实验二说明文档

姓名：姚治宇　　班级：软 52　　学号：2015013228

一. 实验目标

本此实验包含两个部分，一部分是构建平衡树与构建倒排文档，对于 query.txt 文档的关键词进行分析，将结果输到 result.txt 中。另一部分是创建图形化界面，进行用户操作。本文档将先介绍第一部分，然后再详细介绍第二部分与如何对界面进行操作。

特殊说明：本程序应在 Release 条件下运行，用 Debug 模式特别缓慢。

二. 实验环境

系统环境：Windows 10 64 位

IDE：Visual Studio 2012

三. 抽象数据结构的说明

特殊说明：本实验二是在实验一的基础上来编写的，字符串类、栈、字符串链表类在实验一说明文档中都详细的介绍了，这里不再 重复介绍了。但是这里的一些数据结构在实验二中添加了一些新的函数，我在程序的注释中已详细的介绍了，文档中不再介绍。

（1） hashtable(哈希表类)

功能:实现一个链式哈希表

类说明：

成员变量

```
StringLink *elem;                                   以字符串链表为元素的哈希表
int size;                                           哈希表长度
```

成员函数

```
void insert_ptr(CharString &aim);                   往哈希表中插入元素
bool search(CharString &aim);                       哈希表搜索元素，如果找到就返回TRUE反之返回FALSE
unsigned int hash_value(CharString & s);            哈希函数
```

（2） fileLink（文档链表类）

功能：实现倒排文档

类说明：

结构体file（文档）：

```
filename                                            文档名
rank                                                文档地址
fre                                                 词频
next                                                下一个文档节点
class fileLink:
file*head                                           文档链表头结点
file*tail                                           文档链表尾节点
int size                                            文档链表长度
```

成员函数

```
Add                                                 文档链表添加节点
Remove                                              文档链表删除节点
Search                                              文档链表搜索节点
```

Edit	编辑文档链表

（3） AVL平衡树

功能：构建平衡树

TreeNode类：

成员说明	
data	节点储存的字符串
height	节点的高度
frequent	词的频率
left	左孩子
right	右孩子
fileSet	节点的倒排文档

AVLTree类：

成员说明	
root	根节点
成员函数	
Insert_ptr	添加节点
Search_ptr	查找节点
Remove_ptr	删除节点
height	求节点高度
SingleRotateLeft	单左旋
SingleRotateRight	单右旋
DoubleRotateLR	双旋转从左到右
DoubleRotateRL	双旋转从右到左
Max	求最大值
Insert	总体的插入节点
Search	总体的搜索
Remove	总体的删除
Destroy	析构整个平衡树

四. 核心算法说明

实验二主要是在分词过程中将新分出的词插入到平衡树中，在平衡树中构建倒排文档，记录每个网页含有该词的个数。网页分析分词完以后，读取query中的关键词进行分词，将分词结果存到一个字符串链表中。对该链表进行从表头到表尾的遍历，在平衡树中搜索到链表节点所代表的词的倒排文档，将这些关键词的倒排文档进行整合，得出每个含有这些关键词的网页含有这些关键词的个数，然后按要求对节点排序后将结果输出到result.txt中。

五. 流程概述

【加载字典】->【遍历并且记录 input 文件夹下所有的 HTML 格式的文件的文件名->【分析网页内容】-> 【中文分词建立平衡树与倒排文档】- >【对query.txt进行处理将结果输出到result.txt中】->【析构平衡树与哈希表】

六. 输入与输出以及操作的相关说明

将带解析的网页文件存放在 input 文件夹中，在程序目录下添加query.txt文件，将词典文件放在可执行程序目录执行即可。输出时需要耐

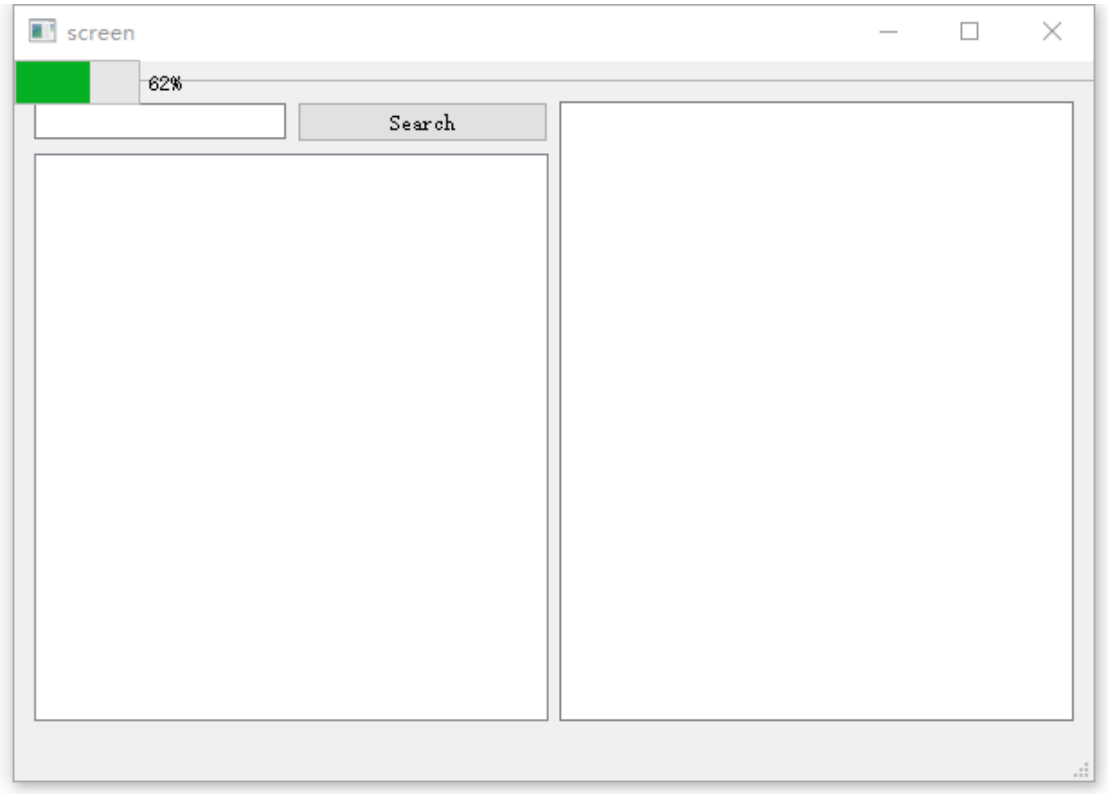
心等待19或20秒钟，便可在result.txt中的到结果。

七. 实验结果

result.txt会显示符合实验要求的结果，每行的实验结果按照关键词出现次数从大到小排序。

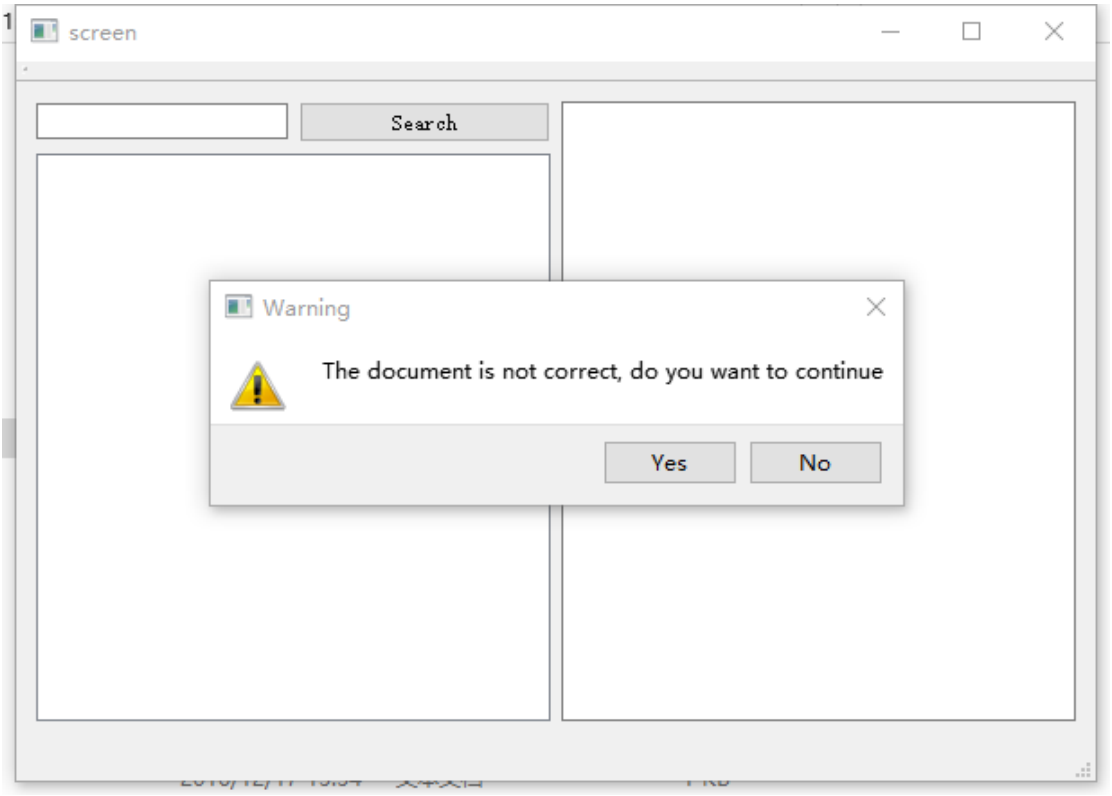
八. UI设计与怎么使用此UI检索关键字

此实验的gui是用QT编写，界面开始出现时会出现一个进度条，只有进度条进度到百分之百时界面才可以使用。如图，



该 UI 除了相应的水平与垂直布局外，主要有这四个部分，搜索框 QLineEdit，文件显示框 QListWidget，搜索按钮 QPushButton 和显示文章内容的 textBrowser。操作的特别说明：

在搜索框输入关键词时，看以往搜索框中输入一整段话，系统会对这段话进行分词，然后根据分词分出的关键字，在左边的框中显示出含有这些分出的关键词的文档名。在输入关键词以后，点击搜索按钮便会在界面左边的窗口显示包含这些关键词的文件的文件名。注意双击这些文件的文件名便可在右边的矿中的到文章的信息，由于没有加载图片资源所以文章内容不会出现图片。文章内容中用户所搜的关键字会显示红色显示出关键词所在的位置。如果想搜其他的关键词清空搜索框再输入关键字点击搜索即可，再次点击新的文件之前右边框会显示上个打开的文章的文章内容。特殊说明：文章内容的网页链接无法被打开。当输入为空或者词没有出现在已载入的网页中会弹出一个对话框进行提示，此时点是或否对话框就会消失然后在搜索框中重新输入即可，如图



九.　实验亮点

（1） 独特的 UI 设计：程序加载会出现进度条显示程序的进度进程，文章内容中的关键词会变红提示关键词位置。

（2） 实验加载 520 个网页的速度比较快，只用了 18 秒（虽然可以在优化）。

（3） 分词算法中可以有效的过滤掉不是中文的单字。

（4） 如果在 gui 中搜索词不在已知网页中出现，会有对话框提醒。

十. 实验体会

通过这次实验我了解到了搜索引擎的原理，也复习了 C++相应的知识与 QT 的使用，学习了 AVL 平衡树与倒排文档的使用，更认识到内存管理的重要性，提高了编程能力。