实验报告

软件学院 2016013258 王泽宇

一、 实验目标

本学期实验目标是实现一个给予文本内容的销售线索检索。

本次实验的目标是在分析网页的基础上，使用二叉搜索树或哈希表实现对解析内容的检索，并进行可视化显示

二、 实验环境

操作系统：Win10

IDE：Visual Studio 2015

编程语言：C++（算法部分），Javascript，Electron 1.7.10（GUI 部分）

三、 抽象数据结构说明

DSCP 项目是 query.exe 的源代码，在此仅介绍相对上次实验结果增加的数据结构。

Inverted_DOC 类：倒排文档类，用平衡二叉树实现。

成员变量：

根节点 root、空节点 NIL，倒排文档大小 size。

成员函数：

1、**插入函数 Insert**（参数：单词，单词 ID，文档 ID，单词出现次数）用于向倒排文档中插入新的节点，并维护平衡二叉树性质；

2、**删除函数 Remove**（参数：节点指针）用于删除一个节点，并维护平衡二叉树性质；

3、**调整函数 Adjust**（参数：节点指针）调整该节点使之不违反平衡二叉树性质。

4、**修改函数 Edit**（参数：单词，文档 ID，修改后出现次数）修改某个单词在特定文档的出现次数；

5、查找前驱后继节点函数 precursor，successor（参数：节点指针），返回该节点的前驱节点或者后继节点；

6、**查找函数 Search**（参数：单词），用于返回该单词的节点指针，如果未找到则返回空节点指针 NIL；

7、**重载[]符号**，参数（下标）：单词，实现与查找函数 Search 类似的功能，但可以提高的可读性。

Inverted_DOC_Node 类：单排文档节点类，是平衡树的节点类。

成员变量：

单词 Term，单词 ID：TermID，单词在多少文章中出现 DF，单词出现总次数 Occur，以该节点为

根的子树大小 size，左右孩子节点指针 son，父结点指针 p，文档链表 doclist。

成员函数：

1、**插入函数** Insert（参数：文档编号，出现次数）向文档链表增加新的文档。

2、**删除函数** Remove（参数：文档编号），在文档链表中删除给定文档

3、**编辑函数** Edit(参数:文档编号,出现次数)，修改单词在给定文档里的出现次数

4、**获取文档链表函数** get_doclist，返回文档链表指针。

DocList 类：文档链表类。

成员变量：

单词 ID TermID，文档链表头指针 head。

成员函数：

1、**添加节点函数** Add(参数:文档编号,出现次数)，在保证链表顺序的情况下，向文档链表中插入新节点。

2、**删除节点函数** Remove（参数：文档编号），在文档链表中删除对应的文档。

3、**修改节点函数** Edit(参数:文档编号,出现次数)，将文档链表中的该文档出现次数修改为特定值。

4、**查找函数** Search（参数：文档编号），返回该文档节点的指针。

5、**获取文档链表头指针函数** get_head，返回文档链表头指针。

6、**实现拷贝构造函数并重载=**，保证 DocList 类型的变量在复制时执行值复制。

DocList_Node 类：文档链表节点类。

成员变量：

文档编号 DocID，单词出现次数 Times，下一个节点的指针 nxt。

成员函数：

1、**获取文档编号** get_DocID，返回文档编号；

2、**获取单词出现次数** get_Times，返回单词出现次数

3、**获取下一个结点指针** get_next，返回下一个节点指针；

4、**实现拷贝构造函数并重载=**，保证 DocList_Node 类型的变量在复制时执行值复制。

Inverted_DOC_HASH 类：倒排文档类，哈希表(RK 哈希)实现

成员变量：

HASHN,HASHM 分别是哈希的基数和模数，哈希表 h，单词在多少文章中出现 DF，单词出现总次数

Occur，文档链表 doclist。

成员函数：

1、**插入函数 Insert**（参数：单词，文档编号，出现次数），将此单词插入到哈希表中。

2、**查找函数 Search**（参数：单词），查找单词，返回该单词的文档链表。

3、**修改函数 Edit**（参数：单词，文档编号，出现次数），在文档链表中修改次单词的出现次数。

ThreadPool.h 实现了多线程池，在代码中用于多线程解析网页结构，ThreadPool.h 参考自 https://github.com/progschj/ThreadPool。

*DSCPdll 项目*生成与 GUI.exe 交互的动态链接库 DSCPdll.dll，所有的库函数均与 DSCP 项目相同。

*GUI 项目*制作了 gui 界面，用 electron 实现，因本次实验重点在算法部分，故不作过多阐释，框架参考自 https://github.com/sunziping2016/zhihu-search-engine。

四、 算法说明

从上次实验得到的网页解析内容，建成检索表（需选择使用平衡二叉树还是哈希表），再从 query.txt 中读取查询信息，对于一条查询，从检索表中得到各个关键词的文档链表并合并，按出现种类由大至小、出现次数由大至小的顺序排序输出导 result.txt。

哈希表实现：采用 RK 哈希，选定进制 K 和模数 m，将单词$S_0S_1S_2S_3\ldots S_n$映射为整数$\sum K^i S_i \bmod m$，并使用线性探测法处理冲突，期望复杂度$O(1)$。

平衡二叉树实现：采用红黑树，每个节点分为红色黑色，保证不存在相邻的红色节点，并且根节点和每个叶节点都是黑色，以此保证树高是$O(\log n)$级别，在插入和删除节点时，分情况讨论，对相关节点进行旋转、重染色等操作使红黑树重新平衡，限于篇幅，具体情况不在详细讨论，可参考 https://en.wikipedia.org/wiki/Red%E2%80%93black_tree。

五、 流程概述

[载入词库]->[抓取网页]->[解析网页]->[提取关键信息并分词]->[根据分词结果建立索引表]->[读取 query.txt 的查询]->[在索引表中查找结果并整合输出]

六、 输入输出及相关操作说明

query.exe： 双击运行，程序会从./input.txt/url.csv 中读取地址并解析好网页，随后提示要求输入检索方式，输入 BBT 表示平衡二叉树索引方式，输入 Hash 表示哈希表索引；程序最终会针对同级目录下查询文件 query.txt，在同级目录下给出查询结果文件 result.txt。

gui.exe： 双击运行，在顶部加载进度条消失后，可在输入框内输入任何想要搜索的内容，点击按钮或回车进行搜索，程序会先对搜索内容分词，并显示搜索内容。

注意：由于 gui.exe 是读取./outout/result.csv 作为输入的，在运行 gui 之前请保证./outout/result.csv 存在且正确

七、 实验结果

实验结果基本符合预期，可以实现发帖内容的搜索与可视化。

在平衡二叉树的索引表建立大约花了25ms,哈希表索引的建立大约花了50ms。

八、 功能亮点说明

多线程爬取网页，加快速度。

采用哈希表和平衡二叉树两种方式建立索引并比较。

Gui 界面里对输入的内容进行分词，使用者可以输入日常化语言进行查询。

九、 实验体会

本次实验的基础算法部分实现起来比较容易。在扩展项目中，动态链接库的建立还是让我产生了不少困惑。由于需要在两种语言中进行交互，所以类型的统一等问题也给我制造了不少困难。但是，第一次用前端语言写 GUI 的经历还是让我受益颇多。总之，这次实验还是非常有收获的。

另外，按照理论复杂度，用哈希表建立索引表的理论复杂度要低于平衡二叉树，但是实验结果却相反，原因应是 2000 个网页的分词结果数据量过小，哈希表还不能体现出其自身的优势。