《数据结构实验报告》

2015080121 软件工程52班 李在弦

一、实验目标

- 实验实现网页解析，正文分词，载入词库到AVL，输入关键词之后对知乎日报文章进行搜索，用户界面，等等

二、实验环境

- 开发环境

操作系统：Windows10

IDE：Visual Studio 2012

编程语言：C++, MFC

三、抽象数据结构说明

- 字符串, 字符串链表数组, 栈 (实验1)

- 平衡二叉树(AVL)

对当前结点的’ID’和要插入的单词的’ID’进行比较，找空间保存单词。保存后如果AVL不平衡，就通过旋转使这AVL平衡。

- 文档链表

每个平衡二叉树的结点里都有一个文档链表，保存当前结点的单词出现的文档名字，出现次数。

四、算法说明以及流程概述

- 网页解析 (实验1)

1.先对标签进行处理，提取有效标签。（同时有‘<’和‘</’的标签）

2. 查找“input”文件夹里面的“*.html”文件。

3. 开始做网页解析（遇到有效标签的‘<’部分就进展，然后开始保存内容。遇到‘</’退栈。）

- 中文分词以及载入词库（有一些改变）

1. 查找“词库”文件夹里面的“词库.dic”文件。

2. 把文件里面的单词都保存在平衡二叉树里。保存方法是这样的，先提取单词的长度，然后一直给‘numWord’加 （一个字的数字是 * (index+1)）的值，index值为长度时停止。然后做 numWord = numWord % 100000, numWord += (单词首字母的相应的数字 * 100000), numWord += (单词首字母的相应的数字 % 1000) * 1000000000, numWord += 单词长度 * 1000000000000 就得到单词的 “ID”值，再把单词保存在平衡二叉树里合适的位置。 （这时，numWord的变量形式为 long long int）

（如果对“联赛”进行处理，（设‘联’= 24635，‘赛’= 36452，‘ID’是‘numWord’）

（下面是我的程序里求‘ID’的部分的代码，‘numWord’表示‘ID’值，‘check.w_size’是单词的长度，‘check.w_str’是单词的字符串）

```
for (i = 1; i <= check.w_size; i++)
    numWord += (i*(check.w_str[i - 1]));

numWord = numWord % 100000;
numWord += check.w_str[0] * 100000;
numWord += (check.w_str[i-1] % 1000) * 1000000000;
numWord += check.w_size * 1000000000000;
```

所以，numWord = ((24635 + 2*36452)%100000) + (24635*100000) + (36452%1000)*1000000000 + (2*1000000000000) = 2454463597539;

3. 取单词同时取文件里面中单词的最大长度，然后打开“*.info”文件读入一行一行。读入一行 之后把文章保存在某个字符串里，然后对它进行分词。做分词用逆向最大匹配

4. 分词操作中如果分词成功，则用文档链表的‘Add’函数添加文档名字和单词在这文档出现次数（如果没有当前文档的文档链表结点，则新建一个结点，出现次数为1，否则出现次数加1（这时候用‘Edit’函数）

- 批量搜索

1. 通过读取“query.txt”（query.exe），还是输入关键词（gui.exe）进行批量搜索。

2. 对关键词再次进行分词，然后把它们的文档链表信息保存在某个string里面，然后如果文档名字一样，则把它们两个的出现次数相加，再赋值给其中一个，然后删除另外一个的。通过这样的操作删除重复的。最后再次排序（排序方法跟说明文档中说的方法一样）。

3. 操作结果输出到“result.txt”(query.exe)，还是输出到输入框中和“result_gui.txt”。

五、输入输出及操作相关说明

1. query.exe

- 把做‘网页解析’功能时需要的输入文件（如 “*.html”）放在“input”文件夹里。

- 把做批量搜索时需要的关键词写在“query.txt”文件里**（保存文件时，文件内容必须要用“UTF8”保存！！）**

- 点击“exe”文件夹里面的“query.exe”可执行文件。（不要改变这文件的位置）

-‘网页解析’和‘中文分词’的结果会生成在“output”文件夹里。

- ‘批量搜索’的结果会生成在跟“query.exe”同目录下。结果文件名是“result.txt”

2. gui.exe

- 把做‘网页解析’功能时需要的输入文件（如 “*.html”）放在“input”文件夹里。

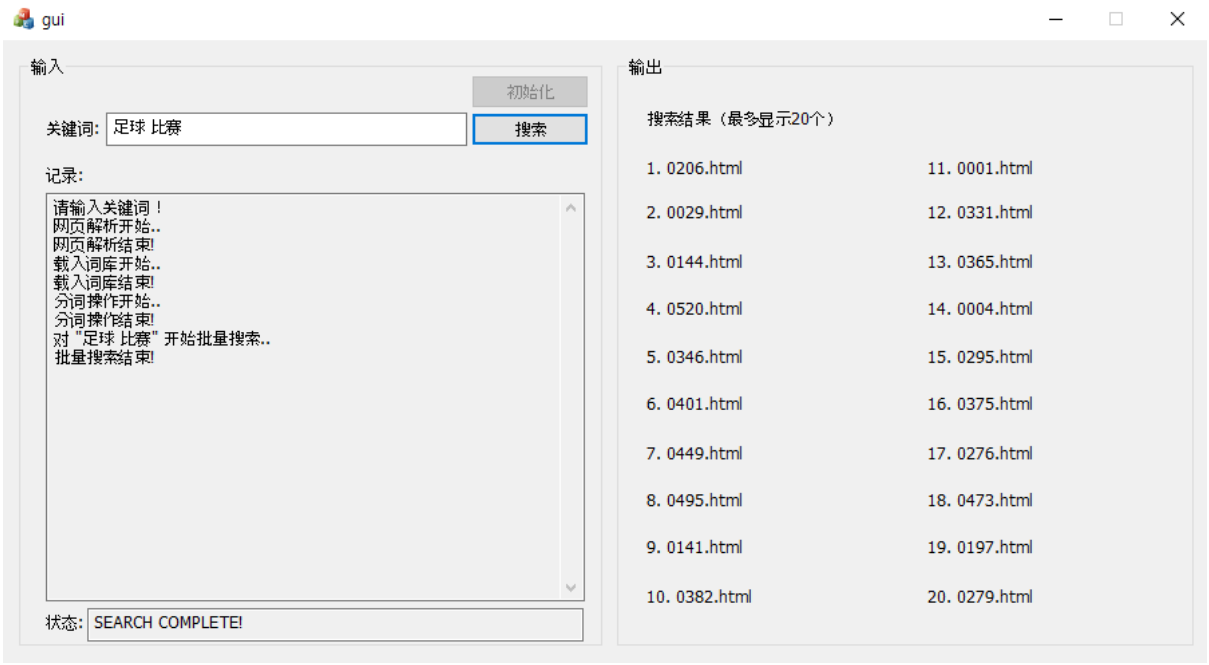
- 执行“gui.exe”之后，先要点击“初始化”按钮进行初始化（网页解析，中文分词，载入词库，等）

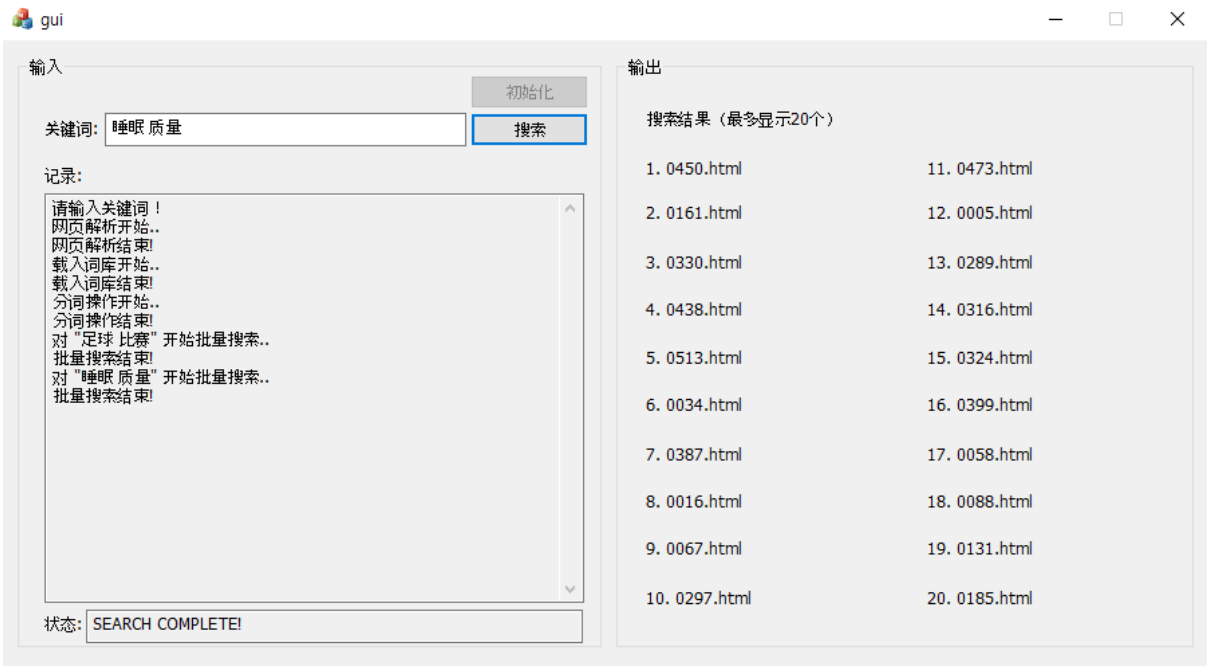
（在进行初始化的过程中，这个用户界面可能会看起来停止了，但是实际上没停止，大概等1分钟左右就初始化完成了。（等到在输入框最下面的状态变成“INITIALIZAION COMPLETE!”，出现这句话表示初始化完成，出现这句话之前什么操作都不要做（移动窗口，点击按钮，输入关键词，等等），直接看着记录。初始化完成之后可以做了））

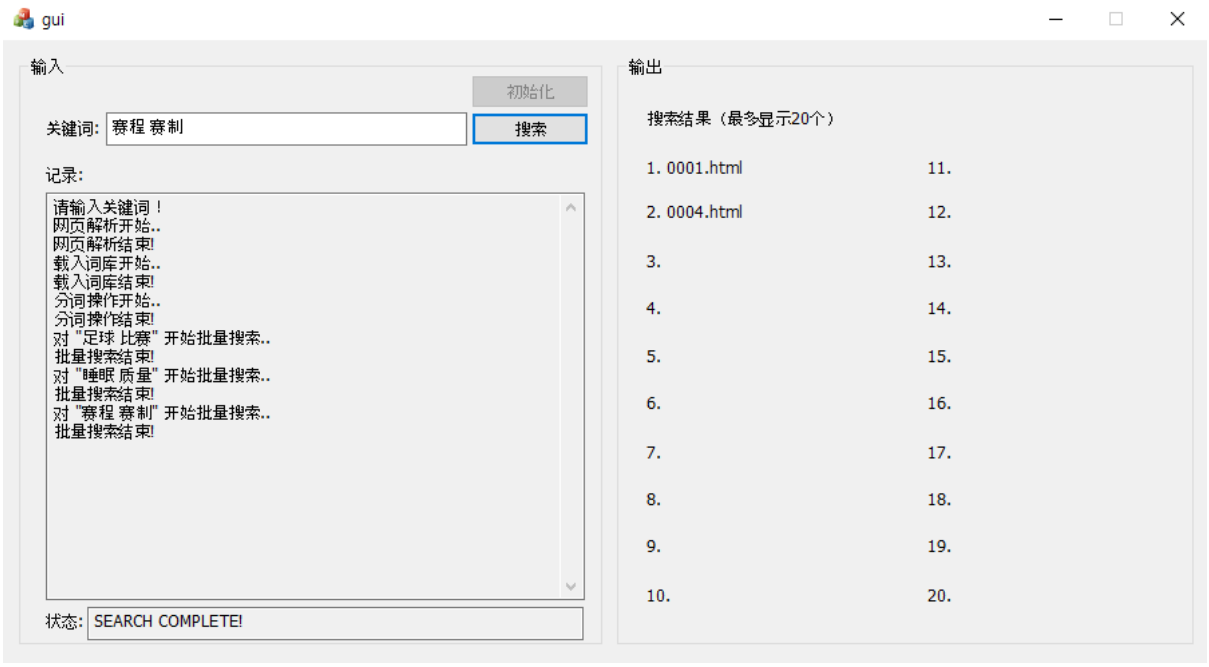
- 出现“INITIALIZATION COMPLETE!”之后‘网页解析’和‘中文分词’的结果会生成在“output”文件夹里。

- 出现“INITIALIZATION COMPLETE!”之后，可以点击‘搜索’按钮用‘批量搜索’功能（进行初始化前不能按‘搜索’按钮）。**（这时候输入格式跟在“query.txt”用的输入格式一样）**

- 搜索结果







（点击文件名（比如“0001.html”），就能打开相应的文件）

- ‘批量搜索’的结果会生成在跟“gui.exe”同目录下。结果文件名是“result_gui.txt”

六、实验结果

- 可以自动查找“input”文件夹里面的“*.html”文件，然后开始做网页解析。

- 做完网页解析之后自动开始查找“词库”文件夹里面的“词库.dic”文件，然后把“词库.dic”里面的所有单词都保存到字符串链表的数组里。

- 保存好之后自动做分词算法。（网页解析，中文分词的结果在“output”文件夹里）

- 通过读取“query.txt”的内容，进行批量搜索。搜索结果在“result.txt”文件里

- 通过输入关键词惊醒批量分词。搜索结果在用户界面的输出框里和“result_gui.txt”文件里。

八、功能亮点说明

七、实验体会

通过这次实验，我会大概理解了像“百度”这种的网页的搜索功能是怎么实现的，还有大概知道了怎么做用户界面的。我确定这两次实验提高了我的编程，设计能力。

（另外，我谢给助教老师…我问的太多,但是助教老师全都回答我了!!^_^ 谢谢！）