数据结构第一次实验报告

一、实验目标

1. 本学期数据结构课程实验希望实现基于挖掘机论坛网页抓取分析的销售线索检索
2. 第一次实验的基本目标包括实现数据结构：栈、字符串、字符串链表
3. 使用自实现的数据结构实现网页的抓取与栈结构分析，提取关键内容
4. 使用自实现的数据结构实现文本内容的分词

二、实验环境

1. 基本功能模块开发（数据结构的实现与测试、栈结构分析网页、中文分词）

操作系统：macOS High Sierra
编译器：g++
编程语言：C++

2. 编码转换模块

操作系统：Windows7
IDE：Visual Studio 2012
编程语言：C++

3. 各功能模块整合、测试、发布

操作系统：Windows7
IDE：Visual Studio 2012
编程语言：C++

三、抽象数据结构说明（仅列出重要成员）

1.栈（类模板） class Stack;

私有成员变量：

1. T* ptrtop; 指向栈顶的指针
2. int size; 栈当前的容量
3. int num; 栈内元素的数量

公有类方法：

1. Stack(); 构造函数，初始栈空间为 init_size
2. void push(T ele); 向栈顶添加元素
3. void pop(); 弹出栈顶元素
4. T top(); 返回栈顶元素值

2.字符串 class CharString;

私有成员变量：

1. int size; 申请的空间大小
2. int num; 字符串的长度，即使用的空间大小
3. char* base; 指向字符串的指针

公有类方法：

1. CharString(***); 构造函数，可以使用的参数如下：空（默认申请 init_size 的空间）、char、char , int、string&、char*
2. CharString(const CharString& cs); 拷贝构造函数，防止编译器生成的默认拷贝函数的直接复制指针的行为

3. int indexOf(CharString& targetString,int pos)；返回从 pos 开始的第一个匹配 targetString 的位置的 index（pos 从 0 开始数，以下表示 index 的都是从 0 开始数）
4. CharString subString(int pos,int len)；返回从 pos 位置开始长度为 len 的字符串（包含第 pos 个）
5. CharString* concat(CharString& targetString)；将 targetString 连接到本身之后，返回一个连接好的字符串的指针，这个操作并不会改变本身或者 targetString
6. void assign(CharString& targetString)；赋值，将 targetString 赋值给 this
7. void append(***)；将某个字符（串）添加到末尾，可传入的参数如下：char、char*、string&、CharString&
8. char getIndex(int index) const；返回第 index 个字符
9. bool is_equal(CharString target) const；如果两字符串相同返回 true，否则返回 false
10. friend std::ostream& operator << (std::ostream& out,CharString& obj)；重载标准输出流操作符

3.双向循环链表（类模板）class Link;

私有成员变量：

1. T elem；当前节点存储的信息
2. Link* prev；前一个节点的指针
3. Link* next；后一个节点的指针

公有类方法：

1. Link()；构造函数，初始化操作，将 prev 和 next 指针指向自己
2. void add(T& elem)；向链表最后添加一个节点
3. T remove(int index)；将第 index 个节点删除
4. T remove(Link* rm)；将 rm 指向的节点删除
5. Link* search(T& target)；查找链表中是否有值等于 target 的节点，如果有就返回该节点指针，否则返回 nullptr
6. void insert(Link* pos,T& elem)；在 pos 节点之前插入一个节点，元素为 elem
7. Link* getNode(int index)；返回第 index 个节点的指针
8. void printLink(std::ostream& os)；打印链表
9. T getElem()；返回当前节点存储的数据

4.Unicode 到 GBK 转码类 class UnicodeToGBK;

私有成员变量

1. std::map<int,int> uni2gbk；unicode 与 GBK 的编码对应表

公有类方法

1. UnicodeToGBK()；默认构造函数，读取 UnicodeToGBK.csv 文件，初始化 Unicode 与 GBK 的对应 std::map
2. ~UnicodeToGBK()；析构函数，释放空间
3. int* convert_int(char* unicode,int length 传入的字符串格式为"&#xxxxx;..."，返回值为每个中文字对应的两个窄字符的取值
4. int* convert_single(int uni)；传入的为 unicode 编码，返回值为该编码的中文
5. 字对应的两个窄字符的取值

5.中文分词类

私有成员变量

1. std::set<CharString> dic; 存储普通词典的词条
2. std::set<CharString> super_dic; 存储专业词汇词典的词条

公有类方法

1. wordSegmentation(string filename,string super_filename); 初始化词典，第一个参数为普通词典，第二个参数为专业词汇词典
2. ~wordSegmentation(); 析构函数，释放空间
3. Link<CharString>* divideWords(CharString& incstring,int max); 参数为待分词的字符串以及逆向最大匹配法中的最大匹配字符数，返回一个字符串列表

四、算法说明

1.网页解析

从头开始遍历存储在CharString类型中的*html*文件，每当遇到<就进入函数label get_label_info(CharString& html,int pos);，返回当前标签的信息（标签类型，class 值），将返回的 label 压栈。如果遇到目标标签（如 ）就进入相应函数提取信息（如 int find_basic_info(Stack<label>& m_stack,CharString& html,int pos,CharString* info); 传入参数为栈的引用，字符串引用，当前遍历到的位置，存储提取信息的字符串数组，返回值为目标标签闭合位置）。如果遇到</或者/>的结构就调用 CharString get_close_info(CharString& html,int pos); ，传入待分析字符串引用和当前遍历位置，返回该闭合标签的信息，与栈顶的标签进行匹配，如果匹配成功就弹出栈顶。遍历结束时提取信息完成。

2.中文分词

采用逆向最大匹配法。从句子末尾开始，每次从目标句中取出长度为 len 的词组，在词典中进行匹配，如果成功则从成功匹配的词组前一个字重复匹配过程，如果失败则取出长度为 len-1 的词组，在词典中匹配。如果 len==1 直接将这一个字单独作为分词的一个部分。

在匹配时优先匹配专业词汇，再匹配普通词典。

在处理中文的时候有一点要特别注意，每一个中文字符占 *2* 个 char，所以每次要取出长度正确的 CharString 保证里面存储的是完整的中文字。

3.Unicode 转 GBK

包装成一个类 class UnicodeToGBK;从网络上找到一张 *Unicode* 与 *GBK* 的映射表，在初始化的时候使用 std::map 加载映射表。将待转换的字符的 *Unicode* 编码传给类方法，取出对应的 *GBK* 编码。将对应的 *GBK* 编码高八位和低八位取出来分别存入 int 型中，返回两个 int 型。在使用的时候需要将这两个 int 型转为 char 型，高八位对应的 char 在先低八位对应的 char 在后依次放入 CharString 中。

五、流程概述

1. 初始化词典和 Unicode 到 GBK 转换表
2. 从待解析 url 列表中获取一个 url，获取页面源代码后存入暂时文件中。
3. 从暂时文件中读取 html 源代码存入 CharString 中，开始栈结构解析页面
4. 将提取内容中的 Unicode 编码部分转换为 GBK 编码
5. 将标题部分和发帖内容做中文分词
6. 将所有信息输出到输出文件中

六、输入输出及相关操作说明

1. 请将输入文件 url.csv 放置在 exe/input 文件夹下，输入文件格式请和示例文件一致，否则将导致输入错误。

2. 放置好输入文件后直接运行 Experiment1.exe 可执行文件。
3. 请在在 exe/output 文件夹下读取输出结果。

七、实验结果

1. 给定的 100 个网页的解析结果表明，发帖大类，发帖小类，发帖标题，发帖内容，发帖人，发帖日期，发帖类型的解析结果出色，基本实现了关键信息的无遗漏提取。
2. 通过逆向最大匹配法，使用助教所给词典得出的结果并不十分理想，一方面是因为发帖内容的语言比较零碎，口语化；另一方面是因为词典的词组具有局限性。在使用了自己的词典（功能亮点部分将会介绍）后结果稍微理想一些。

八、功能亮点说明

1. 通过 python 爬虫爬取 20000 个帖子的文本内容，进行分词词频统计，将高频词提取后扩充词库。具体代码与词频结果请参见 src/scraping 文件夹下 README.md 文件。
2. 通过 GBK 与 Unicode 对照表自实现了转码器。

九、实验体会

1. 网页分析部分和数据结构实现其实并没有什么难度，也很快就写完了。
2. 困难的部分在于网页中文本编码不一致造成的处理困难，以及该网站的不同帖子有时会有不同的 html 结构，导致了处理上的一些麻烦。
3. 中文分词在不介入机器学习，词库不理想的情况下，很难做到比较理想的结果，鉴于本人对于机器学习并不熟悉，所以选择了通过抓取大量的网页数据进行词频统计扩充词库。
4. 个人有一点建议，建议之后可以找些外国论坛或者是文本编码比较一致的中文论坛，减少同学们耗费在文本编码处理上的时间，将时间更多的放在分词算法，网页结构解析算法的优化，以及更多有趣的数据分析上。

姓名：洪方舟
学号：2016013259
班级：软件 62
E-mail：hongfz16@163.com
Tel：15062727188