

实验报告

目录

| | |
|------------|---|
| 实验环境 | 1 |
| 实验结果 | 2 |
| 实验分析 | 4 |
| 小结 | 5 |

实验环境

硬件环境

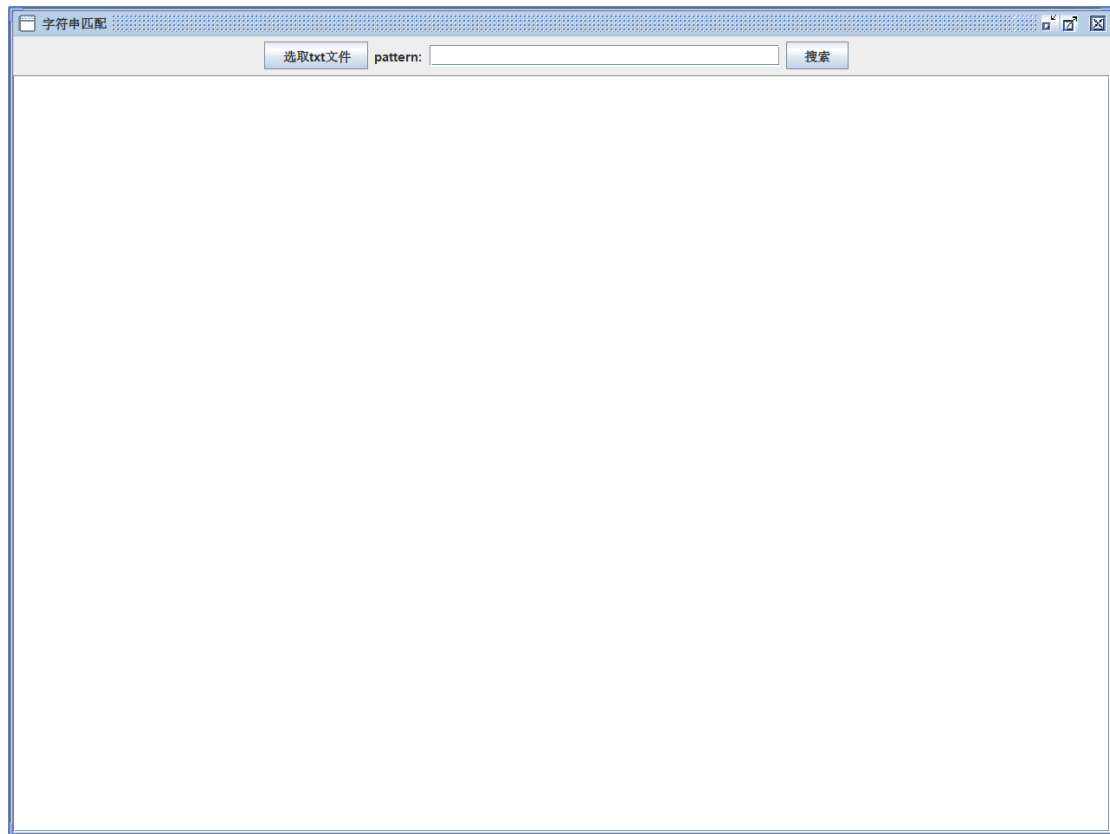
轻薄笔记本：联想小新 Air13.3
CPU：Intel 酷睿 i5 8265U
内存：8GB
硬盘：256GB SSD

软件环境

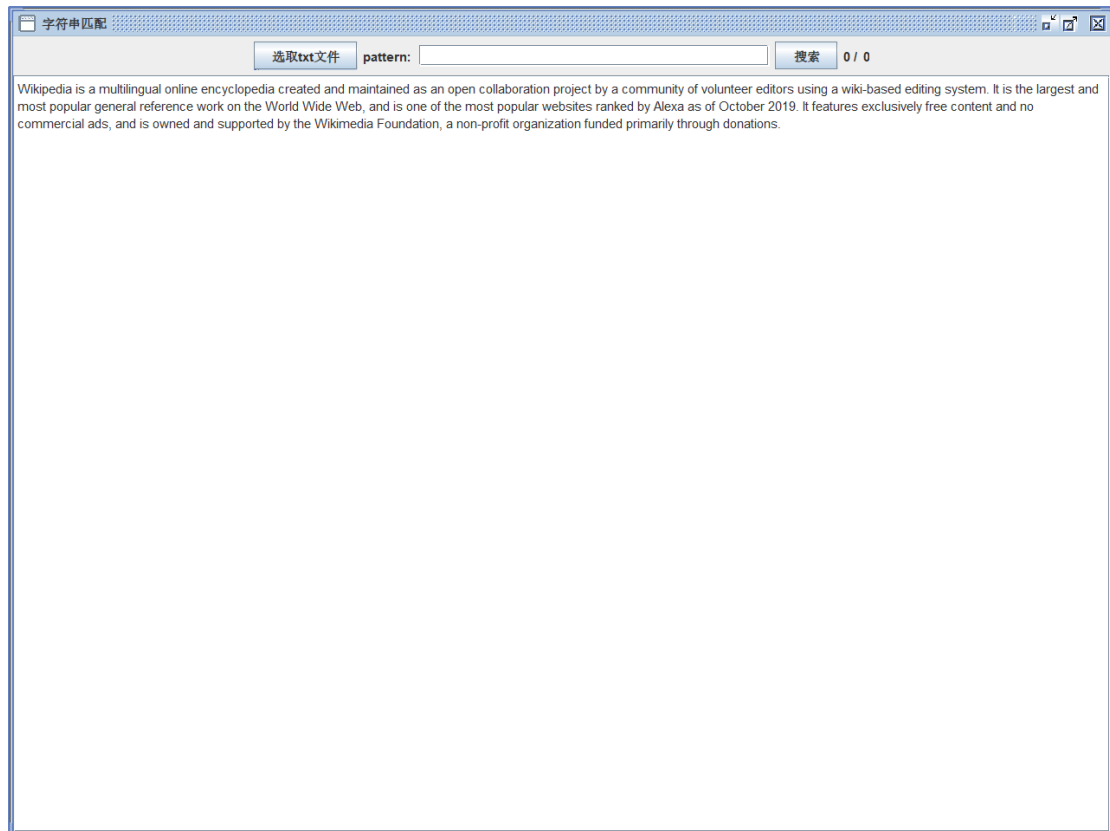
操作系统：Windows 10 1903 家庭中文版
IDE：IDEA
编程语言：JAVA
JDK 版本：JDK 13
执行命令：java -jar out\artifacts_2019214540_10_jar\胡亮-2019214540-算法第 10 次作业.jar

实验结果

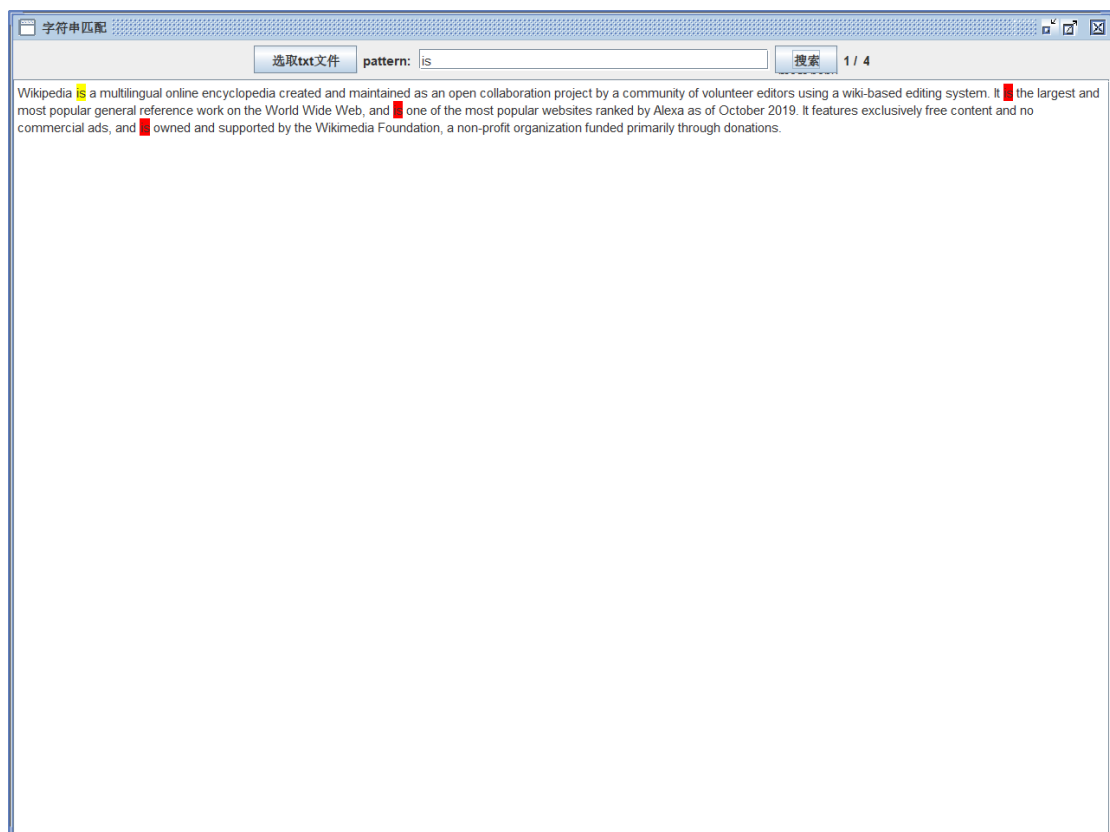
主界面



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

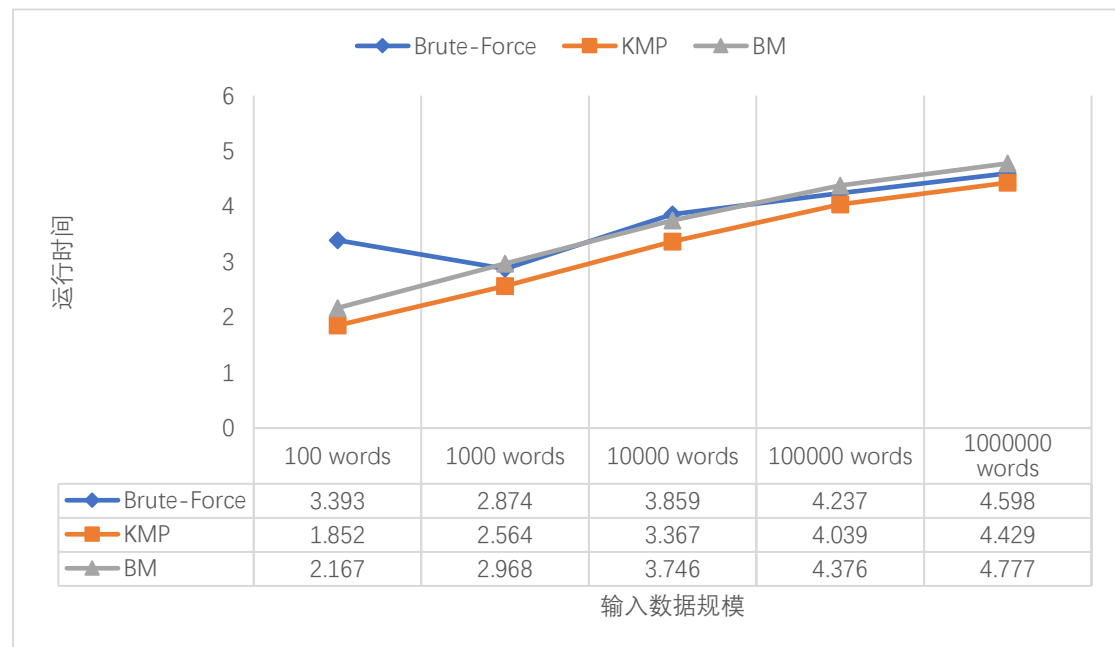
Pattern = “is”

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。