当前作业

- 22级第六次作业（查找与排序）
- 22级第五次作业（树）
- 2022级（信息大类）数据结构综合作业（正确性和性能）
- 2022级（信息大类）数据结构综合作业（可扩展性）

历史作业

- 22级第四次作业（栈和队）
- 22级第三次作业（线性表）
- 22级第二次作业
- 22级第一次作业
- 21级第七次作业（图）
- 21级第六次作业（查找与排序）
- 21级第五次作业（树）
- 2021级（信息大类）数据结构综合作业（正确性和性能）
- 2021级（信息大类）数据结构综合作业（可扩展性）
- 21级第四次作业（栈和队）
- 21级第三次作业

2021级（信息大类）数据结构综合作业（正确性和性能）

作业时间： **2022-04-17 09:00:00** 至 **2022-07-01 02:00:00**

本作业是一个综合性能测试作业，测试数据较大，综合考查学生对数据结构与算法的掌握情况，涉及知识点可能包括顺序表、链表、二叉查找树及查找（索引及Hash等）、排序等。两题描述完全一样，只是测试数据集不同！

作业评分标准为：最终成绩为两题分数之和，最高得分为4.5分！
1.第一题是较小数据集测试题目，课程平台下载区中文件project2022.rar中包含了用于测试该题的字典文件（dictionary.txt）、停用词表文件（stopword.txt）、文本文件（article.txt）、及样例运行结果文件（results(example).txt），学生可用它们来调试程序。本题无性能测试，结果正确即可得2.5分。
2.第二题是较大数据集测试题目，程序运行正确（通过测试点）才能得分，其中运行正确占40%分，性能占60%。性能评判标准是以所有程序中运行最快的前15%平均时间为基准，依次计算得分。程序运行无结果或结果错误将不得分。最高得分为2.0分。
建议同学们尝试用不同方法来实现，以体会不同数据结构与算法的结合如何影响程序性能的。同时，建议同学们用本课程所学到的知识来解决问题。

编程题

#	题目	分值	批阅信息
1.	基于关键词的大规模文档搜索（综合-小数据）	2.50	下载源文件

最后一次提交时间:2022-04-23 10:29:19

共有测试数据:2

平均占用内存:129.883K　平均CPU时间:0.17759S　平均墙钟时间:0.17760S

测试数据	评判结果
测试数据1	完全正确
测试数据2	完全正确

【问题描述】

百度、谷歌等互联网搜索引擎提供高效的网页、文档搜索功能，用户可以通过一个和多个关键词查询感兴趣的网页信息。要实现超大规模的文本文档搜索，通常需要借助高效的索引和查询算法。编程实现一个基于关键词的文档搜索程序，实现对大规模文本文档的快速搜索和排序。具体方法如下：

1、对给定的文档（网页）集合（含N个文档）中每个文档进行单词（英文）提取，并统计每个单词k在每个文档d出现的频次（即出现次数）TN_{kd}（该文档总词数为 $TN_d=\sum_k TN_{kd}$ ），由此可以计算其词频TF_{kd}

$$TF_{kd}=\frac{TN_{kd}}{TN_d}\times 100$$

为了提高算法的准确性，在此只统计字典中出现且不为停用词（stop-word）的单词。单词为仅由字母组成的字符序列，包含大写字母的单词应将大写字母转换为小写字母后进行词频统计。

在课程网站下载区提供了字典“dictionary.txt”文件和英文停用词表“stopwords.txt”文件(文件中只包含单词，不含其解释，且已按字典序排序）。

说明：在自然语言处理中，停用词（stop-word）指的是文本分析时不会提供额外语义信息的词的列表，如英文单词a，an，he，you等就是停用词。

2、统计每个单词k在文档集合中出现的次数（DN_k，即出现该单词的文档数），并计算其逆文档频率IDF_k（log以10为底）。定义如下：

$$IDF_k=log(\frac{N}{DN_k})$$

3、针对输入的关键词$K_1,K_2,...,K_m$，按照TF-IDF对文档集合中的文档进行相关度打分。对任意一个文档d，针对所输入的关键词，其相关度计算公式如下：

$$Sim_d=\sum_{k_i} TF_{kd}\times IDF_{k_i}$$

#	题目	分值	批阅信息

若某个关键词未在文档集合中出现，则不用计算其IDF_k，其对所有文档的相关度都为0。

4、依据相关度给出检索结果按由高至低进行排序，返回Top-N的结果。

为了简化搜索引擎的实现，从互联网上爬取（Web Crawling）相关网页(文档)的工作已经完成，并将爬取的网页文档数据已存入一个文本文件（aritcle.txt）中，其中每个网页第一行为网页标识号（如XX-XXXX，可按字符串来输入），然后为网页内容，网页文档间以换页符\f分隔。在课程网站下载区提供了一个用于测试的article.txt文件。

【输入形式】

从命令行输入作为需要返回的检索结果数量NUM和作为检索的关键词串$K_1,K_2,..,K_m$

具体形式如下：

search NUM K_1 K_2 .. K_m

其中search为搜索引擎运行程序，NUM与关键词之间以一个空格分隔。根据当前目录下的“dictionary.txt”文件、停用词文件“stopwords.txt”以及网页数据文件“article.txt”，按上面要求对网页文档进行相关度计算和排序。

注意：

1. 输入串K_1 K_2 .. K_m中的停用词及非字典中单词将不进行相关度分析。

2. 由于Windows系统下文本文件中的’\n’回车符在（评测环境）Linux系统下会变为’\r’和’\n’2个字符，建议用fscanf(fp,”%s”,…)来处理字典文件和停用词文件中英文单词。

【输出形式】

先将Sim值排名前5（TOP 5）的网页信息输出到屏幕上，输出时先输出相关度Sim值（小数点后保留六位）、相应网页序号（从article.txt文件中读入网页文档时按序从1开始编号）及在文件article.txt中的标识号，三者之间由一个空格分隔，最后有一个回车。

同时将Sim值排名前NUM（TOP N）的网页信息输出到results.txt文件中，输出时先输出相关度Sim值（小数点后保留六位）、相应网页序号（从article.txt文件中读入网页文档时按序从1开始编号）及在文件article.txt中的标识号，三者

#	题目	分值	批阅信息

之间由一个空格分隔，每个网页信息后有一个回车；若找到的网页文档数（即Sim值大于0的文档数，即包含所给关键词的文档数）少于NUM，则按实际数目输出（屏幕输出也如此）。

注意：如果相关度Sim值相同，则按照网页序号由小到大的顺序输出！

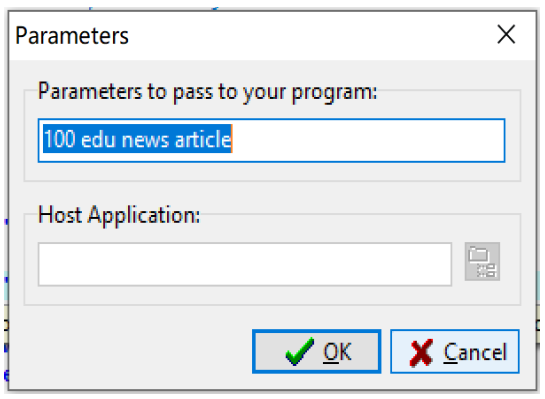
【样例输入】

假设search.exe为搜索引擎程序，以下面方式运行该程序：

search 100 edu news article

（运行程序前，从课程网站下载区下载文件：article.txt, dictionary.txt, stopwords.txt, results(样例).txt到本地）

说明：若本地编程环境为dev-C++，可点击菜单Execute\Parameters…，在下面对话框中输入相应命令行参数。



【样例输出】

程序运行后，屏幕上输出Top-5的结果为：

0.581744 24 1-24

0.466224 230 1-230

0.447891 543 1-543

0.446951 54 1-54

0.440138 87 1-87

所生成的结果文件“results.txt”内容应与下载区文件“results(样例).txt”完全相同。

【样例说明】

样例屏幕输出为按相关度排序由高到低排名前5的结果。其中每一行第一部分为网页文档的相关度（Sim）值，第二部分为相应文档在文件中的序号，第三部分为文档在文件中的标识号。文件results.txt中为按相关度排序由高到低排名前100的结果。

【评分标准】

#	题目	分值	批阅信息
	本综合功能测试题，其评分标准为通过测试数据即可得满分。程序运行无结果或结果错误将不得分。		
2.	基于关键词的大规模文档搜索（综合-大数据）	2.00	下载源文件

最后一次提交时间:2022-05-21 16:20:31

成功编译,但有警告信息.

```
solve13.c: In function 'main':
solve13.c:47:9: warning: array subscript has type 'char' [-Wchar-subscripts]
switch(alpha[*c]){
^
solve13.c:71:17: warning: array subscript has type 'char' [-Wchar-subscripts]
temp[t++]=alpha[*c];
^
solve13.c: In function 'Init':
solve13.c:164:9: warning: array subscript has type 'char' [-Wchar-subscripts]
if (alpha[*s]){
^
solve13.c:181:9: warning: array subscript has type 'char' [-Wchar-subscripts]
if (alpha[*s]){
^
```

共有测试数据:1

平均占用内存:217.723K　平均CPU时间:0.60809S　平均墙钟时间:0.60812S

测试数据	评判结果
测试数据1	完全正确

性能排行榜