

《数据结构与算法》课程实验

基于文本内容的音乐检索与推荐

教师：张力老师

助教：陈凯

2015 年 10 月 14 日

1、 实验目的

本次实验通过实现一个基于文本的音乐检索与推荐系统，对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。

实验要求实现以下功能：

- 根据 HTML 语法使用栈结构分析网页结构
- 提取网页中的关键信息
- 中文分词
- 倒排文档及查询系统的构建
- 推荐系统的简单实现

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的软件开发中，体会数据结构的重要性和广泛用途；
- 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

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、系统实现。

系统运行是指助教根据运行可执行文件的结果进行评分，包括系统的是否可执行，输出结果是否正确，系统效率等；

系统实现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代码实现进行给分。

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，请按照文件夹进行组织。在实验材料中，将包含一个提交样例目录，根据样例目录的格式，在子目录下放置对应内容。

1. 源代码：放置 VS 项目工程，删除 .sdf 等大文件、编译产生结果文件。
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 说明文件及配置文件，说明如何使用可执行文件。
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；言简意赅阐述清楚即可，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，实现网页音乐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 300 个（暂定）网页作为数据库，实现根据输入内容检索音乐功能，并能够针对特定音乐根据不同的规则进行推荐。

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音乐的关键信息。

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音乐的标题、歌手、歌词等信息；信息提取完成后，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. 要求实现的功能：

1. 网页文件解析：通过栈结构，对 html 文件的语法结构进行解析；
2. 关键信息提取：在解析 html 文件语法结构的同时，根据特定的 html 标签提取网页中的关键信息；
3. 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，姓名匹配等）
4. 分词处理：去掉停用词（自行选择停用词表）、将同一歌曲中出现频率较大的词添加进词库等。

C. 数据结构及算法要求：

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

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。

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)。其中每项数据结构需要实现的基本操作有：

栈：push（压栈）、pop（退栈）、top（获取栈顶元素）等操作；

字符串：indexOf（获取对应下标字符）、substring（截取字符串）、concat（连接字符串）等操作；

字符串链表：创建、添加、删除、查找等操作；

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，但不能使用与其相关的系统函数。

本次实验验中，要求同学们实现网页文件的解析，这部分内容将在 5.1（F）中说明；中文分词算法将在课堂上说明。

D. 测试方案：

输入数据：

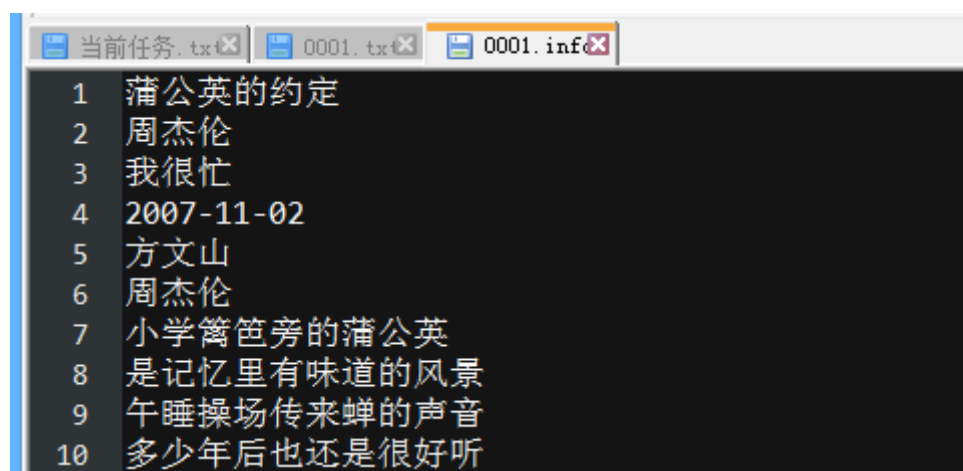
- 10 个网页文件 (*.html)；
- 若干分词文件 (*.txt)；
- 其它自定义数据文件；

输出结果：

- 10 个音乐信息文件：[file_name].info；
- 10 个分词结果文件：[file_name].txt；

其中每个网页文件都生成对应的音乐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，分别保存对应的音乐信息和分词结果。

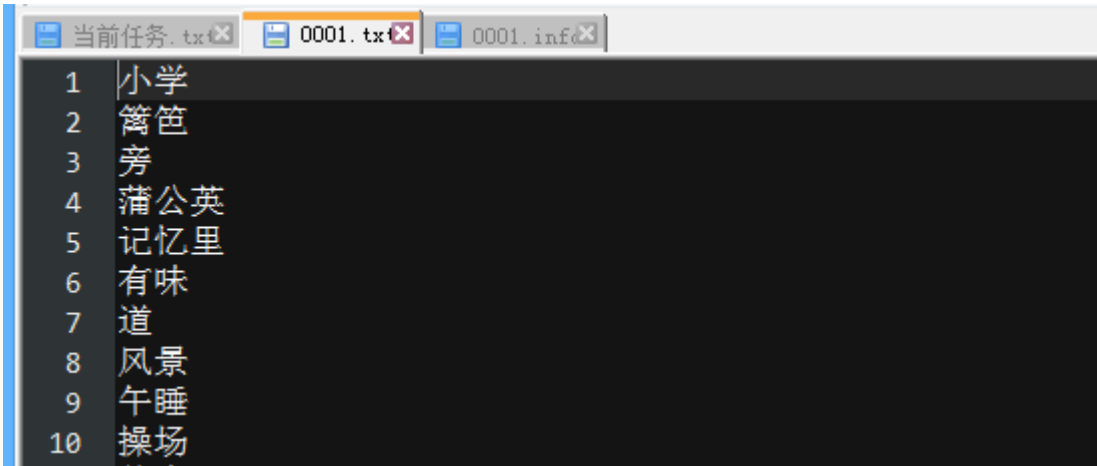
音乐信息文件的格式如下：文件中每行内容依次为音乐名称、歌手名称、专辑名称、发行时间、作词、作曲、歌词。比如对于 0001.html 文件提取音乐信息文件后，得到的 0001.info 文件中内容如下：



```
1 蒲公英的约定
2 周杰伦
3 我很忙
4 2007-11-02
5 方文山
6 周杰伦
7 小学篱笆旁的蒲公英
8 是记忆里有味道的风景
9 午睡操场传来蝉的声音
10 多少年后也还是很好听
```

分词结果文件中，每一行为一个词。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，只要求对歌词内容进行分词。

测试方法：

测试时，助教运行的命令为：`executable.exe arg1 arg2 arg3`

`executable.exe` 为可执行文件；

`arg1` 为同学自定义的配置文件。通过该文件可以读取到词库文件路径及其它所需要定义的参数，请将该配置文件放置在与 `executable.exe` 同级目录；

`arg2` 为测试文件夹的路径；

`arg3` 为输出结果路径；

举例，执行命令：`executable.exe mr.config E:\\Lionel\\input E:\\Lionel\\output`

其中，input 文件夹下将放置 10 个网页文件 0001.html, 0002.html, ..., 0010.html; 最终在 output 文件夹中出现 0001.info, 0002.info, ..., 0010.info, 0001.txt, ..., 0010.txt 等 20 个结果文件。

注：如果助教无法使用上述命令运行程序得到对应结果，最终所得分数将扣除 30%。

E. 评分细则：

模块	内容	分数
数据结构	栈	15%
	字符串	20%
	字符串链表	10%

功能	文本解析	20%
	信息提取	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%

注：助教将根据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。

F. 实验说明

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音乐关键信息，包括歌曲名称、歌手、专辑、歌词内容等。我们处理的是来自“搜狗音乐”的网页，具体分析搜狗音乐网页的 HTML 源代码，可以发现如下内容（代码的其它内容以被忽略）：

```
<div class="song_info_area">
  <div class="album_cover">
    
  </div>
  <div class="song_info">
    <div class="song_tit">
      <h2 title="蒲公英的约定">蒲公英的约定
      <a uigs="out_song_tiny_mv" title="蒲公英的约定的MV" class="video_icon" target="_blank" hr />
    </div>
    <div class="song_btns">
      <a uigs="out_song_tiny_play" onclick="play(event,'[[#102340965#,#2#,#http://cc.stream.qqm" />
    </div>
    <ul class="song_detail">
      <li>歌手: <a uigs="in_song_tiny_singer" href="/tiny/singer?singer_id=4558&query=%D6%DC%BD" />
      <li>语言: <span>国语</span></li>
      <li>所属专辑: <a uigs="in_song_tiny_album" href="/tiny/album?album_id=33021&album_name=%C" />
      <li>发行时间: <span>2007-11-02</span></li>
    </ul>
  </div>
</div>
```

通过对网页文件源代码的分析我们可以发现，音乐信息主要是指包含在<div class=" song_info_area">标签中的部分内容。注意，当除去多余的html 代码后，可以发现其中<h2 title=" ……"></h2>标签内部的文字即是音乐的标题

(歌曲名称), 而标签中包含了歌手、语言、专辑等信息。

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, 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<div>、<h2>、<a>、、、、<p>等, 注意是一个非法的标签, 遇到时可以直接去掉。

为简化实验难度, 本次实验中可以直接提取 “<div class=“song_info_area”>” 到 “<div class=“music_list_area”>” 之间的内容进行解析。即, 从网页文件中读取内容后, 可以先使用自定义字符串的 indexOf 功能定位以上 2 个目标字符串的位置, 然后截取其中的字符串作为有效内容, 进行后续的提取操作。

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。助教尝试提取后的音乐信息如下图所示。



```
10 int main() {
11
12     string pageContent = readWholeContentFromFile("data/page/0001.html");
13
14     MusicInfo musicInfo;
15     musicInfo.parseHtmlContent(pageContent);
16     musicInfo {title="蒲公英的约定" singer="周杰伦" album="我很忙" ...}
17
18     title Q - "蒲公英的约定"
19     singer Q - "周杰伦"
20     album Q - "我很忙"
21     publishDate Q - "2007-11-02"
22     lyricist Q - "方文山"
23     composer Q - "周杰伦"
24     lyrics Q - "小学篱笆旁的蒲公英\n是记忆里有味道的风景\n午睡操场传来蝉的声音\n多少年后也还是很好听\n将离
```

注: 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。

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%。

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音乐文件的分词结果，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；关于如何保存词库，可以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”这样的操作效率低下。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。**

分词的结果需要使用自定义的字符串链表进行保存。

G. 预留接口

在实验 1 完成后，需要为实验 2 预留 3 个接口：

- 一、initDictionaryInfo(...): 该接口执行载入词库等初始化操作；
- 二、extractMusicInfoFromPage(...): 该接口执行解析网页操作，返回结果自行定义，需要包含网页音乐的关键信息；
- 三、divideWords(...): 该接口执行分词操作，返回结果保存为一个字符串链表。

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，并获取每个页面的音乐信息和分词结果，为实验 2 构建倒排文档做好了充分的准备。

5.2 实验 2——音乐检索与推荐（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，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。