# 实验 2 实验报告

## 一、 实验目标

实现基于文本内容的知乎日报文章检索系统，通过从静态网页中提取文章信息并分词，以此为基础构建倒排文档索引，实现输入关键词查询相关文章的检索系统。

## 二、 实验环境

a) 操作系统：Windows 10

b) IDE：Visual Studio 2013

c) 编程语言：C++

（发誓以后一定用和助教环境一样的 IDE 写大作业==）

## 三、 抽象数据结构说明

### a) AVLTree 二叉平衡树

主要实现的操作有

1. Insert 插入结点
2. Search 搜索结点
3. height 获取树的高度
4. Remove 移除该结点
5. Delete 析构函数中释放该对象空间
6. Adjust 旋转树使其平衡的函数

实现的时候由于旋转有多种情况 故写了四个具体的调整函数 依具体情况调用

i. LL_Adjust LL 单旋

ii. RR_Adjust RR 单旋

iii. LR_Adjust LR 双旋

iv. RL_Adjust RL 双旋

b) docList 倒排文档类

主要实现的操作有

1. Add 添加文档 累加该文档的出现次数
2. Search 搜索文档
3. Edit 修改文档 编辑文档出现次数
4. Remove 移除文档
5. Output 向文件中输出链表内容
6. isEmpty 判断当前链表是否为空
7. merge 合并多个关键词搜索结果的链表
8. sort 将搜索结果按照出现关键词的个数以及出现次数降序排列
9. clear 清空文档列表

c) HashTable 哈希表

主要实现的操作有

1. HashTable 提供默认的以及可指定长度的构造函数
2. Hash 用于获取哈希值的函数
3. add 向哈希表中增加因子
4. search 在哈希表中搜索因子

d) fileprocess 一些接口函数

注：部分实验一的函数用哈希表改进提高了效率

| | |
|---|---|
| extractInfo | 提取网页的正文信息 |
| initDictionary | 加载词库 |
| divideWords | 中文分词结果输出 txt |
| init | 初始化存储搜索结果的哈希表 |
| search | 根据给定的关键词 搜索在哪些文档中出现 |

e) Stack（实验一）

f) StringList<CharString>（实验一）

g) CharString（实验一）

## 四、 算法说明

i. 词典索引：使用实验一提供的接口函数获取分词结果，初始化所需的哈希表，分词结果作为树的结点，初始化二叉平衡树

ii. 文档链表更新：向二叉平衡树的结点添加文档，简单来说就是每个结点挂一串它出现的文档编号以及次数（若树中未出现过此关键词，则添加关键词结点并调整平衡树，如果树中有此关键词但其对应文档链表中没有相应文档， 则添加文档结点，否则，文档结点出现次数累加）

iii. 倒排文档以及关键词查询：query.txt 中输入一行待查询的关键词，把这行关键词分解成单个关键词后一次搜索查询该结点挂着的文档。最后将所有结果合并，用倒排文档类中的 merge 函数合并所有出现的文档编号及出现次数，以及 sort 函数将这些信息按照关键字出现次数降序排列，输出结果显示在

exe\\result.txt 中

iv. 用户搜索界面文档信息输出：将原始网页与正文无关的 tag 及其下的内容删除，在目标网页上搜索定位关键词，并 highlight 标识出来（去除无关信息的算法与实验一获取正文信息相同，故此处不展开）

## 五、 流程概述

实验一：载入字典 → (对每个网页) → 读取网页源码 → 利用栈解析成标签列表 → 遍历标签列表获取关键信息 → 中文分词 → 输出分词结果文件 →实验二：利用分词结果建立二叉平衡树 → 向分词结点添加文档并记录出现次数 → 读取待查询关键词的文件并分解关键词 → 搜索对应的结点，将(docID,OccurTimes)结果合并并排序 → 输出得到的查询结果（用户界面/txt 文本）

## 六、 输入输出及操作相关说明

a) 输入

在 exe\\query.txt 中输入待查的关键词 每行可有多个关键词 用空格隔开 可有若干行搜索关键词

b) 输出

运行 query.exe 对给定的 input 文件夹中的网页进行搜索，运行完毕程序自动退出后便可点开 result.txt 查看具体的所在文档及出现次数信息

注：每行显示的是该行对应关键词总共出现的位置和次数，关键词之间是或不是与关系

运行 gui.exe 用户可在搜索框中搜索一行若干个关键词，用空格隔开，点

击搜索键后文档位置及关键词出现次数信息显示在下方的搜索结果框中，若要查看相关原文，双击对应的 docID 便会显示出原文的正文部分

## 七、 实验结果

a) 通过向 query.txt 中输入关键词，运行 query.exe 可在 result.txt 中得到搜索结果

b) 通过运行 gui.exe 模拟搜索引擎查询关键词并得到网页结果

## 八、 功能亮点

a) 用户界面中显示了倒排文档的进度条

b) 用户界面关键词所在原文中凡是出现的地方 highlight 显示

c) 对特殊的需要忽略的文段进行了处理

d) 改进了实验一中的接口函数，实现哈希表管理词库，以及哈希表对词典进行索引，提高查找效率

e) 写了平衡树中的可选函数

## 九、 实验体会

当我准备开始写实验二的时候，突然发现实验一中一些当时觉得不严重的小问题会对实验二产生较大影响，于是只好改（bu）进（jiu）实验一的接口函数（毕竟分词也是影响实验结果时间的），#论实验一写好的重要性

实验最大的收获是以前觉得及时释放空间不是一件重要的事情，但是在数据量变大的时候就显得尤为关键了，不只是拖慢速度，可能还会直接跑不下去了。所以当我莫名其妙debug不出来的时候，条件反射去检查有没有把用完的空间释放掉了。以及debug的时候对于指针和引用也有了更深的认识。

最深刻的体会是 !!! 我高估了自己写实验报告的速度 !!! 作为一个处女座龟毛得要死，折腾半天才写好，不过讲道理写报告是对自己写代码的思路的一个梳理和总结，写完报告才觉得自己对实验的框架有了总体的认识。