

中国地质大学（北京）

数据结构

课程设计任务书

2024 年 12 月

# 基于不同策略的英文单词的词频统计和检索系统

## 课程设计目的：

1. 掌握基于线性表、二叉排序树和散列表不同存储结构的查找算法。
2. 掌握不同检索策略对应的平均查找长度（ASL）的计算方法，明确不同检索策略的时间性能的差别。
3. 掌握相关的排序算法。

## 课程设计内容：

一篇英文文章存储在一个文本文件中，然后分别基于线性表、二叉排序树和哈希表不同的存储结构，完成单词词频的统计和单词的检索功能。最后计算不同检索策略下的 ASL，通过比较 ASL 的大小，对不同检索策略的查找效率做出相应的比较分析。（比较分析要写在课程设计报告中的“总结”部分）

1、读取一篇包括标点符号的英文文章（InFile.txt），从文件中读取单词，将单词转化为小写，过滤掉所有的标点。（除去数字，特殊情况如 China 识别为 china，long-term 整体识别为 long-term）

2、分别基于线性表、二叉排序树和哈希表这三种不同的存储结构，实现单词词频的统计和单词的检索功能。其中，线性表采用顺序表和链表两种不同的存储结构分别实现顺序查找，同时实现基于顺序表的折半查找；哈希表分别实现基于开放地址法的哈希查找和基于链地址法的哈希查找。因此，总计实现 6 种不同的检索策略。

3、不论采取哪种检索策略，完成功能均相同。主要功能如下：

### （1）词频统计

当读取一个单词后，若该单词还未出现，则在适当的位置上添加该单词，将其词频计为 1；若该单词已经出现过，则将其词频增加 1。统计结束后，将所有单词

及其频率按照词典顺序写入文本文件中。其中，不同的检索策略分别写入 6 个不同的文件。

基于顺序表的顺序查找 --- OutFile1.txt

基于链表的顺序查找 --- OutFile2.txt

基于顺序表的折半查找 --- OutFile3.txt

基于二叉排序树的查找 --- OutFile4.txt

基于开放地址法的哈希查找 --- OutFile5.txt

基于链地址法的哈希查找 --- OutFile6.txt

注：如果实现方法正确，6 个文件的内容应该是一致的。

## (2) 单词检索

输入一个单词，如果查找成功，则输出该单词对应的频率，同时输出查找成功所花费的时间和查找长度 ASL。如果查找失败，则输出“查找失败”的提示。

## 实验提示

不同的检索策略所采取的数据结构不一样，算法实现的过程不一样，但查找结果是一样的。下面给出系统运行的部分参考截图。

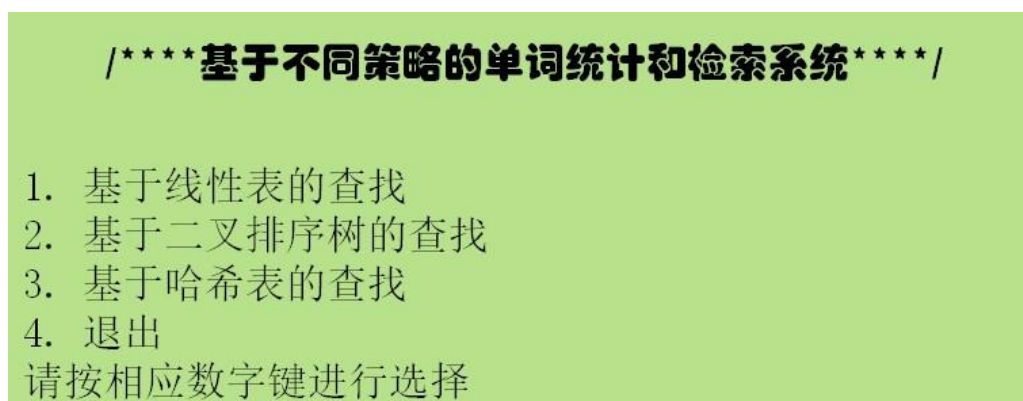


图 1 系统主界面

对于图 1，选择 1 后进入“基于线性表的查找”，结果如图 2 所示。

**/\*\*\*\*基于不同策略的单词统计和检索系统\*\*\*\*/**

**-----基于线性表的查找-----**

1. 顺序查找
  2. 折半查找
  3. 返回上一级
- 请按相应数字键进行选择

图 2 基于线性表的查找

对于图 2，选择 1 后进入“顺序查找”，结果如图 3 所示。

**/\*\*\*\*基于不同策略的单词统计和检索系统\*\*\*\*/**

**-----顺序查找-----**

1. 基于顺序表的顺序查找
  2. 基于链表的顺序查找
  3. 返回上一级
- 请按相应数字键进行选择

图 3 顺序查找

对于图 3，选择 1 后进入“基于顺序表的顺序查找”，结果如图 4 所示。

**/\*\*\*\*基于不同策略的单词统计和检索系统\*\*\*\*/**

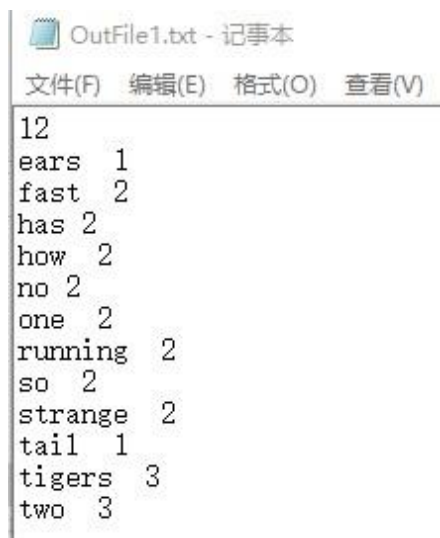
**-----基于顺序表的顺序查找-----**

1. 词频统计
  2. 单词查找
  3. 返回上一级
- 请按相应数字键进行选择

图 4 基于顺序表的顺序查找

对于图 4，选择 1 后进入“词频统计”，完成词频统计功能。要求统计出所有单词的总数和每个单词的词频。统计结果全部输出到对应的文件 OutFile1.txt 中，该文件中的数据总计  $n+1$  行，第一行  $n$  为所有单词的总数，后面  $n$  行为每个单词及其出现的频率（单词和频率用空格分隔）。

文件 OutFile1.txt 的示例如图 5 所示。



```
OutFile1.txt - 记事本
文件(F) 编辑(E) 格式(O) 查看(V)
12
ears 1
fast 2
has 2
how 2
no 2
one 2
running 2
so 2
strange 2
tail 1
tigers 3
two 3
```

图 5 词频统计

对于图 4，选择 2 后进入“单词查找”，输入待查找的单词后，如果查找成功，结果如图 6 所示。首先显示此单词的词频，之后分别给出查找该单词 10000 次累计所花的时间（单位为毫秒）和单次查找长度。（注：由于单次内存查找极快，时间统计值会为 0，请在查找单词时用 10000 次重复查找替代，代码中时间单位使用高精度时钟 ns 纳秒，打印输出时转换成 ms 毫秒，保留四位小数）。如果查找失败，结果如图 6 所示。



图 6 单词查找成功

(注意：这里查找长度是对于单个单词的，不是平均查找长度 ASL)

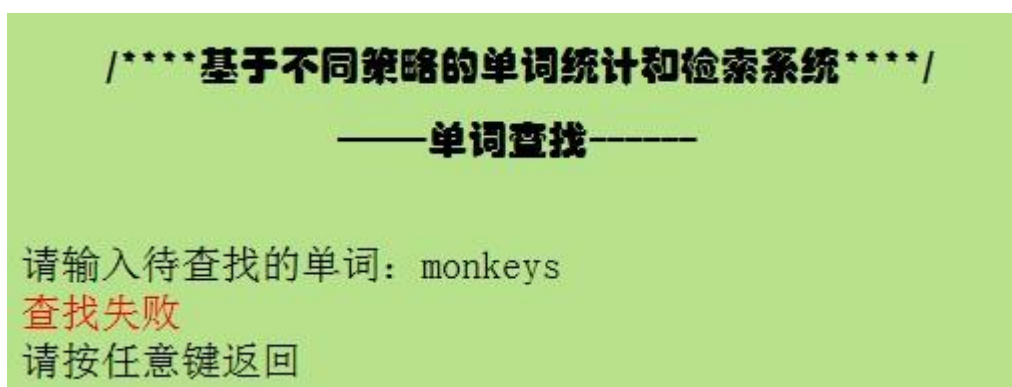


图 7 单词查找失败

对于其他 5 种检索策略，查找结果同图 6 或图 7 所示，只是在查找成功时所花的时间和平均查找长度不同而已。

最后，对词频统计中的所有单词，分别使用上述 6 种查找策略进行循环查找，计算每种策略下的平均查找长度 ASL（保留 4 位小数）。

再次强调：请不要使用 C++ 标准库中的数据结构，请不要使用 AI 生成的代码。相似或雷同的作业，无论是主动还是被动，涉及到的同学成绩都将受到影响!!!

## 选做内容：

1. 用窗体版界面取代控制台界面。
2. 程序中所用的到排序算法选用先进的排序算法（快速排序或归并排序或堆排序），将二叉排序树查找换成平衡二叉树查找。
3. 实现低频词过滤功能，要求将出现词频低于 5 的单词删除，可以选择不同的存储结构分别实现。
  - （1）顺序表
  - （2）链表
  - （3）二叉排序树
  - （4）基于开放地址法的哈希表
  - （5）基于链地址法的哈希表