("link126","<Leave>",show_arrow_cursor)
ybar=tk.Scrollbar(top,orient=tk.VERTICAL)
ybar.config(command=textbox.yview)
textbox.config(yscrollcommand=ybar.set)
ybar.pack(side=tk.RIGHT,fill=tk.Y)     #滚动条至于窗口右端，填充全部
#创建导入文本按钮
ReadBtn=tk.Button(top,
                  text="导入文本",
                  command=originaltext)
ReadBtn.place(x=5,y=10,width=60,height=30)
#创建清除文本按钮
clearBtn=tk.Button(top,
                   text="标准模式",
                   command=lambda:modifiedtext(text))
clearBtn.place(x=80,y=10,width=60,height=30)
#创建词频统计按钮
wordfreBtn=tk.Button(top,
                     text="词频统计",
                     command=lambda:frequency(text))   #lambad 传入参数
是 lambda
wordfreBtn.place(x=155,y=10,width=60,height=30)
#绘图按钮
drawBtn=tk.Button(top,
                  text="绘制图形",
                  command=lambda:draw(text,stopwords))
```

```
drawBtn.place(x=230,y=10,width=60,height=30)
#词云按钮
cloudBtn=tk.Button(top,
                   text="生成词云",
                   command=lambda:drawcloud(text))
cloudBtn.place(x=305,y=10,width=60,height=30)
#高亮按钮
lightBtn=tk.Button(top,
                   text="高亮文本",
                   command=highlight)
lightBtn.place(x=380,y=10,width=60,height=30)
#取消高亮按钮
cancellightBtn=tk.Button(top,
                         text="取消高亮",
                         command=cancellight)
cancellightBtn.place(x=455,y=10,width=60,height=30)
#查找按钮
searchBtn=tk.Button(top,
                    text="标记单词",
                    command=searchword)
searchBtn.place(x=530,y=10,width=60,height=30)
labelIntro=tk.Label(top,
                    text="注:除保存文本&导入文本,其余对文本内容的操作需先保存,后分析",
                    justify=tk.RIGHT,
                    width=1000)
labelIntro.place(x=-125,y=575,width=1000,height=20)
#添加菜单功能
menubar = tk.Menu(top)   #创建一个顶级菜单
```

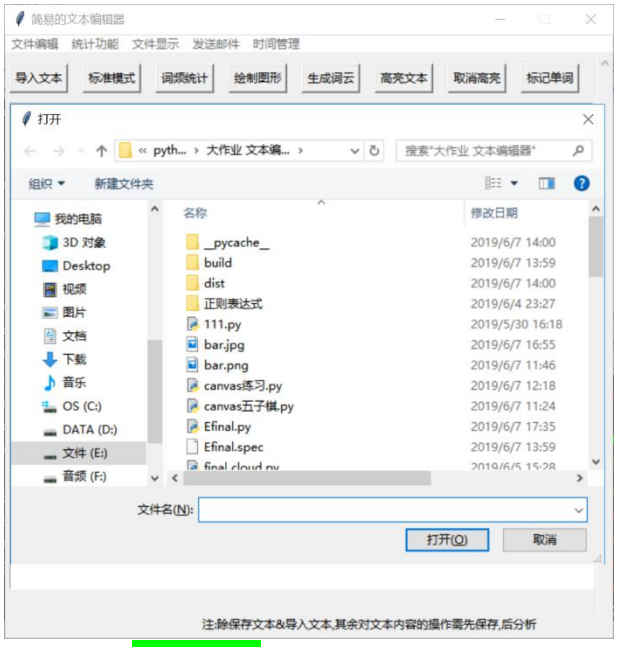
```
#filemenu1 创建主菜单和子菜单
c1 =
[originaltext,save,undo_callback,redo_callback,clear,lambda:modifiedtext(text),o
nSave]
i = 0
filemenu1 = tk.Menu(menubar,tearoff = 0)
for item in ['导入文本','文件保存','撤销操作','恢复操作','清空文本','标准模式','
导出文本']:
    filemenu1.add_command(label = item,command = c1[i])
    filemenu1.add_separator()   #加分割线
    i = i + 1
#filemenu2 创建主菜单和子菜单
c2 =
[strlong,wordlong,lambda:frequency(text),lambda:draw(text,stopwords),lambda:
drawcloud(text)]
i = 0
filemenu2 = tk.Menu(menubar,tearoff = 0)
for item in ['字符统计','词数统计','词频统计','绘柱状图','绘制云图']:
    filemenu2.add_command(label = item,command = c2[i])
    filemenu2.add_separator()   #加分割线
    i = i + 1
#filemenu3 创建主菜单和子菜单
c3 = [highlight,cancellight,searchword,searchword0,lambda:change(text)]
i = 0
filemenu3 = tk.Menu(menubar,tearoff = 0)
for item in ['高亮文本','取消高亮','标记单词','查找单词','修改单词']:
    filemenu3.add_command(label = item,command = c3[i])
    filemenu3.add_separator()   #加分割线
    i = i + 1
```

```
#filemenu4 创建主菜单和子菜单
c4 = [link163,link126]
i = 0
filemenu4 = tk.Menu(menubar,tearoff = 0)
for item in ['163 邮箱','126 邮箱']:
    filemenu4.add_command(label = item,command = c4[i])
    filemenu4.add_separator()   #加分割线
    i = i + 1
#filemenu5 创建主菜单和子菜单
c5 = [caltime]
i = 0
filemenu5 = tk.Menu(menubar,tearoff = 0)
for item in ['时间统计']:
    filemenu5.add_command(label = item,command = c5[i])
    filemenu5.add_separator()   #加分割线
    i = i + 1
#指定主菜单和子子菜单的级联关系
#将 menubar 的 menu 属性指定为 filemenu，即 filemenu 为 menubar 的下拉菜单
menubar.add_cascade(label = '文件编辑',menu = filemenu1)
menubar.add_cascade(label = '统计功能',menu = filemenu2)
menubar.add_cascade(label = '文件显示',menu = filemenu3)
menubar.add_cascade(label = '发送邮件',menu = filemenu4)
menubar.add_cascade(label = '时间管理',menu = filemenu5)
top['menu'] = menubar
tk.mainloop()
```

五、程序运行效果

给出程序 GUI 截图和各种情况下运行结果的截图，并**简要说明**。

注意：不是简单地把截图粘贴到报告里，一定要加以文字说明！



导入文本：点击导入文本，选择需要导入的文本文档或 py 文件(均可)

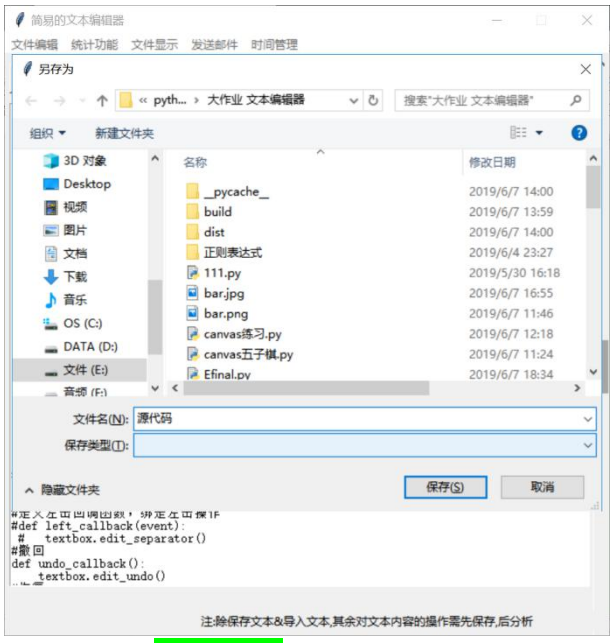
标准模式：小写，取出各种符号

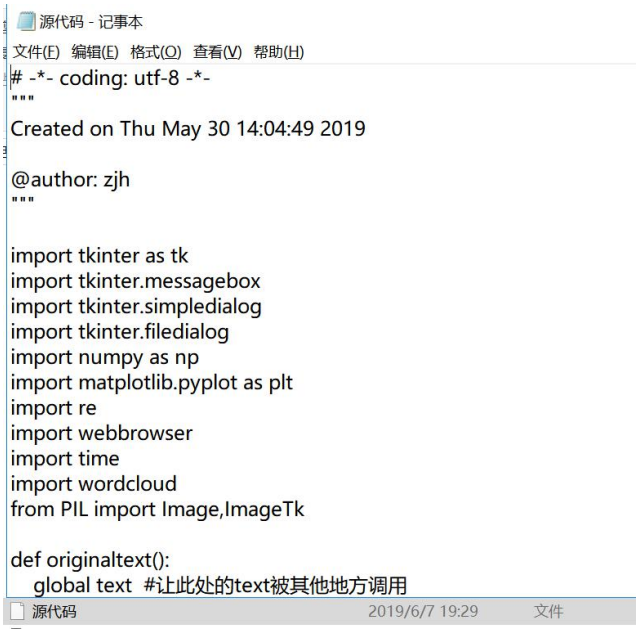
清空文本：弹出确认消息框，点击确认清空文本

撤销操作：点击撤销上次对文本的操作

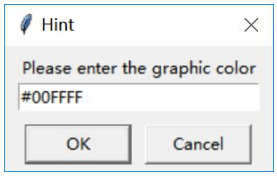
恢复操作：点击恢复上次对文本的操作

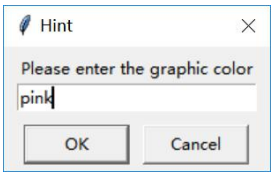
文件保存：保存当前文本框里的文本

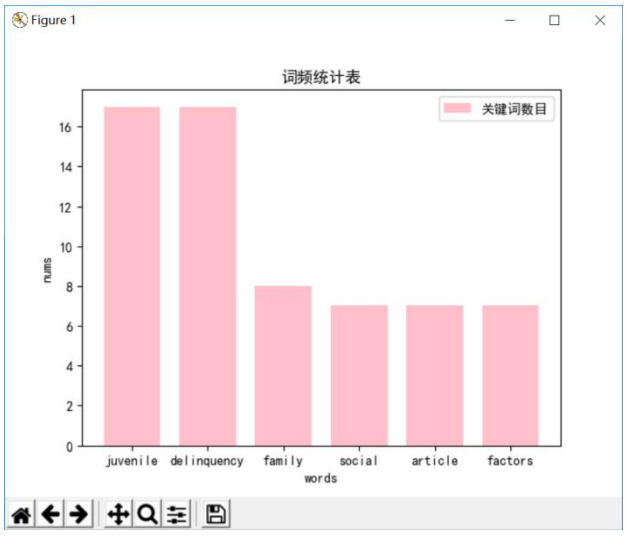




导出文本：将文件导出保存





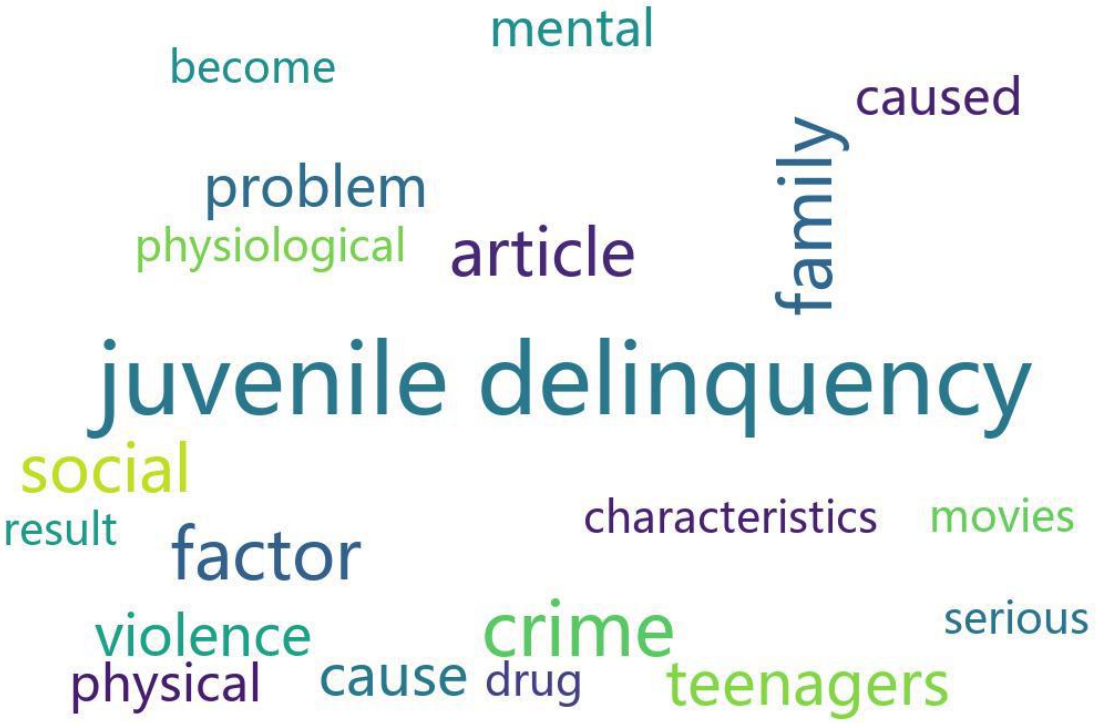


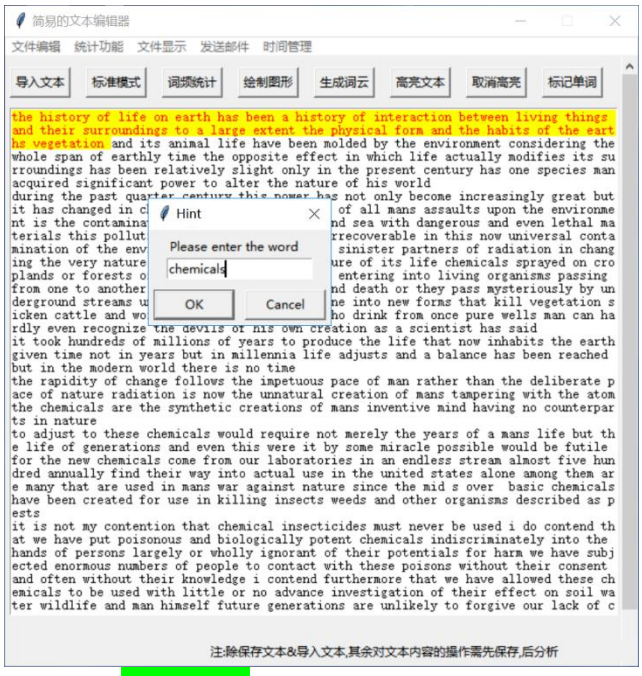
字符统计：统计字符数　词数统计：统计单词数

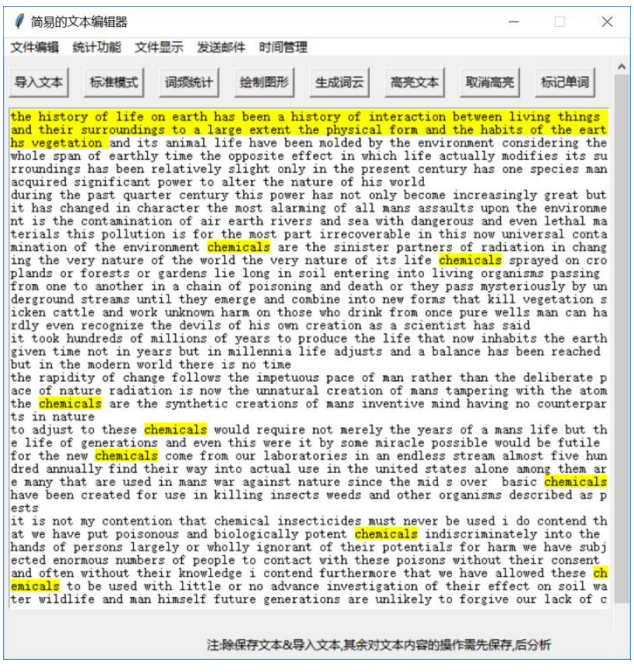
词频统计：点击弹出词频统计结果，下拉至最后有总单词数

绘柱状图：可输入颜色，支持 16 进制颜色代码和颜色单词，得到图像文件

生成云图：点击生成云图得到云图文件



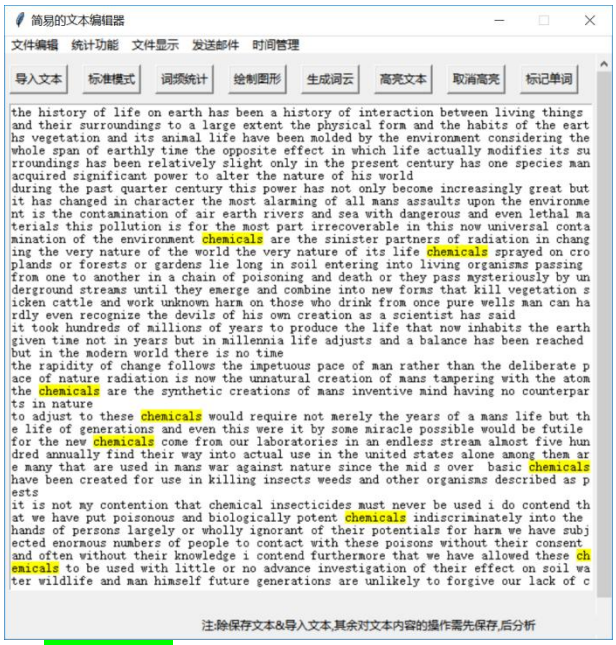


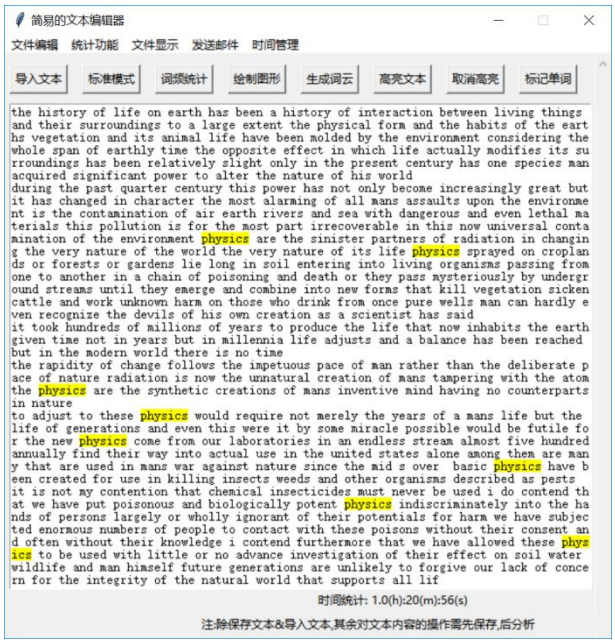


高亮文本：选中需要高亮的文本后点击高亮文本

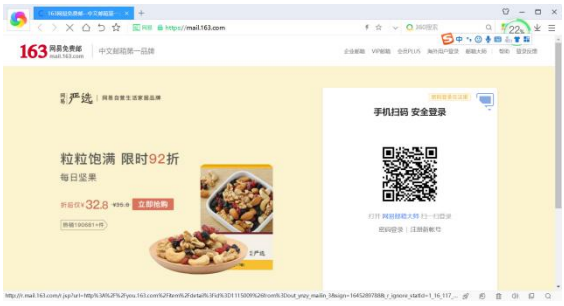
标记单词：点击标记单词，输入要标记的单词和颜色

取消高亮：点击取消高亮，恢复编辑状态





修改单词：点击修改单词，输入修改前后的词，修改后要保存文本，才能让之后的操作基于修改后的文本正常运行



发送邮件：点击选择邮箱，邮箱链接插入首行，点击即可超链接

时间管理：点击时间统计可以获得当前连续操作的时间

六、创新之处

说明你采用了什么独特的方法或者课堂介绍方法之外的方法来解决某个问题；增加了哪些功能；介绍你设计的亮点在于什么地方。

①独特的方法有：通过 Text 的 search 函数配合循环语句，实现搜索全文并高亮的功能；通过正则表达式来实行更高级的匹配功能，在一定程度上解决了诸如搜索 it 会将 with 中的 it 一并返回的问题。

②增加的功能有：高亮指定的单词、生成词云图、保存当前文本、撤销和恢复操作、统计操作时间功能、文件导出功能、将文件打包为小程序……

③亮点有：可以生成词云，获得另一种数据可视化的图形；通过正则表达式提高了查找的准确率；用 PyInstall 库将程序打包为可执行

文件(Windows:exe)，可以让程序在没有安装 Python 解释器的电脑中运行。

七、实验总结

1、在本实验中你遇到了哪些问题？是如何解决的？

①问题：普通通过字符串查询，或通过 tkinter 中的 search 函数查询方式存在着不足，例如我想查询 it，它会将 with 中的 it 一起返回。

解决：通过引入正则表达式的库，用"[\s]*"+指定单词+"[^a-z]"的表达式可以匹配到所需要的空格(或什么都没有)+单词+空格(或换行符)结构的字符串。从而保证匹配到的是一个完整的单词，而不是某个单词中的一部分。

②问题：保存图片时，导出的文件是空白。

解决：将 plt.savefig()语句放到 plt.show()之前

```
plt.savefig(r"bar.jpg")
        plt.show()
```

③问题：在 Spyder 中引用 wordcloud 库的时候，由于电脑上出现了两个 Python，造成在之前在 idle 中装过这个库，但是在 Spyder 中用不了。

解决：在环境变量中添加一个”PYTHONPATH”,路径设置成安装路径，然后把已经装好库的 lib 文件夹复制到 Anaconda 文件夹里面，忽略存在的文件，就可以把 idle 中下载的所有库都弄到 Anaconda 中了。

说明：除了用文字阐述外，最好给出改进前后的源代码。

2、有何收获和体会？你认为大作业题目还可以从哪些角度出题？并说明理由。

①收获和体会：自己上网搜索资料查阅的能力有了非常大的长进；复习了数字，字符串，字典，列表，迭代器等数据结构的使用；练习了进制转换，词频统计等简单的代码；学习了正则表达式的基本用法；学会了 matplotlib，tkinter 等数据可视化处理和 GUI 相结合的方法；尝试用 PyInstaller 库打包了可执行文件……

②可以鼓励同学自己设计游戏，因为我来自福建厦门，我们中秋节有一个特别重要的习俗是博饼(https://baike.so.com/doc/6461758-6675446.html)，它的规则主要是同时掷六个色子，根据不同的点色组合划分不同的奖品等级，每个奖品等级有一定的数量限制，大奖状元又有一系列判别的方法。但是这个游戏在外地的普及

率不高，作为大学生找到博饼需要的色子和碗有一定的难度。这个游戏可以通过计算机来实现，可以实现模拟博饼的过程，有助于推广这项民俗活动。因此我觉得可以鼓励同学自己设计游戏，例如模拟家乡的游戏来推广它，也有助于保护传统文化。

八、课程学习总结

1、课程收获和难点分析

通过学习本课程你收获了什么？你认为现有内容中的难点在于哪些章节？

Python 基本语法和数据结构、栈和队列、算法设计和优化、绘图、GUI 页面

难点：算法章节

2、课程评价

从课程体系的先进性和合理性、课程内容、实验内容、考核环节、课程管理等方面进行评价。

课程较为综合，同时兼顾了计算模型和计算机思维、编程的基本能力、算法设计、人机交互设计，对计算机有了一个初步的认识，培养了对计算机的兴趣。课程内容总体较为紧凑，建议如果可以的话是否可以考虑增加课时或串讲，对实验题进行较为详细的讲解？实验内容有时偏难，但整体还是比较符合课程内容的，总体难度可以延续。考核环节大体合理，但个人觉得大作业的分值与难度相差较大，建议可以做适当的调整。

3、教师授课评价

从工作态度、责任心、是否认真备课、讲课水平、教学质量、是否注重课堂互动、重点难点是否突出、逻辑是否清晰、授课效果、是否关心学生、是否耐心及时解答学生问题等方面进行评价。

工作态度好，富有责任心，备课认真，讲课水平好，教学质量高，课堂互动好，授课效果好，重难点突出，关心学生，在讲解动态规划完后的下课还及时问我是否理解，老师逻辑应该是挺清晰的，但确实作为第一次接触动态规划的我来说理解起来是有一定的难度。

4、助教评价

从工作态度和责任心、是否耐心进行实验辅导、是否认真批改作业并及时反

馈、是否及时解答学生问题等方面进行评价。

工作态度好，富有责任心，实验室能及时给予有效的帮助和辅导，批改作业及时，能准确从中找到需要改进的地方，平常也能及时回答学生学习中存在的问题。

5、课程进一步改进建议

欢迎同学们对于课程体系、教学内容、实验环节、考核方式、大作业形式等各个方面，提出各种建议！我们将认真听取同学们的建议，不断改进课程。

建议能够增加课时数或串讲次数，多在 OJ 上提供习题作为练习，另外希望老师能推荐一些自学是比较好的参考书、网站等，讲解一下提高上网找到所需参考信息的能力。

九、主要参考资料

主要参考百度，CSDN 等网站吧……

感谢刘倩宇助教提供关于用正则表达式匹配时，处理空格和换行符的逆向思路，即匹配非字母的方法。

感谢徐星宇助教提供的关于处理取消高亮的思路，以及查找 matplotlib 和 tkinter 的结合使用的方法。

感谢两位助教扫清了我对于全局变量和局部变量理解上的误区。