

目 录

第一部分 算法实现设计说明.....	2
1.1 题目	2
1.2 软件功能	2
1.3 设计思想	3
1.4 逻辑结构与物理结构	4
1.5 开发平台	4
1.6 系统的运行结果分析说明	4
1.7 操作说明	8
第二部分 综合应用设计说明.....	10
2.1 题目	10
2.2 软件功能	10
2.3 设计思想	11
2.4 逻辑结构与物理结构	11
2.5 开发平台	11
2.6 系统的运行结果分析说明	11
2.7 操作说明	14
第三部分 实践总结.....	17
3.1. 所做的工作.....	17
3.2. 总结与收获.....	17
第四部分 参考文献.....	18

第一部分 算法实现设计说明

1.1 题目

二叉树，完成：

- (1) 建立一棵二叉树，并对它进行先序、中序、后序遍历；
- (2) 统计树中的叶子结点个数；
- (3) 分别对它进行先序、中序、后序线索化；
- (4) 实现先序、中序线索树的遍历；
- (5) 显示该树和线索化后的树。

1.2 软件功能

设置字体：用户可以通过主界面菜单栏 SetFont 进行字体大小的选择。

通过信号槽机制以及 QFont, QMenu, QAction 三个类来实现。

创建二叉树：用户点击主界面“创建”按钮，按照先序遍历的结果并用”#”表示空节点来创建一个二叉树。

通过 QInputDialog 对话框获取创建二叉树的字符串。

先序遍历：用户点击操作界面“先序遍历”按钮，在操作界面文本框中即可输出先序遍历的结果。

通过 BTree:: PreOrderTraverse();进行遍历。

中序遍历：用户点击操作界面“中序遍历”按钮，在操作界面文本框中即可输出中序遍历的结果。

通过 BTree:: InOrderTraverse ();进行遍历。

后序遍历：用户点击操作界面“后序遍历”按钮，在操作界面文本框中即可输出后序遍历的结果。

通过 BTree:: PostOrderTraverse ();进行遍历。

先序线索化：用户点击操作界面“先序线索化”按钮，系统会对二叉树进行先序线索化，在操作界面的左侧显示先序线索化后的二叉树结构以及在右侧的文本框中输出先序线索化遍历的结果。

通过 `BTree::PreOrderThr ()` ;进行线索化, 再通过 `BtreeWindow::PreOrderThrPaint ()` ;进行图形化显示。

中序线索化：用户点击操作界面“中序线索化”按钮，系统会对二叉树进行中序线索化，在操作界面的左侧显示中序线索化后的二叉树结构以及在右侧的文本框中输出中序线索化遍历的结果。

通过 `BTree::InOrderThr ()` ;进行线索化, 再通过 `BtreeWindow::InOrderThrPaint ()` ;进行图形化显示。

后序线索化：用户点击操作界面“后序线索化”按钮，系统会对二叉树进行后序线索化，在操作界面的左侧显示后序线索化后的二叉树结构以及在右侧的文本框中输出后序线索化遍历的结果。

通过 `BTree::PostOrderThr ()` ;进行线索化, 再通过 `BtreeWindow::PostOrderThrPaint ()` ;进行图形化显示。

统计叶子结点数目：用户点击操作界面“统计叶子结点数目”按钮，系统会在操作界面文本框中输出当前二叉树中叶子结点的个数。

通过 `BTree::CountLeafNode ()` ;计算叶子结点个数。

1.3 设计思想

本次所设计的程序允许用户通过自行输入字符串来创建一棵二叉树, 在所建立的二叉树上进行先序、中序、后序遍历; 先序、中序、后序线索化以及遍历先序、中序、后序线索树; 输出二叉树当前叶子结点个数等功能、

字体大小设置是通过菜单栏中不同的动作发出不同信号, 由不同的槽来接收信号从而设置字体。

创建二叉树是通过对话框来实现，对话框返回一个字符串，把这个字符串传入到 BTree 类的对象，进行先序建树。

先序、中序、后序遍历二叉树则是通过按钮和信号槽机制来实现，信号槽收到信号之后开始调用 BTree 的先序、中序、后序遍历函数，得到对应的返回值，再把对应的返回字符串通过文本框输出到操作界面上。

先序、中序、后序线索化通过按钮和信号槽机制来实现，信号槽收到信号之后开始调用 BTree 的先序、中序、后序线索化，生成对应的线索树，紧接着调用先序、中序、后序线索树遍历函数，返回一个字符串，再把对应的字符串通过文本框输出到操作界面上。

统计二叉树中叶子结点个数也是同样的模式，调用 BTree 的统计二叉树叶子结点函数，得到函数的返回值之后通过文本框输出到操作界面上。

程序同时还实现了二叉树以及线索树的可视化。每个结点在建立的时候就会根据该结点的父结点以及左孩子还是右孩子的属性生成一个在操作界面上对应的二维坐标。通过 QPainter 类，利用类似先序遍历的思想进行递归绘制图像，同时根据父子关系绘制箭头。

带箭头的线段是自定义的函数，通过平面几何知识，根据线段的起始两端点在线段的起点处绘制一个实心三角形。

1.4 逻辑结构与物理结构

逻辑结构：树形结构(用户建立的二叉树)

物理结构：链式存储结构(二叉链表、三叉链表)

1.5 开发平台

采用开发平台为 Qt Creator 4.7.0(Qt 版本为 5.11.1)，软件运行环境为 Windows10 64bit。

1.6 系统的运行结果分析说明

主要开发过程：

- (1) 完成界面各层级的设计以及菜单栏各种功能的实现。
- (2) 完成实现类（二叉树）的设计与实现。包括先序、中序、后序遍历；先序、中

序、后序线索化；统计叶子结点个数。

(3) 设计显示图形，并将图形与二叉树进行映射。

(4) 进行整体调试与容错性检测。

软件成果：

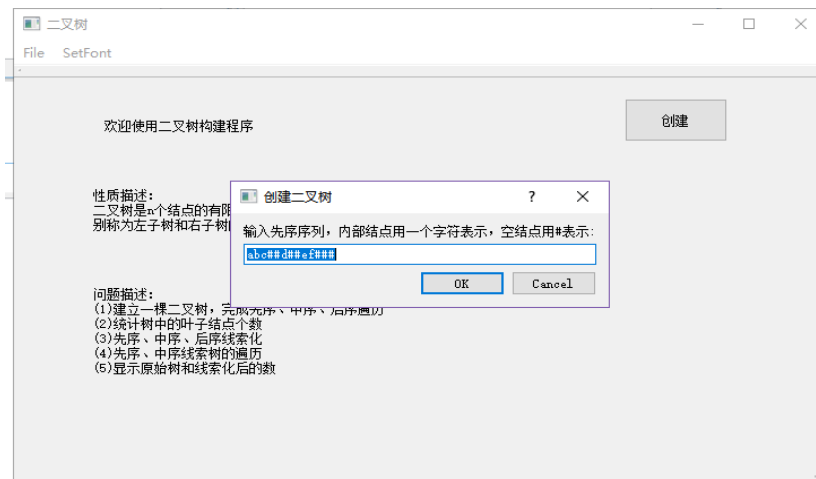
(1) 正确性：经过大量测试，涵盖了四层内所有的二叉树包括多种极端情况，软件各功能的正确性得到了保证。

(2) 稳定性：程序运行稳定，经过认真修改调试之后再未发生过崩溃的情况。

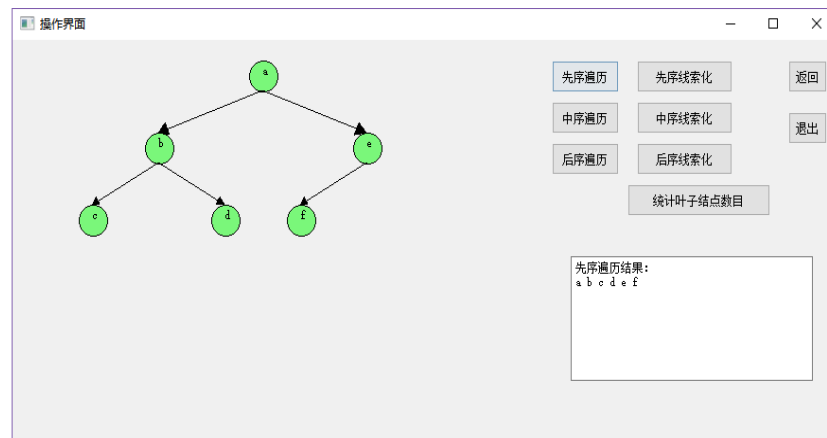
(3) 容错能力：在创建二叉树时，会自行检测输入的字符串是否符合创建要求，二叉树深度是否满足不超过 4 层。如果超过 4 层则会提示错误并取消本次创建活动。

运行结果：

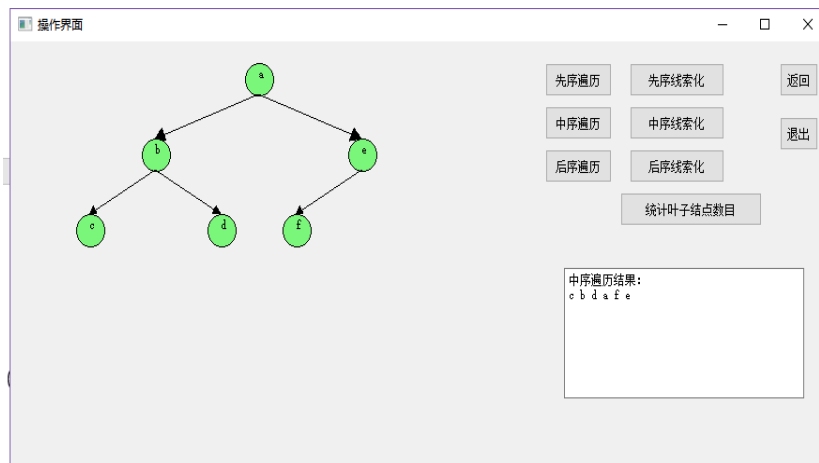
(1) 二叉树创建：



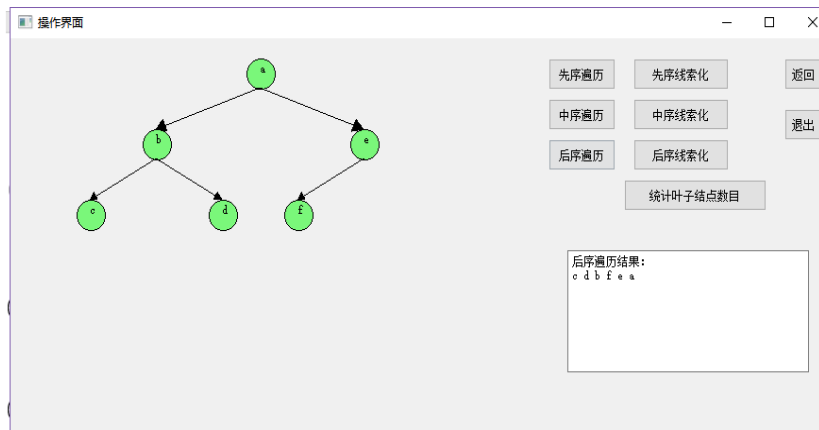
(2) 先序遍历



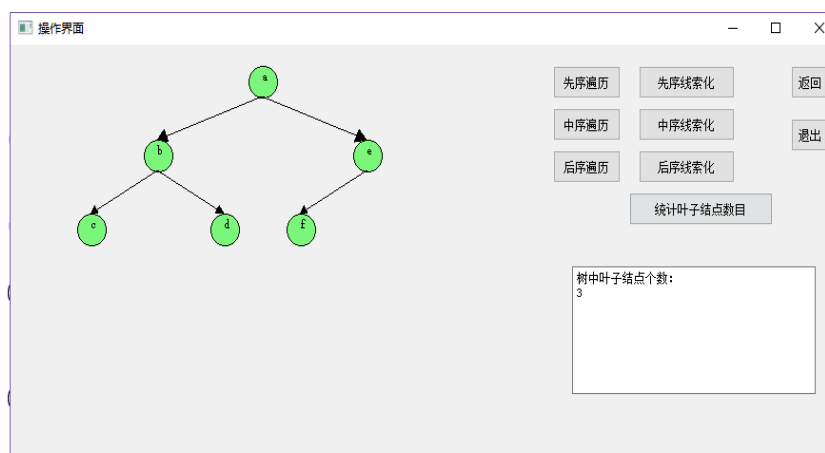
(3)中序遍历



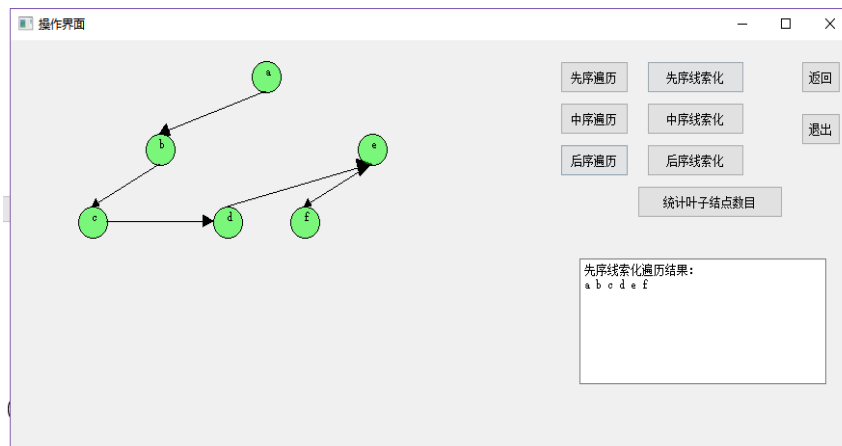
(4)后序遍历



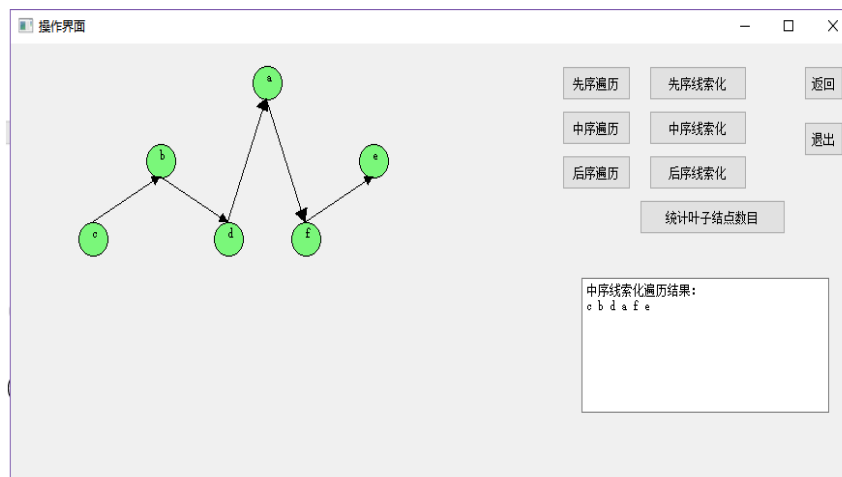
(5)统计叶子结点数目



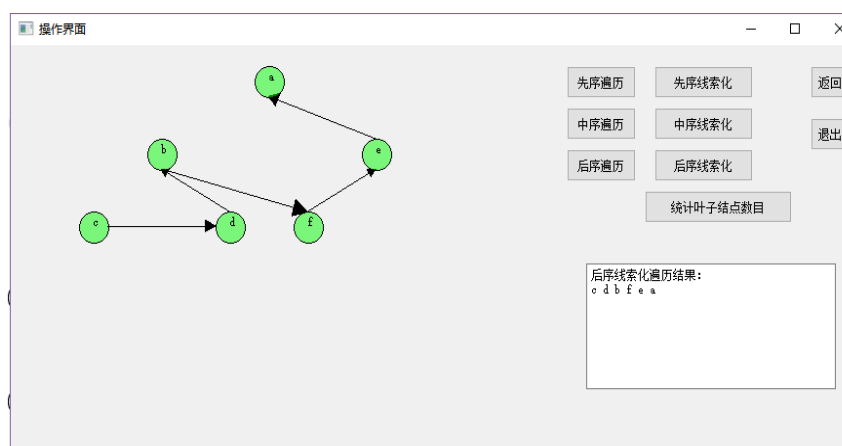
(6)先序线索化



(7)中序线索化



(8)后序线索化

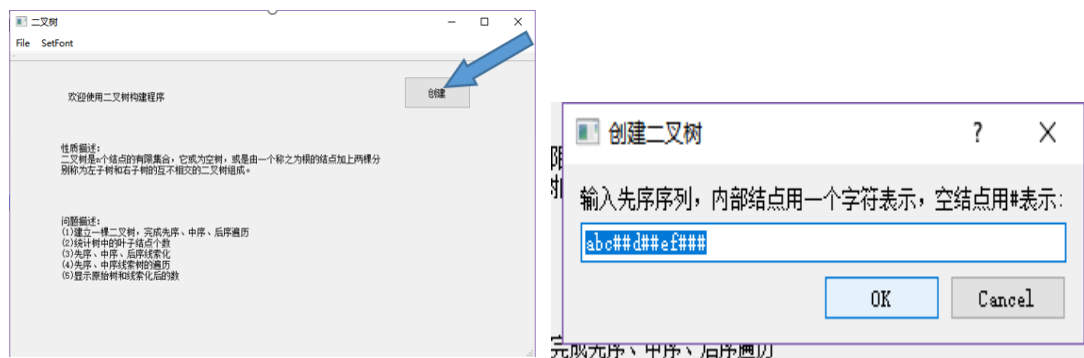


1.7 操作说明

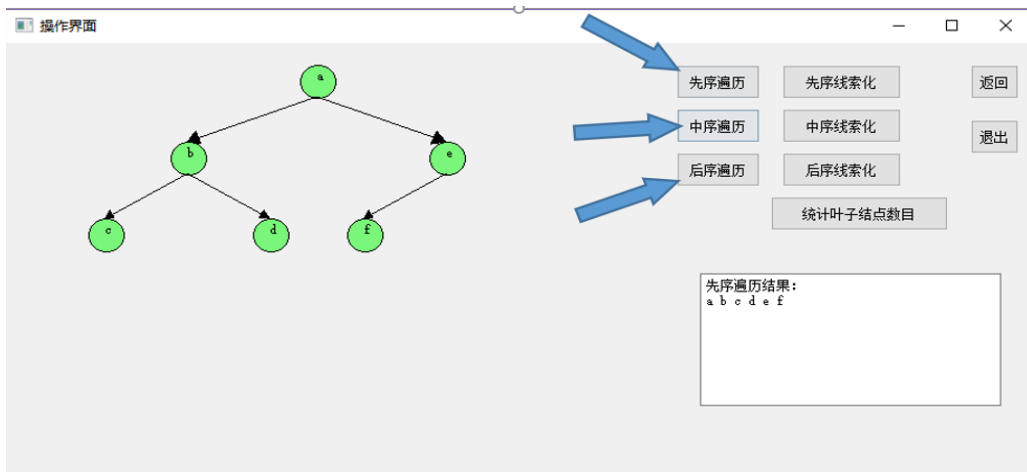
字体设置： 点击“SetFont”菜单栏里, 在菜单选择一种字体大小即可。



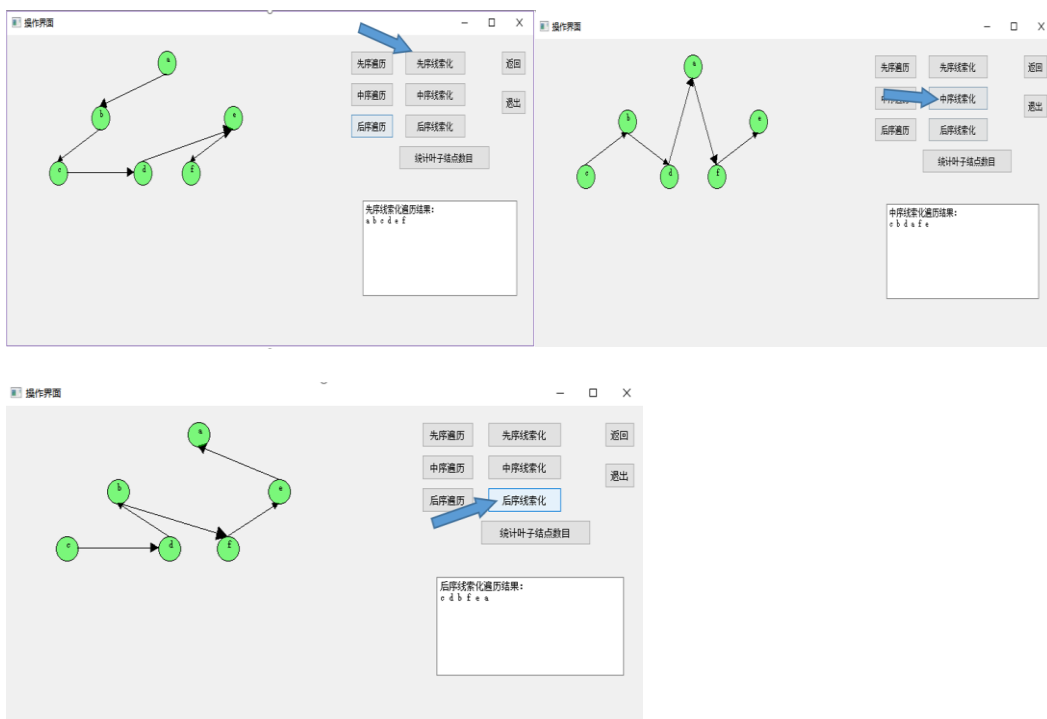
创建二叉树： 点击创建，输入先序序列即可点击“OK”。



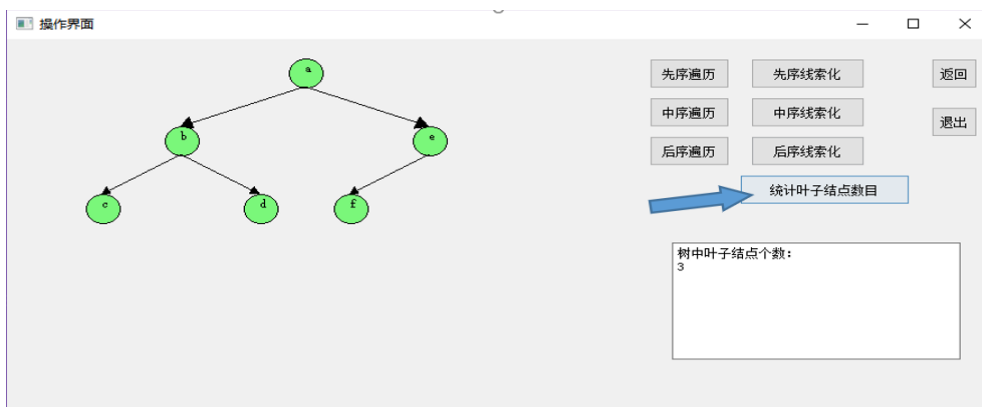
先序、中序、后序遍历： 点击对应按钮，即可在右下侧文本框中得到结果。



先序、中序、后序线索化：点击对应按钮，即可在右下侧文本框中得到结果。同时左侧显示对应的线索树。



统计叶子结点数目：点击对应按钮，即可在右下侧文本框中得到结果。



第二部分 综合应用设计说明

2.1 题目

在某社会关系网络系统中，一个人属性包括所在地区、就读的各级学校、工作单位等，没一个人有众多好友，并可以根据个人兴趣及社会活动加入到某些群组。现在需要设计一算法，从该社会关系网络中某一个人出发，寻找其可能认识的人。例如根据两个人共同好友数量及所在群组情况，来发现可能认识的人；通过就读学校情况发现可能认识的同学。

- (1) 通过图形化界面，显示某一个人的社会关系网络。
- (2) 寻找某一可能认识的人（不是其好友），并查看这些人与其关联度（共同好友数）。
- (3) 根据可能认识的关联度对这些人进行排序。

2.2 软件功能

图信息输出：用户可以通过点击演示界面上的“图信息输出”按钮，在右下角的文本框中即可得到当前无向图的结点姓名、邻接矩阵、邻接表信息。

通过信号槽机制以及 `MGraph::DisplayVextex();MGraph::isplayAdjMatrix();`
`MGraph::DisplayAdjArc();`来实现。

、

添加结点：用户可以通过点击演示界面上的“添加结点”按钮，即可弹出对话框进行结点的添加。

通过信号槽机制、`MGraph::addNode();`、自定义对话框来实现

详细信息输出：用户可以通过点击演示界面上的“详细信息”按钮，即可弹出对话框选择结点，显示指定结点的详细信息以及该结点与其他结点之间的关联度。

通过信号槽机制、`MGraph::getVexs();MGraph::getAssociateExtent();`来实现

图形化界面：系统通过遍历图中结点，根据结点的个数 n 在界面上绘制正 n 边形。再通过每个结点之间时候认识，绘制结点之间的连线。

通过 `QPainter` 和 `MGraph` 来实现的。

2.3 设计思想

社会关系网络可以用无向图来表示,对应的存储结构采用邻接矩阵和邻接表方式。理论上计算两个结点之间的关联度只需要邻接矩阵即可完成,但是为了能够更加深入地理解无向图的结构,把邻接表一种加入图类中。

在图形交换界面,允许用户通过添加结点的方式自行输入结点信息以及结点之间的关系来创建一个新的结点。同时,用户还可以通过下拉式对话框选择查看某个结点的详细信息。

寻找好友即寻找图中所有与当前结点不存在连接的结点。其关联度计算方式是遍历可能认识的人中的所有好友,通过邻接矩阵来判断可能认识的人的好友是否和当前结点为好友关系。如果是,则关联度加1;否则关联度就不变。

关联度排序则是通过冒泡排序法的方式,使得关联度最高的结点对应数组中下标最小的位置。但是由于数组中只是存放每个结点的关联度,因此需要而外增设一个数组来记录结点的序号,两个数组的变动应该保持一致即可。

2.4 逻辑结构与物理结构

逻辑结构: 无向图状结构(用户建立的图)

物理结构: 顺序存储结构(邻接矩阵)、链式存储结构(邻接链表)

2.5 开发平台

采用开发平台为 Qt Creator 4.7.0(Qt 版本为 5.11.1),软件运行环境为 Windows10 64bit。

2.6 系统的运行结果分析说明

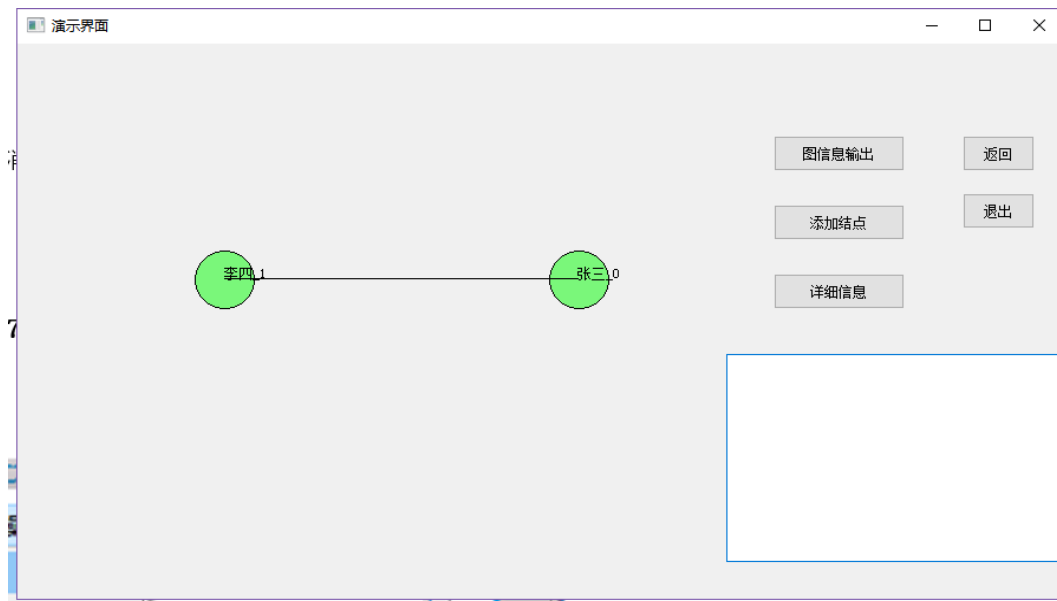
主要开发过程:

- (1) 完成界面各层级的设计以及菜单栏各种功能的实现。
- (2) 完成实现类(无向图)的设计与实现。包括结点插入、结点查询、结点关联度查询、图的邻接矩阵输出、图的邻接表输出等功能。
- (3) 设计显示图形,并将图形与无向图进行映射。
- (4) 进行整体调试与容错性检测。

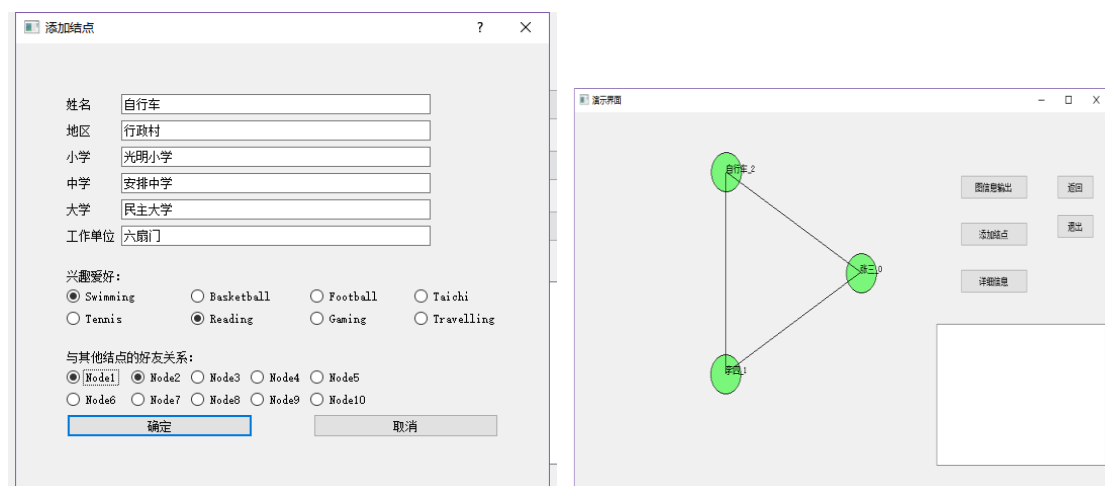
软件成果：

- (1) 正确性：经过大量测试，涵盖了 10 个结点之内的所有无向图，各种极端情况均被包含在内，软件各功能的正确性得到了保证。
- (2) 稳定性：程序运行稳定，经过认真修改调试之后再未发生过崩溃的情况。
- (3) 容错能力：能够自行检测创建结点时“空输入”的问题以及结点数目超过 10 个则自动取消新结点建立。

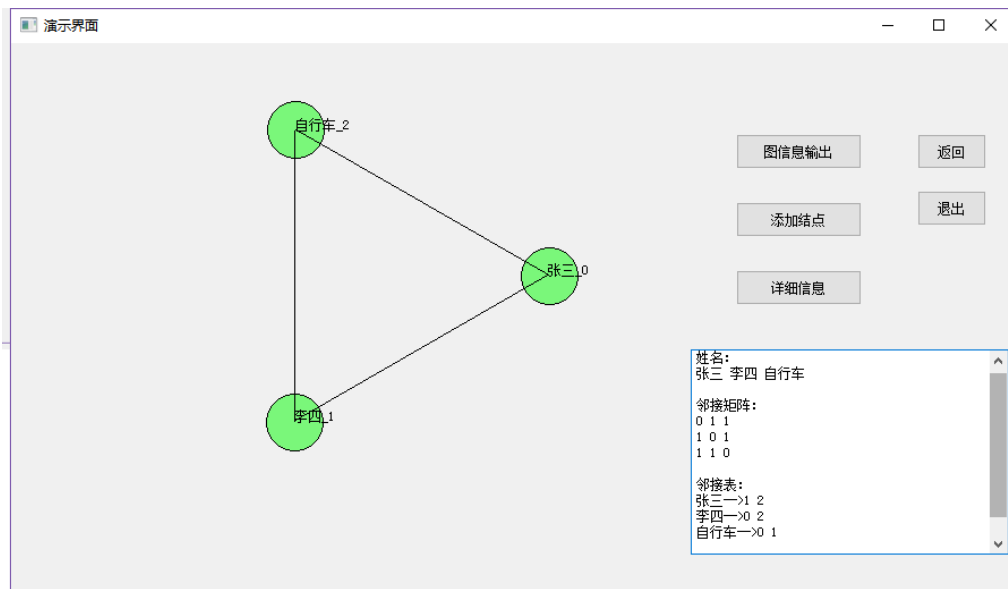
演示界面：



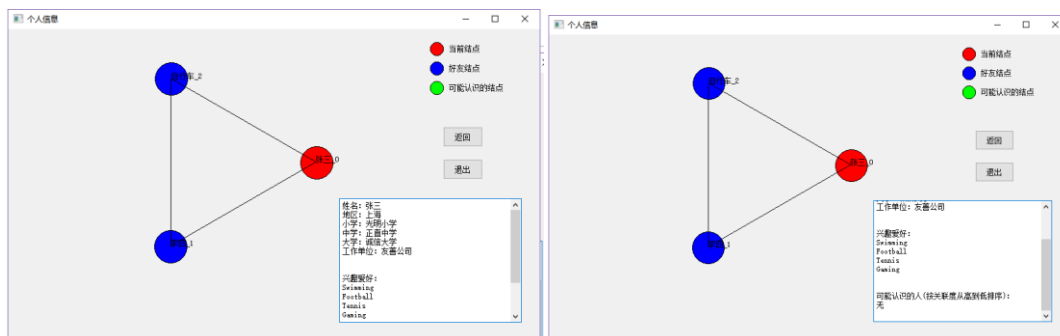
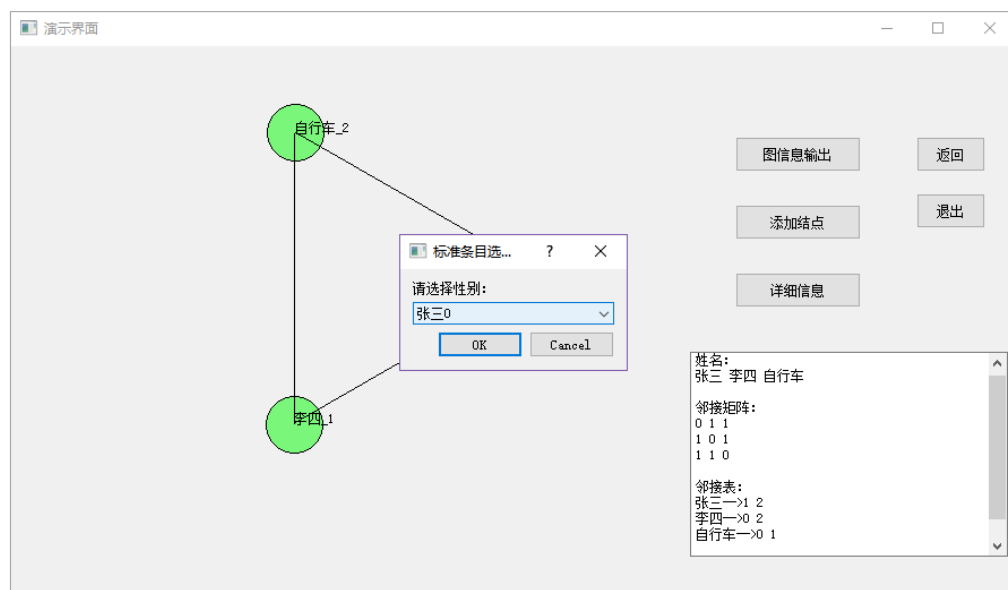
添加结点：



图信息输出：



详细信息：

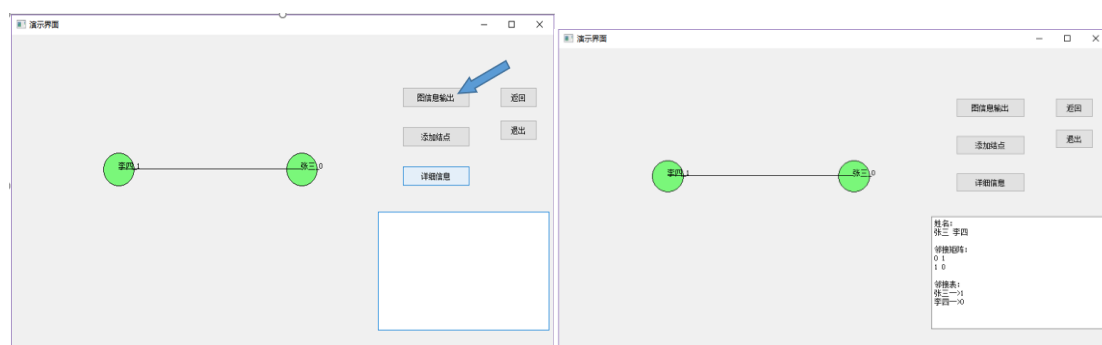


2.7 操作说明

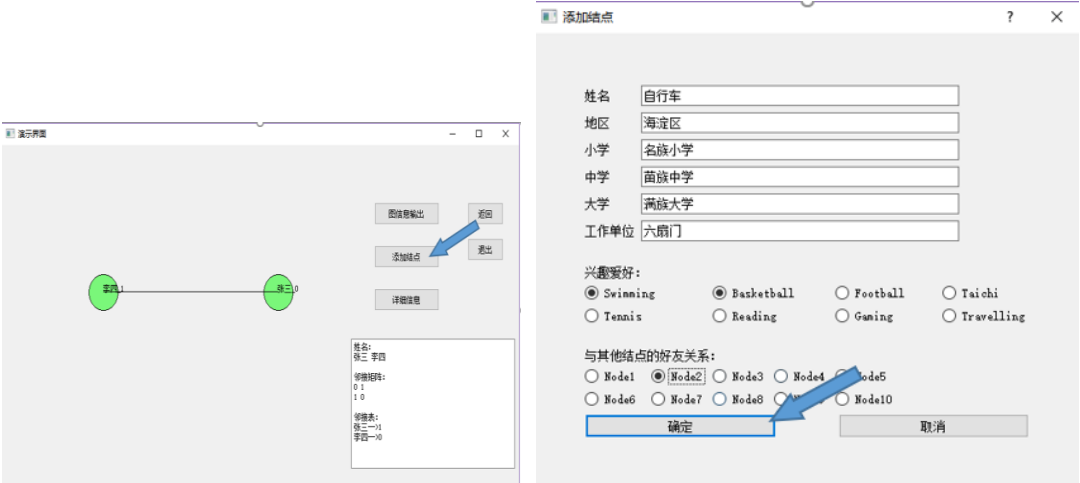
设置字体： 点击“SetFont”菜单栏，从中选择一种字体大小即可。



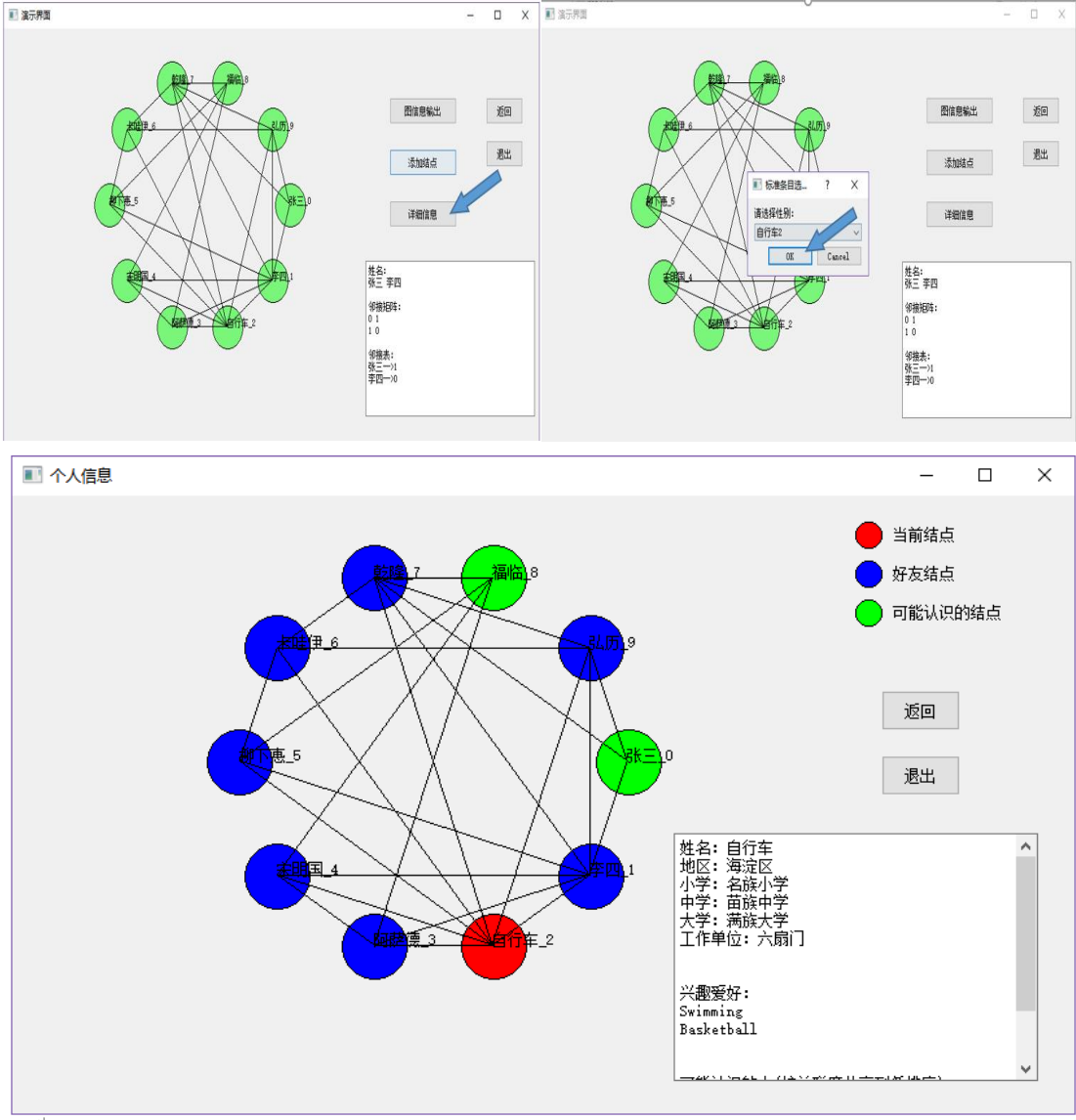
图信息输出： 鼠标点击“图信息输出”按钮即可。

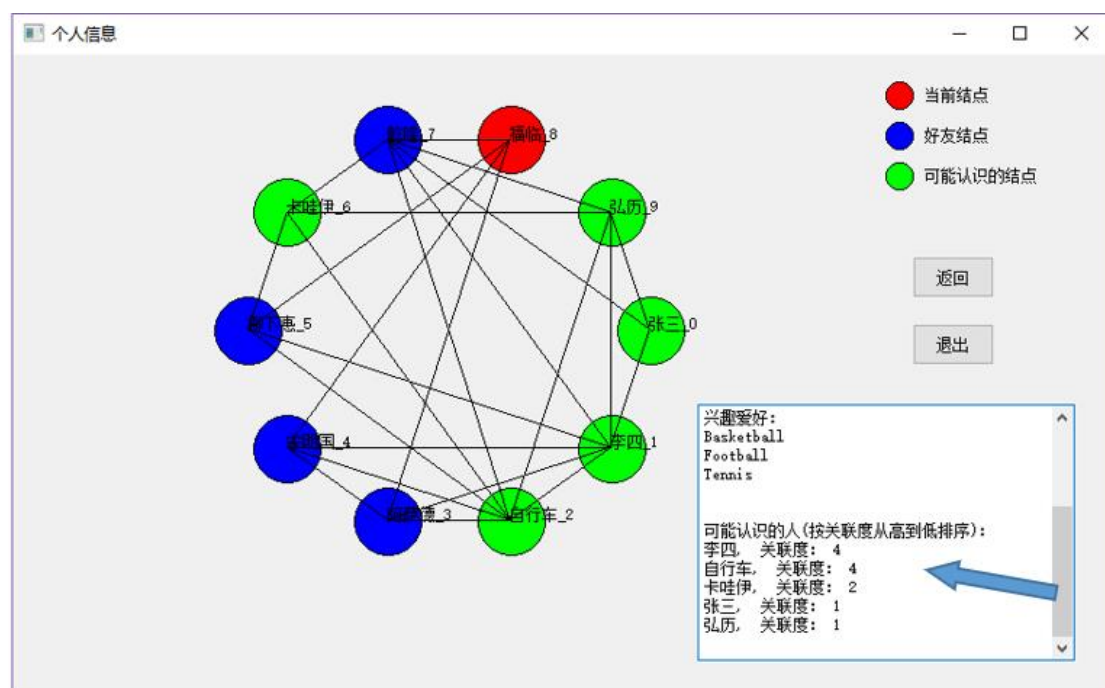
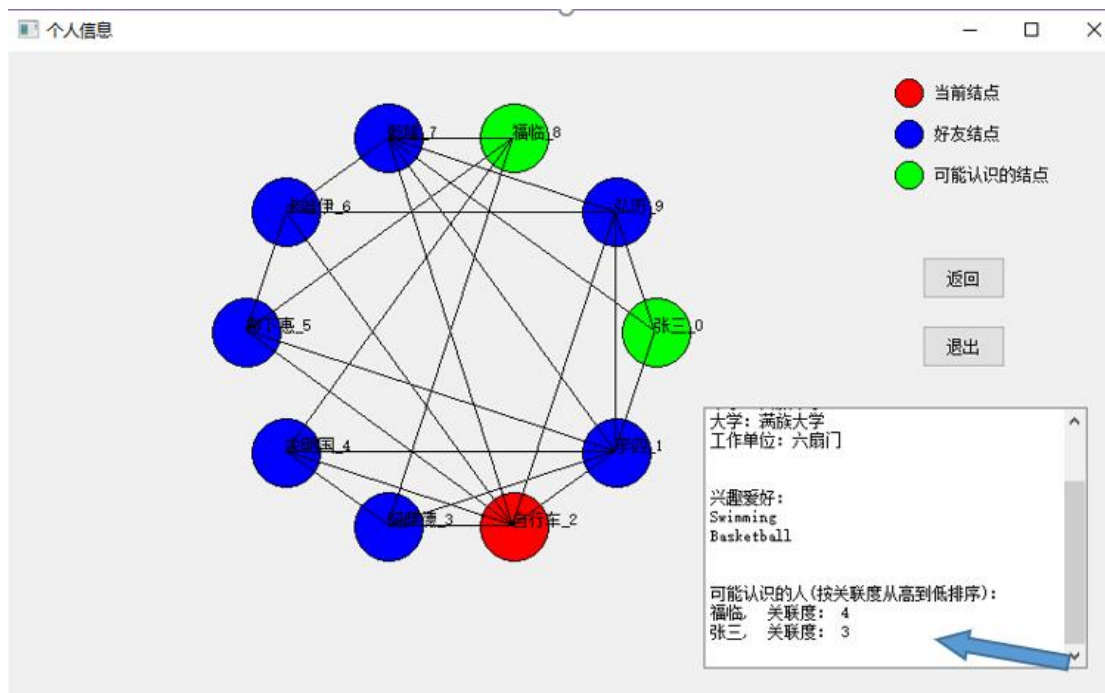


添加结点：鼠标点击“添加结点”按钮即可。



详细信息：鼠标点击“详细信息”按钮即可。





第三部分 实践总结

3.1. 所做的工作

完成的工作：

算法实现题和综合应用题的题目要求基本上都完成了。其中算法实现题完成了二叉树的建立；先序、中序、后序遍历；统计树中的叶子结点个数；先序、中序、后序线索化；先序、中序、后序线索树遍历；图形化二叉树以及线索树等功能。

综合应用题完成了图形化显示一个人的社会关系网络；寻找某一个人可能认识的人并查看这些人的关联度；根据关联度对这些人进行排序等功能。

两道题所完成的功能已基本满足题目要求。

可改进提升的地方：

在综合应用题中结点的选择是通过下拉式对话框来实现的，还不够界面友好，可以通过相应鼠标点击来完成从而达到更好的体验感。但考虑到时间紧张等问题所以尚未完成。

3.2. 总结与收获

本次数据结构课程设计作业前前后后用了半个月的时间来完成，查阅大量资料，积累了一笔关于 GUI 编程的宝贵财富。由于数据结构课程是在大二上学期学习的，因此在做课程设计作业的时候看，关于二叉树和无向图的一些基本概念都变得非常模糊，因此重新把之前的课件和作业拿起来又浏览了一遍，重新理解了一遍之前关于二叉树和无向图的概念，收获颇丰。

GUI 编程分为两个部分，一个部分是实现类的构建，另一类则是交互界面的设计。实现类主要写的是关于二叉树的一些操作，比如建立一棵二叉树；二叉树的先序、中序、后序遍历；二叉树的先序、中序、后序线索化；线索树的先序、中序、后序遍历等等。还有一块是关于无向图的操作，例如建立一张无向图，构造相应的邻接表和邻接矩阵；分析各个结点之间是否直接相连；各个结点之间的关联度等等。

交互界面主要也分两块。一则是如何绘制二叉树，二则是如何绘制无向图。为了绘制二叉树，我在二叉树每个结点生成的同时，就把该结点的圆心坐标对应于操作界面中的位置一同记录在结点信息中，这个信息是根据父亲结点的位置信息以及当前结点是父亲结点的做结点还是右结点来确定的。然后利用类似先序遍历的方法进行递归绘制二叉树。至于无向图，

我是把整张图绘制成一个正 N 边形的结构，其中 N 是结点的个数。通过 QPainter 中的坐标变化改变界面中心，然后通过三角函数和旋转变换矩阵计算出下一个结点对应的圆心位置即可，判断每个结点之间的关系然后进行关系连线。

值得一提的是在无向图添加结点的操作中，为了能够符合添加要求，于是自定义一个对话框，使用了两种布局，在布局中采用了标签、文本编辑、按钮、按钮组等类。最终完成一个定制版的添加结点对话框界面。

在整个设计与实现的过程中，时时刻刻都存在着各种各样的问题。遇见二叉树与无向图的设计问题，就查阅之前的课件与课本，发现有了一个新的理解角度，重新设计一种高效的解决方案。遇见 GUI 交互问题，就面向 GOOGLE 编程，寻找相关资料，查看对应的代码，最后再自行实现。

整个数据结构课程设计完成下来，学会了很多关于工程化的知识，比起之前单纯地在 CONSOLE 控制台下的简单编程上又大大进步了。将 CONSOLE 控制台编程的内容融入到 GUI 编程当中，从而做出一个完整的运用程序。在算法方面，以前写过的算法，通过这次课程设计进行了一次巩固加深，没写过的算法则自己认真写了一遍。可以说，通过这次课程设计，对于数据结构中的树形结构和无向图这两个方面的内容，我的理解变得十分深刻。

第四部分 参考文献

- [1] 严蔚敏，吴伟民. 数据结构 (C 语言版). 北京：清华大学出版社，2007
- [2] 陆文周. Qt 5 开发及实例 (第 3 版). 北京：电子工业出版社，2017
- [3] ynlzqyj. 学习笔记之 Qt 自定义对话框. <https://blog.csdn.net/youarenotme/article/details/53452962>
- [4] 一去、二里. Qt 之格栅布局 (QGridLayout. <https://blog.csdn.net/liang19890820/article/details/51538275>
- [5] Rocky_Sky. Qt 之 2D 绘图 4——坐标系统与坐标变换. <https://my.oschina.net/jintao/blog/80660>
- [6] 古月河图. Qt 学习笔记：画一条带箭头的线. <http://blog.csdn.net/founderznd/article/details/51661523>