

《数据结构》课程设计总结

学号： 1851785

姓名： 唐骏龙

专业： 计算机科学与技术

2021 年 8 月

目 录

第一部分 算法实现设计说明.....	1
1.1 题目.....	1
1.2 软件功能.....	1
1.3 设计思想.....	1
1.4 逻辑结构与物理结构.....	2
1.5 开发平台.....	2
1.6 系统的运行结果分析说明.....	3
1.7 操作说明.....	4
第二部分 综合应用设计说明.....	12
2.1 题目.....	12
2.2 软件功能.....	12
2.3 设计思想.....	12
2.4 逻辑结构与物理结构.....	15
2.5 开发平台.....	16
2.6 系统的运行结果分析说明.....	16
2.7 操作说明.....	18
第三部分 实践总结.....	21
3.1.所做的工作.....	21
3.2.总结与收获.....	22
第四部分 参考文献.....	22

第一部分 算法实现设计说明

1.1 题目

5. 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（此要求可视情况选择是否完成）

1.2 软件功能

- 根据所输入的序列按层次遍历顺序建立一棵二叉树。
- 通过点击二叉树上的结点进行添加子结点以及删除结点操作。
- 对建立好的二叉树进行先序、中序、后序遍历，并生成遍历序列。
- 统计树中的叶子结点个数并显示。
- 对建立好的二叉树进行先序、中序、后序线索化。
- 对三种线索二叉树进行遍历，以表格形式展示遍历序列以及各结点具体信息(包括结点值、左右孩子值、左右标志域等)。

实现方式：Vue+Element-UI

1.3 设计思想

1.技术学习与准备工作

本学期在其他课程中我有用到 Vue 技术进行网站的前端开发，因此此次课程设计我打算使用 Vue 框架结合 Element-UI 组件库进行可视化方面的设计,但由于 JavaScript 语法之前掌握得并不够多，因此在完成此次课程设计之前，我先进行了 JavaScript 相关知识的学习，为实现二叉树的构造以及其一系列基本操作而做准备。

2.设计框架

在准备工作做的差不多时，我便开始对这一题目进行框架设计，这道题是对二叉树的基本概念和操作的考察，具体的算法流程等在数据结构课程里都已经进行过系统的学习，对实现方式也有了一定的了解，因此这道题主要的难点在于如何将这些结果比较直观地呈现出来。

我经过一些思考以及资料查询后，认为可以先让用户根据自己输入的序列来创建一棵二叉树，如果可以的话，还能直接编辑现有的二叉树，并展示二叉树的图像。同时，用户可以通过点击相应按钮来实现二叉树的遍历以及线索化等操作，网站会显示相应操作后的结果，这样就非常直观。

3.实现代码

实现框架设计好后，就可以上手完成代码，此次实现的难点在于利用 JavaScript 中的 canvas 来进行二叉树的绘制，其具体绘制方法我也是先进行了相应的学习。而对于二叉树的一些基本操作只要在相应函数中写好相应的算法，然后将对应结果传到页面进行展示即可。

1.4 逻辑结构与物理结构

逻辑结构：

整道题都是有关二叉树的建立及其基本操作，因此逻辑结构自然是树状结构。

物理结构：

由于是树状结构，因此物理结构采用的是链式存储结构，结点之间以左右孩子的方式进行连接，而对于线索二叉树，还需要设有两个标志域，在未线索化时先初始化为 LINK，具体如下图所示。

```
let bt = function (ele) {  
  this.data = ele;  
  this.lChild = null;  
  this.rChild = null;  
};
```

```
let bt = function (ele) {  
  this.data = ele;  
  this.lChild = null;  
  this.rChild = null;  
  this.leftTag = this.rightTag = LINK;  
};
```

1.5 开发平台

开发平台： Vue.js

Vue 是一套用于构建用户界面的渐进式 JavaScript 框架。与其它大型框架不同的是，Vue 被设计为可以自底向上逐层应用。Vue 的核心库只关注视图层，不仅易于上手，还便于与第三方库或既有项目整合。另一方面，当与现代化的工具链以及各种支持类库结合使用时，Vue 也完全能够为复杂的单页应用（SPA）提供驱动。

第三方库：Element-UI

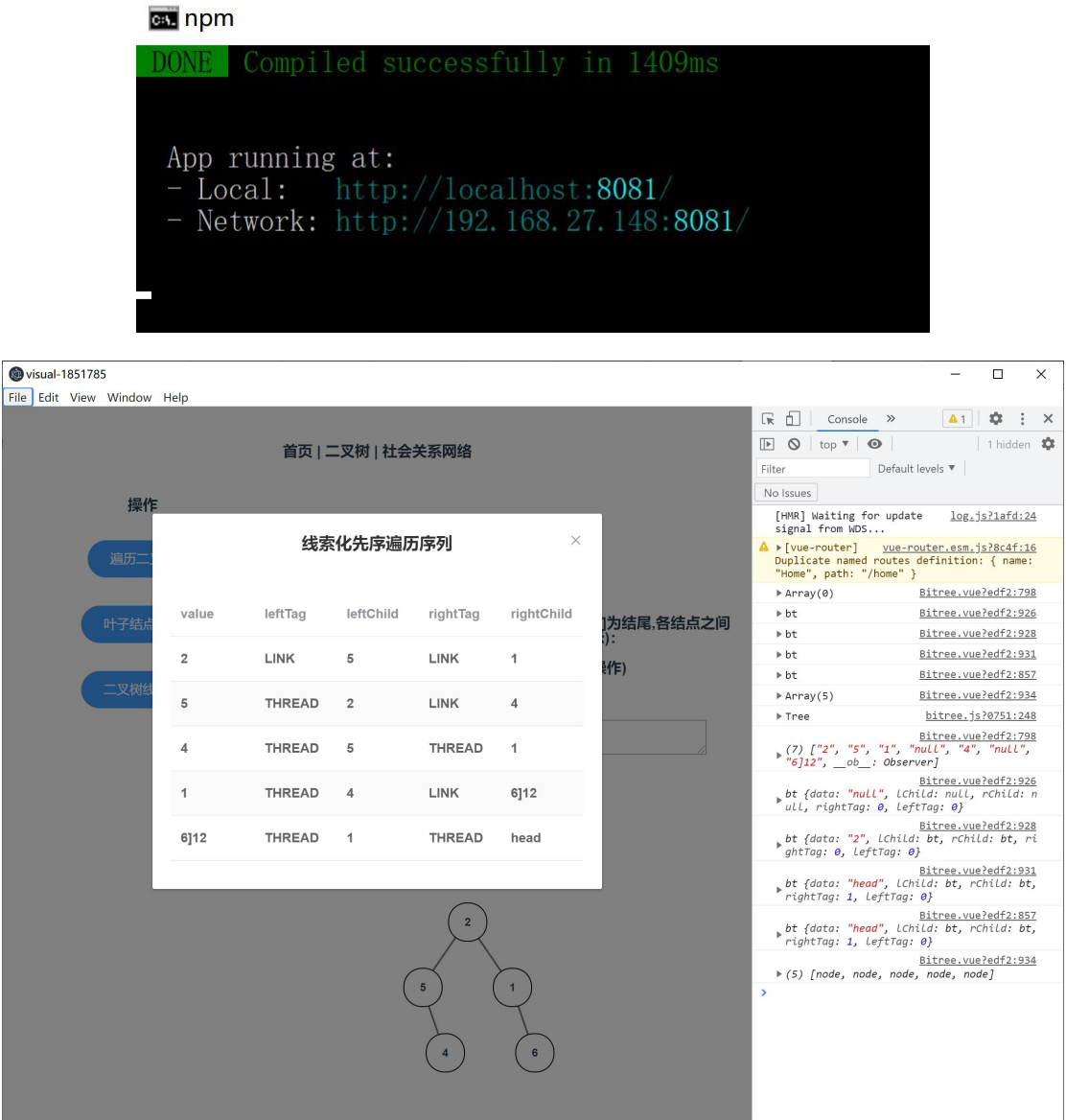
Element-UI 是一套为开发者、设计师和产品经理准备的基于 Vue 2.0 的桌面端组件库，是一款方便快捷，容易上手的网速快速成型工具，其 UI 美观大方，简约干净，受到许多 Vue 开发者的喜爱。

软件的运行环境：Electron/任意浏览器

1.6 系统的运行结果分析说明

调试/开发过程:

先在项目文件夹的命令行中输入”npm run serve”语句来运行项目，通过 VSCode 进行代码开发，而调试主要是通过函数中调用 console.log()语句，并在网页开发者工具的 console 部分查看相应的数据，检查结果是否正确。由于开发过程中项目一直处于运行状态，因此修改代码并保存文件后，页面会自动进行刷新，如图所示。

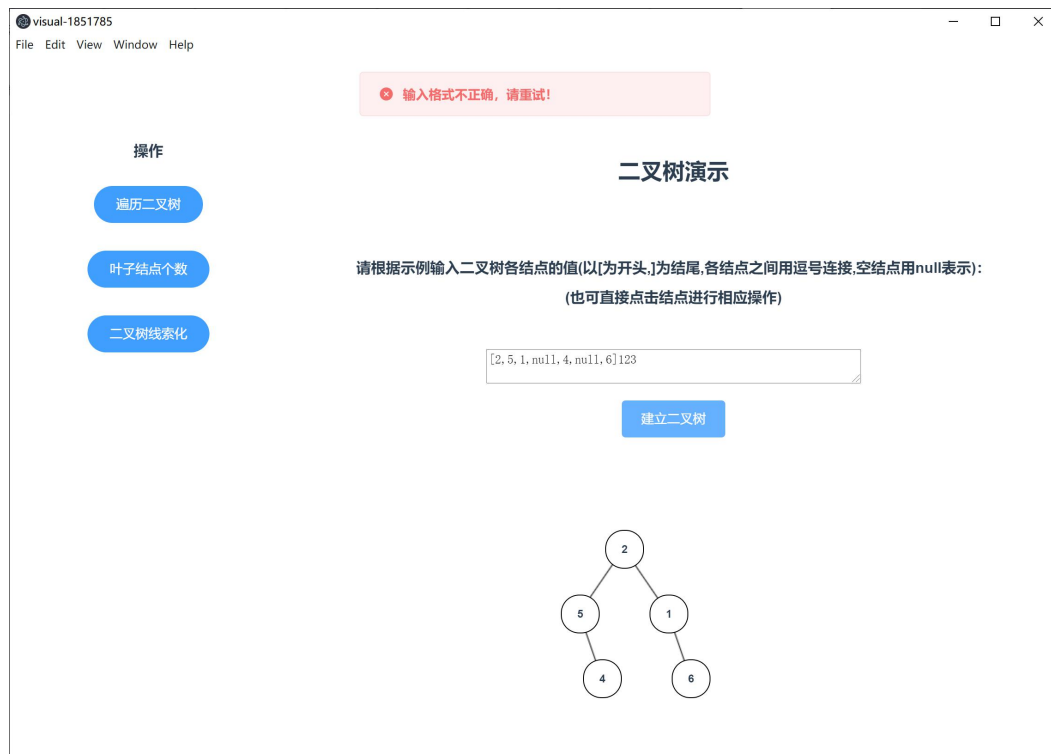


正确性/稳定性:

在开发过程中，我已经建立过几种不同的二叉树，经过验证，其显示的三种遍历序列，叶子结点个数，以及遍历线索化二叉树的序列以及结点信息均是正确的，说明算法的实现函数部分没有错误，具体案例可以见操作说明。

容错能力:

对于所输入的二叉树序列, 会特别检查其输入格式是否满足所给要求, 即“以[为开头,]为结尾,各结点之间用逗号连接,空结点用 null 表示”, 若不满足, 则点击“建立二叉树”按钮时,会出现提示“输入格式不正确, 请重试!”, 且无法生成新的二叉树, 如图所示。



1.7 操作说明

项目启动方法:

方法一:

在源文件所在目录下的命令行中输入“npm run electron:serve”（可能需要提前下载 node.js 和 electron），之后会自动弹出 electron 窗口供用户进行操作。

方法二:

在源文件所在目录下的命令行中输入“npm run serve”（可能需要提前下载 node.js），之后在浏览器内输入命令行中显示的网址（一般为 localhost:8080）也可启动项目。

```
\visual-1851785>npm run electron:serve_
```

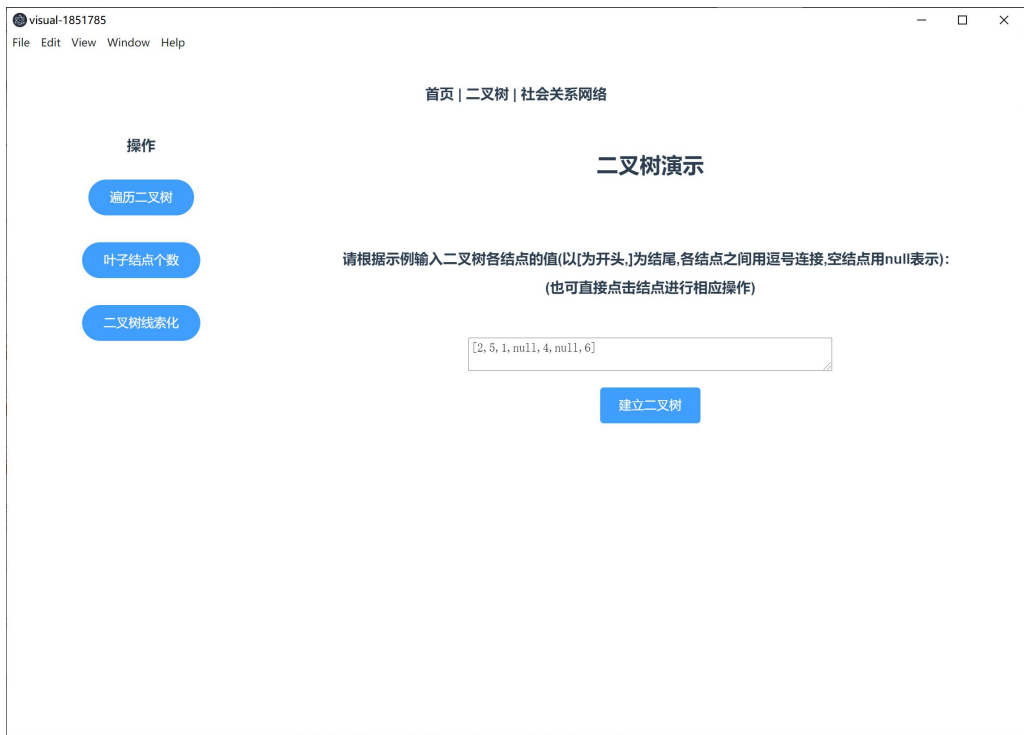
```
sual-1851785>npm run serve_
```

首页界面：



说明：首页为简单的个人信息以及两道题目的链接，可直接点击进入相应的页面。

二叉树主界面：



说明：界面分为两个部分，分别是侧边栏和主体部分，侧边栏中包括一些操作需要用到的按钮，包括遍历二叉树，统计叶子结点个数和二叉树线索化。主体部分为二叉树的创建和显示部分，针对序列的输入格式进行了详细说明，文本框中已经提前初始化好一个二叉树序列，

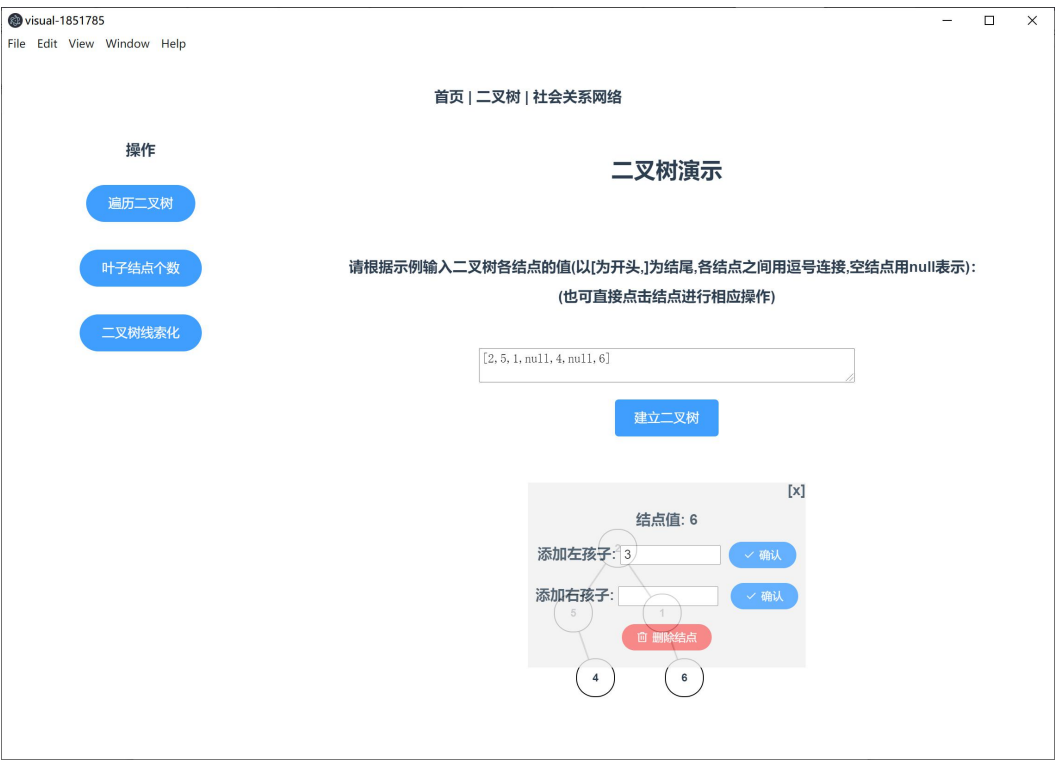
用户可以直接点击“建立二叉树”进行建立。

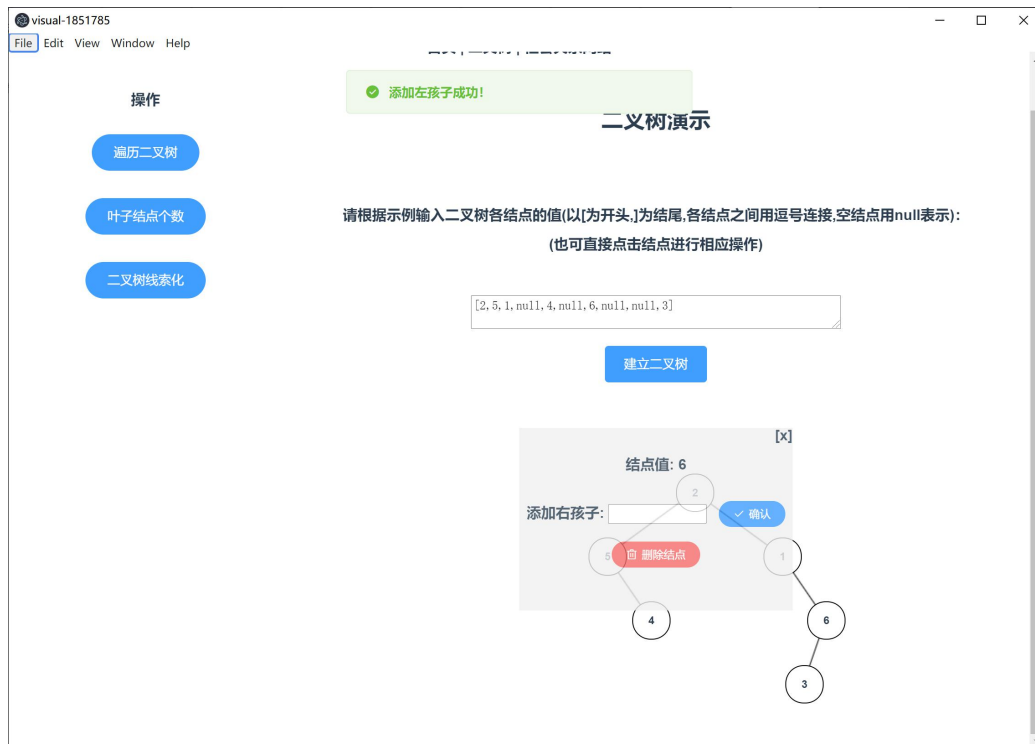
建立二叉树：



说明：若输入格式正确，点击“建立二叉树”按钮后，会绘制出二叉树的图像，并出现相应提示。

添加新结点：

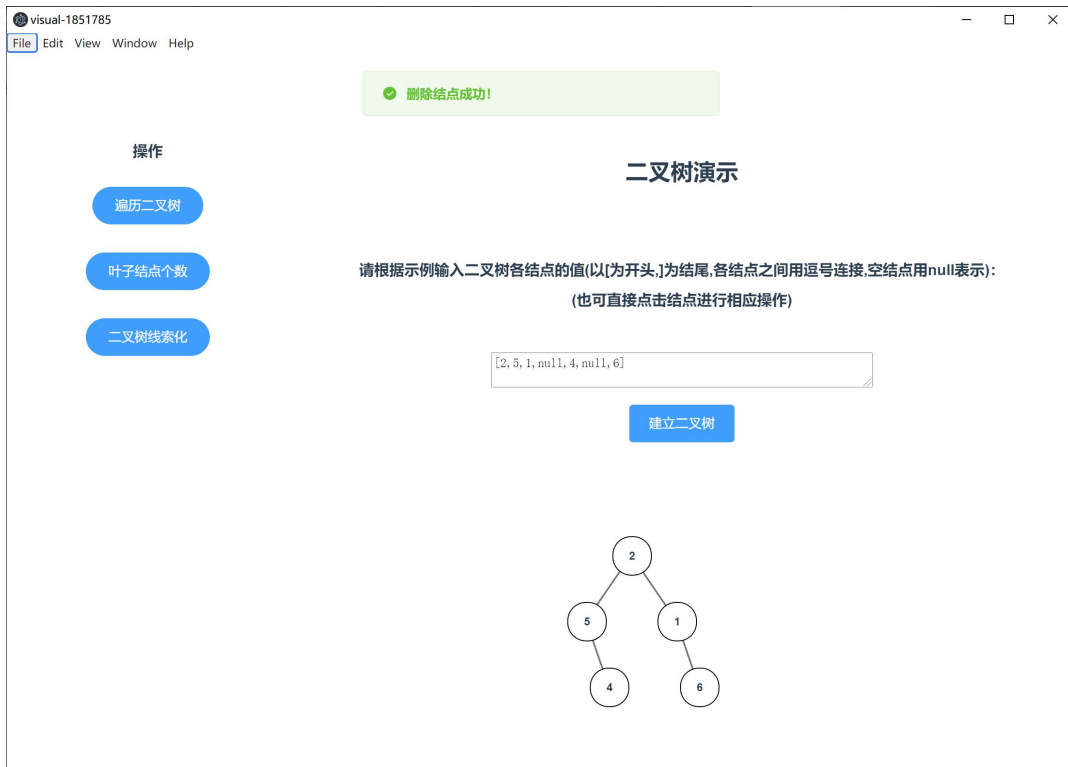




说明：可以直接点击二叉树中某个结点来对二叉树进行编辑，这里点击结点 6 后添加一个左孩子结点 3，添加好后二叉树图像以及输入框中二叉树序列均会同步变化，同时弹出相应提示。

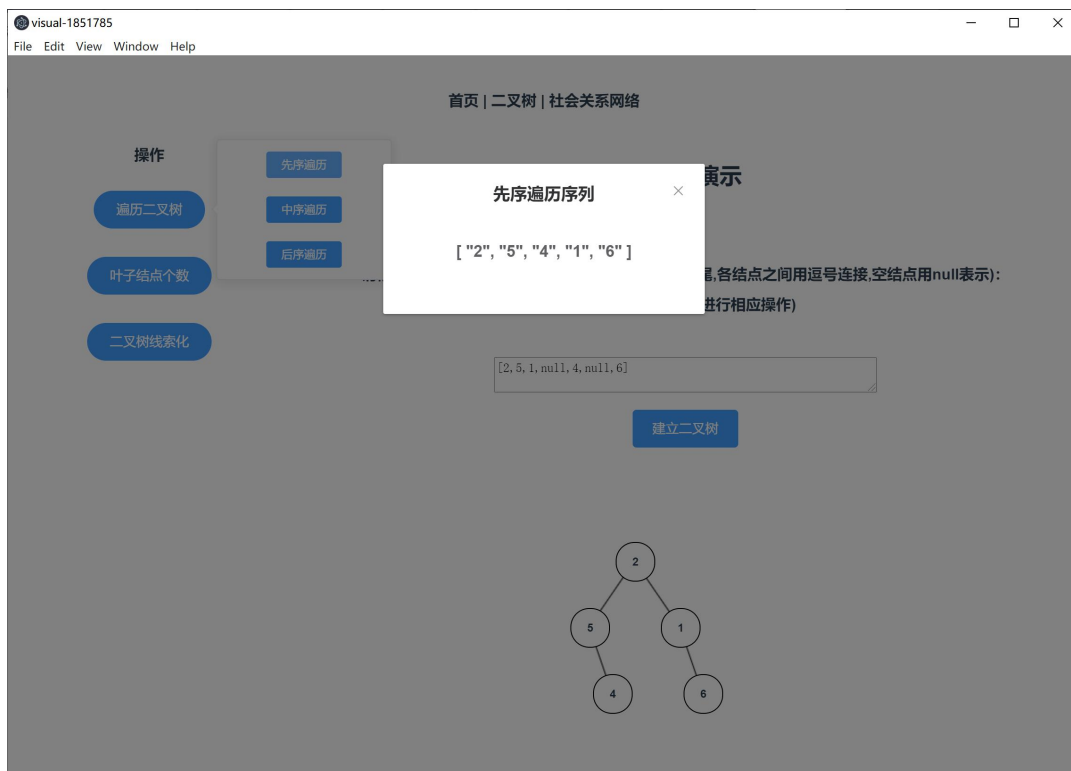
删除结点：





说明：也可以直接点击某结点将其进行删除，这里点击结点 3 并将其删除，操作成功，二叉树图像以及输入框中二叉树序列均会同步变化，并有相应提示。

先序遍历：



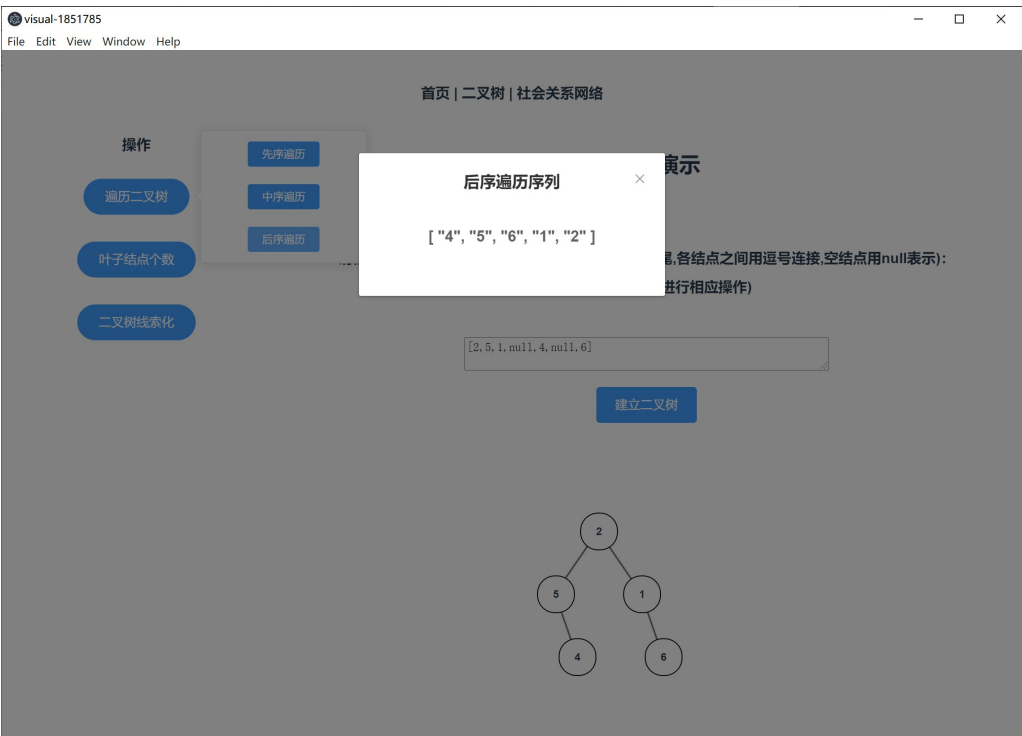
说明：点击侧边栏中“遍历二叉树”中的“先序遍历”按钮，会弹出显示先序遍历序列的消息框。

中序遍历：



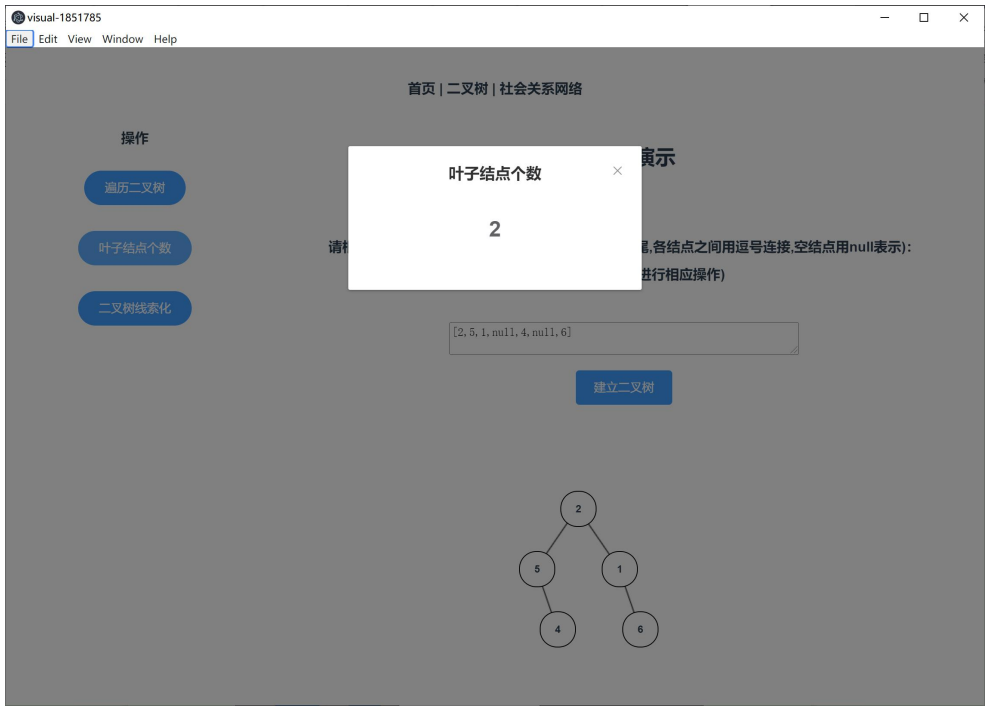
说明：点击侧边栏中“遍历二叉树”中的“中序遍历”按钮，会弹出显示中序遍历序列的消息框。

后序遍历：



说明：点击侧边栏中“遍历二叉树”中的“后序遍历”按钮，会弹出显示后序遍历序列的消息框。

统计叶子结点个数：



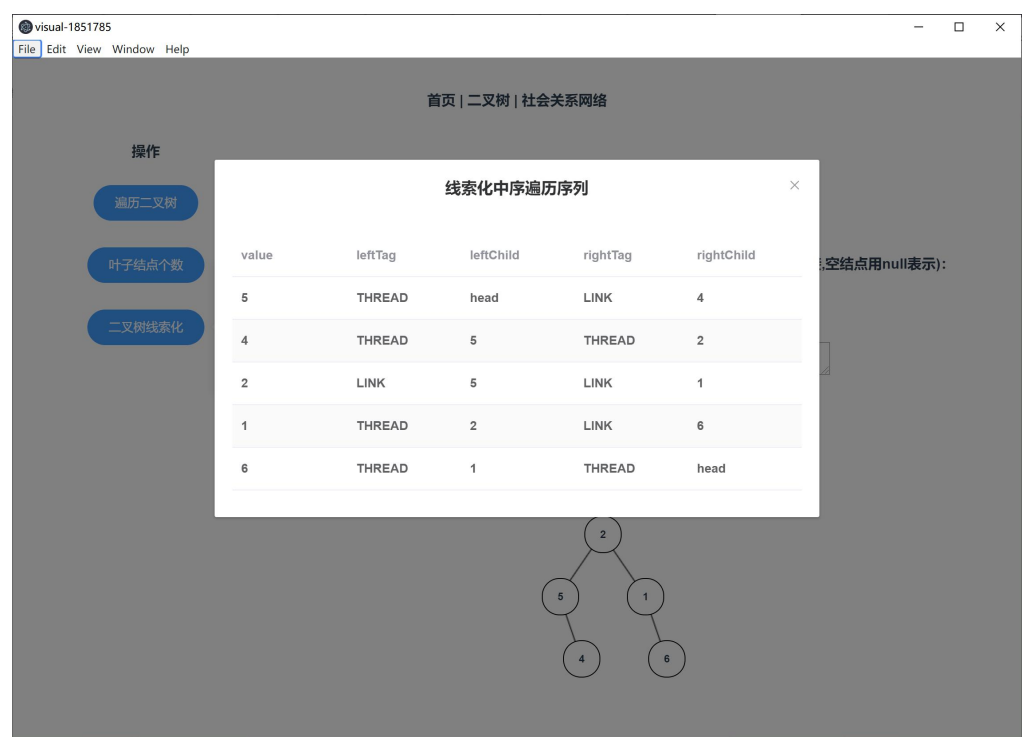
说明：点击侧边栏中“统计叶子结点个数”按钮，会弹出显示叶子结点个数的消息框。

先序线索化二叉树并遍历：



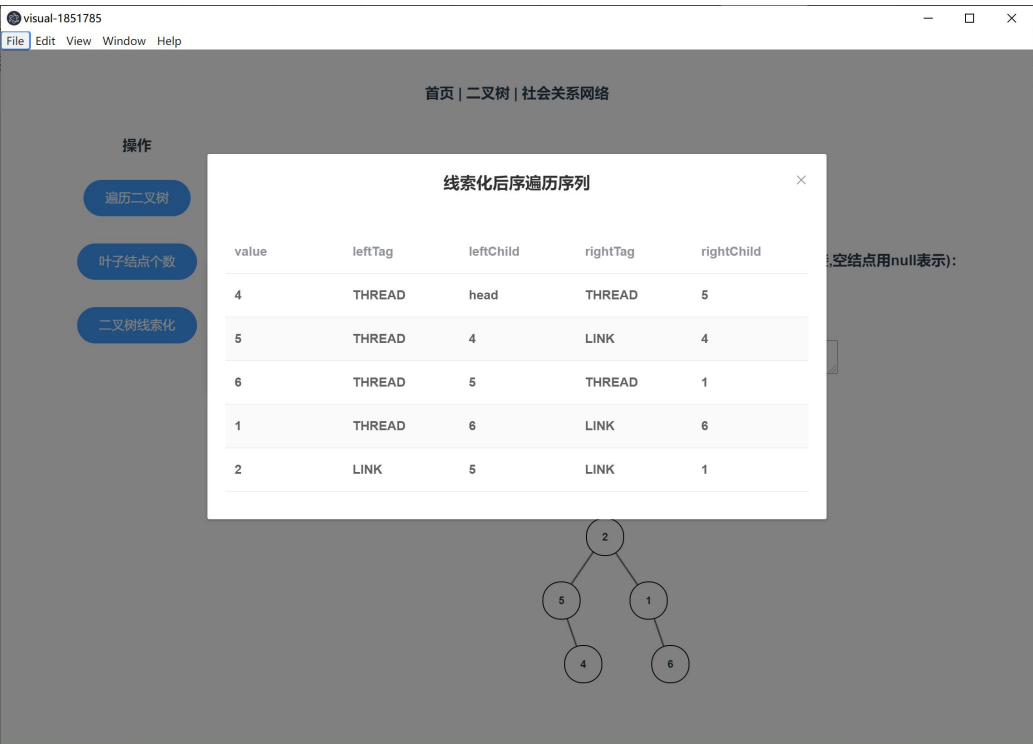
说明：点击侧边栏中“二叉树线索化”中的“先序线索化”按钮，会弹出显示先序线索树遍历序列以及各结点具体信息的消息框。

中序线索化二叉树并遍历：



说明：点击侧边栏中“二叉树线索化”中的“中序线索化”按钮，会弹出显示中序线索树遍历序列以及各结点具体信息的消息框。

后序线索化二叉树并遍历：



说明：点击侧边栏中“二叉树线索化”中的“后序线索化”按钮，会弹出显示后序线索树遍历序列以及各结点具体信息的消息框。

第二部分 综合应用设计说明

2.1 题目

5. ★★★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，每一人有众多好友，并可以根据个人兴趣及社会活动加入到某些群组。现需设计一算法，从该社会关系网络中某一人出发，寻找其可能认识的人。例如根据两个人共同好友的数量及所在群组情况，来发现可能认识的人；通过就读的学校情况发现可能认识的同学。

(1) 通过图形化界面，显示某一人的社会关系网络。

(2) 寻找某一人可能认识的人（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

- 显示某一个人的社会关系网络图，图中结点包括本人，其好友以及可能认识的人，互为好友的两个结点之间会有连线。
- 以表格形式列出某一个人可能认识的人，可以查看的信息有本人与ta的共同好友情况，共同群组情况，以及关联度(共同好友个数+共同群组个数 $\times 2$)，且默认按关联度由大到小进行排序。
- 查看我的个人信息，我的好友以及我的群组，并提供查询功能。
- 切换身份，以查看不同人的社会关系网络图和可能认识的人等信息。

实现方式：Vue+Element-UI+ECharts

2.3 设计思想

实现思路：

1.技术学习与准备工作

由于我打算将两个题目放到一个 Vue 项目里进行构建，这样可以直接在页面上进行切换，非常方便，因此所用到的技术与上一道题大部分相同，即 Vue+Element-UI，以及 JavaScript 等。但是，此题要求画出某个人的社会关系网络图，为了改善显示效果，我查阅了相关资料，发现 JavaScript 中有一个叫做 ECharts 的数据可视化图表库，画出来的图表效果非常好，因此，我又去学习了一下使用 ECharts 画图的方法。

2.设计框架

在准备工作做的差不多时，我便开始对这一题目进行框架设计，这道综合应用题的基本要求是寻找某个人可能认识的人，要求并没有给的太具体。我的想法是先在 JavaScript 中直接初始化好一定的人以及群组，给他们随机分配好友和群组。寻找一个人可能认识的人就通过看他们是否有共同好友或者位于相同群组中，并根据数量计算其关联度，再按照关联度进行排序，查看最有可能认识的人。

3.实现代码

实现框架设计好后，就可以上手完成代码，此次实现的难点主要在于利用 JavaScript 中的 ECharts 来进行社会关系网络图的绘制，我设计的网络图中会显示某个人，其好友以及可能认识的人，再通过结点之间的连线来表示互为好友关系。而对于每个人的信息，我是通过 Json 和 sessionStorage 来进行相应的传递，总的来说，也是通过 JavaScript 来进行实现。

数据结构：

```
"userList": ["王伟", "张一", "刘武", "徐斌", "秦策", "顾泽", "任宇", "谭勋", "曾芸", "陈路"],
"groupList": ["高数交流群", "大物交流群", "跑步交流群", "篮球交流群", "音乐交流群"],
"userInfo": [{
  "id": "0",
  "name": "王伟",
  "friend": [
    "张一",
    "刘武"
  ],
  "group": [
    "高数交流群"
  ]
}, {
  "id": "1",
  "name": "张一",
  "friend": [
    "王伟",
    "刘武",
    "秦策",
    "顾泽",
    "陈路"
  ],
  "group": [
    "高数交流群",
    "大物交流群"
  ]
}, {
```

图 1 用户信息(userData.json)

说明：直接在 userData.json 文件中静态定义好所有人名，群组名，以及顺序存储每个人的具体信息，即姓名，好友和所在群组。静态定义虽然比较死板，但好处在于便于操作、遍历以及传递信息。

```

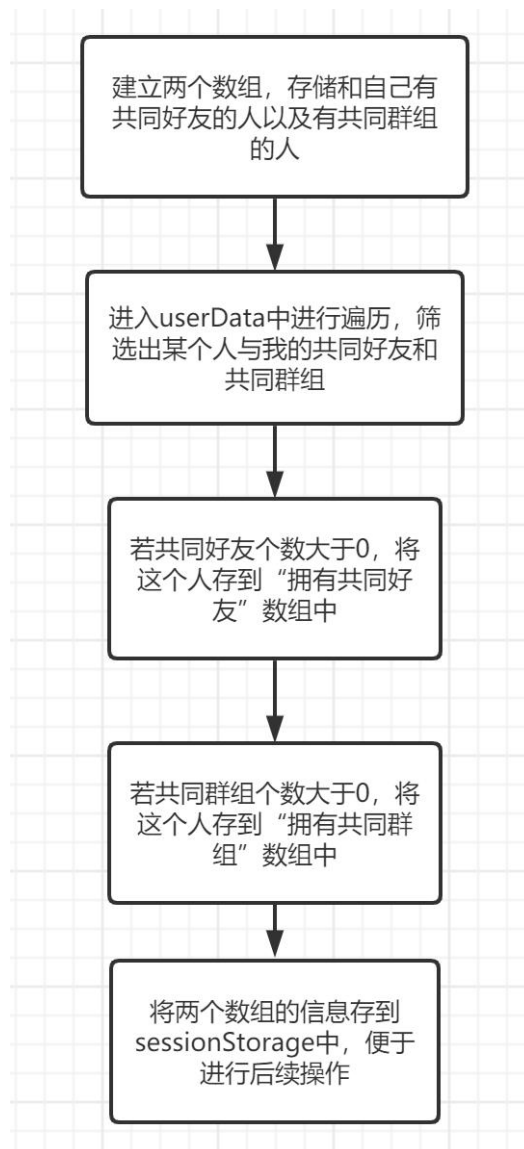
"nodes": [],
"links": [],
"categories": [{
  "name": "我",
  "symbolSize": 40,
  "label": {
    "textBorderColor": "inherit",
    "textBorderWidth": 1,
    "fontSize": 14
  }
},
{
  "name": "好友",
  "symbolSize": 36,
  "label": {
    "color": "#ffffff",
    "fontSize": 12,
    "textBorderColor": "#91cc75",
    "textBorderWidth": 1
  }
},
{
  "name": "可能认识的人",
  "symbolSize": 24,
  "label": {
    "fontSize": 12,
    "color": "#ffffff",
    "textBorderColor": "#fac858",
    "textBorderWidth": 1
  }
}
]

```

图 2 网络图信息(graphData.json)

说明：直接在 graphData.json 文件中定义好存储结点的数组以及存储连线的数组，并提前定义好三个类别，分别为“我”“好友”“可能认识的人”，在图中这三类会分别以不同的颜色进行展示，使得网络图更加直观。

算法设计基本流程：



2.4 逻辑结构与物理结构

逻辑结构：

由于某个人存在好友和可能认识的人，而可能认识的人与我的好友之间又可能互为好友关系，也即与我有共同好友，因此此题的逻辑结构为图形结构。

物理结构：

对于用户信息的存储，使用的物理结构是顺序存储结构，将所有人的信息直接以结构体的方式存储到 `userInfo` 数组中。而考虑到图形显示效果问题，此题中我初始化的人数为 10 人，这样在寻找某个人的可能认识的人时，就可以直接通过遍历 `userInfo` 数组，统计每一个人与我的共同好友和共同群组情况，以此判断是否为可能认识的人。

2.5 开发平台

开发平台: **Vue.js**

Vue 是一套用于构建用户界面的渐进式 JavaScript 框架。与其它大型框架不同的是，Vue 被设计为可以自底向上逐层应用。Vue 的核心库只关注视图层，不仅易于上手，还便于与第三方库或既有项目整合。另一方面，当与现代化的工具链以及各种支持类库结合使用时，Vue 也完全能够为复杂的单页应用（SPA）提供驱动。

第三方库：Element-UI

Element-UI 是一套为开发者、设计师和产品经理准备的基于 Vue 2.0 的桌面端组件库，是一款方便快捷，容易上手的网速快速成型工具，其 UI 美观大方，简约干净，受到许多 Vue 开发者的喜爱。

第三方库：ECharts

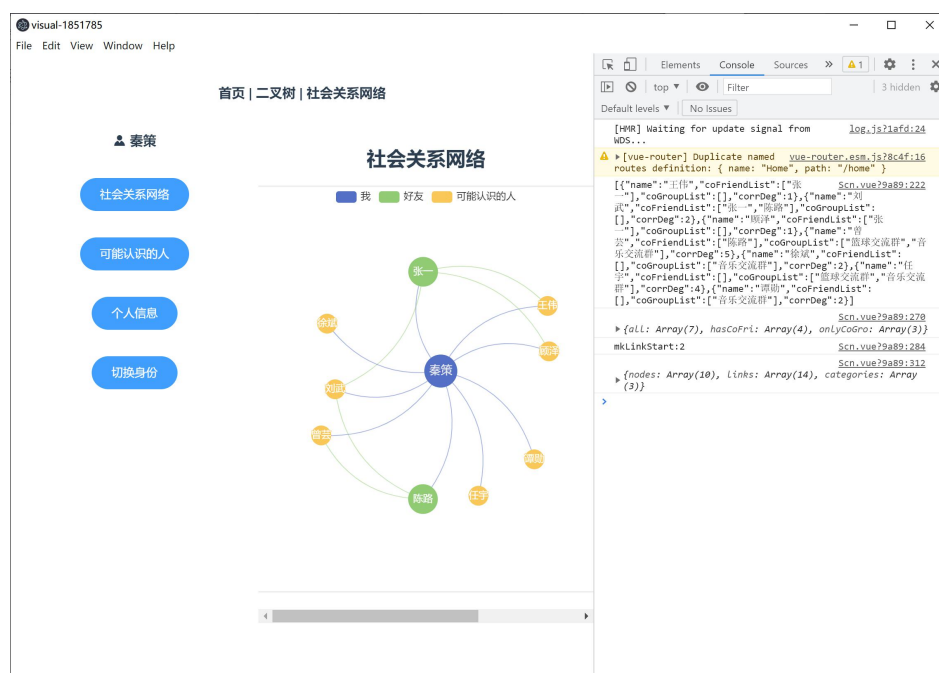
ECharts 是一个使用 JavaScript 实现的开源可视化库，可以流畅的运行在 PC 和移动设备上，兼容当前绝大部分浏览器（IE8/9/10/11，Chrome，Firefox，Safari 等），底层依赖轻量级的矢量图形库 ZRender，提供直观，交互丰富，可高度个性化定制的数据可视化图表。

软件的运行环境: Electron/任意浏览器

2.6 系统的运行结果分析说明

调试/开发过程:

由于处于同一个项目中，所以调试开发过程与上题一致，如图所示。



正确性/稳定性:

经过不同身份的切换验证, 每个人的好友情况和可能认识的人情况均正确, 且绘出的社会关系网络图均符合情况, 具体案例可以见操作说明。

容错能力:

有一个操作失误的情况已经被考虑, 就是若用户切换的身份就是当前身份时, 页面不会发生改变, 同时右上角会有弹窗消息, 提示已是当前身份, 无法切换, 如图所示。



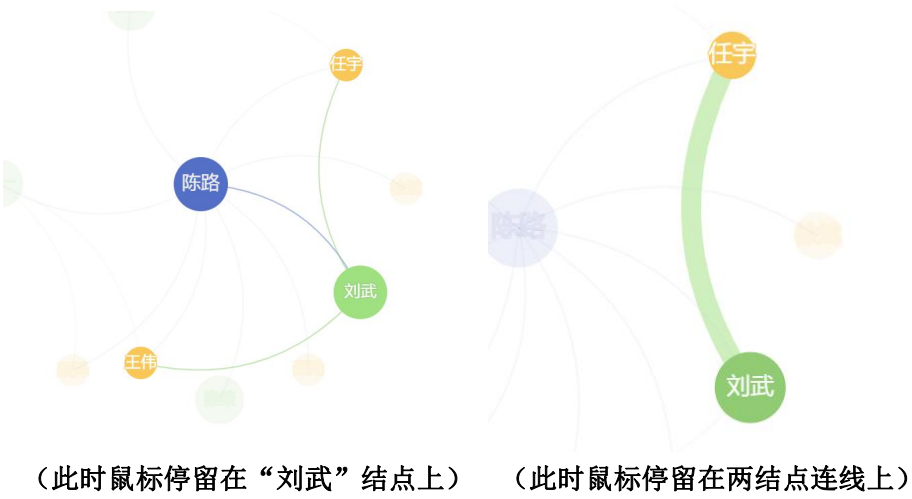
2.7 操作说明

社会关系网络界面：



说明：界面分为两个部分，分别是侧边栏和主体部分，侧边栏中包括一些操作需要用到的按钮，包括查看社会关系网络，可能认识的人，个人信息以及切换身份。主体部分为每个页面的具体内容，点击进来的默认页面为社会关系网络页面，会自动绘制出我的社会关系网络图。其中，我位于中间蓝色结点，好友为绿色结点，可能认识的人为黄色结点。连线部分，除了我与其他结点之间都有连线以外，两个互为好友关系的结点之间也会有连线。

同时，可以通过滚动鼠标滚轮来调整网络图的大小，还可以通过鼠标的移动来单独查看某一结点或某一连线的情况，如图所示。

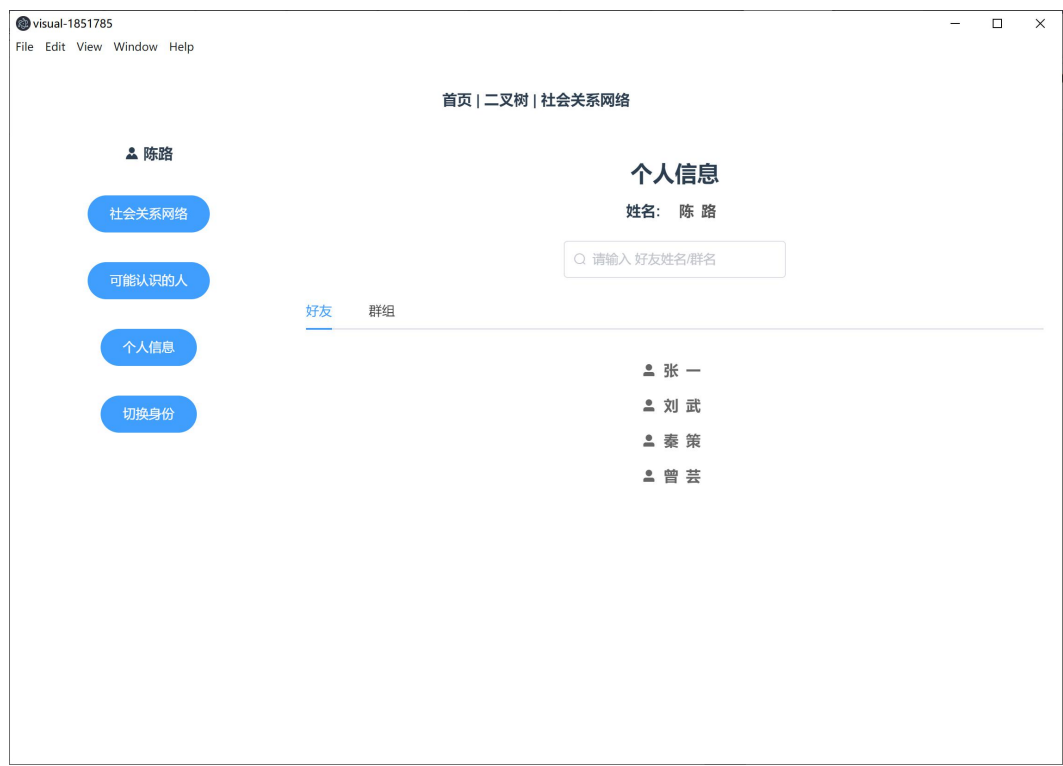


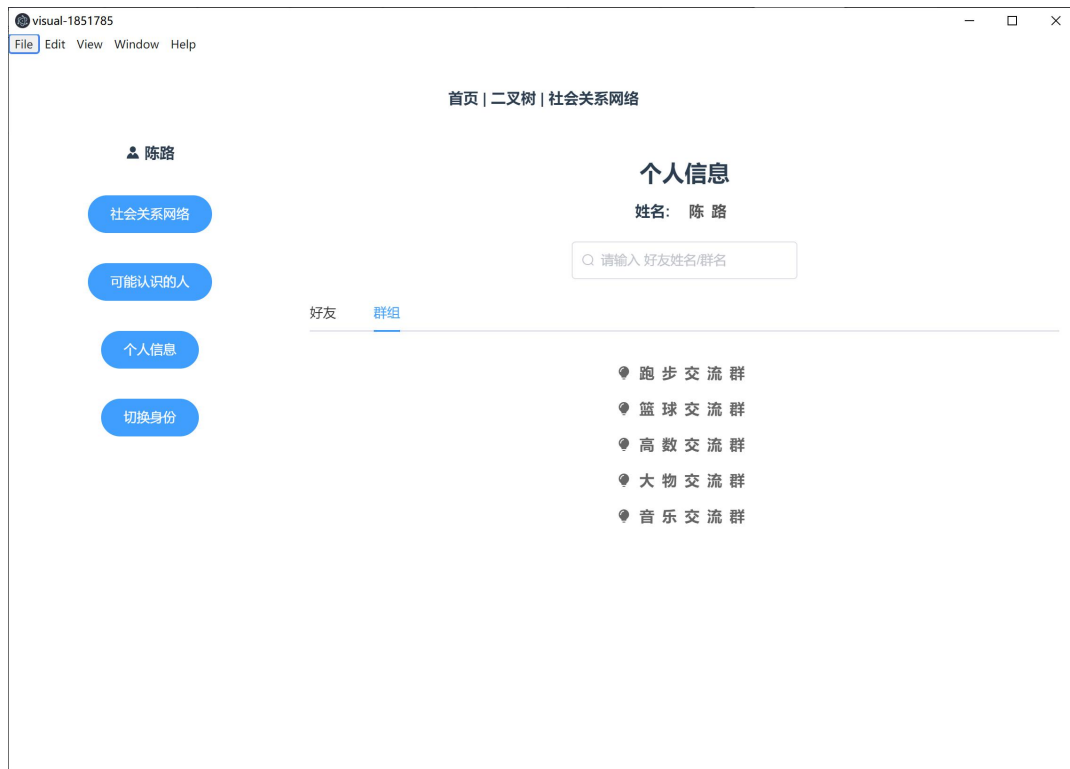
可能认识的人界面：



说明：此页面中，以表格形式展示我可能认识的人，其中包括姓名，共同好友信息以及共同群组信息，以及关联度(共同好友个数+共同群组个数×2)，并且已经默认按照关联度由高到低进行排列。

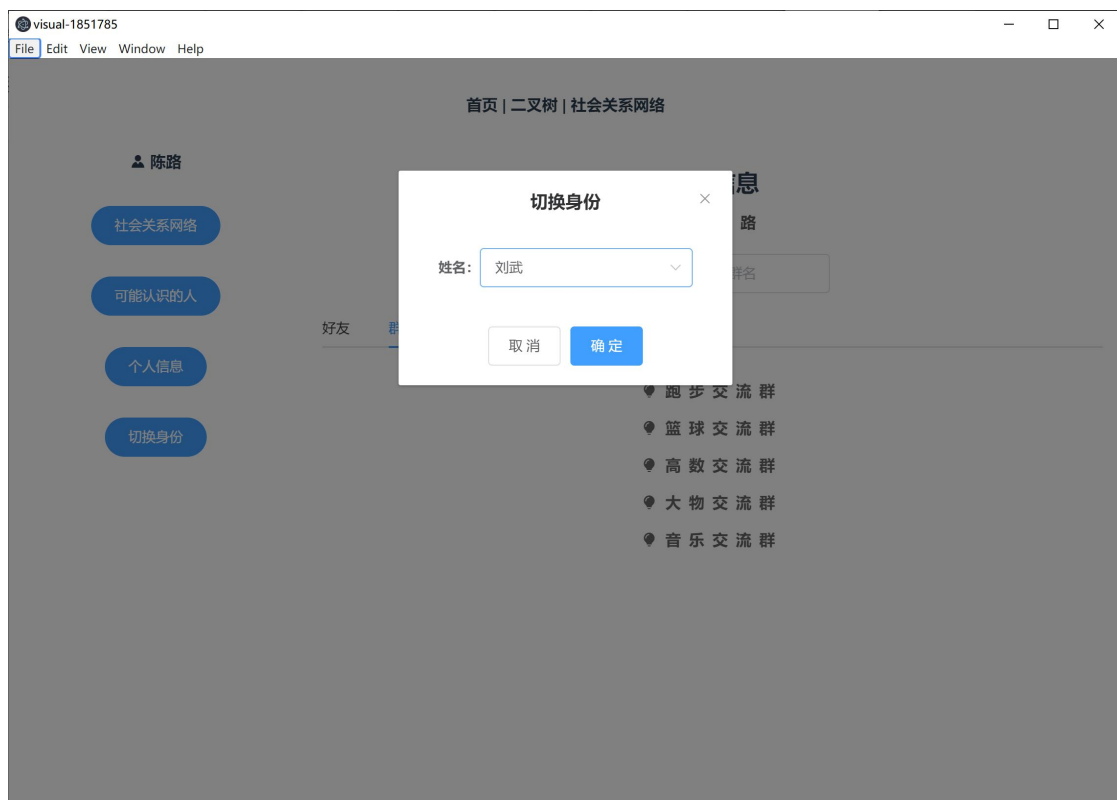
个人信息界面：

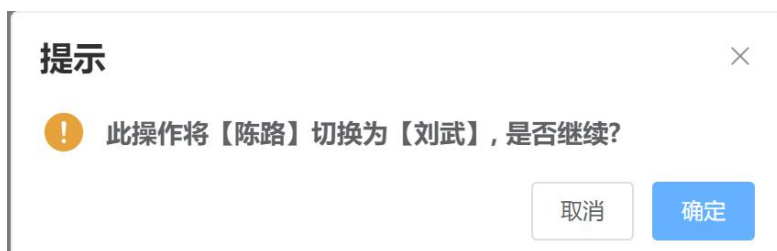




说明：个人信息界面可以查看姓名，以及好友情况和群组情况，还可以直接进行检索。

切换身份界面：





说明：点击切换身份按钮，可以选择十个人中的任意一个人进行切换，切换成功后，会自动跳转到该人的社会关系网络页面，此时也同样可以查看其可能认识的人和个人信息。

第三部分 实践总结

3.1.所做的工作

- 学习了利用canvas来绘制二叉树。
- 学习了利用ECharts来绘制社会关系网络。
- 利用Javascript进行数据传递以及算法构建。
- 利用Vue+Element-UI搭建了简单的前端页面，进行二叉树的构建和相关操作展示，以及社会关系网络的构建和展示。

3.2.总结与收获

能力提升:

完成此次课程设计之后,我的各方面能力都得到了明显的提升。首先,是分析解决问题的能力,尤其是对于综合应用题,题目中并未给出太多的限制性条件,而只是简单的给出了题目的背景,并简要说明寻找可能认识的人这一要求,然而具体怎么找,如何展现,都并未给出明确说明,这就要求我在熟悉题目以后,自行设计数据结构和页面框架,这非常考验解决实际问题的能力,在实现的过程中也的确得到了有效提升。还有一个就是自我学习的能力,由于这两道题目对于绘图全部都有要求,在这之前,我从未学习过绘图相关的技术和方法,只能从网上查找相关教程,从零开始学起,这对于自主学习的能力是一个非常高的要求,相信通过这种能力的培养,也会对我们未来的个人发展大有裨益。

课程收获和心得体会:

这次数据结构课程设计无论对于数据结构知识还是前端相关技术知识都是一个不小的考验,我个人觉得它对于灵活运用可视化技术的要求甚至高于对于数据结构知识本身的掌握要求。就拿算法实现题来说,其实像二叉树的遍历和线索化等等知识,我们都已经在课堂上进行了详细学习,有了一定的掌握,因此,课设的主要难度就在于如何将这些数据和信息进行友好的可视化。而可视化中画图又成为了我这次面临的主要困难,因为之前从未接触过相关技术。通过此次课设,自己从头开始学习 ECharts 这样的画图工具,并利用其画出效果很好的社会关系网络图,这对我来说就是一个巨大的收获。

此次课程主要的要求就在于标准的交互式图形界面,由于技术是自选的,所以我就选择了曾经使用过的 Vue 框架结合 Element-UI 组件库来搭建前端页面,在实践的过程中,自己搭建网页的经验也进一步得到了增加,现在对于 Vue 的使用也可以说算是小有心得。

第四部分 参考文献

- [1]杨晓波,陈邦泽. 二叉树的可视化实现[J]. 软件, 2011(12):24-27.
- [2]高攀,郑瑶,赵恒斌,等. 基于算法融合思想的交互式可视化实验案例设计二叉树综合应用[J]. 当代教育实践与教学研究, 2019(08):63-64.
- [3]苏慧哲,刘杰. 浅谈如何输出二叉树图像[J]. 电脑知识与技术:学术版, 2018, 014(036):247-249.
- [4]王子毅,张春海. 基于 ECharts 的数据可视化分析组件设计实现[J]. 微型机与应用, 2016, 35(014):46-48.