《数据结构与算法》课程实验

基于文本内容的知乎日报文章检索

（第一部分）

教师：张力老师

助教：陈凯、李璇

2016 年 10 月 13 日

0、摘要：

- 使用 C++语言，实现实验要求的数据结构和相关算法。
- 完成知乎日报文章网页信息的提取及分词；
- 按时提交实验结果、源代码、实验文档。

1、 实验目的

本次实验通过实现基于文本内容的知乎日报文章检索系统。**概括地讲，首先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。**

实验中需要实现的算法和功能有：分析网页结构并提取信息、中文分词、索引机制的实现、倒排文档及查询系统的构建。希望通过常用的数据结构与算法进行训练，锻炼同学们的实际编程能力。

实验中涉及到的数据结构有：字符串、栈、链表、树、哈希表等。

总体来讲，通过课程实验，希望达到以下三个目标：

1）对课堂上的基础数据结构类型进行训练；

2）将数据结构知识应用到实际软件开发中，体会其重要性和广泛用途；

3）通过实验培养学生全方面思考的能力，面对困难解决实际问题的能力。

2、 实验环境

开发环境（建议）

- 操作系统：Windows7/8
- IDE：Visual Studio 2012（建议）/ Visual Studio 2010
- 编程语言：C++

测试环境（检查标准）

- Windows 8 企业版 64 位
- CPU：Intel® Core(TM) i7-2600 CPU @ 3.40GHz 3.70GHz
- 内存：8.00GB
- IDE：Visual Studio 2012

3、评分方案

首先，遵守学术职业素养的基本原则：如果在提交的实验结果中发现相互抄袭现象，被抄袭和抄袭者的本次实验分数均为 0 分。如果发现使用第三方代码的情况，若未直接注明出处，则视为抄袭，抄袭者的本次实验分数为 0 分，若注明了，则根据使用情况酌情考虑扣分。

实验评分将依照两部分进行：系统运行结果、系统实现内容。

系统运行结果是指助教正确运行提交的可执行文件，并根据运行结果进行评分。评分标准包括系统的是否可执行，输出结果是否正确，系统的效率等；

系统实现内容是指代码是否实现了要求的数据结构与算法，助教将会检查实验报告及源码实现进行综合给分。

具体的实验评分项将在实验内容中说明。

实验中鼓励创新，在完成基础任务的情况下，任何与实验相关的、有意义的创新都将有机会获得额外加分。加分项上不封顶，但与基础得分的总分不超过 110 分（基础满分 100 分）。

4、实验提交

最终实验要求提交 3 部分内容，请参考以下说明按照文件夹进行组织。

在实验材料中，包含一个提交样例目录[提交样例]。提交作业时请根据样例目录中文件夹[2014213455_陈凯_实验 1]的组织和命名格式，在其子目录下放置对应内容。

1. 源代码：放置 VS 项目工程，务必删除.sdf 等大文件和编译产生的结果文件。
2. 可执行文件：放置可以直接运行的可执行文件，该目录下应该同时包含 readme 说明文件及相关配置和输入文件。具体配置文件放置方式请参考第 5 章 C 节测试方案的要求。
3. 实验报告：pdf 格式，不超过 4 页，正文使用宋体小四号字，单倍行距；实验报告中要求提供包括但不仅限于以下信息：实验目标、实验环境、抽象数据结构说明、算法说明、实验流程、操作说明、实验结果、功能亮点、实验体会；言简意骇阐述清楚即可，不要复制代码或截图代码。鼓励图文并茂辅助说明，但注意引用图片的版权。

注：未按照要求格式提交的作业，会酌情扣分。

5、实验内容

本次实验将有两次实验组成。

实验 1 主要实现一些基础数据结构，并通过对网页文件的解析，实现知乎日报文章信息的提取与文本分词；

实验 2 在实验 1 的基础上进行，利用实验 1 的接口，以 500 个（暂定）网页作为数据库，实现根据输入内容在数据库中检索文章的功能。

5.1 实验 1——网页信息的提取与分词

A. 实验目标：

本次实验是整个课程实验的第一部分，目标是从网页文件中提取文章的关键信息。

具体来讲，给定 10 个特定规则的网页，要求程序**使用栈结构标签全遍历地解析网页语法结构**，提取网页的关键信息。在本次实验中只需要提取文章标题、问题名称、作者名称、文章正文等信息；当信息提取完成后，需要用分词算法对关键信息进行分词，将最后的结果保存到文件。

B. 数据结构及算法要求：

首先，在实验中涉及到的数据结构与算法有：

- 数据结构：字符串、链表、栈
- 算法：网页解析、中文分词

本次实验中，要求同学们实现三种数据结构：栈(Stack)、字符串(CharString)和字符串链表（CharStringLink)。每项数据结构需要实现的基本操作如下：

数据结构	函数名称	函数功能
栈	push	压栈
	pop	退栈
	top	获取栈顶元素
	empty	判断栈是否为空
字符串	indexOf	查找子串的位置
	substring	截取字符串
	concat	连接字符串
	assign(或重载操作符=)	赋值
字符串链表	add	添加元素
	remove	删除元素
	search	查找某元素位置

实验评分过程中，将严格参照上述数据结构的功能进行评分。建议每种数据结构单独创建一个类，在类中实现上述函数，并添加相关注释。

在之后的功能实现中，除特殊声明的步骤外，所使用的数据结构必须是以上自行实现的数据结构类型。

注：在执行文件读写等最基本的操作时，可以使用C++自带的字符串类型进行读写，但不能使用与其相关的系统函数对其进行其它变化操作。例如，读取完成后，应立刻使用自定义的字符串数据结构对文件内容进行保存，此后的操作都在自定义字符串数据结构上完成。

完成基础数据结构的实现后，可进行本次实验任务的开发。本次实验要求实现的功能如下：

1. 网页文件解析和提取：要求栈结构，对 html 文件的语法结构进行标签全遍历的解析；并在解析 html 文件语法结构的同时，根据特定的 html 标签及属性提取网页中的关键信息；

2. 分词算法：使用分词算法对提取到的信息进行分词；(可以对分词算法和分词的词库进行优化，例如数字匹配，姓名匹配、去掉停用词(自行选择停用词表)、将同一文章中出现频率较大的词添加进词库等)

注：除特殊声明的相关实验步骤外，以上数据结构和算法需要自行实现。

其中第 1 部分要求同学们使用栈结构实现网页文件的解析，这部分内容将在附录 1 中说明；中文分词相关算法参考附录 2 相关说明。

C. 测试方案：

实验完成时，助教需要能够使用提交的可执行文件**直接**获取实验结果。

输入数据：

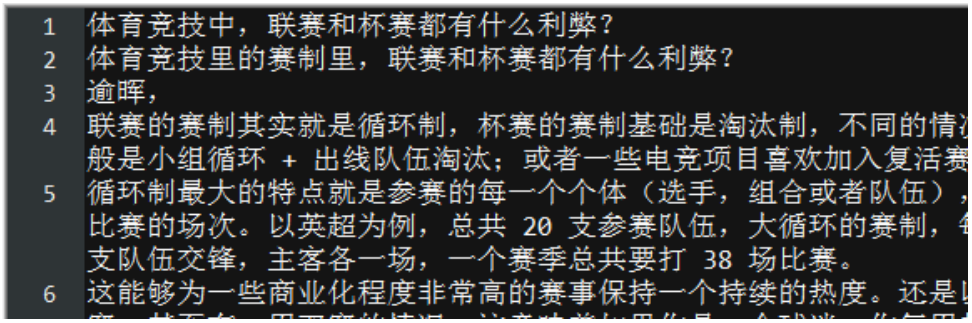
- n 个网页文件（[file_name].html）；

输出结果：

- n 个文章信息文件：[file_name].info；
- n 个分词结果文件：[file_name].txt；

即，可执行程序可为其中每个网页文件都生成对应的的文章信息文件和分词结果文件。例如文件名为 0001.html 的网页文件将生成 0001.info 和 0001.txt 两个文件，分别保存对应的文章信息和分词结果。

文章信息文件的格式如下：文件中每行内容依次为文章标题、问题名称、作者、文章正文。比如对于 0001.html 文件提取文章信息后，得到的 0001.info 文件中内容如下：



分词结果文件的格式如下：每一行为一个词语。比如对于 0001.html 文件的信息进行分词后，得到的 0001.txt 文件中内容如下

注：第一次实验中，只要求对文章正文内容进行分词。

测试方法：

为确保助教能够顺利对实验进行测试，请**务必注意以下几点：**

1. 助教将在可执行程序的同级目录下，放置一个 input 文件夹，其中包含若干个输入文件[file_name].html。（注意，这里文件数量并不确定，且文件名不一定是 0001-0010，需要同学们实现获取当前文件夹下所有文件的功能。）

2. 在可执行程序的同级目录下，自定义放置所需要的配置文件、词库等其它所需输入数据。放置路径可自定义，但确保使用的是**可执行程序的相对路径**，以保证可移植性。（即，项目移植到另外一台电脑上，也可以顺利执行。）

3. 预留一个 output 文件夹，用于放置输出文件。结果文件的命名方式及内容格式参考上述要求。

测试时，助教将使用脚本自动执行可执行程序，并读取 output 文件夹内的结果文件内容进行评分。因此，请大家务必按照上述要求组织提交内容。

注：如果助教无法通过上述配置获取实验结果，最终所得分数将扣除 30%。

D. 评分细则：

助教将根据以下模块对实验进行评分。

模块	内容	分数
数据结构	栈	15%
	字符串	20%
	字符串链表	10%
功能	网页解析	20%
	信息提取结果	5%
	分词算法	15%
文档与代码风格	相关文档	10%
	代码风格与注释	5%
*亮点与加分项	相关特色功能点	10%

助教将根提交代码和文档对上述功能进行评分，并根据程序运行的结果得到最终分数。如之前提及，若程序无法正常运行，将在初始得分的基础上乘以 0.7 得到最终分数。

亮点与加分项需要在文档中说明，加分将会根据实现的亮点进行评判。

G. 预留接口

在实验 1 完成后，需要为实验 2 预留 3 个接口：

1. extractInfo(…)：该接口执行解析网页操作，返回结果自行定义，需要包含网页文章的相关信息；
2. initDictionary(…)：该接口执行载入词库等初始化操作；
3. divideWords(…)：该接口执行分词操作，返回结果保存为字符串链表。

这样，在实验 2 开始时，只需要使用上述 3 个接口，就可以完成初始化操作，并获取每个页面的文本信息和分词结果，为实验 2 构建倒排文档做好了充分的准备。

5.2 实验 2——知乎日报文章检索（待补充）

6、其它事项

实验报告：

除了代码工程之外，实验报告是体现你工作量的重要工具，**也是助教进行实验评分的主要依据之一**。请同学们合理分配写代码和实验报告的时间，实验报告以简洁清晰为主。

代码注释：

在实际工程开发中，代码注释非常重要。在此不给同学们规定哪里一定要写注释，但希望同学们在关键的变量、方法、算法步骤处使用注释进行简单说明，帮助他人（很可能是几年以后的你自己）理解代码的功能。

作业迟交：

作业若未能按时在网络学堂上提交，可通过邮件或其他方式提交给助教。迟交的时间点按照助教确认为准。若出现迟交作业，需要在作业评分的基础上扣除相应分数，按照迟交的天数，扣分依次为 5%、15%、30%、50%、70%、100%。迟交天数按照向上取整计算。

其它未尽事宜，将在网络学堂上补充通知，谢谢。

附录 1：网页解析方法说明

网页解析依据的是 HTML 文件所具有的规则。

一般来说，HTML 语法由不同的标签组成，如 head、body、p、div 等。HTML 文件可利用栈结构进行解析。HTML 文件的具体语法及相关知识可从互联网上获得，这里不再赘述。

本次实验中，我们需要提取的是网页的文章关键信息，包括文章标题、问题名称、作者、文章正文等。我们处理的是来自知乎日报的网页，具体分析知乎日报页面 HTML 源代码，可以发现部分内容如下：

```
65  <div class="question">
66    <h2 class="question-title">体育竞技里的赛制里，联赛和杯赛都有什么利弊？ </h2>
67
68  <div class="answer">
69
70  <div class="meta">
71    <img class="avatar" src="http://pic1.zhimg.com/ec3de3cee61114c1ae0f4303256d0ed4_is.jpg">
72    <span class="author">逾晖，</span><span class="bio">羽毛球爱好者</span>
73    </div>
74
75  <div class="content">
76    <p>
      联赛的赛制其实就是循环制，杯赛的赛制基础是淘汰制，不同的情况可能会有不同的变化，比如世界杯欧洲
      杯一般是小组循环 + 出线队伍淘汰；或者一些电竞项目喜欢加入复活赛的赛程，不一而足。</p>
77    <p>
      循环制最大的特点就是参赛的每一个个体（选手，组合或者队伍），都能够与所有的对手逐一比赛，大大提
      高了比赛的场次。以英超为例，总共 20 支参赛队伍，大循环的赛制，每一支队伍都能够和其他 19
      支队伍交锋，主客各一场，一个赛季总共要打 38 场比赛。</p>
78    <p>
      这能够为一些商业化程度非常高的赛事保持一个持续的热度。还是以英超为例，除了休赛期，每个周末都会
      有比赛，甚至有一周双赛的情况。这意味着如果你是一个球迷，你每周都可以花钱去球场支持你的球队，球
      员每周都有比赛可打能够维持良好的竞技状态，球队每周都可以挣一次门票钱，赞助商的广告每周都会持续
      曝光，赛事转播权可以卖很多场。</p>
```

通过对网页文件源代码的分析我们可以发现，文章信息主要是指包含在一些特殊标签中的内容。比如，可以发现其中<h2 class=”question-title”></h2>标签内部的文字即是文章对应的问题名称，而 标签中包含了文章正文信息。

在此次作业中，网页的解析由学生自行实现，具体的语法结构解析需**使用栈结构并遍历全部标签**，以便处理标签嵌套的情况，从中提取相应的文本信息。

总体思路为：通过扫描源码字符串，发现<**的结构便压栈，发现**/>或者</**的结构则退栈；当遇到特定匹配的标签时，提取其内部的关键信息；标签内部的文本将在解析的过程中提取出来。

在网页解析过程中，有可能出现标签未正常关闭，或者网页解析结束时栈不空等异常情况，同学们需自行寻找规律，想办法进行应对。实验中可能遇到的标签如 、<h2>、<a>、<ul>、<li>、、<p>等。

基本的扫描流程可以归纳如下：（参考）

第一步：查找下一个“<”的位置和“</”的位置，进行比较；

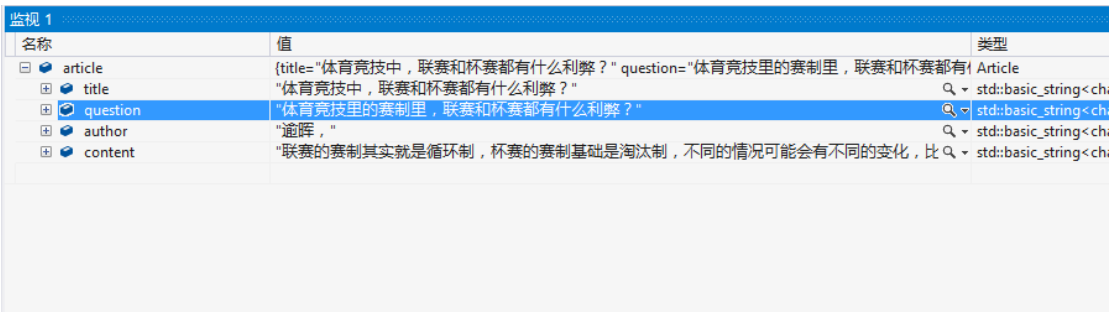
第二步：查看栈顶状态，观察是否需要提取当前位置至下一个标位置之间的

内容；

第三步：如果接下来的标签是“<”，通过查找“ ”或“>”定位标签的类型，比如“<div”或“<h2”，执行对应标签符号的进栈操作；如果是“</”，执行退栈操作；

过程中可能需要依赖一些自定义的规则，具体细节同学们自己去发掘。所有给定的数据已经经过测试，可以完成信息的提取操作。

助教尝试提取后的文章信息如下图所示。



注意，网页中可能会有一些额外信息，正确实现对这些信息的去除，是保证信息正确性的关键。例如，在网页最后，都有一个 标签中的内容是“客官，这篇文章有意思吗？”，这句话实际上并不是正文内容，在提取时需要注意如何避免错误地加入。另外，像文章作者部分貌似多出了一个中文逗号，同学们也可以探索网页的规律，将这个多余的逗号去掉。另外文章中还可能出现一些转义字符，处理得当应该将它们转换为对应文字或符号。类似的优化手段可自行尝试，在实验文档中可作为加分点论述。若信息提取效果太差，也可能适当扣除对应分数。

截取的信息中可能包含多余的空格和换行，自行处理。注意，虽然本次实验只要求提取少量标签中的关键信息，但解析算法执行时需要遍历所有 html 标签，然后根据特定的标签特征及栈顶状态进行信息提取。

注：如果实在无法实现栈结构解析网页，可以直接使用字符串匹配的方式定位关键信息的位置。这种方案没有体现栈的使用，解析算法的通用性也较差。使用这种方案的话，评分项【文本解析】的评分将不超过其评分项总分的 30%。

附录 2：中文分词算法说明

关键信息的中文分词算法将在课堂上补充。

中文分词算法可以很粗糙，也可以做到非常精致。其中有很多功能点可以挖掘，大家可以尝试分析不同文章的分词结果，针对一些缺陷进行完善，这些都可以作为功能亮点，作为加分项。

另外，在执行中文分词的过程中，需要预先载入词库。于是问题来了，如何保存词库中的所有单词？

其中一种方案是使用定义的字符串链表结构，但这样将导致“查找一个词是否在词库中”操作效率低下。由于此时课程暂时未提及哈希表，**此处允许同学们使用系统哈希表进行保存和查找操作，但鼓励自己实现哈希表，此处有加分。**

对于一段文本的分词结果，需要使用自定义的字符串链表进行保存。