



## 数据结构课程设计2022

## ——测试报告

小组成员： 韩旭、陈可儿、齐雨婷

学 院 ：计算机学院（国家示范性软件学院）

专 业 ： 计算机科学与技术

班 级 ： 2020211307

指导教师： 张海旸老师

注：本报告只对部分重点功能进行详细测试报告，其余功能模块只进行基础测试

1. 课程信息管理

* 测试内容一：排序

使用排序方法：快速排序---挖坑版

性能：时间复杂度O(Nlog2N)，空间复杂度O(log2N)

优点：平均情况下比较快，不需要额外空间，优化后使用栈不需要递归。

缺点：属于不稳定排序，最坏的情况的情况下时间复杂度为O(N^2)

测试结果：本周优先与每周。



* 测试内容二：解压缩

压缩算法：哈夫曼压缩

测试结果如下：



优点：可以对多种类型文件压缩，doc，bmp，txt，cpp，py，exe文件压缩效果相对较好，其中bmp最优。

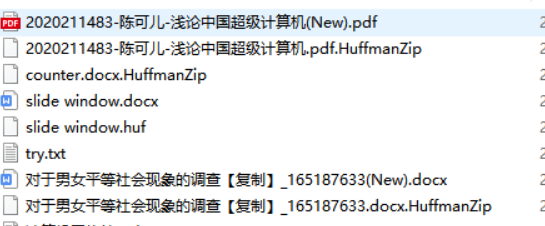
缺点：文件规模较小时传入的辅助信息量相对较大压缩率差。

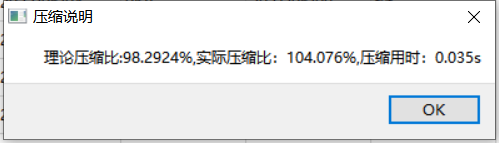
解压部分：对应上部分的操作进行对应读取，主要思路是将压缩时传入的辅助信息提取，建立一个新的Map映射，根据这个映射，将原文件还原。

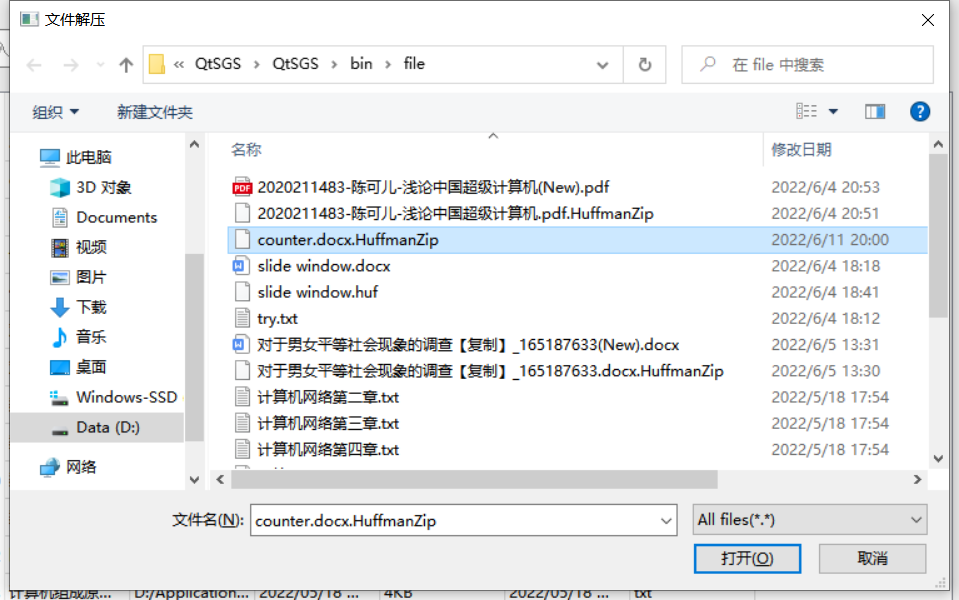
测试结果：

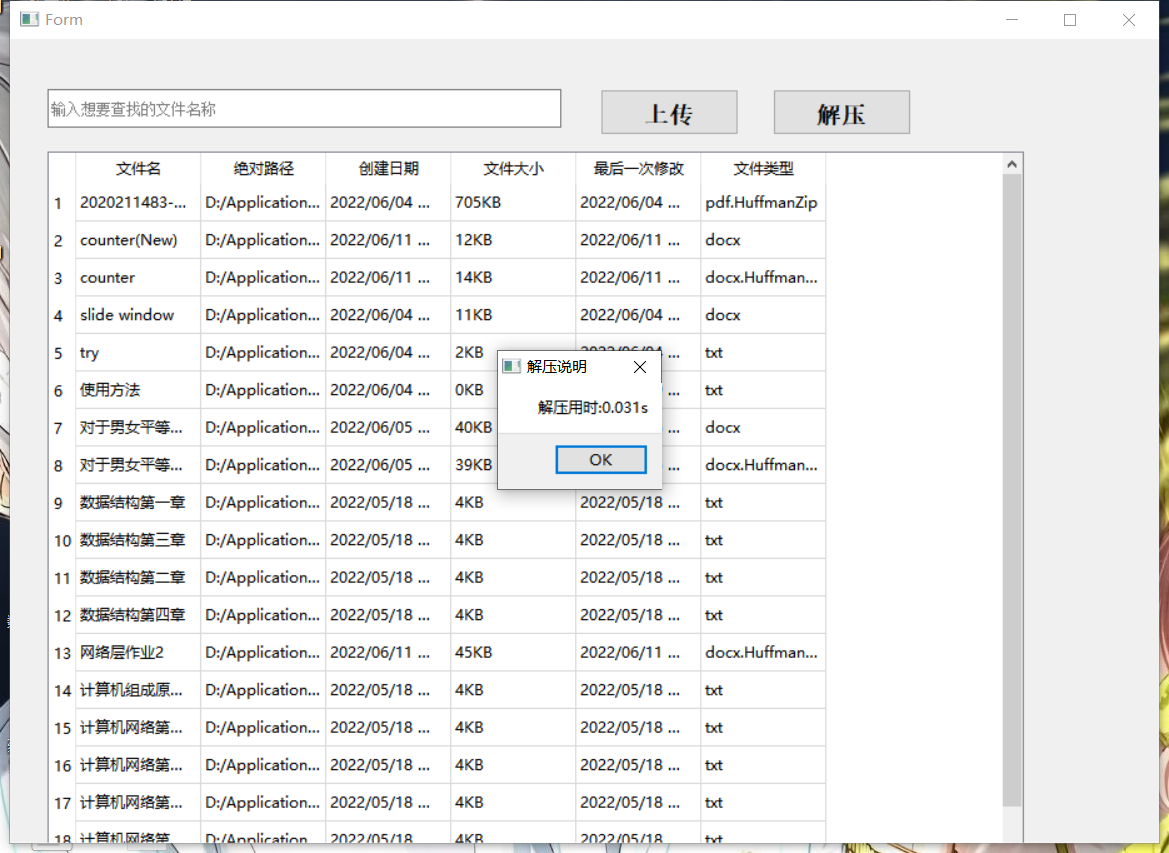
压缩后文件后缀加上了.HuffmanZip

解压后在文件名后加上（New）









* 测试内容三：查找

算法思想：根据课程表名称的首字母或者首字符创建哈希表，然后根据输入课程名称首字母或者首字符，利用哈希查找到对应表项再利用KMP算法逐个匹配。

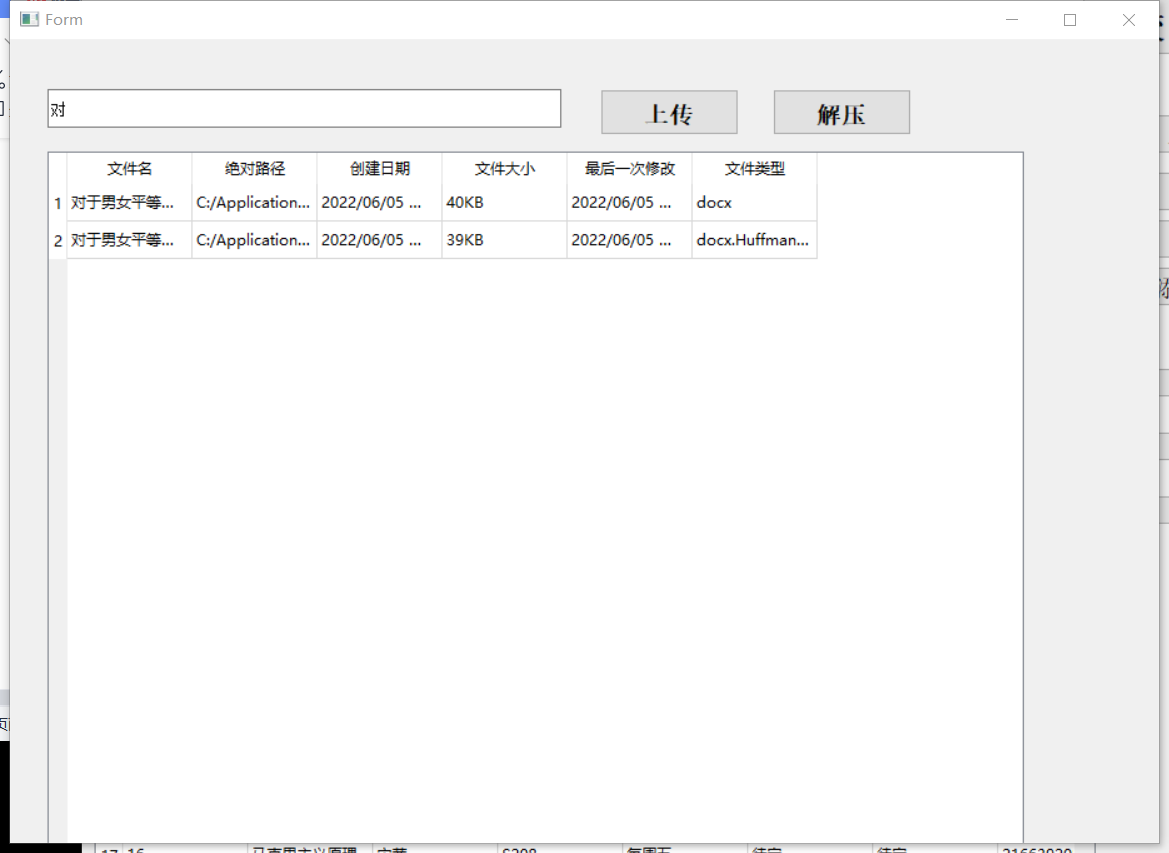
测试结果如下：

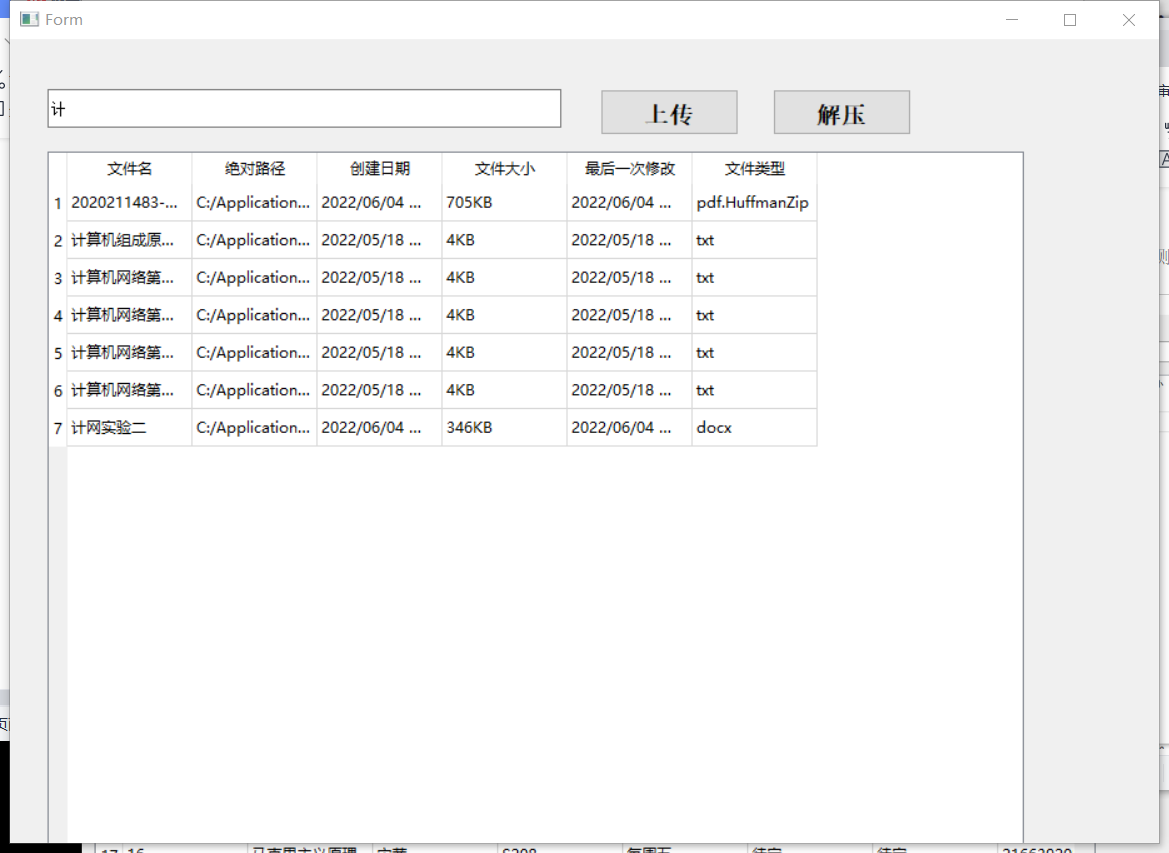
（1）课程查询





1. 文件管理查询





优点：插入和查找速度快，时间复杂度为O(1)。

缺点：扩展性差，需要提前预测数据量的大小，不能有序遍历数据。

* 测试内容四：时间冲突检测

算法思想：将用户输入的活动开始时间分周次和时间分别存储，读取数据库每一项记录，将time字段分周次和时间存，最后进行比较活动开始时间是否落在某一记录的时间区间中，若是，则弹窗提示。

测试结果：





优点：能检测到的冲突比较全面

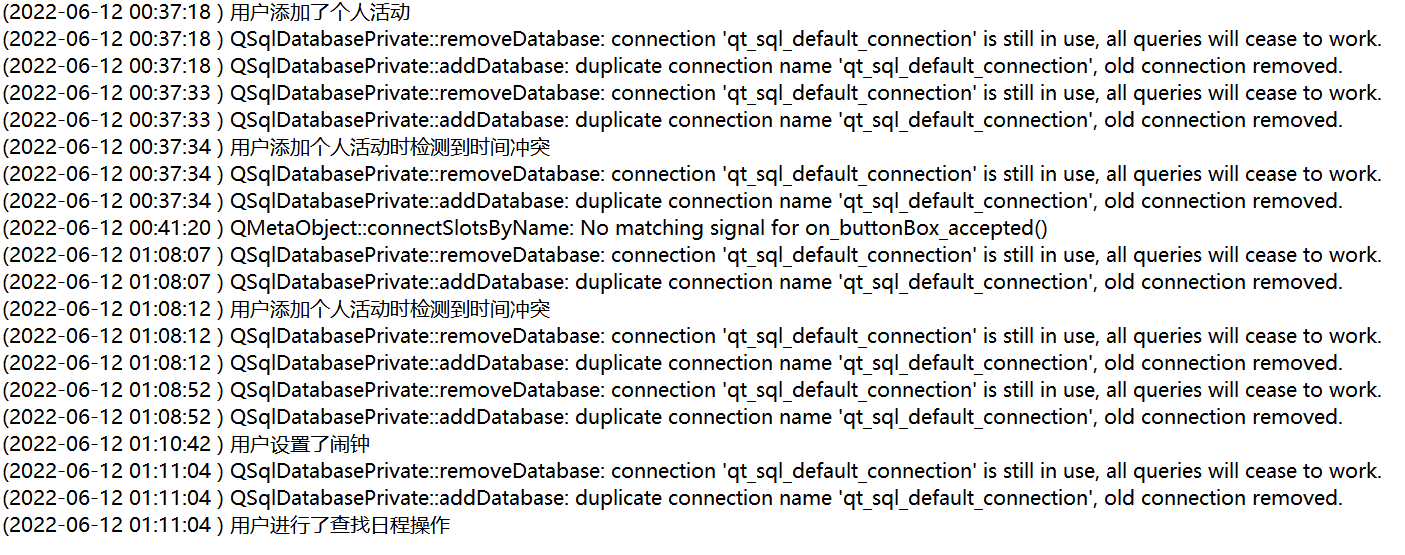
缺点：顺序遍历查找，效率不高

* 测试内容五：日志功能

算法思想：

每次用户进行某操作的时候，用qDebug()<<“操作信息”记录下来；使用 qInstallMsgHandler 函数重定向qDebug、qInfo、qWarning、qCritical、qFatal等宏，输出到txt文件。

测试结果：



优点：除了操作还能输出调试信息，且Qt自带重定向到txt文件的函数qInstallMsgHandler，使用简单。

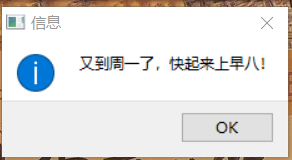
缺点：需要在设计时，在用户的每个操作后添加qDebug()<<“操作信息”。

测试内容六：闹钟

* 算法思想：

使用QDialog类，弹出对话框让用户选择时间段来设置闹钟。判断虚拟系统的时间是否为设置的时间，是的话用QMessageBox弹出提示消息。

测试结果：



优点：实现起来简单，逻辑比较直接。

缺点：只能实现对特定时间的闹钟的设置且闹钟无法精确到秒（默认为00秒）。

1. 校园导航

* 功能设计：从主界面进入导航系统以后，选择当前位置，途经位置，终点位置和导航策略，点击开始导航后系统显示出规划线路。

主要为时间最短策略和距离最短策略。其中时间最短策略采用对道路拥挤度进行标记。

* 测试结果：

当选择【最短距离】时，因校园设计方正，所以可能有多种路径







当选择【最短时间】时，因每条道路拥挤度不同，所以基本上只有一条最佳路径。





测试结果：导航规划基本符合预期

优点：简单易于实现，不容易出错，并且易修改

缺点：算法效率较低，若面对更为复杂的道路结点效率会大打折扣

1. 总结

通过以上测试，可以看出，本系统可实现课程设计基本要求，但在某些算法上还有优化空间。