

文本索引

一、实验目的

编写一个构建大块文本索引的程序，然后进行快速搜索，来查找某个字符串在该文本中的出现位置。

二、实验内容

你的程序应该使用两个文件名作为命令行参数：文本文件（我们称为语料库）和包含查询的文件。假设这两个文件只包含小写字母、空格和换行符，查询文件中的查询由换行符分隔。这不是一个限制，因为你可以使用一个过滤器将任何文件转换为此格式。你的程序应该读取语料库，将其存储为（可能巨大）字符串，并可能为其创建索引，如下所述。然后它应该逐个读取查询（假设在命令行中的第二个命名文件中，每行有一个查询），并打印出语料库中每个查询在文本文件中首次出现的位置。

三、代码实现

代码如下：

```
#include <iostream>
#include <vector>
#include <string>
#include <fstream>
#include <streambuf>
#include <iomanip>

using std::cin;
using std::cout;
using std::endl;
using std::ifstream;
using std::string;
using std::vector;
using namespace std;

const int MAX_CHAR_ASCLL = 255;
int shift[MAX_CHAR_ASCLL];
// Sunday 算法的实现
int Sunday(const string &thestring, const string &mystring)
{
    int length1 = thestring.size();
    int length2 = mystring.size();
    for (int i = 0; i < MAX_CHAR_ASCLL; i++)
    {
        shift[i] = length2 + 1;
    }
    for (int i = 0; i < length2; i++)
    {
        shift[mystring[i]] = length2 - i;
    }
}
```

```

    }
    int i = 0;
    int j;
    while (i <= length1 - length2)
    {
        j = 0;
        while (thestring[i + j] == mystring[j])
        {
            j++;
            if (j >= length2)
                return i;
        }
        i += shift[thestring[i + length2]];
    }
    return -1;
}

int calculate_index(int k, string txt)
{
    if (k == -1)
        return -1;
    int j = 1;
    for (int i = 0; i <= k; i++)
    {
        if (txt[i] == ' ')
            j++;
    }
    return j;
}

// 使用更加高效的 Sunday 算法来进行模式匹配
int main()
{
    std::ifstream sin("C:\\Users\\lenovo\\Desktop\\message.txt");
    string txt((std::istreambuf_iterator<char>(sin)),
std::istreambuf_iterator<char>());
    string mystring;
    ifstream in("C:\\Users\\lenovo\\Desktop\\in.txt", ios::in);
    if (!in.is_open())
    {
        cerr << "open error!" << endl;
        exit(0);
    }
    vector<string> vec;

```

```

while (!in.eof())
{
    in >> mystring;
    vec.push_back(mystring);
}
for (int i = 0; i < vec.size(); i++)
{
    int k = Sunday(txt, vec[i]);
    k = calculate_index(k, txt);
    cout << k << "    " << vec[i] << endl;
}
return 0;
}

```

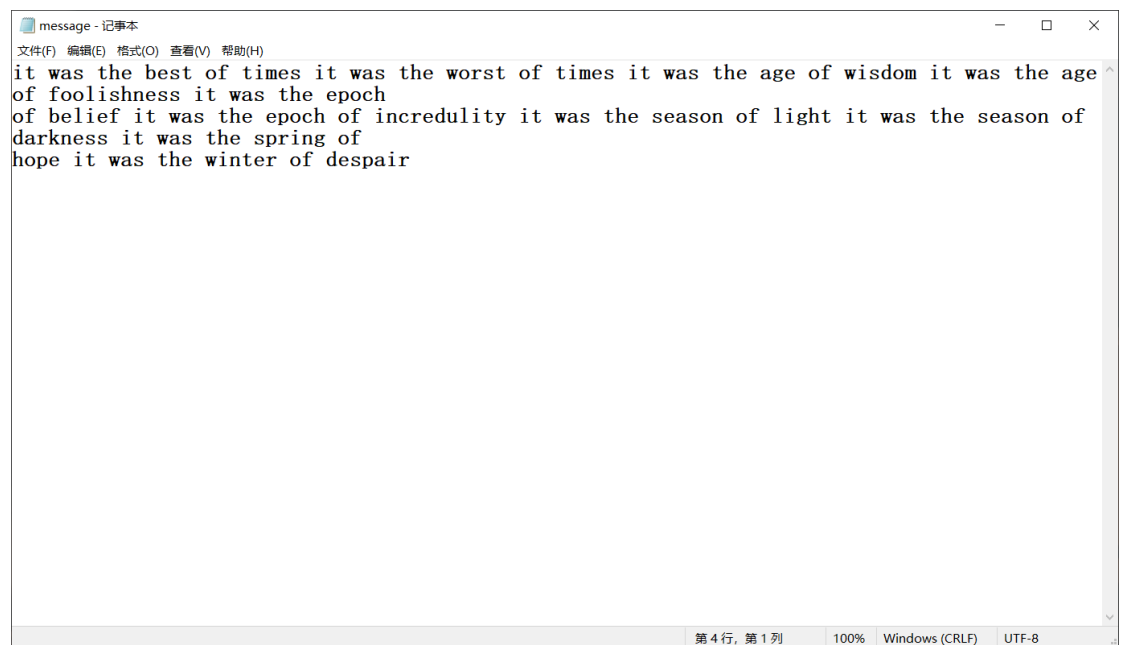
代码实现思路：

我在实现本题的算法时，并没有采用 PDF 中所提供的思路，而是直接采用了一种比较优秀的检索算法——Sunday 算法，假设文本长度为 n ，所需要检索的字符串的长度为 m ，该算法的平均性能为 $O(n)$ ，最好情况为 $O(n/m)$ ，最坏情况为 $O(nm)$ ，该算法的性能优于 KMP 算法和 BM 算法，因此在本次题目中我使用了该算法来检索。如果查找到了一个单词，则返回其首字母所在的指针位置，否则返回值为 -1。

为了方便检验，我将所有的输入写入到了一个文件之中，以此来提高输入的效率。

四、 实验结果

文件的输入如图所示：



```
in - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
this
spring
was
the
第 4 行, 第 4 列 100% Windows (CRLF) UTF-8
```

程序的输出结果如下所示：

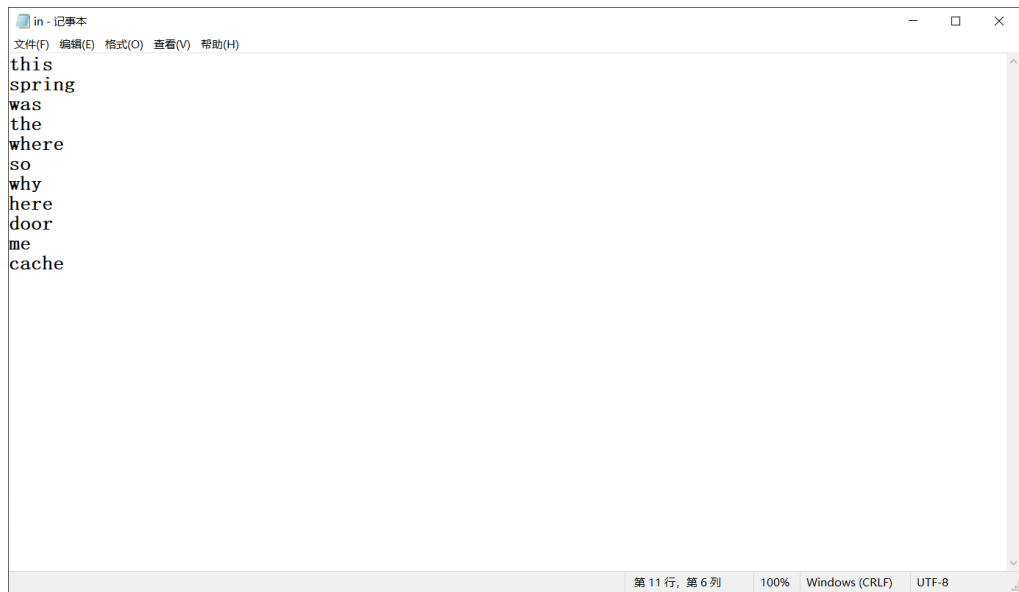
```
问题 输出 终端 调试控制台
PS D:\GitRepo\Algorithm\algorithm4> cd "d:\GitRepo\Algorithm\algorithm4"; if ($?) { g++ algorithm4.cpp -o algorithm4 }; if ($?) { .\algorithm4 }
-1 this
52 spring
2 was
3 the
PS D:\GitRepo\Algorithm\algorithm4>
行 50, 列 19 空格: 4 UTF-8 CRLF C++ Win32
```

五、 结果分析

根据程序的结果不难看出程序输出是正确的，当然，为了验证结果更为广泛的正确性，我又修改了 message 文件，将其规模扩大，选用了很大的语料库。

```
message - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
It was the best of times it was the worst of times it was the age of wisdom it was the age of foolishness it was the epoch of belief it was the epoch of incredulity it was the season of light it was the season of darkness it was the spring of hope it was the winter of despair.
In clinical practice, management of iliotibial band (ITB) pain often includes passive treatments that aim to reduce "tension" in the ITB. Various methods are used to estimate the mechanical properties of the ITB. Comparison of such methods in pain-free and injured populations may provide a better understanding of the mechanical and morphological properties of the ITB, and its implication in musculoskeletal injuries. This study aimed to systematically review the mechanical properties and morphology of ITB using a range of experimental methods and evaluate differences between individuals with and without lower limb musculoskeletal conditions.It was the best of times it was the worst of times it was the age of wisdom it was the age of foolishness it was the epoch of belief it was the epoch of incredulity it was the season of light it was the season of darkness it was the spring of hope it was the winter of despair.
In clinical practice, management of iliotibial band (ITB) pain often includes passive treatments that aim to reduce "tension" in the ITB. Various methods are used to estimate the mechanical properties of the ITB. Comparison of such methods in pain-free and injured populations may provide a better understanding of the mechanical and morphological properties of the ITB, and its implication in musculoskeletal injuries. This study aimed to systematically review the mechanical properties and morphology of ITB using a range of experimental methods and evaluate differences between individuals with and without lower limb musculoskeletal conditions.
It was the best of times it was the worst of times it was the age of wisdom it was the age of foolishness it was the epoch of belief it was the epoch of incredulity it was the season of light it was the season of darkness it was the spring of hope it was the winter of despair.
第 1 行, 第 1 列 100% Windows (CRLF) UTF-8
```

改变搜索内容如下：

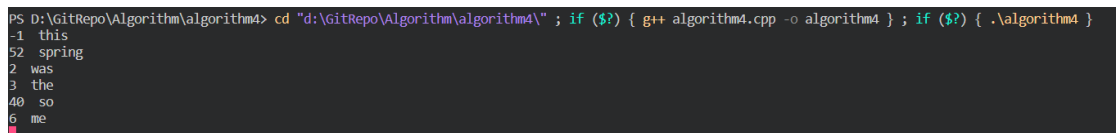


```
in - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

this
spring
was
the
where
so
why
here
door
me
cache

第 11 行, 第 6 列 100% Windows (CRLF) UTF-8
```

搜索结果如下：



```
PS D:\GitRepo\Algorithm\algorithm4> cd "d:\GitRepo\Algorithm\algorithm4\" ; if ($?) { g++ algorithm4.cpp -o algorithm4 } ; if ($?) { .\algorithm4 }
1 this
52 spring
2 was
3 the
40 so
6 me
```

可见程序是可以应用于大型数据的。