**实验报告**

目录

[实验环境 1](#_Toc20753485)

[实验结果 2](#_Toc20753486)

[实验分析 4](#_Toc20753487)

[小结 5](#_Toc20753488)

## 实验环境

#### 硬件环境

轻薄笔记本：联想小新Air13.3

CPU：Intel酷睿i5 8265U

内存：8GB

硬盘：256GB SSD

#### 软件环境

操作系统：Windows 10 1903 家庭中文版

IDE：IDEA

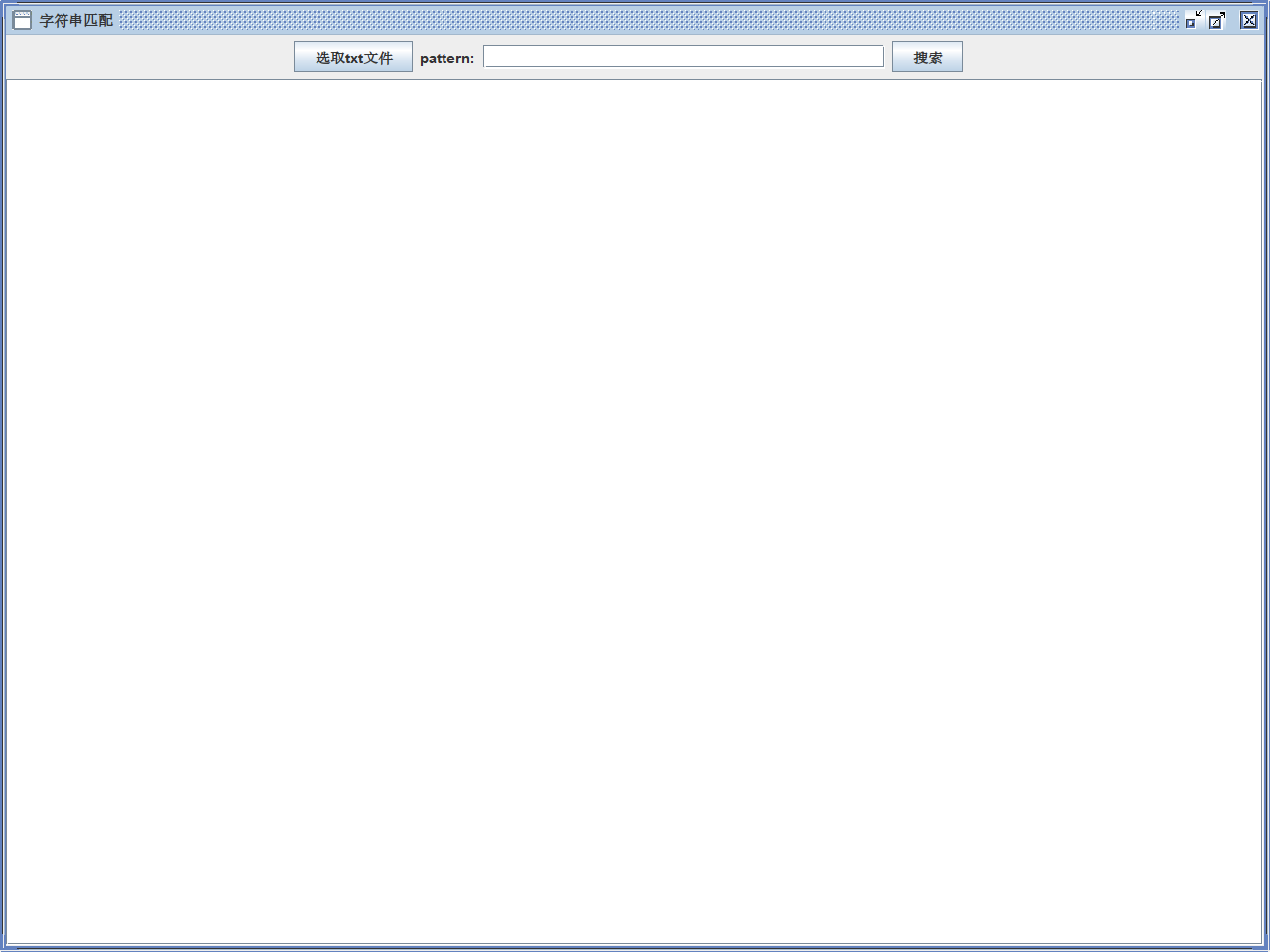
编程语言：JAVA

JDK版本：JDK 13

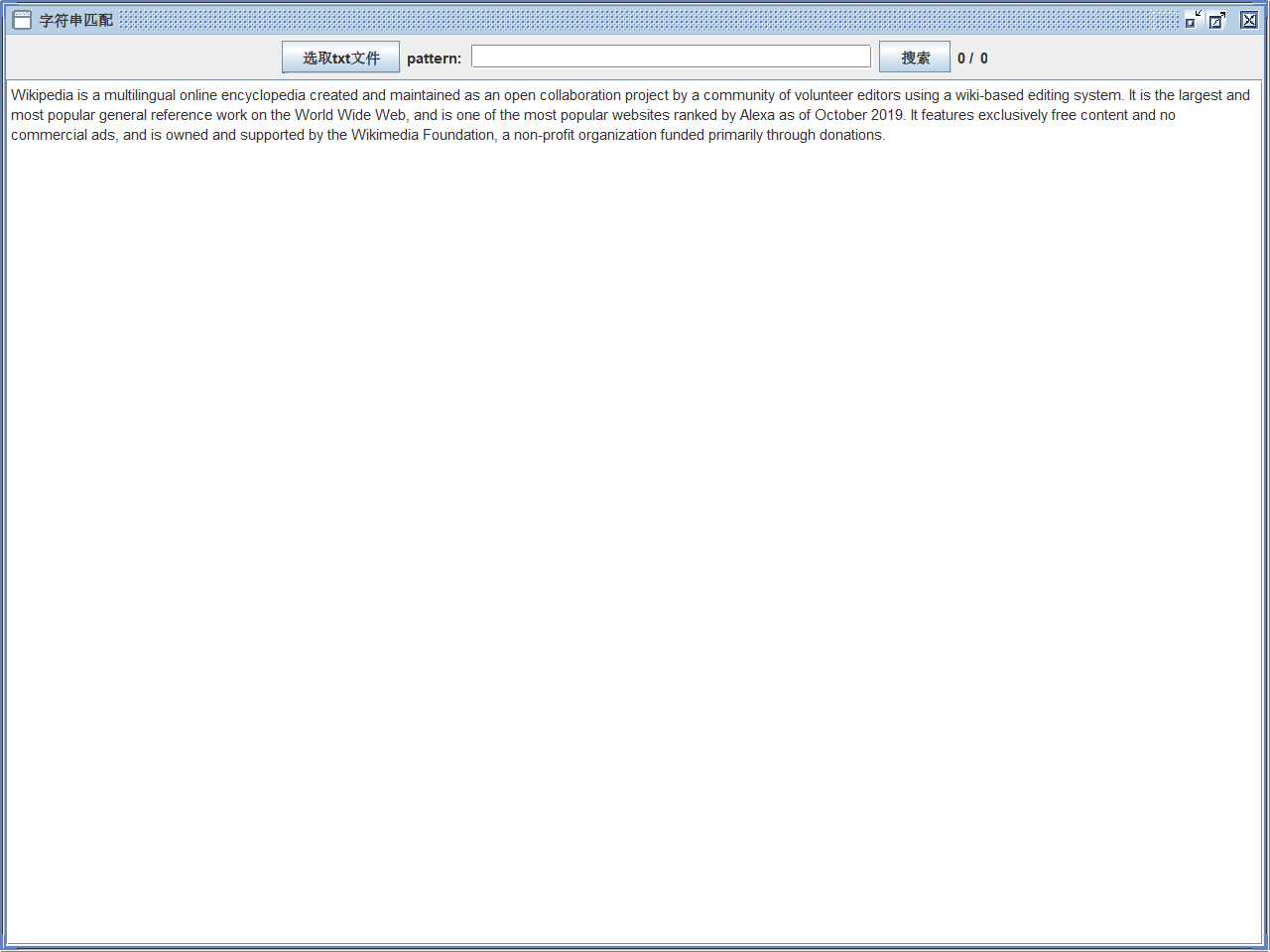
执行命令：java -jar out\artifacts\\_2019214540\_10\_jar\胡亮-2019214540-算法第10次作业.jar

## 实验结果

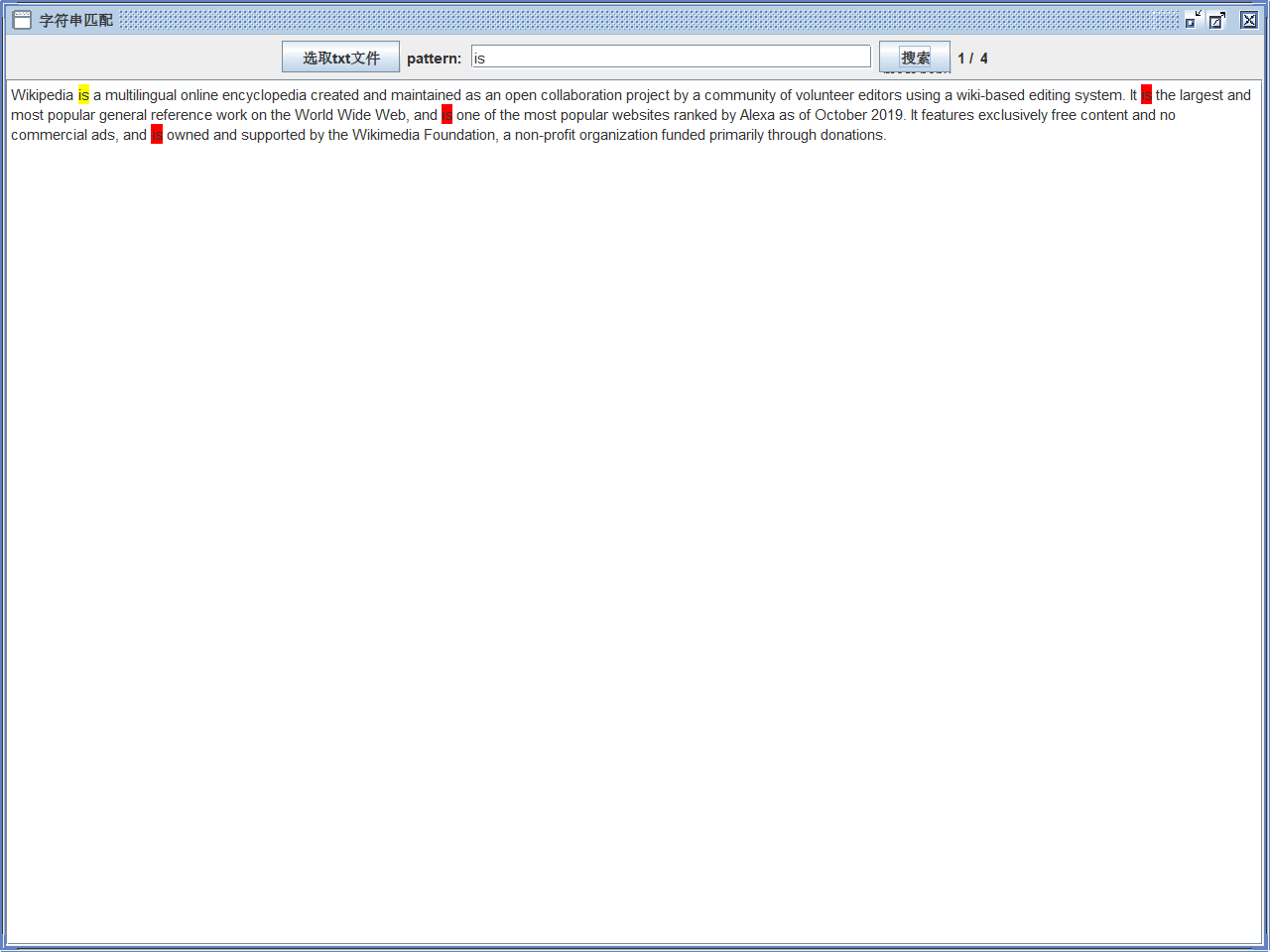
##### 主界面



选取一个txt文件后，内容会显示在下方



接着，在pattern中输入要查找的内容，这里输入“is”，然后点击搜索



如图，全部的”is”会用红色和黄色的背景表示出来，当前的is用黄色背景表示，继续点击搜索，会跳转到下一个”is”

##### 不同输入规模下的运行时间

**Pattern = “is”**

**Text = 源码中rsc文件夹下的txt文件**

这部分实验没有放在UI中，直接用控制台跑，一共跑5次，可查看源代码，下面附表格和图。

|  |  |  |  |
| --- | --- | --- | --- |
| 文本规模(单位:词数) | 消耗时间(5次的平均时间) | | |
| Brute-Force | KMP | Boyer-Moore |
| 100 | 2.47ms | 71.10us | 146.80us |
| 1000 | 748.30us | 366.70us | 928.80us |
| 10000 | 7.23ms | 2.33ms | 5.57ms |
| 100000 | 17.24ms | 10.94ms | 23.76ms |
| 1000000 | 39.60ms | 26.86ms | 59.84ms |

以us为基本单位，将同等输入规模的时间取平均值，再以10为底取对数后，画出如下图形。

## 实验分析

Boyer-Moore算法没优势，KMP算法最快，Brute-Force位于两者之间，与预想不符。

## 小结

###### 关于数据

100 words和1000 words的txt文件是在网上找的，再大的就不好找了，所以10000的是用1000的复制了10遍得到的，以此类推。

###### 关于结果

执行时可能有额外的initial操作。因为发现Brute-Force的第一次时间明显不合理，猜测是刚开始执行，虚拟机要执行一些额外操作，浪费了时间。