《数据结构》课程设计总结

学号：2152118

姓名：史君宝

专业：计算机科学与技术

2023 年 7 月

目　录

一、 算法实现

一、 题目：

1. 题目展示：

8. 二叉排序树的建立和删除

输入一组关键值，建立相应的二叉排序树，完成结点的查找和删除操作。

要求：

(1) 可以实现删除根结点、叶子结点以及其它任意结点的功能；

(2) 可随时显示操作的结果。

2. 题目说明：

（1）题目解读：

我们先解读一下题目，从题意我们可以知道，我们需要完成一个二叉排序树的算法设计，这需要我们利用 Qt 的交互化窗口设计出对应的程序。

它能够实现：对于给定的输入数据，能够根据输入顺序，建立对应的二叉排序树。建立完成后，我们应该将树中的结点储存起来，能够实现对结点的查找和删除的操作。同时对于上面所有的操作，我们能够始终保持着二叉排序树的结构，并实时显示对应的图像化结果。

（2）二叉排序树的定义：

对于一棵空树或者是一棵二叉树，若它的左子树不空，则左子树上的所有结点的值均小于根节点的值；若它的右子树不空，则右子树上的所有结点的值均大于根节点的值；同时左右两个子树也都是二叉排序树。

二、软件功能：

1. 开始窗口：

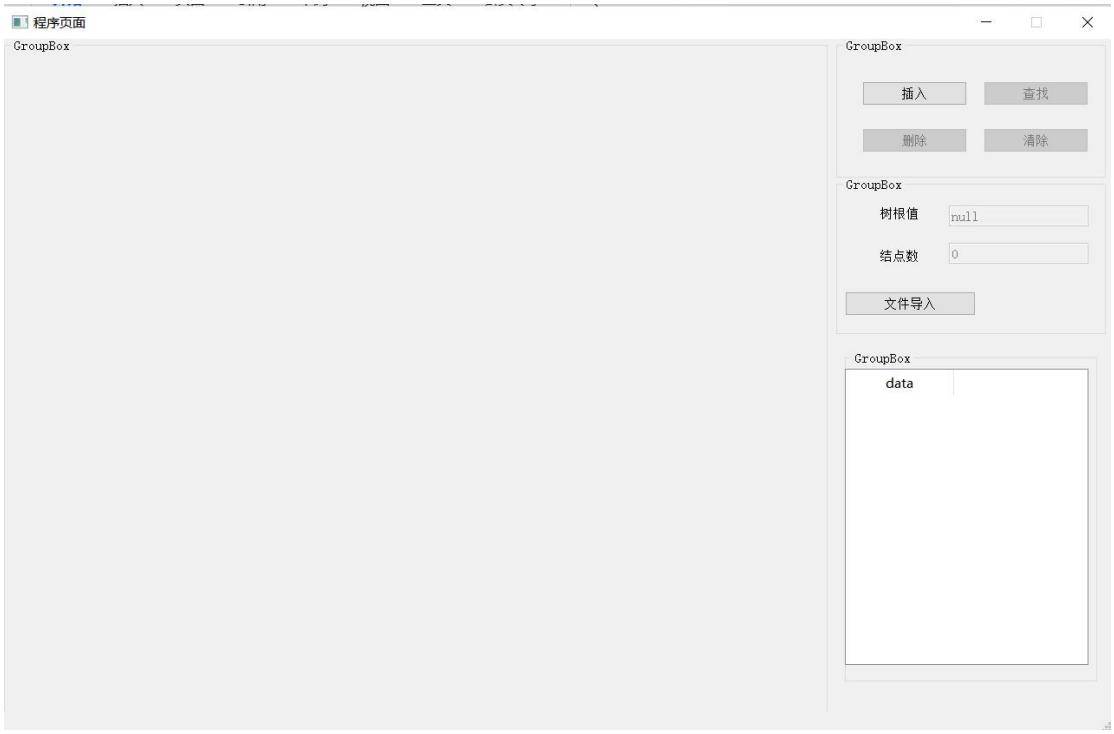
（1）功能说明：

在 Qt 运行程序之后，会进入一个辅助的窗口界面，上面分别介绍本次实验的一些基本信息，比如学号、姓名、项目名称等等。在最下面使用了一个 Qt 的基本部件 QpushButton，我们将其做成了一个“Start”的按钮，在点击之后，就会进入程序的主窗口。

（2）效果展示：

开始窗口效果：

点击 START 按钮之后会进入主窗口：



2. 主窗口：

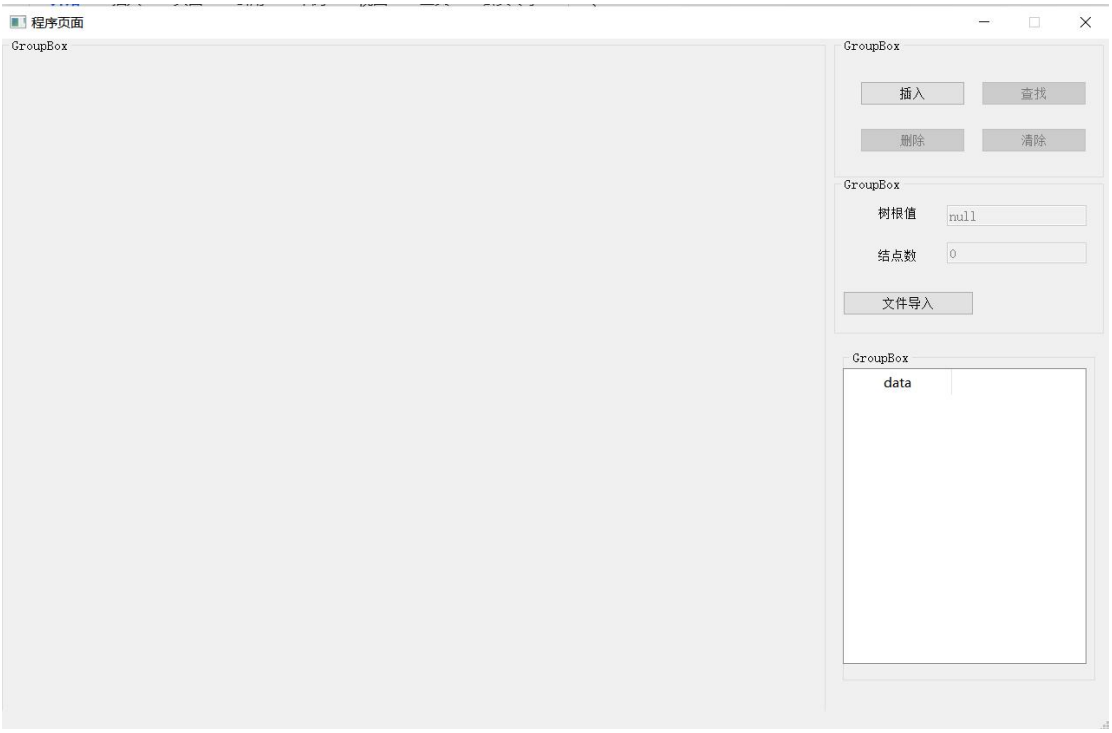
（1）功能说明：

在开始窗口点击“Start”按钮之后，会进入一个我们的主窗口，在最左边是一个 Qt 的展示板，我们将在上面展示我们画出来的二叉排序树。

在右边，上面是一个功能菜单，分别罗列本次实验的一些基本操作，比如插入、删除、查找、清楚和文件导入等等功能。下面是一个信息框，其中有当前的根结点和结点个数，还储存了我们在二叉树中的所有数据，帮助我们使用。

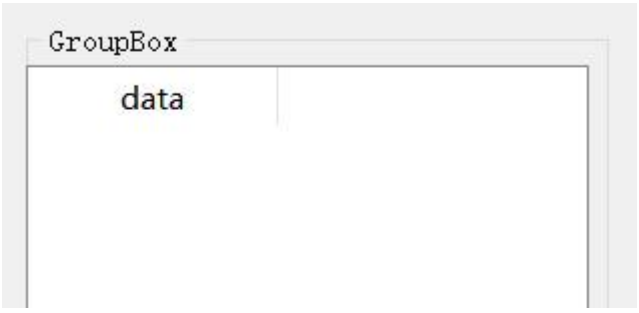
（2）效果展示：

主窗口效果：



功能菜单效果：

信息栏效果：



3. 插入功能：

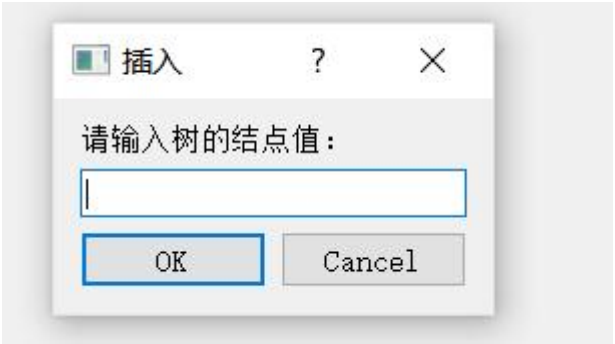
（1）功能说明：

在功能菜单上有插入按钮，使用 Qt 的 QpushButton 部件，点击之后会出现一个对话框，输入数据之后，会在二叉树中查找，如果没有相关结点，就会插入到二叉排序树中，并在画布中实时显示结果，也会更新信息栏中的相关信息。

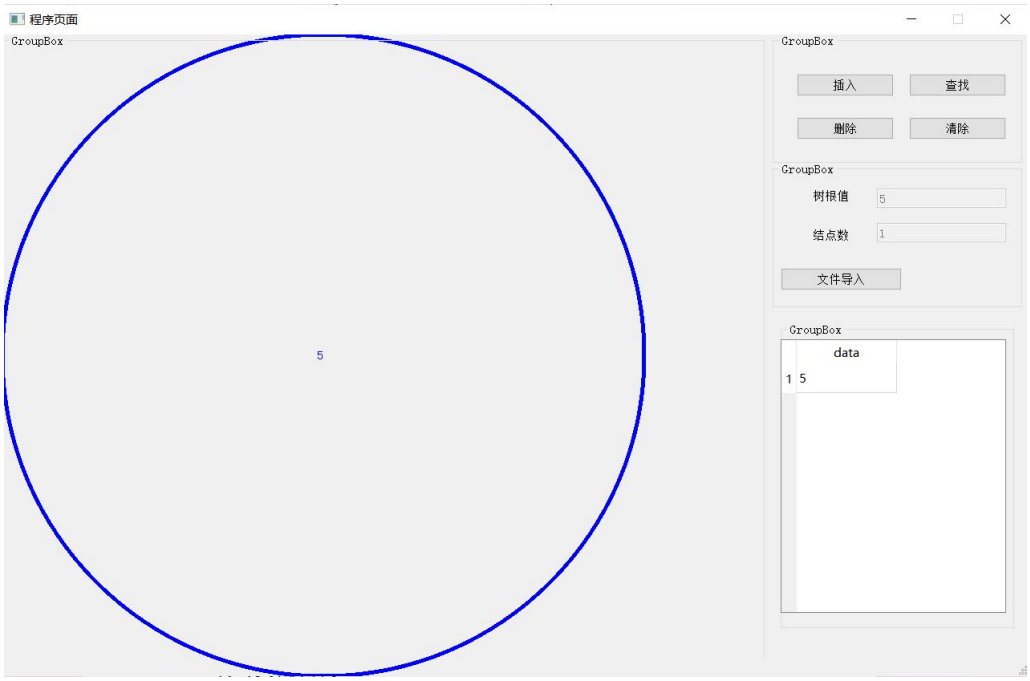
注：由于为了充分利用画布，我们设计了结点的大小，在二叉树的层数增加的时候，结点的大小会发生变化。

（2）效果展示：

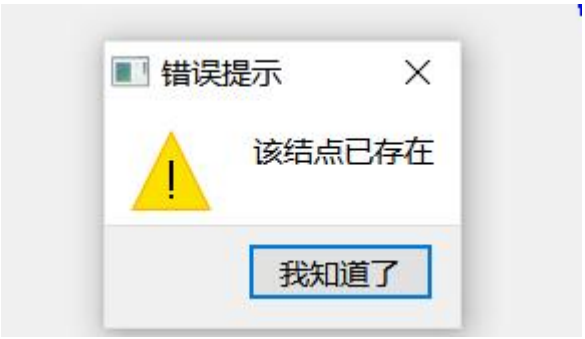
点击按钮之后弹出对话框：



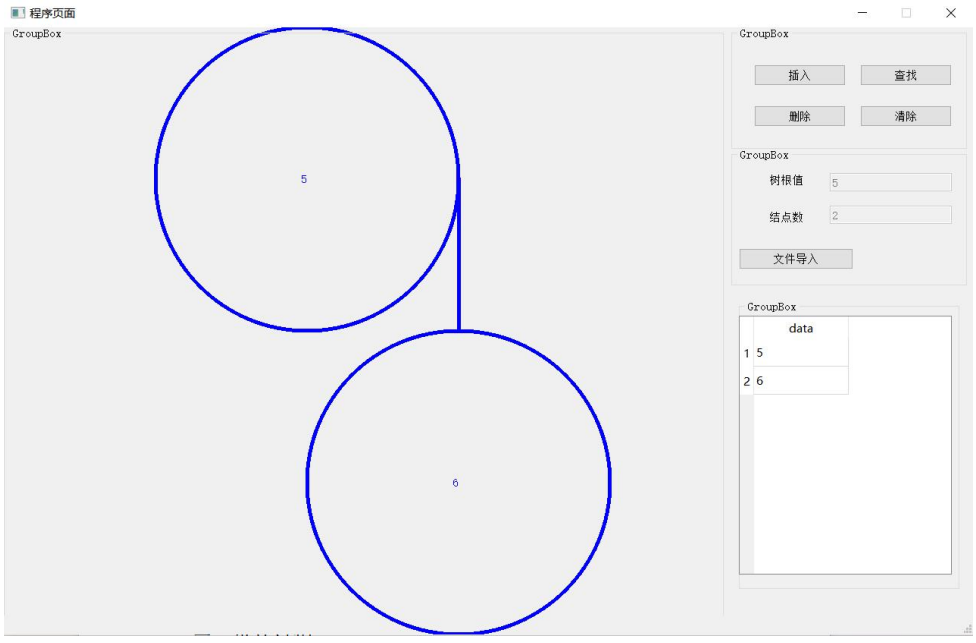
输入的数据画出对应的结点：

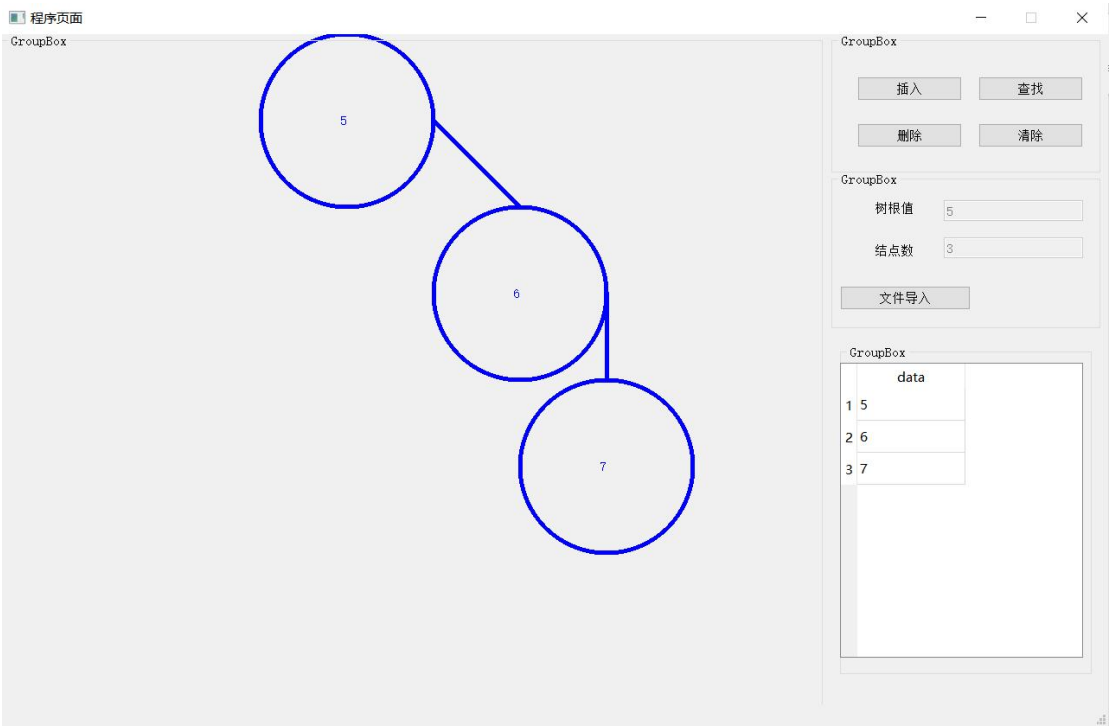


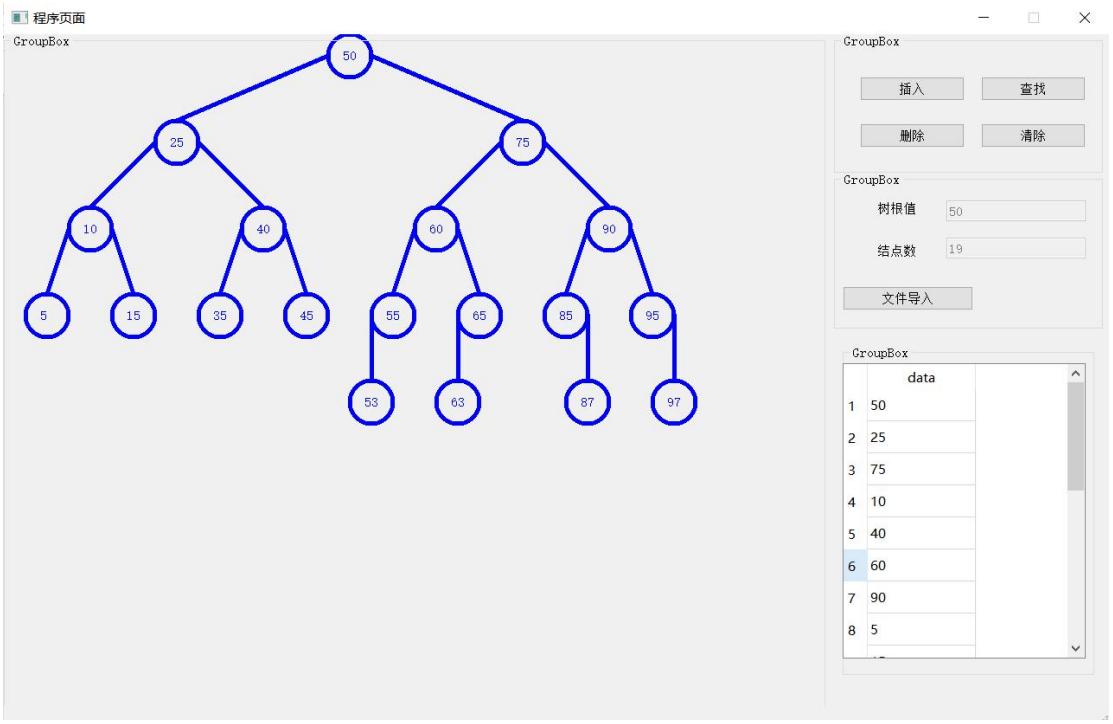
输入二叉树中已经有的数据：



继续插入数据：







4. 查找功能：

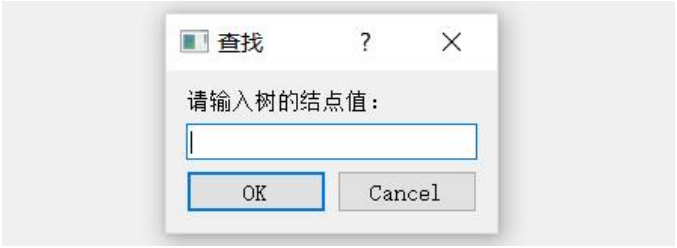
（1）功能说明：

在功能菜单上有查找按钮，使用 Qt 的 QpushButton 部件，点击之后会出现一个对话框，输入数据之后，会在二叉树中查找，查找结果会弹出对话框。

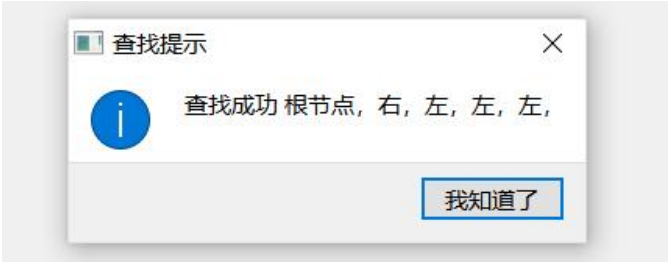
该功能在有数据之后才会解锁。

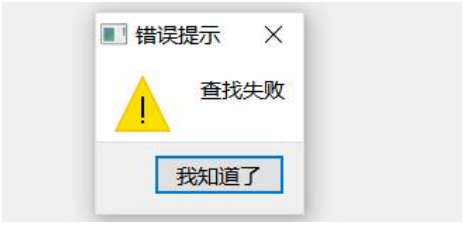
（2）效果展示：

点击按钮之后弹出对话框：



查找结果：





5. 删除功能：

（1）功能说明：

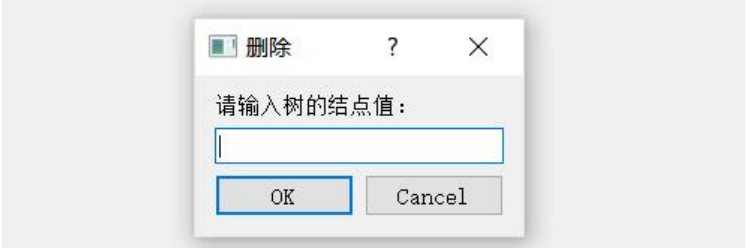
在功能菜单上有删除按钮，使用 Qt 的 QpushButton 部件，点击之后会出现一个对话框，输入数据之后，会在二叉树中查找，查找到结点会进行删除操作，并显示，未查找到会显示错误。

该功能在有数据之后才会解锁。

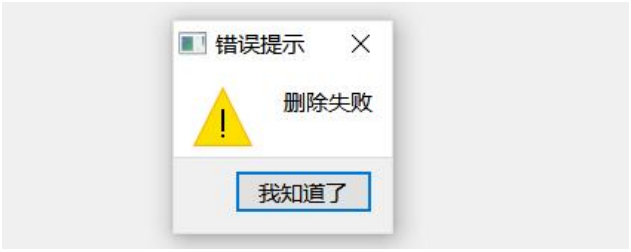
注：在过程中，时刻保持着二叉排序树的结构，非叶子结点也能够正确的删除。

（2）效果展示：

点击按钮之后弹出对话框：

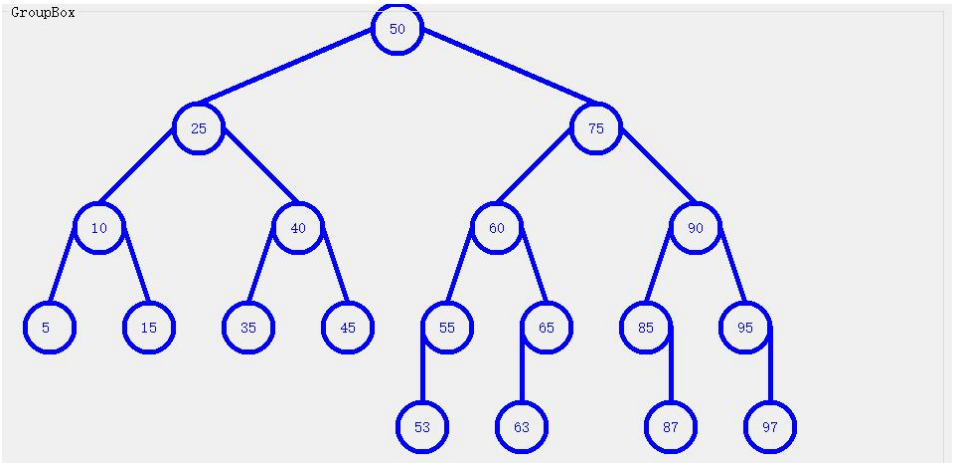


二叉树中没有该结点：

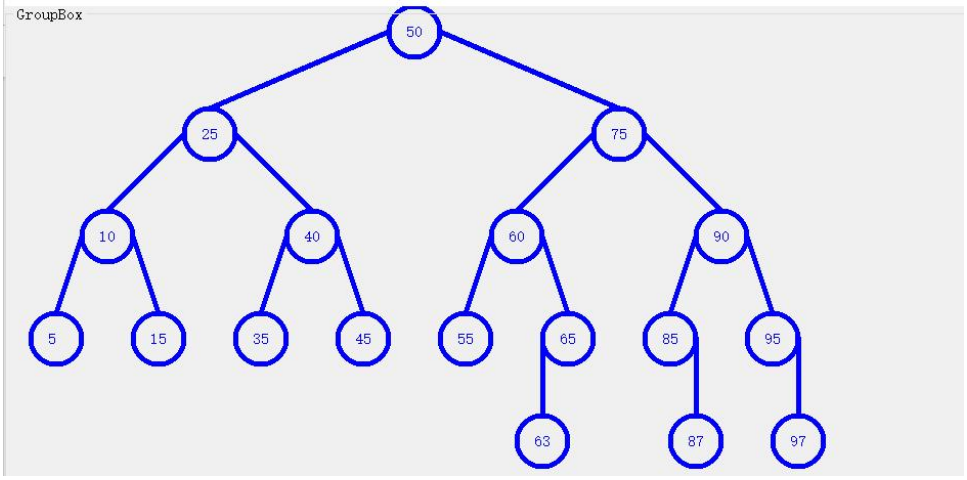


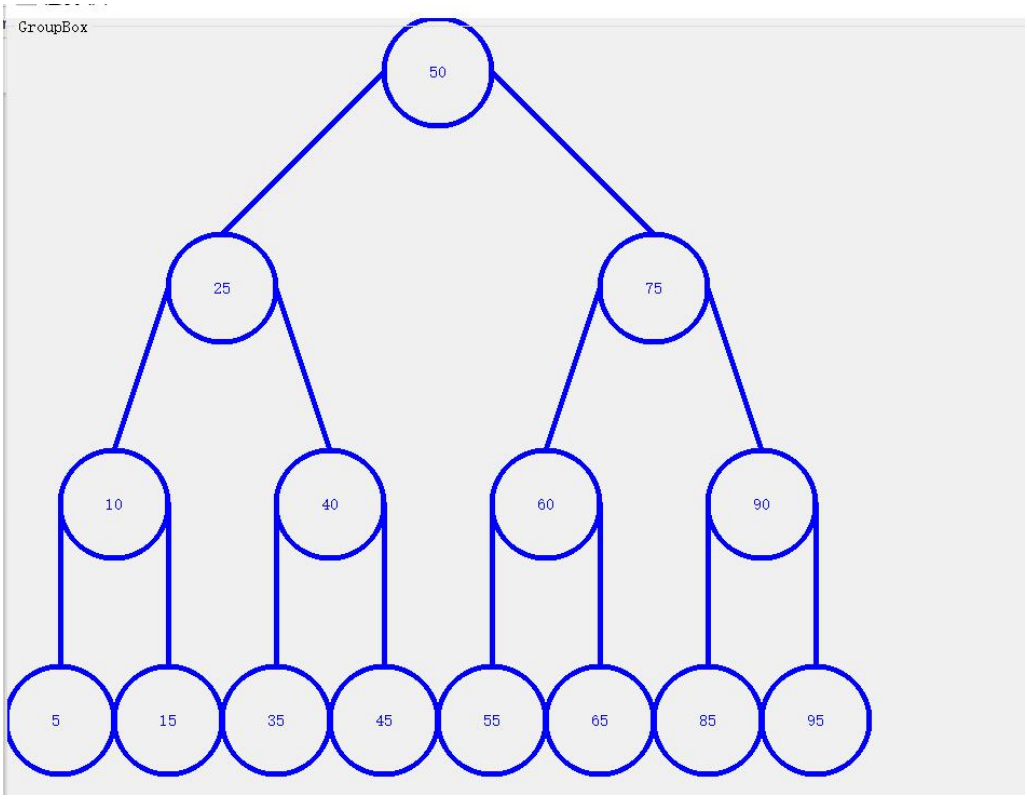
删除效果：

删除前：



持续删除数据：





6. 清除功能：

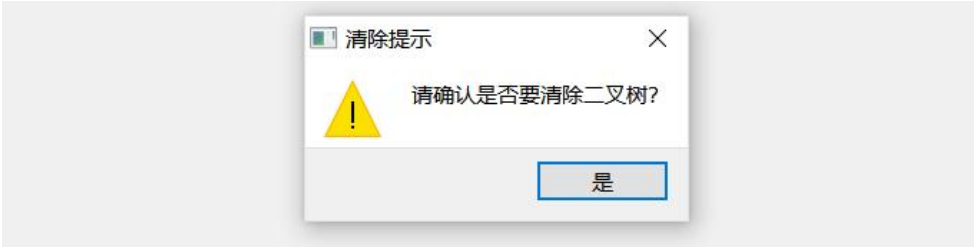
（1）功能说明：

在功能菜单上有清除按钮，使用 Qt 的 QpushButton 部件，点击之后会出现一个对话框，点击确定之后，会清除所有数据，并将左边的画布清空。

该功能在有数据之后才会解锁。

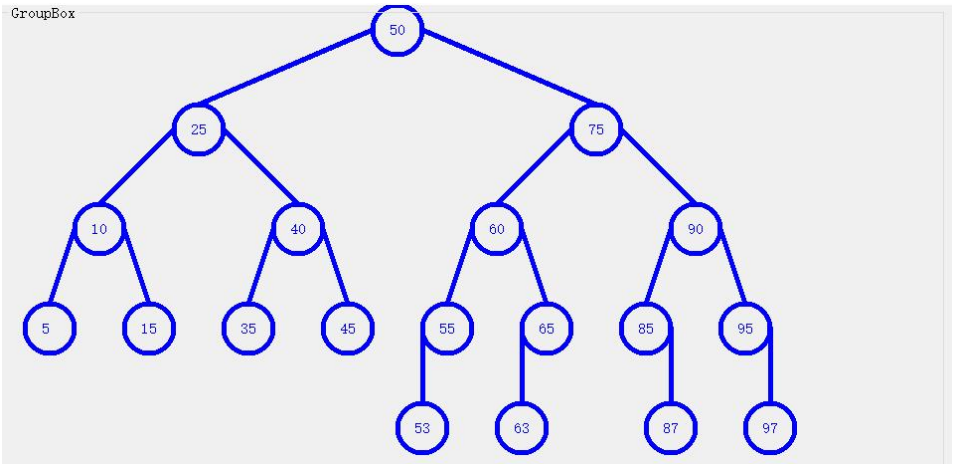
（2）效果展示：

点击按钮之后弹出对话框：



清除效果：

清除前：



清除后：

7. 文件导入功能：

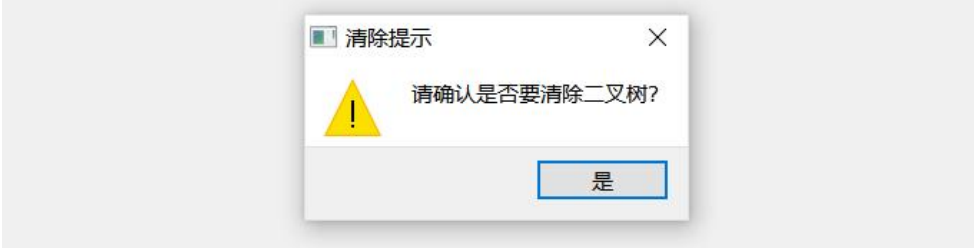
（1）功能说明：

在功能菜单上有文件导入按钮，使用 Qt 的 QpushButton 部件，点击之后会出现一个对话框，点击确定之后，会清除所有数据，并将左边的画布清空。以方便新文件中数据的导入。

该功能在有数据之后才会解锁。

（2）效果展示：

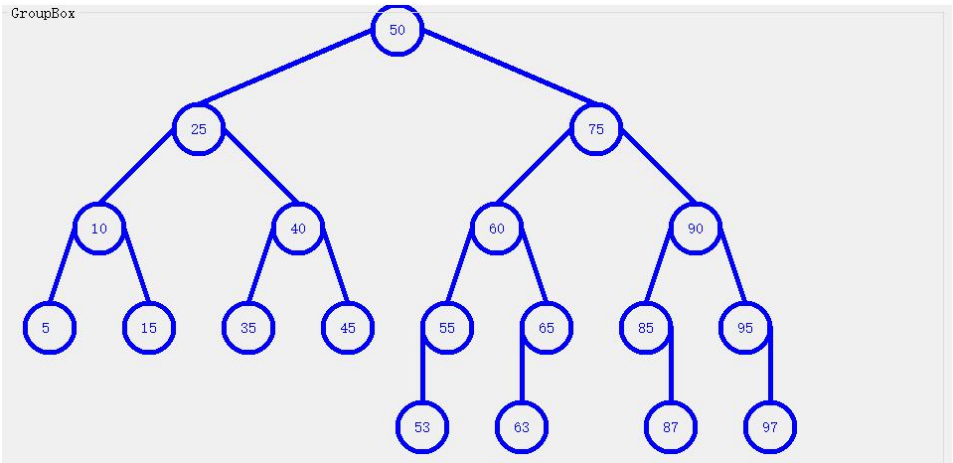
点击按钮之后弹出对话框：



文件导入效果：

导入前：

导入后：



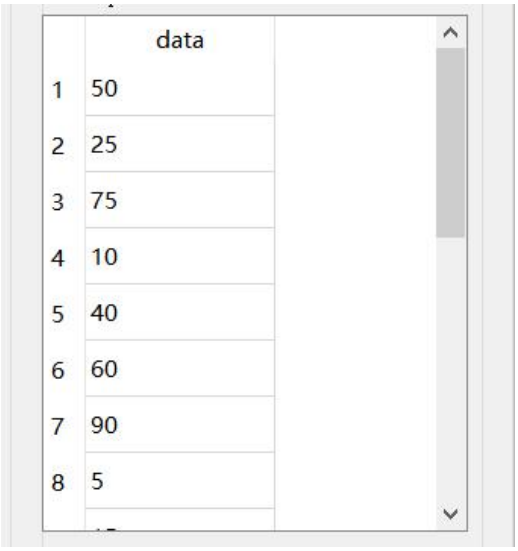
8. 信息框功能：

（1）信息框一：

会实时显示当前二叉树中的根节点的值和二叉树中的所有结点个数。

（2） 信息框二：

会实时显示当前二叉树中的所有数据。



三、设计思想：

1. 设计过程：

（1）在整个的设计过程中先确定需要完成的任务和功能操作，然后根据对应的应用要求我们选择对应的 Qt 的部件，并确定主窗口中的所有功能和操作，并对主窗口进行 UI 设计，确定整体的布局。

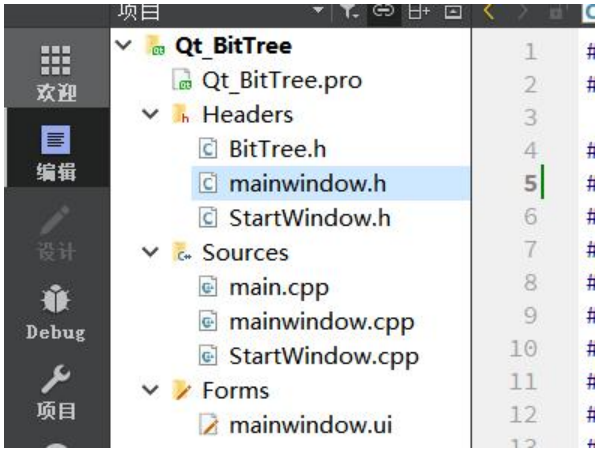
（2）然后我们根据所确定的 UI 设计选择我需要用到的所有 Qt 的部件，比如用到对话框，我们使用 QMessageBox，对于对话框的输入我们使用 QInputDialog，对于文件的读写，我们使用 QFile，我们进行画图我们要使用 QPainter 部件。

（3）之后我们根据二叉排序树的要求，我们构建 BitTree.h 在其中编写二叉树的数据结构，并声明相应的变量，确定对应操作的函数。

（4）之后我们具体的编写函数，实现对二叉树的插入、删除、查找、清除等等所有的功能，并在其中使各个函数相互关联起来，并与对应的按钮等等 UI 元素相关联，建立 connect 的信号和槽。

（5）最后我们依次进行调试，并对 UI 等等进行美化设计，比如我们可以加一个开始的窗口，使效果更加美观。

（6）编写的内容架构显示：



四、逻辑结构及物理结构：

1. 逻辑结构：

在整个编写的过程中，我们使用了所有的逻辑结构，顺序结构、选择结构和循环结构。

2. 物理结构：

（1）二叉树结点的物理结构：

二叉树的结点是结构体类型，其中储存了很多信息，有 int 型的结点值；有对应的结构体指针 LeftChild 和 RightChild，对应的左右子结点；有 bool 型的左右子结点的标志位；有 int 型当前结点的深度；还有 double 型的 x 和 y 坐标帮助我们之后画图。

（2）二叉树结点间的连接方式：

二叉树的储存并不是使用顺序存储结构，而是使用链式存储结构，彼此之间依靠指针来连接。

（3）函数的结构：

void paintEvent(QPaintEvent *)override; 用于在画布中画二叉树。

void cal_size(); 用于综合二叉树的深度和结点个数，选择一种适宜的结点大小。

void on_pushButtonSearch_clicked(); 实现查找功能。

void on_pushButtonInsert_clicked(); 实现插入功能。

```
void on_pushButtonDelete_clicked();  实现删除功能。

void on_pushButtonClear_clicked();  实现清除功能。

void on_pushButtonFilein_clicked();  实现文件导入功能。

void dataInsert(QString key);    帮助实现文件中数据的插入。
```

五、开发平台：

开发平台：QT 5.15.0 (MSVC 2017)
运行环境：QT Creator 4.9.2 Community
其它的库：C++ STL (纯 C++ 方式实现时)

六、系统的运行结果分析说明：

1. 调试过程和调试方法

在使用 Qt 编程调试的时候，并不像我们平时所用的编译器，其调试并不容易，而且对于报错的信息我们也经常由于知识欠缺而不太懂，所以我这里给了一些调试的方法。

（1）qDebug()<<“信息”调试：

有时候我们的程序会因为一些指针访问错误而发生异常的中断，这时候我们可以采用在程序中添加 qDebug()<<“信息”代码，来确定我的代码是否已经执行到某个位置了，通过一系列的 qDebug()我们可以大致确定程序异常中断的位置。

（2）设置断点调试：

确定程序异常中断的大致位置之后，我们就可以利用 Qt 中的调试模式，通过设置断点来逐步执行程序的语句。我们可以观察程序执行过程中的变量值，确定是发生了什么样的错误。

（3）具体实例，寻找函数中的问题：

上面解决的是程序语法和程序运行过程指针越界的问题，对于函数不能达到我们想要实现的具体功能，我们可以输入具体的实例，通过观察程序执行的现象，来查找程序中的逻辑错误。

2. 开发程序所达到的成果。

（1）正确性：

在程序的执行中，我们已经完成了二叉排序树的创建和绘制，同时能够正确的查找其中的结点。在向其中插入和删除数据的时候能够保证新的二叉树仍然满足二叉排序树的定义。

在信息框中，能够实时更新当前的根节点信息和二叉树总结点树的信息，同时储存数据的 TableEdit 中的数据信息与二叉树相对应，能够实现数据随插入和删除的实时更新。

（2）稳定性：

在程序执行过程中，不会发生数组或指针越界的情况，程序也不会无关终止，基本上实现了程序的稳定。

（3）容错能力：

在所有输入的对话框中，均可以对无输入这种情况进行检测，容错能力较强，对于当前不能进行的操作，会自动将其按钮关闭，比如在二叉树中没有结点的时候，不可以进行查找、删除和清除的操作，对应的按钮会关闭。在文件导入过程中会自动将之前的二叉树清除，不会发生误操作的情况。

但是，为了实现结点的大小清晰可辨别，我们在树的深度为 6 以下的时候，每当总深度变化的时候，结点大小会动态变化，尽可能的占据整个画布，可以实现 5 层满二叉树的清晰表示。但是在深度 6 及 6 以上（私以为是树太偏某一侧了），所以换了一种动态调节结点的方法，可能会发生结点相重叠，如果想避免可以找到下面函数（但是结点可能过小）

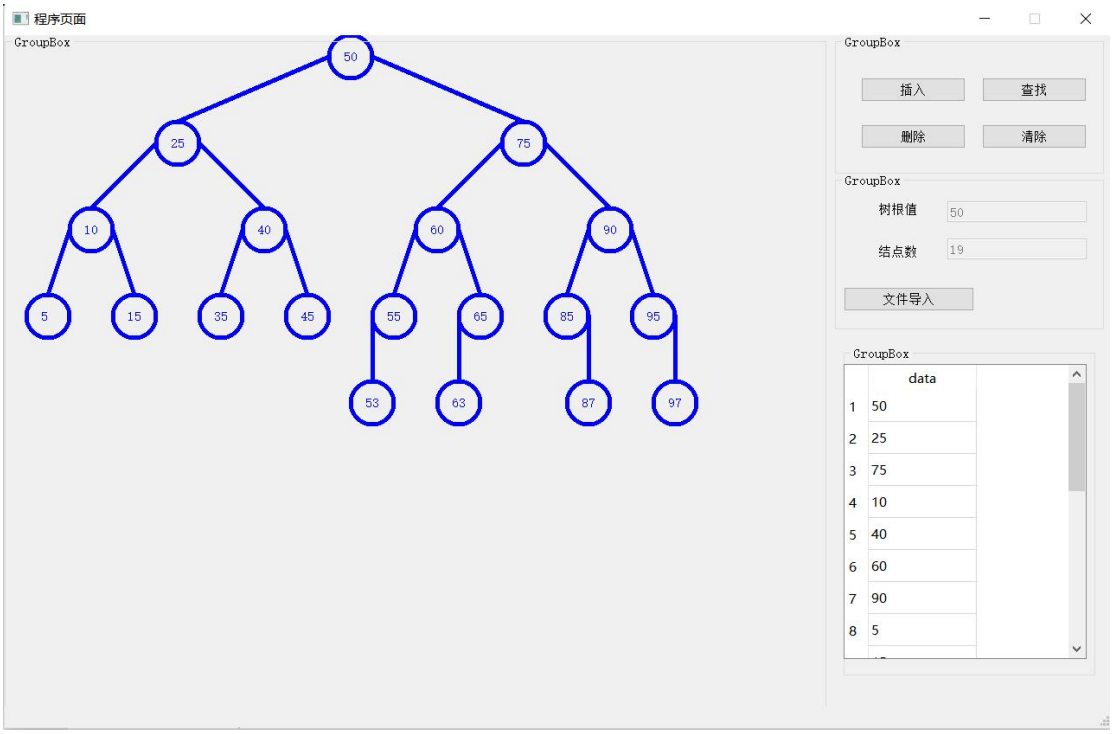
```
81  void MainWindow::cal_size(){
82      int sum = 1;
83
84      if(head->LeftChild){
85          for(int i=0; i<head->LeftChild->Depth-1; ++i){
86              sum*=2;
87          }
88
89          size = 800/sum;
90
91          if(size<800/Tree_num && head->LeftChild->Depth>=6)
92              size = 800/Tree_num;
93      }
94
95      return;
96  }
97
```

将其修改为：

```
void MainWindow::cal_size(){
    int sum = 1;

    if(head->LeftChild){
        for(int i=0; i<head->LeftChild->Depth-1; ++i){
            sum*=2;
        }

        size = 800/sum;

//        if(size<800/Tree_num && head->LeftChild->Depth>=6)
//            size = 800/Tree_num;
    }

    return;
}
```

3. 具体案例分析。

会附带一个 test_one.txt 的文件，将其按文件导入后会有下面的效果：



之后可以自行进行操作，由于不具有典型性，在这里就不予演示了。

七、操作说明

具体的操作说明之前已经展示过了，在这里就不赘述了。

二、综合应用

一、 题目：

1. 题目展示：

8. ★★现需开发一个简单的文本编辑器（支持 200 字以内的输入即可），能支持输入数据的形式和范围包括大写、小写的英文字母、任何数字及标点符号。该编辑器需要实现以下功能：
（1）对输入的文字进行统计，统计出文字、数字、空格的个数；
（2）统计某一字符串在文章中出现的次数，并输出该次数；
（3）删除某一子串，并将后面的字符前移；
（4）保存文本修改内容，可以撤销修改和恢复修改。

2. 题目说明：

（1）题目解读：

我们先解读一下题目，从题意我们可以知道，我们需要完成一个文本编辑器的综合应用，这需要我们利用 Qt 的交互化窗口设计出对应的程序。

它能够实现：对于我们在文本框中输入的字符串，我们能够实现对于任意字符的查找，能够不断地查找下一个，直到全部查找完成才可以。

对于字符串中文字、数字和空格我们能够实现统计，并输出对应的统计个数。

对于所输入的字符串中的某个字符串进行统计，并输出对应的统计个数。

对于所输入的字符串中的某个字符串进行删除，并将后面的字符前移。

对于所输入的字符串进行保存，在我们对文本进行修改的时候会进行保存，在之后我们可以撤销修改和恢复修改。

二、软件功能：

1. 开始窗口：

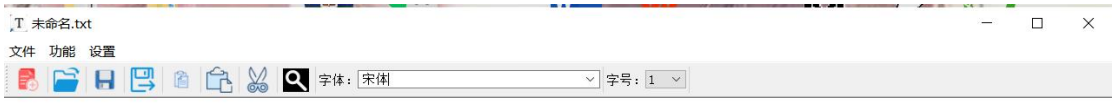
（1）功能说明：

在 Qt 运行程序之后，会进入一个辅助的窗口界面，上面分别介绍本次实验的一些基本信息，比如学号、姓名、项目名称等等。在最下面使用了一个 Qt 的基本部件 QpushButton，我们将其做成了一个“Start”的按钮，在点击之后，就会进入程序的主窗口。

（2）效果展示：

开始窗口效果：

点击 START 按钮之后会进入主窗口：



2. 主窗口：

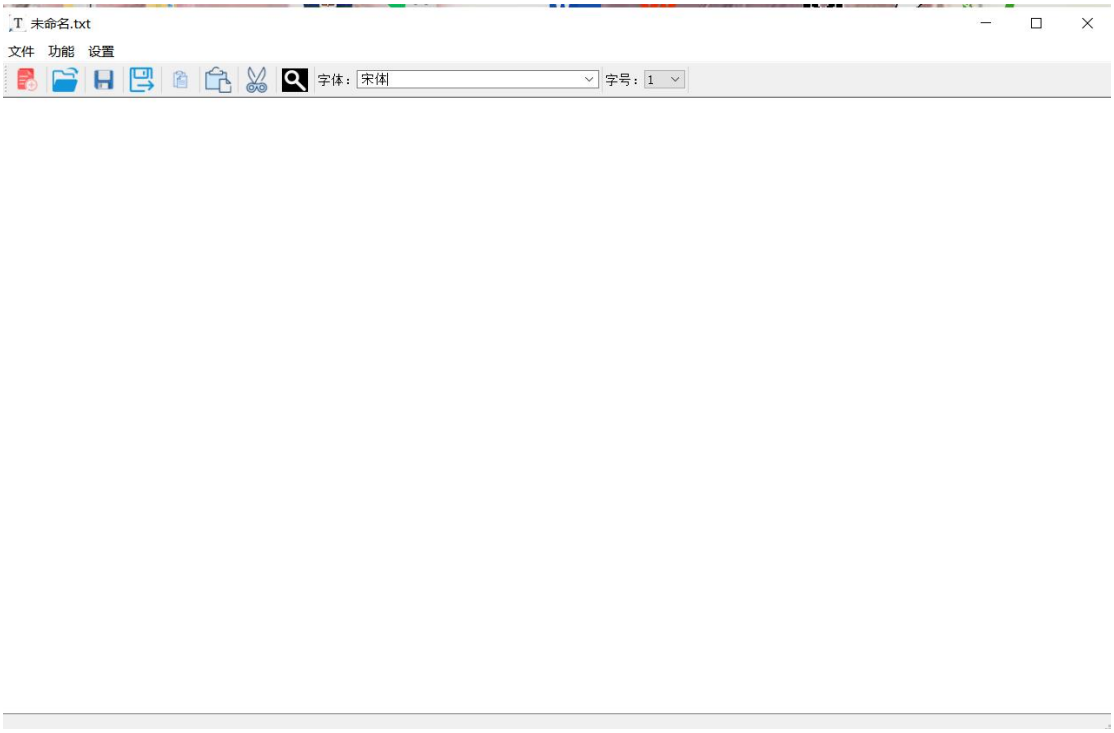
（1）功能说明：

在开始窗口点击“Start”按钮之后，会进入一个我们的主窗口，最上面有两栏，分别是菜单栏和工具栏，其中包含很多我们所实现的功能。

然后是一个 Qt 部件 QTextedit，用于文本框我们之后可以在其中输入信息，实现文本编辑功能。

（2）效果展示：

主窗口效果：



工具栏菜单效果：

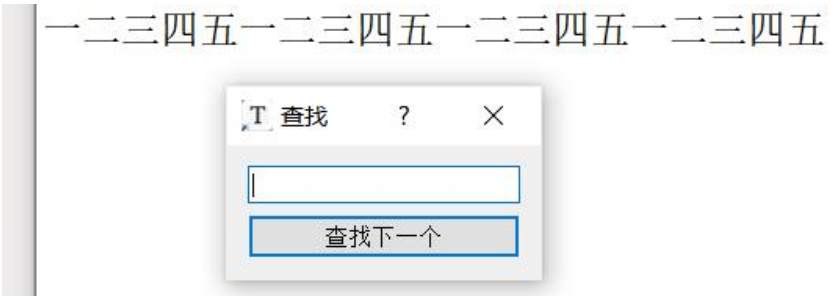
3. 查找功能：

（1）功能说明：

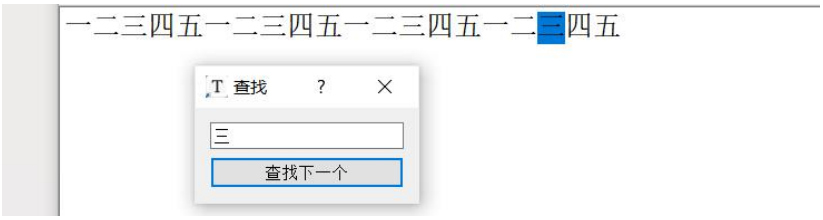
在菜单栏上有查找按钮，或者使用工具栏中的放大镜图标的按钮，它使用 Qt 的 QpushButton 部件，点击之后会出现一个对话框，让你输入要查找的字符。输入数据之后，会在文本框中进行查找，会依次将找到的所有数据用高亮光标显示出来，点击查找下一个后，会继续查找下一个。

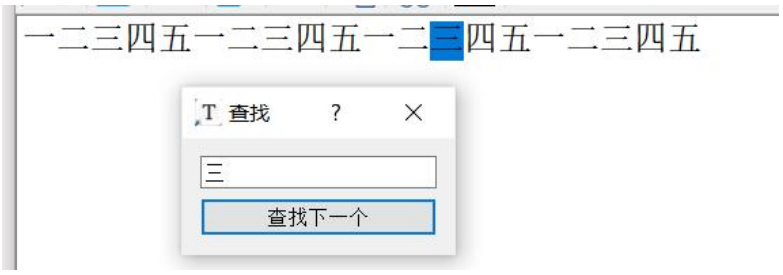
（2）效果展示：

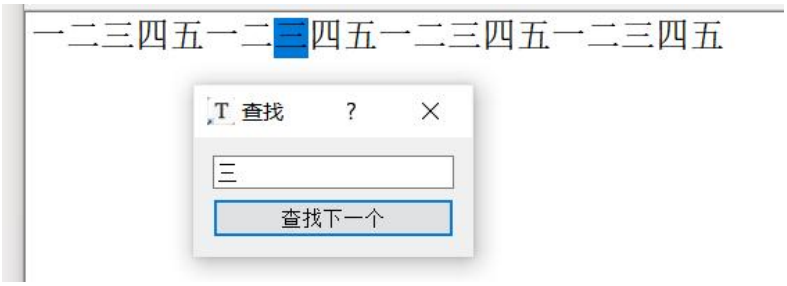
点击按钮之后弹出对话框：

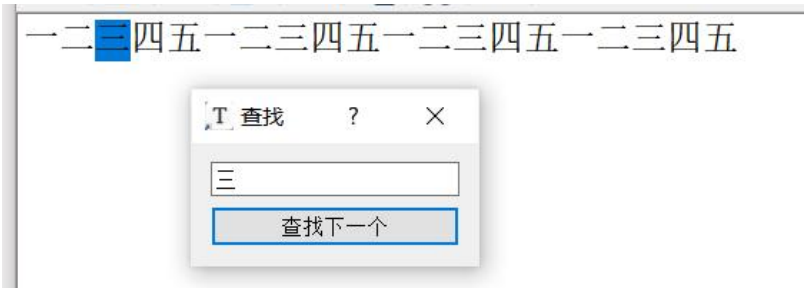


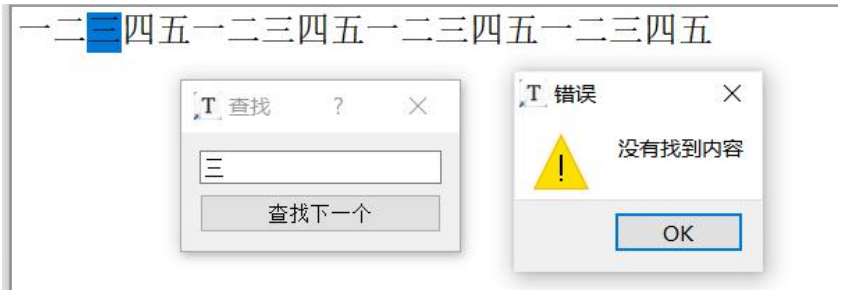
持续进行查找：











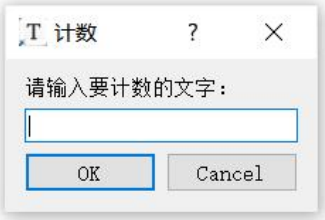
4. 统计功能：

（1）功能说明：

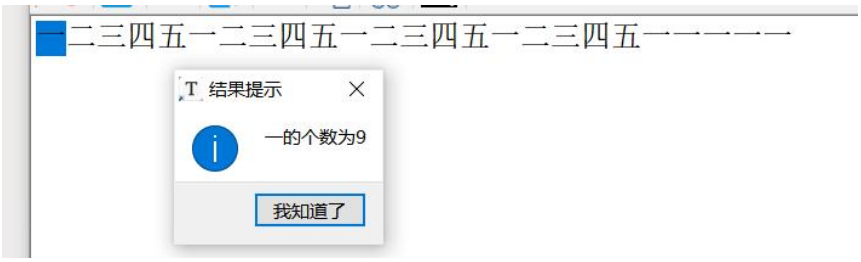
在功能菜单上有统计相关按钮，使用 Qt 的 QpushButton 部件，点击之后会出现一个对话框，输入对应字符之后，会在文本框中查找，查找结束会弹出对话框。告知所查找的字符的个数，即统计

（2）效果展示：

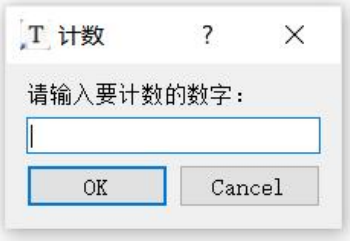
统计文字：



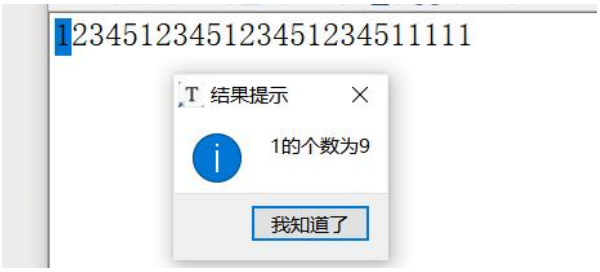
统计结果：



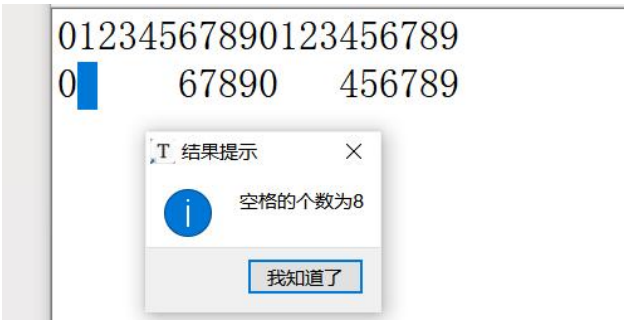
统计数字：



统计结果：



统计空格：



5. 统计字符串功能：

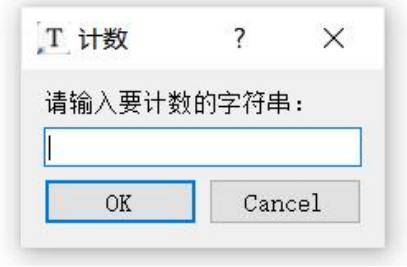
（1）功能说明：

在功能菜单上有统计相关按钮，使用 Qt 的 QpushButton 部件，点击之后会出现一个对话框，输入对应字符串之后，会在文本框中查找，查找结束会弹出对话框。告知所查找的字符串的个数，即统计

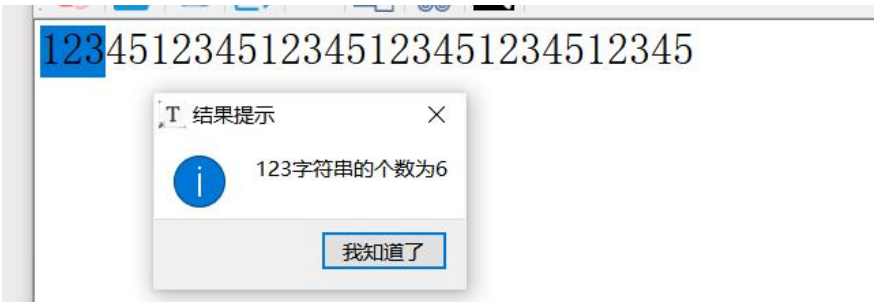
注：在过程中如果查找 aa，对于 aaa 算找到 2 个，即能找到的所有子串。

（2）效果展示：

点击按钮之后弹出对话框：



统计结果：



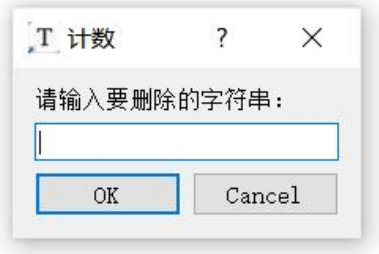
6. 删除字符串功能：

（1）功能说明：

在功能菜单上有删除字符串按钮，使用 Qt 的 QpushButton 部件，点击之后会出现一个对话框，输入对应字符串之后，会在文本框中查找，查找结束会将文本框中对应的字符串删除，并将后面的字符放到前面。

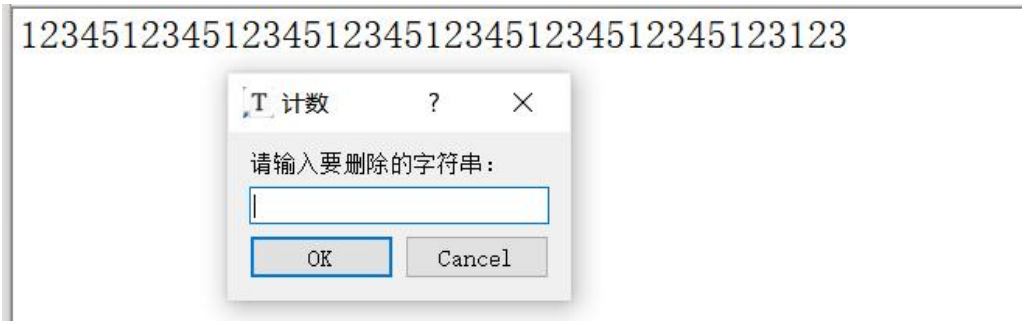
（2）效果展示：

点击按钮之后弹出对话框：

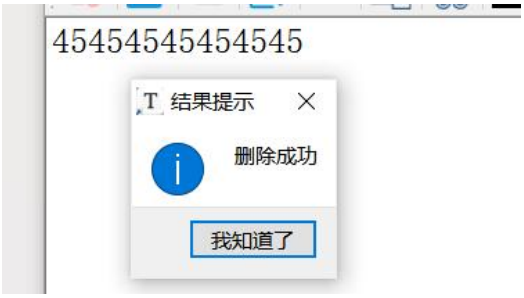


删除效果：

删除前：



删除后：



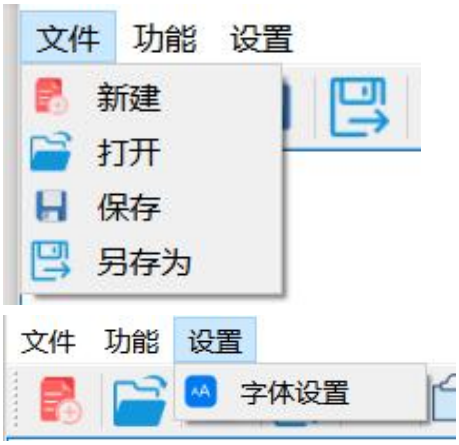
7. 信息框功能：

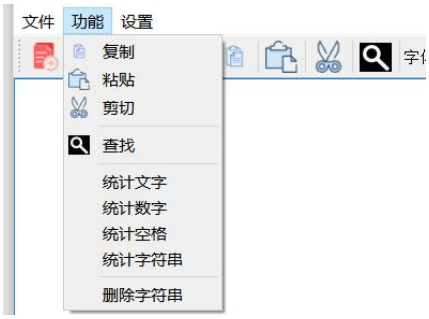
（1）工具栏：

实现对于一些操作的简化，比如查找功能。

（2） 菜单栏：

包含所有的功能





三、设计思想：

1. 设计过程：

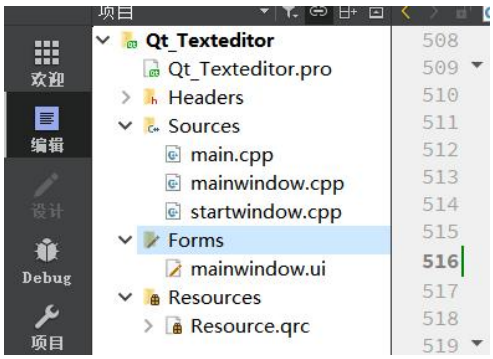
（1）在整个的设计过程中先确定需要完成的任务和功能操作，然后根据对应的应用要求我们选择对应的 Qt 的部件，并确定主窗口中的所有功能和操作，并对主窗口进行 UI 设计，确定整体的布局。

（2）然后我们根据所确定的 UI 设计选择我需要用到的所有 Qt 的部件，比如用到对话框，我们使用 QMessageBox，对于对话框的输入我们使用 QInputDialog，对于文件的读写，我们使用 QFile 部件。

（3）之后我们具体的编写函数，实现对文本框的查找、统计、删除等等所有的功能，并在其中使各个函数相互关联起来，并与对应的按钮等等 UI 元素相关联，建立 connect 的信号和槽。

（4）最后我们依次进行调试，并对 UI 等等进行美化设计，比如我们可以加一个开始的窗口，使效果更加美观。

（5）编写的内容架构显示：



四、逻辑结构及物理结构：

1. 逻辑结构：

在整个编写的过程中，我们使用了所有的逻辑结构，顺序结构、选择结构和循环结构。

2. 物理结构：

（1）内容存储结构：

内容的储存我们使用的是 QString 的数据结构，即字符串类型来储存所有的文本内容。

（2）函数的结构：

void Creat_Menu();　用于创建菜单栏。

void New_File();　用于创建新文件。

bool maybeSave();　是否保存。

bool Save_File();　用于保存文件。

bool Save_Other_File();　用于另存为文件。

bool Save_File(const QString &fileName);　用于保存文件于指定位置。

bool Open_File(const QString &fileName);　用于打开指定位置的文件。

void Creat_Tool();　用于创建工具栏。

void Set_Font(const QFont &font);　用于设置字体。

void Set_FontSize(int index);　用于设置字号。

void connectImpl();　用于连接槽函数。

void Find_Text();　用于在文本中查找字符。

void Count_characters();　用于统计文字。

void Count_number();　用于统计数字。

void Count_space();　用于统计空格。

void Count_chars();　用于统计字符串。

void Delete_chars();　用于删除字符串。

五、开发平台：

开发平台：QT 5.15.0 (MSVC 2017)
运行环境：QT Creator 4.9.2 Community
其它的库：C++ STL (纯 C++ 方式实现时)

六、系统的运行结果分析说明：

1. 调试过程和调试方法

在使用 Qt 编程调试的时候，并不像我们平时所用的编译器，其调试并不容易，而且对于报错的信息我们也经常由于知识欠缺而不太懂，所以我这里给了一些调试的方法。

（1）qDebug()<<“信息”调试：

有时候我们的程序会因为一些指针访问错误而发生异常的中断，这时候我们可以采用在程序中添加 qDebug()<<“信息”代码，来确定我的代码是否已经执行到某个位置了，通过一系列的 qDebug()我们可以大致确定程序异常中断的位置。

（2）设置断点调试：

确定程序异常中断的大致位置之后，我们就可以利用 Qt 中的调试模式，通过设置断点来逐步执行程序的语句。我们可以观察程序执行过程中的变量值，确定是发生了什么样的错误。

（3）具体实例，寻找函数中的问题：

上面解决的是程序语法和程序运行过程指针越界的问题，对于函数不能达到我们想要实现的具体功能，我们可以输入具体的实例，通过观察程序执行的现象，来查找程序中的逻辑错误。

2. 开发程序所达到的成果。

（1）正确性：

在程序的执行中，我们已经完成了文本框的删除和输入，并进行修改。同时能够正确的查找其中的字符信息。在其中进行统计汉字、数字和空格的时候，能够保证统计始终从文本末尾开始向前查找。进行字符串的删除的时候，能够实现所有字符串的查找并删除，并将后面的字符提前。

（2）稳定性：

在程序执行过程中，不会发生数组或指针越界的情况，程序也不会无关终止，基本上实现了程序的稳定。

（3）容错能力：

在所有输入的对话框中，均可以对无输入这种情况进行检测，容错能力较强，对于当前不能进行的操作，会自动将其按钮关闭，比如在二叉树中没有结点的时候，不可以进行查找、删除和清除的操作，对应的按钮会关闭。在文件导入过程中会自动将之前的二叉树清除，不会发生误操作的情况。

3. 具体案例分析。

文本编辑器属于尽情发挥，没有具体的案例。

七、操作说明

具体的操作说明之前已经展示过了，在这里就不赘述

三、实践总结

1.所做的工作

（1）Qt 的下载安装和学习：

在这次的课程设计中，自己选择了 Qt 这一工具进行开发，在刚开始的时候根据教程成功下载并安装了 Qt 软件，配置了对应的环境。并对 Qt 的基础知识进行了深入的学习，基本了解 Qt 的常用的部件，成功实现了 Qt 的 UI 设计。并利用设计中的部件编写函数，比如槽函数等等，实现了对应的功能。

（2）与数据结构知识的结合：

在这次的课程设计中我们进行算法实现，与之前学到的数据结构知识相结合，比如开发过程中的二叉排序树的知识，在文本编辑器中的字符串匹配算法 KMP 算法等等。与之前的知识相结合，活学活用，加深了理解。

（3）实现了程序的实时显示：

在二叉排序树这一题中，需要对二叉树进行实时的绘制。我们使用 Qpainter 进行绘制，需要对其函数进行一定的了解，并对结点大小和位置进行一定的设计。同时要实现实时表示，这是比较困难。

2.总结与收获

这次的程序开发获得的收获还是很多的。我们之前所有的操作主要还是在 visual studio 中进行开发，而这一次我们结合其他软件进行开发，并将画面实时显示。Qt 本身的内容就比较多，我们学习到了很多部件的知识，是一次十分有趣的过程。

总结下来，我们还是完成了任务，长达一两周的开发学习，收获颇丰。本次工作后还是希望能够学习到其他同学的优秀代码，汲取所长。

四、实践总结

【1】霍亚飞 北京航空航天大学出版社 2022 《Qt Creator 快速入门》

【2】徐野，赵星宇，黄海新 北京航空航天大学出版社 2017 《Qt 平台体系与应用》

【3】CSDN 网站

https://blog.csdn.net/m0_65635427/article/details/130780280?ops_request_misc=%257B%2522request%255Fid%2522%253A%252216943472741680018218923 7%2522%252C%252

https://blog.csdn.net/weixin_45589030/article/details/119086044?utm_medium=distribute.pc_relevant.none-task-blog-2~default~baidujs_baidulandingword~default-0-119086044-blog-78947024.235^v38^pc_relevant_sort_base3&spm=1001.2101.3001.4242.1&utm_relevant_index=1