

四川大学计算机学院（软件学院、智能
科学与技术学院）

实验报告

学号：2023141460321 姓名：孙谦昊 专业：计算机科学与技术 班级：行政七班 第 8 周

课程名称	数据结构与算法分析课程设计	实验课时	第 2 次
实验题目	简单文本编辑器	实验时间	2024 年 10 月 22 日
实验目的和要求	<p>实验基本要求：</p> <p>R:读取文本文件到缓冲区中,缓冲区中以前的任何内容将丢失,当前行是文件的第一行。</p> <p>W:将缓冲区的内容写入文本文件,当前行或缓冲区均不改变。</p> <p>I:插入单个新行,用户必须在恰当的提示符的响应中键入新行并提供其行号。</p> <p>D:删除当前行并移到下一行。</p> <p>F:从当前行开始,查找包含有用户请求的目标串的第一行。</p> <p>C:将用户请求的字符串修改成用户请求的替换字符串,仅在当前行中有效。</p> <p>Q:退出编辑器。</p> <p>H:显示解释所有命令的帮助消息,程序也接受?作为 H 的替代者。</p> <p>N:下一行,在缓冲区中进一行。</p> <p>P:上一行,在缓冲区中退一行。</p> <p>B:开始,到缓冲区的第一行。</p> <p>E:结束,到缓冲区的最后一行。</p> <p>G:转到缓冲区中用户指定的行号。</p> <p>V:查看缓冲区的全部内容,显示到终端上。</p> <p>实验扩展：</p> <p>U:撤销,即回到上一步操作之前的状态。</p> <p>Z:重做,即撤销后可以恢复撤销前的状态。</p>		
实验环境	12th Gen Intel(R) Core(TM) i7-1260P 2.10 GHz Visual Studio 2022 Debug x64 本地 Windows 调试器		
算法描述	<p>本程序采用 KMP 算法等实现了简单文本编辑器的功能，具体实现方式如下：</p> <p>一、程序开始</p> <p>进入程序，展示作者和指令集。允许用户 R 读取文件，H 或？查看指令，Q 退出程序。</p> <p>二、SearchAndReplacement 类</p> <p>在 SearchAndReplacement 类中，GetNext()函数和 Search()函数共同组成 KMP 算法。GetNext()函数用于构建部分匹配表，而 Search()函数则用于在原始文本中搜索模式串。</p> <p>GetNext()函数用于预处理模式串（matchTxt），构建一个称为“部分匹配表”（或“失败函数”）的数组 next。这个表记录了模式串中每个位置的最长相同前缀和后缀的长度。例如，对于模式串"ABCDABD"，部分匹配表将告诉我们，如果当前字符不匹配，我们应该将模式串向右移动多少个位置，以便能够利用已经匹配的前缀信息。</p>		

	<p>构建部分匹配表的过程如下：</p> <ol style="list-style-type: none">1.初始化 j 为 0，next[0]为 0，因为空字符串是任何字符串的前缀。2.遍历模式串，对于每个位置 i： 如果 matchTxt[i]与 matchTxt[j]相同，则 j 加 1，next[i]设为 j。 如果 matchTxt[i]与 matchTxt[j]不同，且 j 不为 0，则 j 更新为 next[j - 1]，即 matchTxt 中 j-1 位置的失败函数值。 如果 j 为 0 或 matchTxt[i]与 matchTxt[j]不同且 j 为 0，则 next[i]设为 0。 <p>Search()函数使用 GetNext()函数构建的部分匹配表来在原始文本（originalTxt）中搜索模式串（matchTxt）。搜索过程如下：</p> <ol style="list-style-type: none">1.调用 GetNext()函数构建部分匹配表。2.初始化两个指针 i 和 j，分别指向原始文本和模式串的起始位置。3.遍历原始文本，对于每个位置 i： 如果 originalTxt[i]与 matchTxt[j]相同，则 j 加 1。 如果 matchTxt[j]到达模式串末尾，即找到了一个匹配，返回模式串在原始文本中的起始位置索引。 如果 originalTxt[i]与 matchTxt[j]不同，且 j 不为 0，则 j 更新为 next[j - 1]。 如果 j 为 0 或 originalTxt[i]与 matchTxt[j]不同且 j 为 0，则 i 加 1。4.如果遍历完原始文本后仍未找到匹配，则返回-1。 <h3>三、TxtEditor 类</h3> <p>在 TxtEditor 类中，定义了各种有关于文本编辑操作的函数。</p> <p>PageAndTotal()和 Screen(string command)函数用于计算和输出可视化版本的文本编辑器，并将当前行用绿色字体标注。</p> <p>ReadTxt(string tmpFileName)函数用于读取.txt 文件。在运行程序后，如果成功读取文件则打开文件，反之则创建新文件。在程序运行过程中，如果用户读取另外一个文件则询问是否打开？若用户选择继续打开则将当前文件存储随后读取另外一个文件。</p> <p>WriteTxt(string file)函数用于快速将文本编辑器中的文本输入到文件中。</p> <p>SaveTxt()函数用于保存文件。</p> <p>InsertTxt()函数用于插入行。首先将询问用户意图在哪一行插入。合法插入行为第 1 行至在最后一行插入。如果用户输入的为合法插入行，则在对应行前插入新的一行；反之则报错误提醒。</p> <p>DeleteTxt()函数用于删除行。首先将询问用户意图将哪一行删除。合法删除行为第 1 行至在最后一行。如果用户输入的为合法删除行，则在删除对应行；反之则报错误提醒。</p> <p>FindTxt()函数用于寻找文本中用户想要查找的内容。在 FindTxt()函数中调用 KMP 算法。查找找到则将当前行定义为当前行；反之则报错。</p> <p>ChageTxt()函数用于用户请求的字符串首次在文本中出现处修改成用户请求的替换字符串。在 ChageTxt()函数中调用 KMP 算法。查找到内容后允许用户替换；反之则报错。</p> <p>GetTxt(int getLineNo)函数用于将获取某一行的内容。</p> <p>ViewTxt()函数用于展示文本内容。在此函数中用户可以输入查看某一页中的全部文本内容。</p> <p>Run()函数用于获取用户输入的指令并调用上述对应的函数。</p>
源程序清单	<ol style="list-style-type: none">1.main.cpp，用于主程序和文本操作函数的调用2. SearchAndReplacement.h，用于定义 KMP 算法3. TxtEditor.h，用于定义了各种有关于文本编辑操作的函数

二、程序实际运行结果

请读取一个文件(R)，或退出(Q): |

请输入指令：

请输入指令：|

	<div><div>四川大学计算机学院 2024-2025 学年数据结构与算法分析课程实验课作业二</div><div>姓名: 孙谦昊</div><div>学号: 2023141460321</div></div> <div><div>1 a</div><div>2 d</div><div>3 c</div><div>4 d</div><div>5 e</div><div>6 f</div><div>7 g</div><div>8 h</div><div>9 i</div><div>10 j</div><div>11 k</div><div>12 l</div><div>13 m</div><div>14 n</div><div>15 o</div><div>16 p</div><div>17 q</div><div>18 r</div><div>19 s</div><div>20 t</div></div> <div>Page: 1 ,Total: 2</div> <div>----- 请输入指令: r 输入文件名: 3 读取 3.txt 将保存并删除缓冲区中 2.txt 中的内容, 是否要继续操作 (Y/N) ? y</div>
实验运行情况分析(包括算法、运行结果、运行环境等问题的讨论)	<div>一、程序运行情况分析</div> <p>本程序采用的算法和编写的程序能够正确地运行出所期望的对应结果。可视化的界面相较于实验基本要求而言更用户友好型。</p> <div>二、运行环境讨论</div> <p>本程序的编写、编译、运行和调试皆是借助 Visual Studio 2022 Debug x64 本地 Windows 调试器。若将本程序使用 DEV C++ 等软件上编译、运行和调试时可能报错, 而这是因为各软件对于支持的 C++ 版本不同所导致的。例如在 Visual Studio 2022 中可以使用 nullptr, 而在部分 DEV C++ 中仅能使用 NULL。</p> <p>本程序因为采取的写作思路与事例程序有一定差别, 因此代码长度显著长于事例代码。借助 Visual Studio 2022 Debug x64 本地 Windows 调试器强大的功能能够快速编译运行, 而在部分 DEV C++ 上则速度较慢。表明程序仍有一定优化空间。</p> <div>三、个人心得</div> <p>在课上听老师讲述运算过程时以为自己听明白了, 可是实际上手时发现问题很多。上课时其实对于 KMP 算法有一定的疑问, 但是通过自己上手编程的方式使我对于 KMP 算法有了更深的理解。</p> <p>同时, 在最初写好的版本中是没有像现在这样子的可视化界面。比如说跳转到第 10 行后很快就会忘记了第 1 行的内容是什么。来回的反复跳转查看耗时且麻烦, 于是打算开始写可视化界面。</p> <p>在可视化界面的编写过程中新了解到了#include <iomanip>和#include <cstdlib>中的很多函数。将可视化界面编写和调试好后, 确实让我能够更加便捷的编辑文本。</p> <div>四、其他</div> <p>在编写完成程序后, 邀请了同学们帮忙测试程序。期间也发现了许多 Bug, 后面都逐一进行了相关修正。</p> <p>在编写程序过程中的相关版本代码发布已经在 Github 中, 链接为: https://github.com/StrayerSQH/Learning/tree/main/%E6%95%B0%E6%8D%AE%E7%BB%93%E6%9E%84%E4%B8%8E%E7%AE%97%E6%B3%95%E5%88%86%E6%9E%90%E5%AE%9E%E9%AA%8C%E8%AF%BE/%E5%AE%9E%E9%AA%8C2</p>
指导老师评议	<div>成绩评定:</div> <div>指导教师签名:</div>

