

**《数据结构》课程设计总结**

**学 院 ： 电子与信息工程学院**

**专 业 ： 计算机科学与技术**

**学 号 ：** 1651574

**姓 名 ：** 贾昊霖

**指导老师 ：** 叶晨

**2018 年 9 月 11 日**

目录

[第一部分 算法实现设计说明 1](#_Toc524468554)

[1.1 题目 1](#_Toc524468555)

[1.2 软件功能 1](#_Toc524468556)

[1.2.1 图形化实现图，给用户很好的直观的体验 1](#_Toc524468557)

[1.2.2 显示出其邻接表 2](#_Toc524468558)

[1.2.3输出确定起点的最短路径问题 2](#_Toc524468559)

[1.2.4用不同算法显示出一个图的最小生成树 2](#_Toc524468560)

[1.3 设计思想 3](#_Toc524468561)

[1.3.1 整体程序框架 3](#_Toc524468562)

[1.3.2 index.html 3](#_Toc524468563)

[1.3.3 font.css 4](#_Toc524468564)

[1.3.4 forceGraph.js 5](#_Toc524468565)

[1.3.5 algorithm.js 7](#_Toc524468566)

[1.3.6 highlightLine.js 12](#_Toc524468567)

[1.3.7 link\_algorithm\_graph.js 12](#_Toc524468568)

[1.4 逻辑结构与物理结构 14](#_Toc524468569)

[1.5 开发平台 14](#_Toc524468570)

[1.6 系统的运行结果分析说明 16](#_Toc524468571)

[1.6.1调试遇到的问题一 16](#_Toc524468572)

[1.6.2调试遇到的问题二 16](#_Toc524468573)

[1.7 操作说明 17](#_Toc524468574)

[第二部分 综合应用设计说明 20](#_Toc524468575)

[2.1 题目 20](#_Toc524468576)

[2.2 软件功能 20](#_Toc524468577)

[2.3 逻辑结构与物理结构 21](#_Toc524468578)

[2.3.1 工程组成 21](#_Toc524468579)

[2.3.2数据结构 22](#_Toc524468580)

[2.4 开发平台 23](#_Toc524468581)

[2.5 系统的运行结果分析说明 24](#_Toc524468582)

[2.5.1 SocialNetWork.js: 24](#_Toc524468583)

[2.5.2 AddEdge.js: 27](#_Toc524468584)

[2.5.3 AddNode.js: 29](#_Toc524468585)

[2.5.4 index.html 29](#_Toc524468586)

[2.7 操作说明 32](#_Toc524468587)

[第三部分 实践总结 34](#_Toc524468588)

[3.1所做的工作 34](#_Toc524468589)

[3.2 总结及感悟 34](#_Toc524468590)

[第四部分 参考文献 34](#_Toc524468591)

# 第一部分 算法实现设计说明

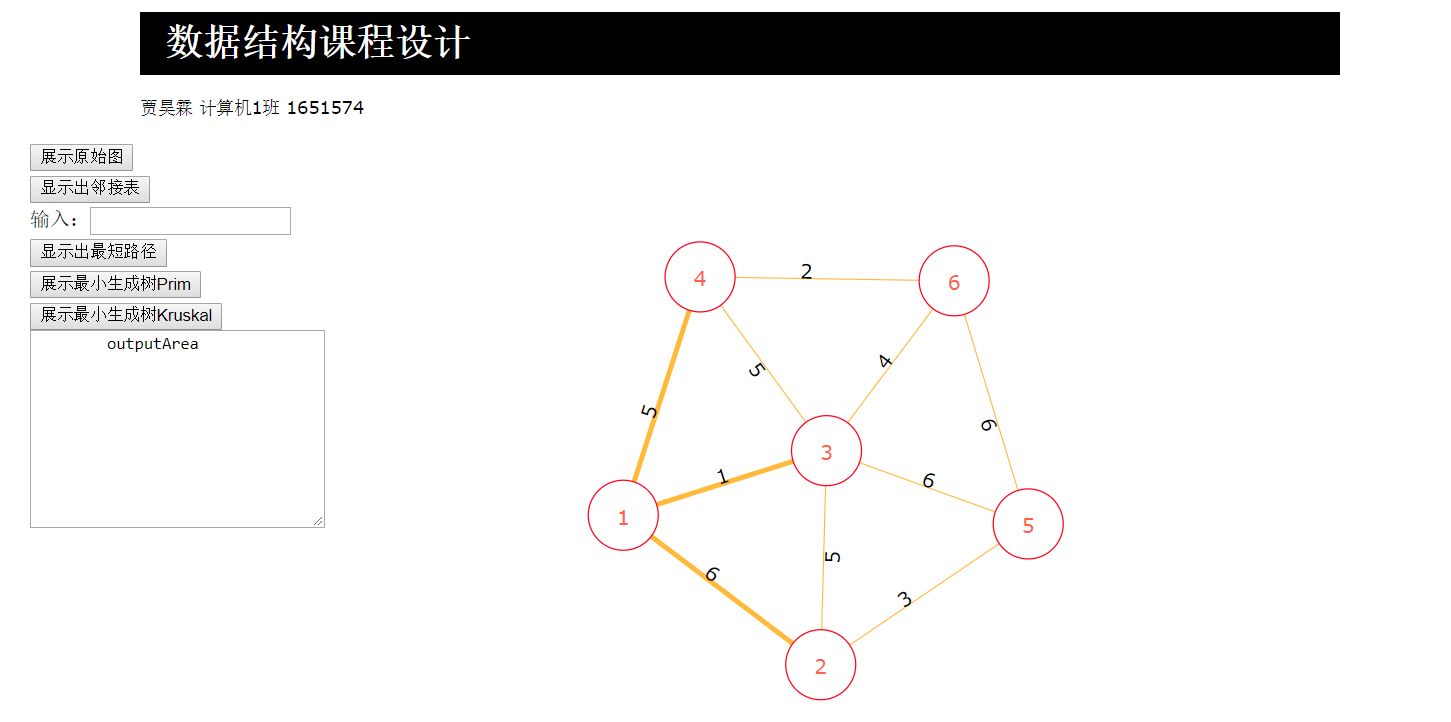
## 1.1 题目

给定一个图，完成：

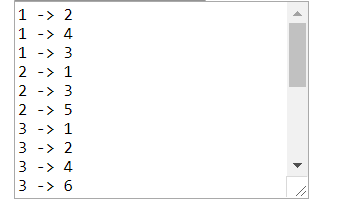
1. 建立并显示出他的邻接链表
2. 分别用Prim算法和kruskal算法构造最小生成树，随时显示其构造的过程
3. 给出某一确定定点到所有其它定点的最短路径
4. 给出每一对顶点之间的最短路径

## 1.2 软件功能

### 1.2.1 图形化实现图，给用户很好的直观的体验

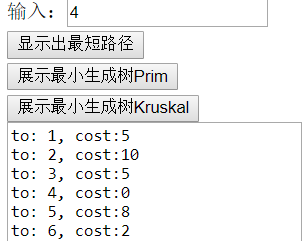


### 1.2.2 显示出其邻接表



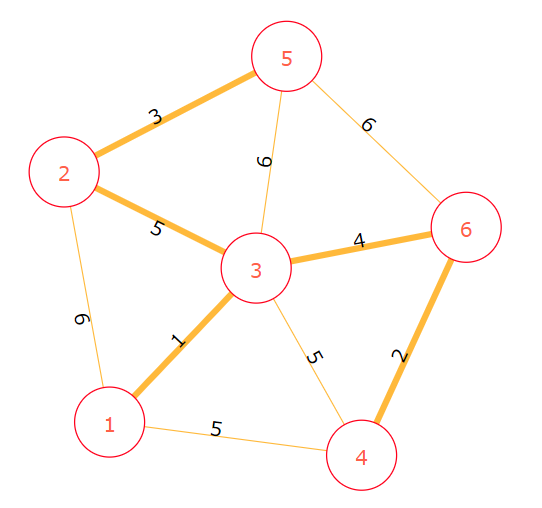
邻接表，存储方法跟树的孩子链表示法相类似，是一种顺序分配和链式分配相结合的存储结构。如这个表头结点所对应的顶点 存在相邻顶点，则把相邻顶点依次存放于表头结点所指向的单向链表中。

### 1.2.3输出确定起点的最短路径问题



　　在输出框中输出可以显示从一个源点到达其他节点的最短路径。

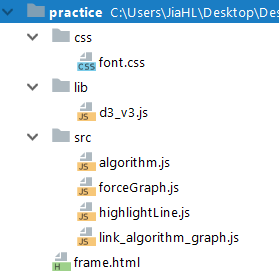
### 1.2.4用不同算法显示出一个图的最小生成树



最小生成树的边由加粗的黄色边显示

## 1.3 设计思想

### 1.3.1 整体程序框架



由单个frame.html 调用各个js,css文件，其中css文件夹为css库的调用，其中的css为自己编写的排版程序，主要为了界面的美观

lib文件夹下的d3\_v3为大数据可视化d3库的库文件，下载源为d3官网的：

src文件夹为所有代码文件，叙述如下：

algorithm.js 为算法主程序，里边包含了Graph类的实现

forceGraph.js 为d3库的调用即实现可视化的主要js程序

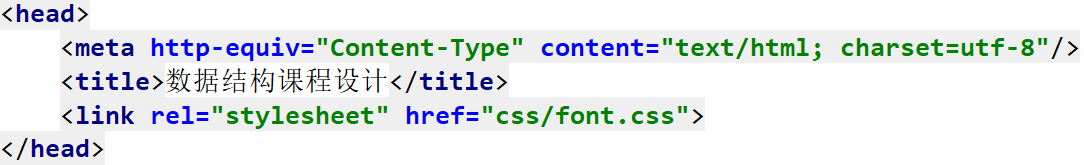
highlightLine 为但拿出来的函数文件

link\_algorithm\_graph.js为串联algorithm与forceGraph程序的重要关联程序

整个软件用html构建，在网页中显示出图以及相应算法的结果，

主要实现数据的输入判断、打印输出与6种图算法，为便于相互间配合实现子对话框，动态显示等功能。各函数相应作用如下：

### 1.3.2 index.html

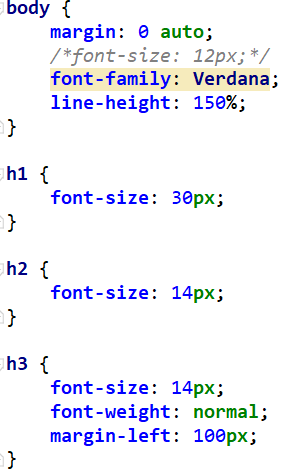


head部分的html代码，并用link调用了css文件

如图所示为，Webstorm下html种buttons类下的所有成员

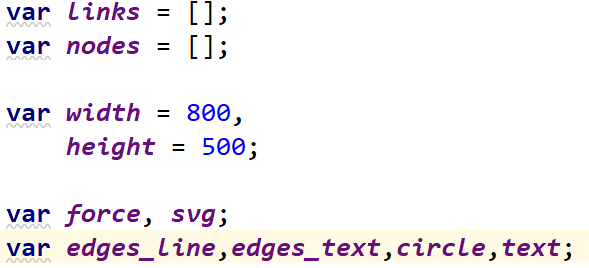


### 1.3.3 font.css

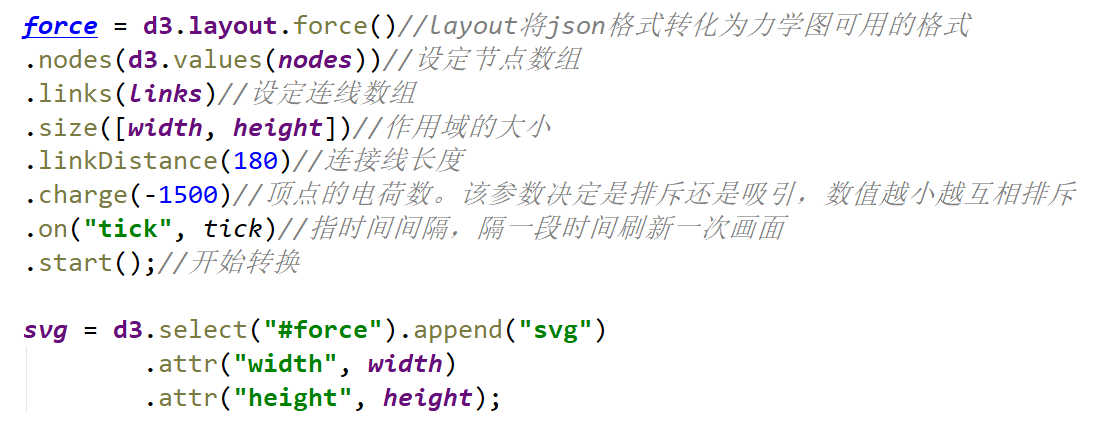


自己编写的css文件，实现朴素的界面，简朴风格

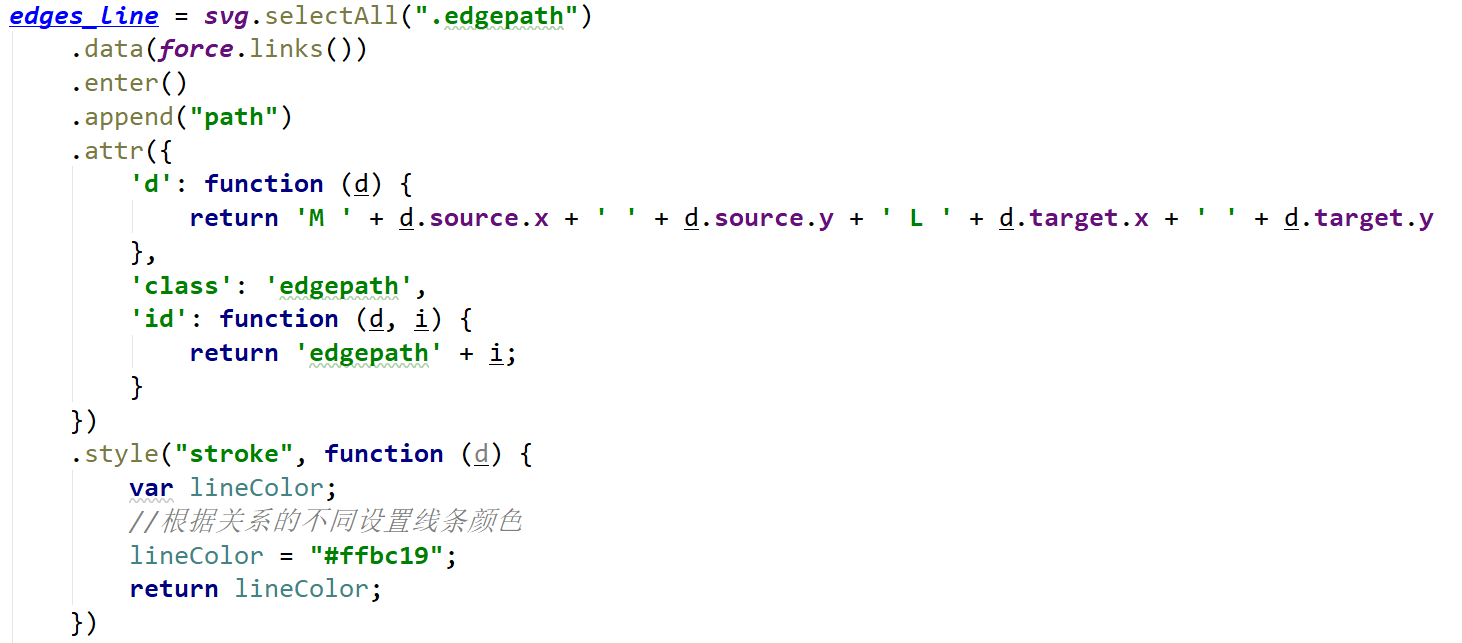
### 1.3.4 forceGraph.js



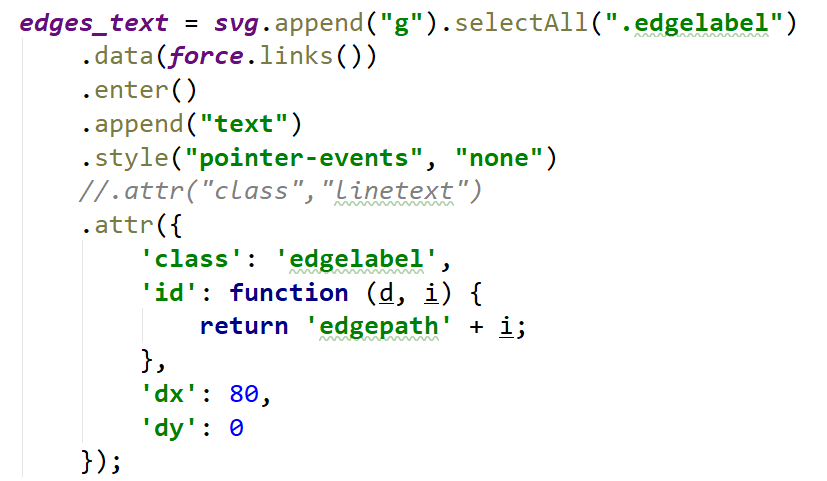
此为全局变量，用于传输变量数值，使link\_algorithm\_graph文件得以调用



此为将json格式的文件抓换成力学图，分别设定设定节点数组，连线数组以及作用域的大小



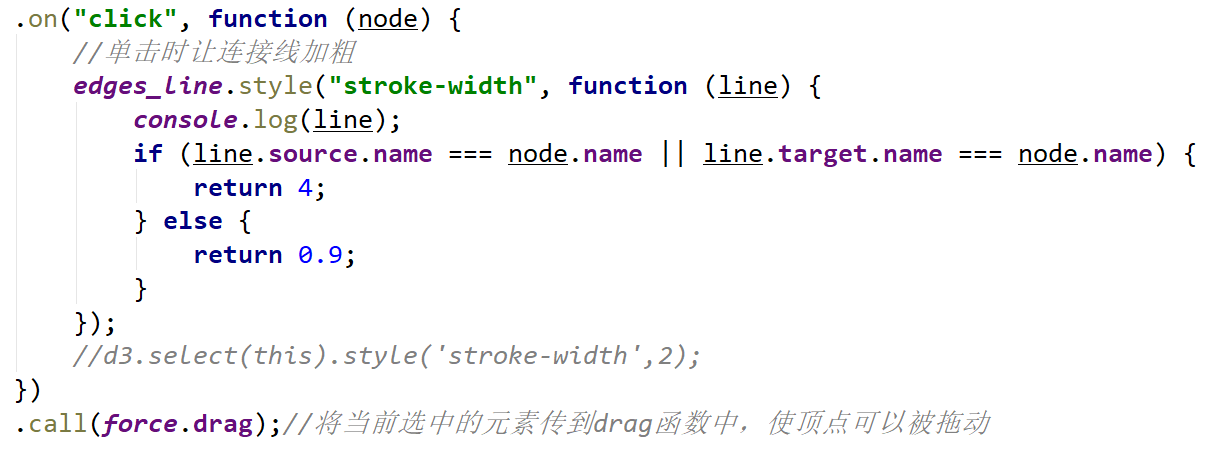
edges\_line 变量主要为d3.svg种的边变量，因此很多通过这个变量可以用attr函数增加<edgepath>中的属性值。



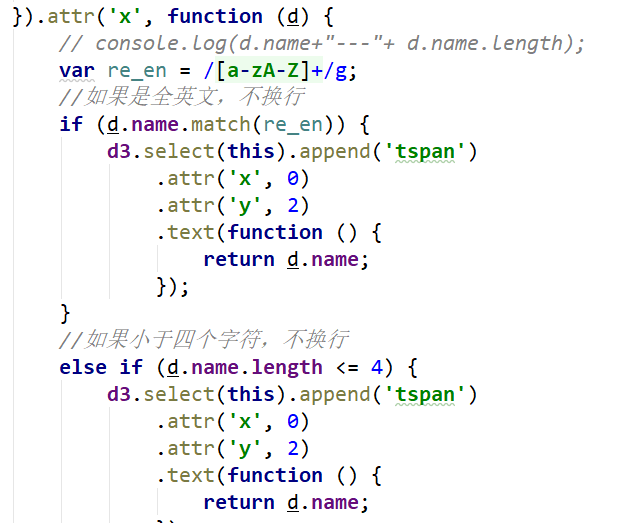
同理此为边上的文字操作，其中dx为可以查看的文字距离定点的距离



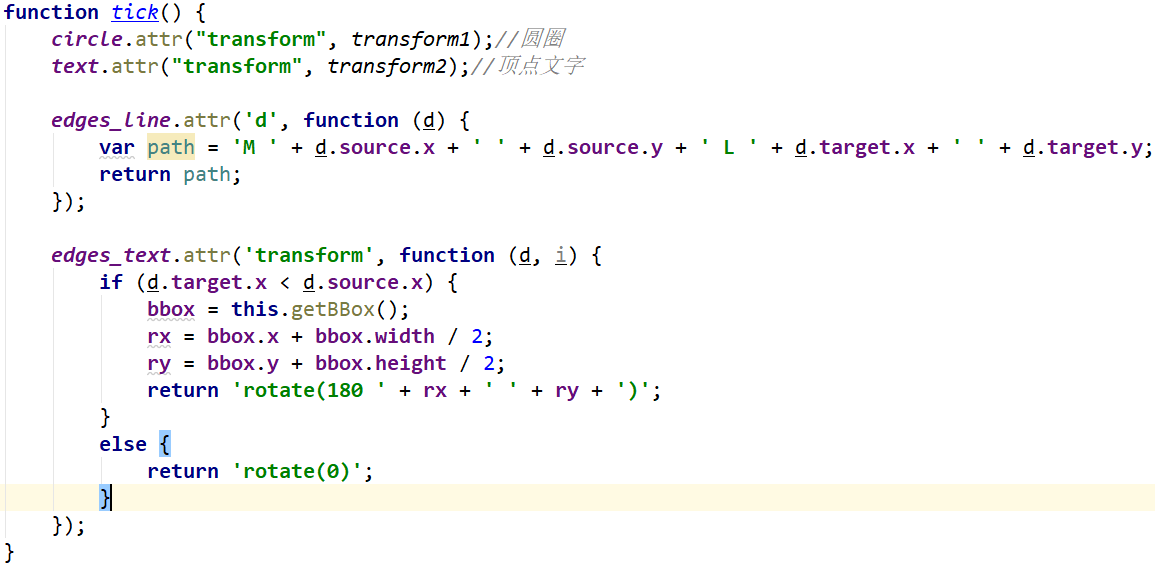
此为增加的圈变量，用来记录图中相应的节点，其中增加了如下的方法：



即点击可以加粗线的操作，具体为function一个高阶函数传入一个node变量为系统默认的点，然后对其进行操作，即对于点击的节点而言，edges\_line变量的style方法种 stroke-with又是一个高阶函数，传递进入每一个edges\_line的line，此line为所有edges\_line中的一个，对于这个line进行判断，如果两个点中有一个点为点击的node，即满足加粗操作。



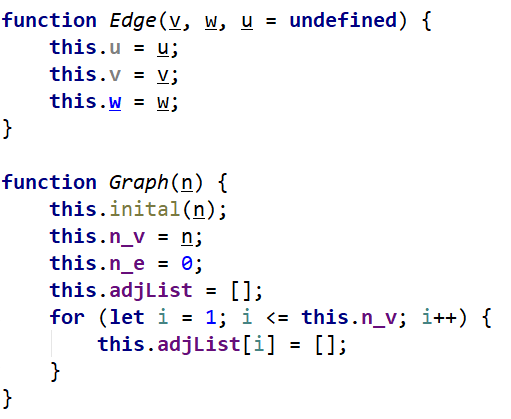
此函数实现较为繁琐，保证了文字在旋转中可以实现夹角小于90°的效果，增加了程序的友好性



tick函数用于随时更新圆圈的位置，根据刷新间隔更新

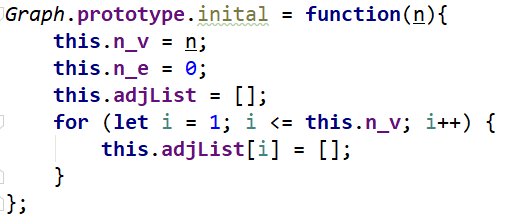
### 1.3.5 algorithm.js

此为算法函数，最重要也是最基本的数据结构到汇聚在次js文件中

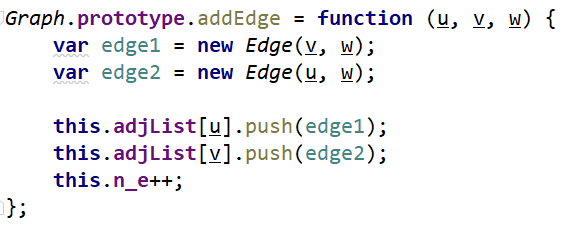


定义了两个类，分别为Edge、Graph

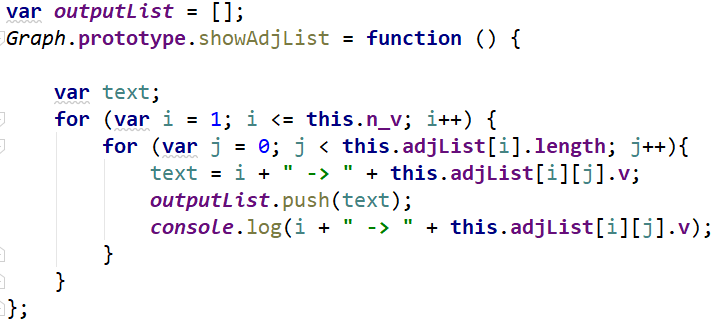
用于保存边的u,v,w 以及一张图的所有信息以及求解方法



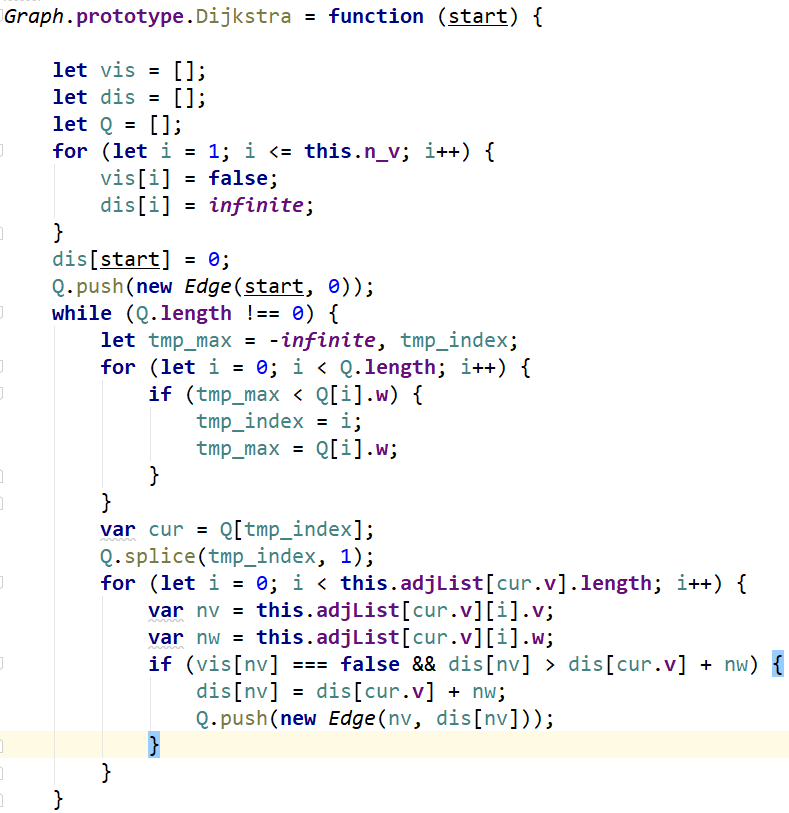
函数初始化操作，adjList为邻接表，该表的索引为起始点，索引对应的值为指向的点



增边操作，简单易懂，在邻接表中增加边



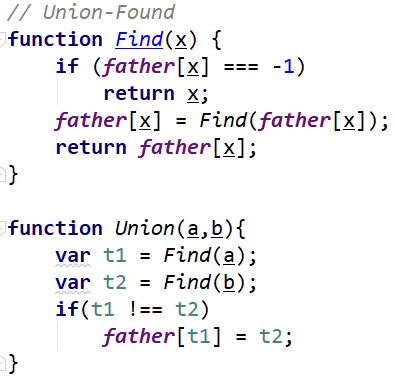
输出邻接表，此为题目要求1.



Dijkstra最短路径求解，传入一个起始点，通过算法求解到各个点的最短路径长度，并通过shortest\_list输出



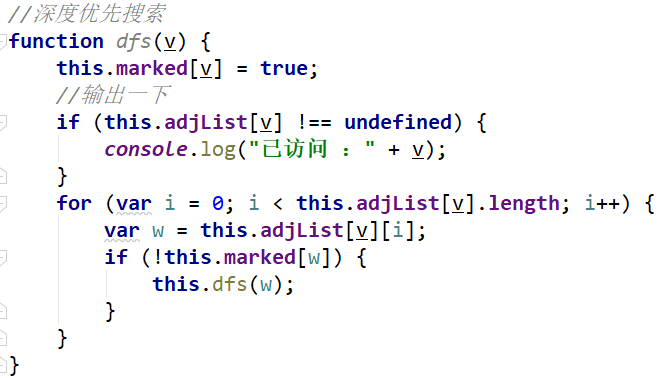
最小生成树算法，得到最小生成树，通过prim\_list传出相连的各个点，然后遍历求得相连的点对应的边集



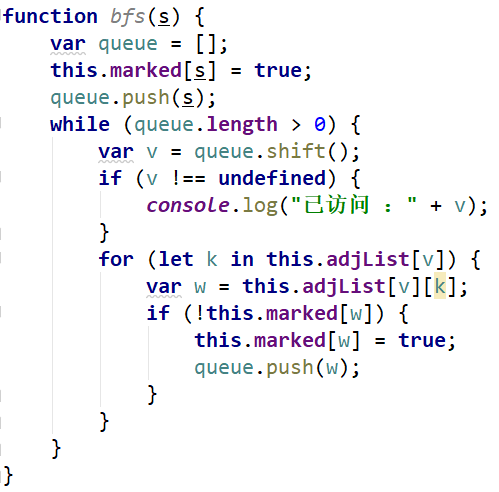
并查集，判断是否形成环



kruskal算法，配合并查集，先排序后得到最小生成树



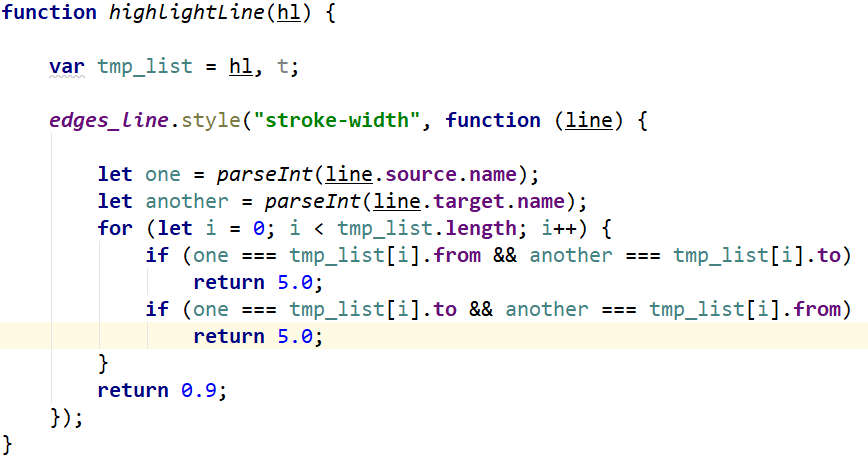
此为自己写的深度优先搜索



此为自己联系写的宽度优先搜索

### 1.3.6 highlightLine.js

特殊增加了一个加粗边的操作



时间复杂度为O(n2)，由于时间紧迫没有考虑算法优化，走暴力了

### 1.3.7 link\_algorithm\_graph.js

链接枢纽文件



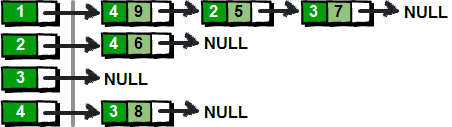


三个函数分别为链接邻接表，链接最短路径，链接prim，链接kruskal函数

并得到html中的id，用document.getElementBtId()等朴素JavaScript语法实现（没有用jQuery。

## 1.4 逻辑结构与物理结构

逻辑结构：邻接表（非邻接矩阵）



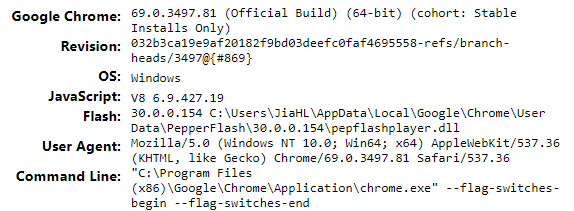
## 1.5 开发平台

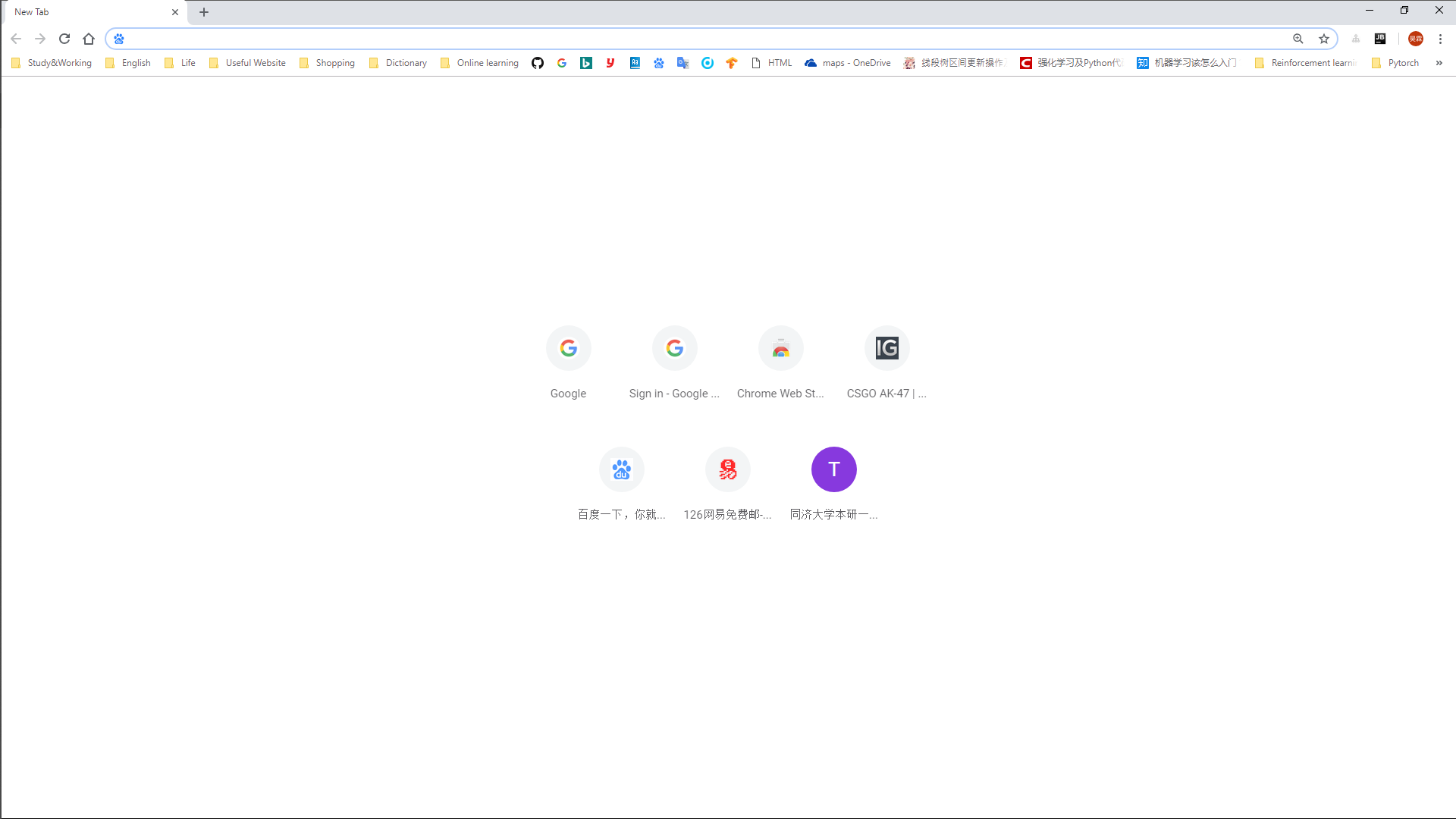
程序开发平台：

Webstorm 2018.2



Chrome

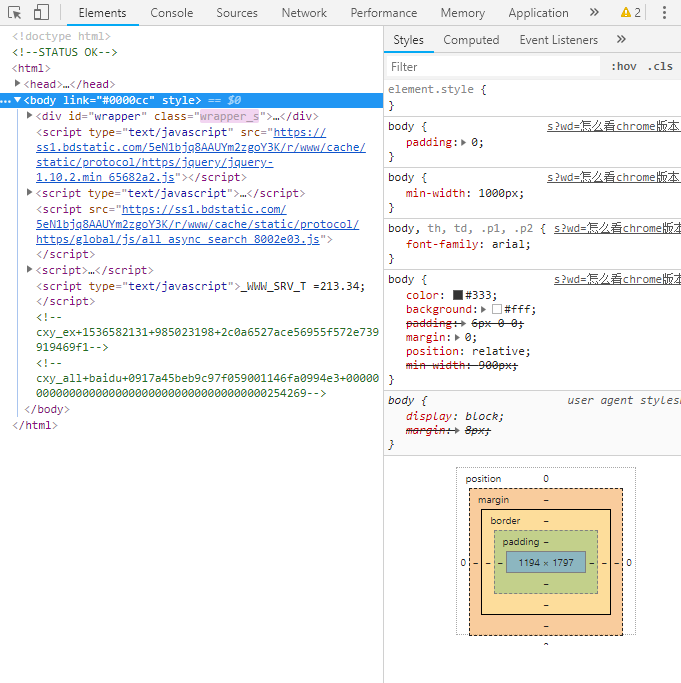




## 1.6 系统的运行结果分析说明

### 1.6.1调试遇到的问题一

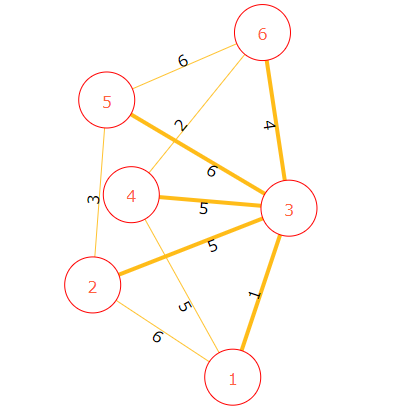
javascript 运行html时Chrome无法兼容信息，直到现在仍没解决



按网上所说，命令行中加入--flag-switches-begin --flag-switches-end，也无济于事..

### 1.6.2调试遇到的问题二

d3库中的图片无法显示，这个问题困扰了我1整天，从早上7：30到凌晨1：30浪费了我大量的时间，导致不能满足部分提议要求，促使我不得不放弃一些功能

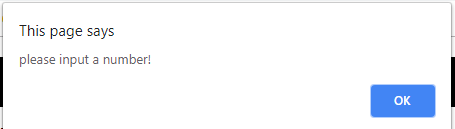


例如对动态生成最小生成树的展示等..

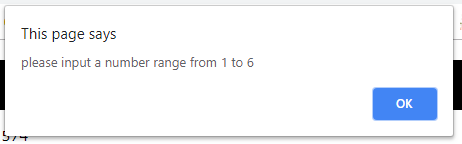
## 1.7 操作说明

算法实现设计软件主要由主对话框与子对话框两个部分构成。

主对话框中主要由静态文本（输入提示）、编辑框（输入区域）、五个按钮等控件，以及一个标题栏，项目栏构成。初始状态下用户通过主对话框中的编辑框随机输入一组数据后，软件判断输入数据是否合法：若输入错误，软件将警告错误类型

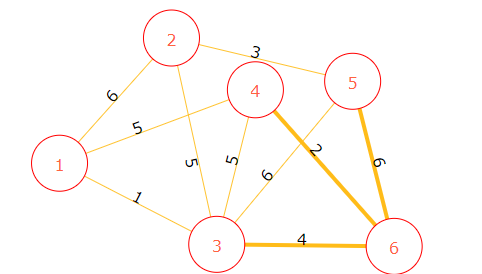
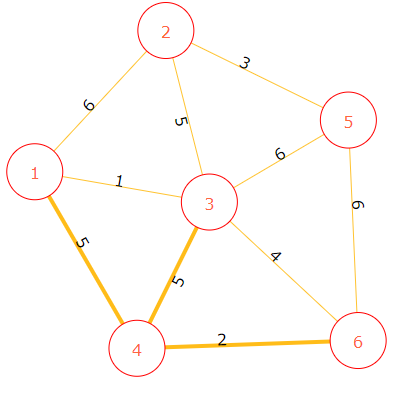


以及给定的图中判断是否输入的数字有效：

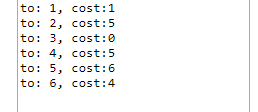


并提示用户重新输入合法数据，同时清空编辑框中输入内容，排序菜单仍保持禁止状态；若输入正确，软件将提示用户在菜单栏中选择相应排序类型，同时保存并禁止向编辑框中输入内容，禁止“确定”按钮，排序菜单则呈现使能状态。

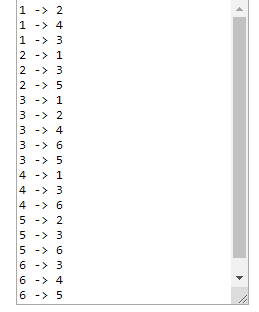
如图所示，可以随意拉动d3力导向图，每个节点以及相连的边清楚地显示如下。

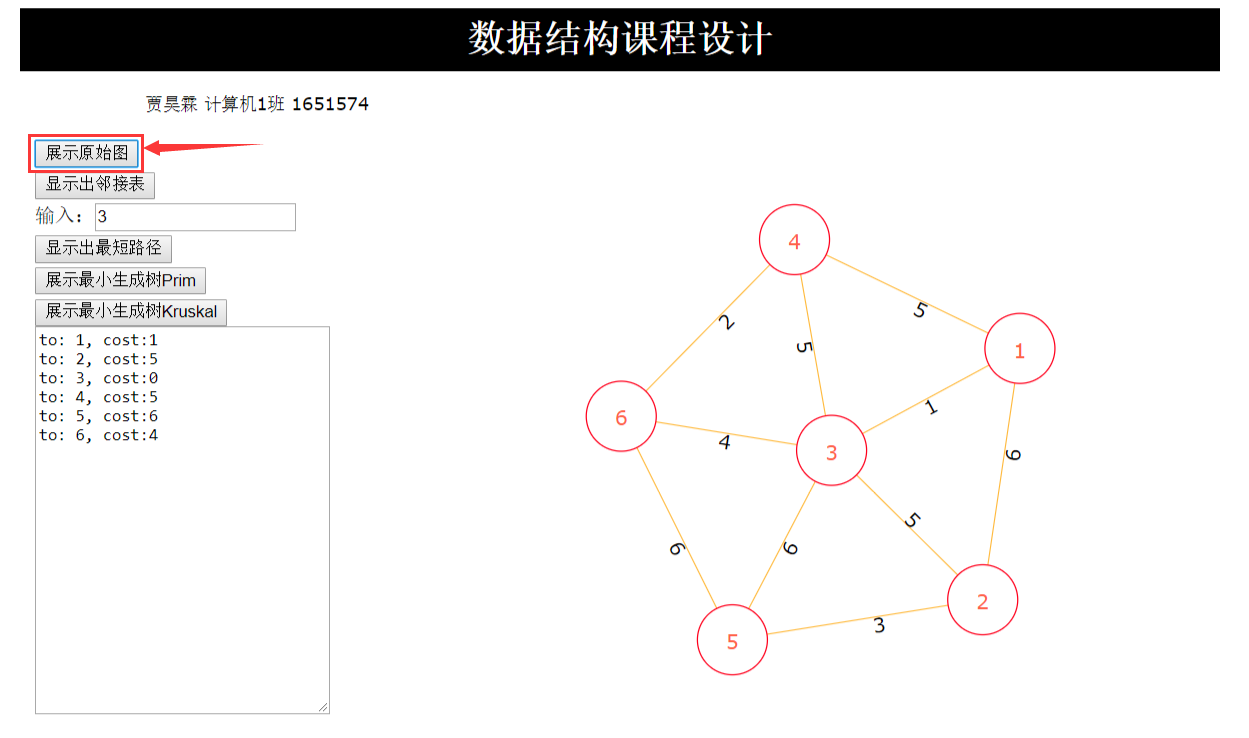


生成地最短路径输出如下：



生成地邻接表打印如下：



若点击

可以恢复最初地状态，即初态，重新检阅该程序

# 第二部分 综合应用设计说明

## 2.1 题目

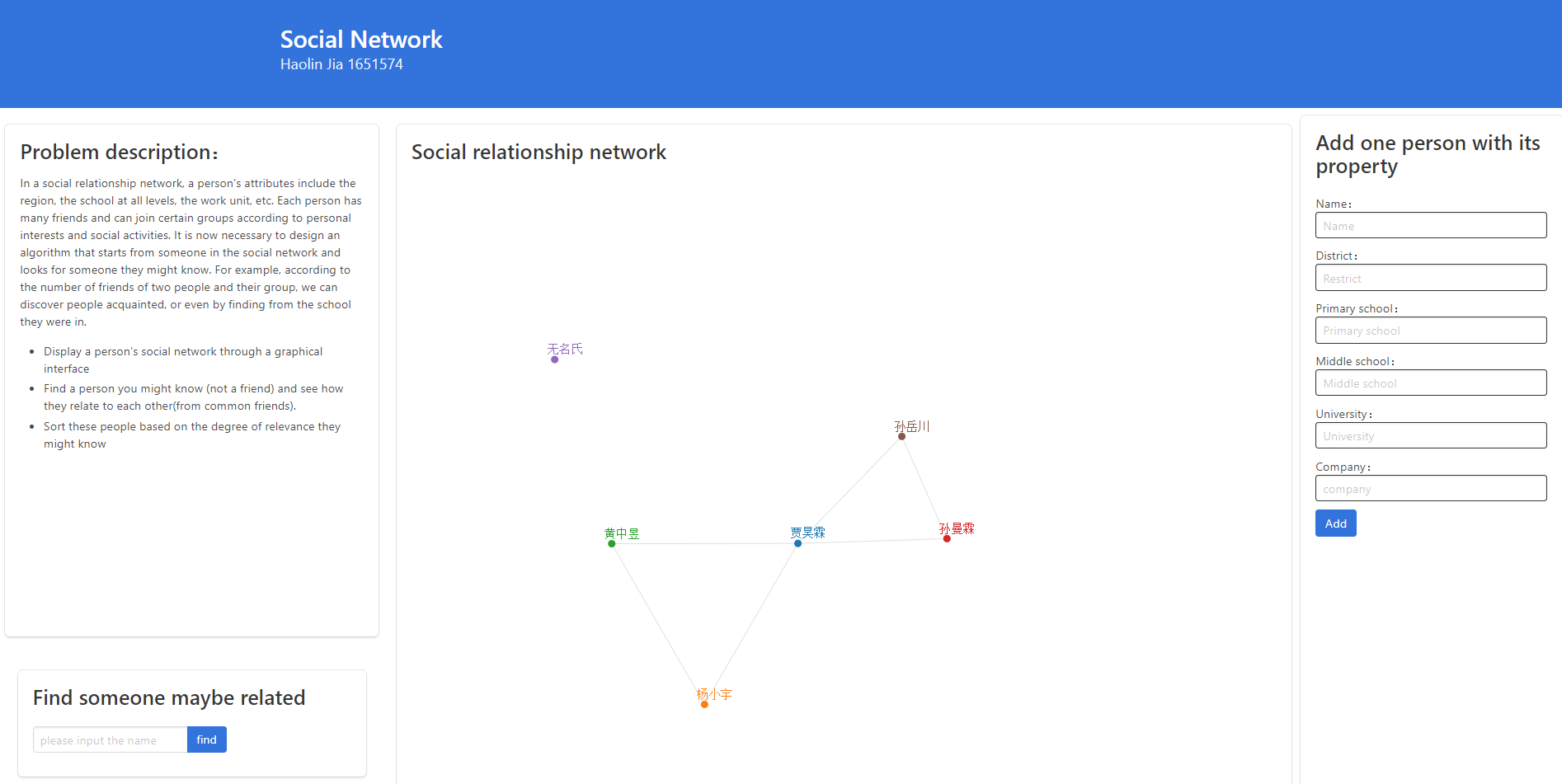
5 ★★★在某社会关系网络中，一个人属性包括所在地区、就读的各级学校、工作单位等，每一人有众多好友，并可以根据个人兴趣及社会活动加入到某些群组。现需设计一算法，从该社会关系网络中某一人出发，寻找其可能认识的人。例如根据两个人共同好友数量及所在群组情况，来发现可能认识的人；通过就读的学校情况发现可能认识的同学。

（1）通过图形化界面，显示某一人的社会网络

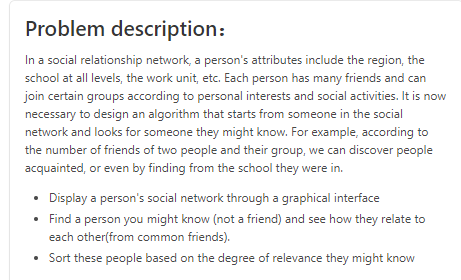
（2）寻找某一人可能认识的人（不是其好友），并查看这些人与其关联度（共同好友数）

（3）根据可能认识的关联度对这些人进行排序

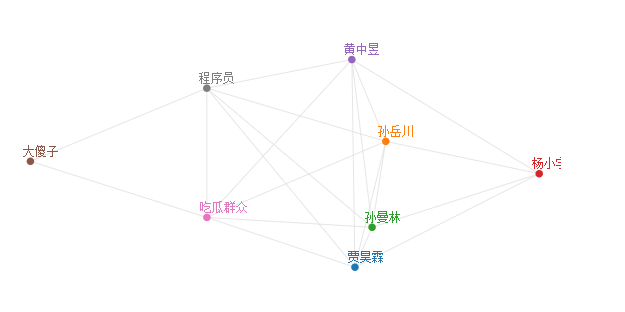
## 2.2 软件功能

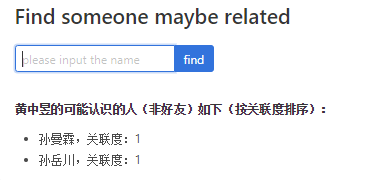


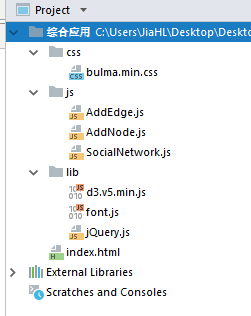
给出题目的英文提示



通过输入构建社会关系图，可视化并用相似度算法计算其相关度。





输入名称得到可能认识的人的关联度排序

## 2.3 逻辑结构与物理结构

### 2.3.1 工程组成

**整个文件目录由这些文件组成，其中：**

**bulma.min.css为css所需要用到的css排版库文件**

**AddEdge.js为增加边操作函数**

**AddNode.js为增加节点操作函数**

**SocialNetwork.js为d3库所需要调用的函数类**

**d3.v5为d3的第五版本，具体更新时间为2017年3月（由于前一个吸取了教训，选用了低版本导致很多函数无法实现，这个项目我选用了最新版本，并根据官方文档学习并灵活运用，构造出力导向图）**

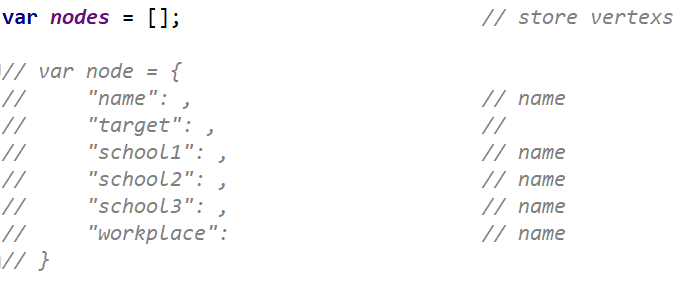
**font.js为额外调用的格式排版js文件**

**jQuery为jQuery库函数文件**

**index.html为主要网页构成HTML文件**

### 2.3.2数据结构

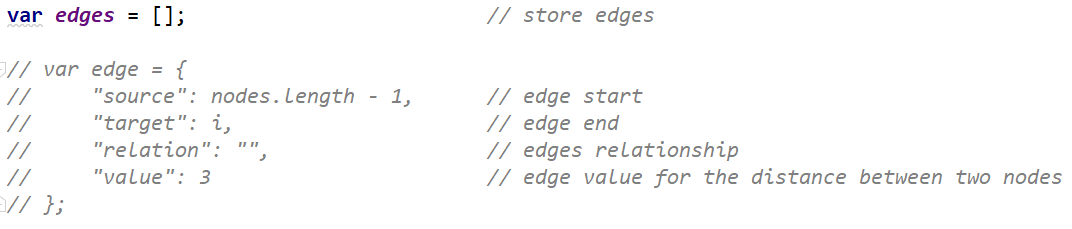
用到的数据结构如下图所示：



**顶点类包含6个属性，分别为：**

**姓名、家乡、小学、中学、大学、工作单位（设计灵感来源于学长github开源代码！学长用的C#编写：**

**（**[**https://github.com/LiuChangFreeman/SocialRelationshipNet**](https://github.com/LiuChangFreeman/SocialRelationshipNet)**）**



**边类包含4和属性：**

**分别为源点、指向点、关系字符串以及关系系数**

**其中关系字符串用来描述由这个点相连的两个定点的关系，而关系系数用来描述两个顶点的关系紧密度，关系系数越大，关系越紧密，算法由后面具体讲述**



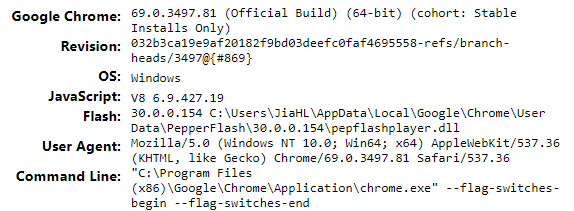
**用一个临界矩阵，来表示两两节点之间的相对系数关系，利用一个临界矩阵来表示两个点的关联，每个index表示点的序数，相应地值为边权值，即关联系数**

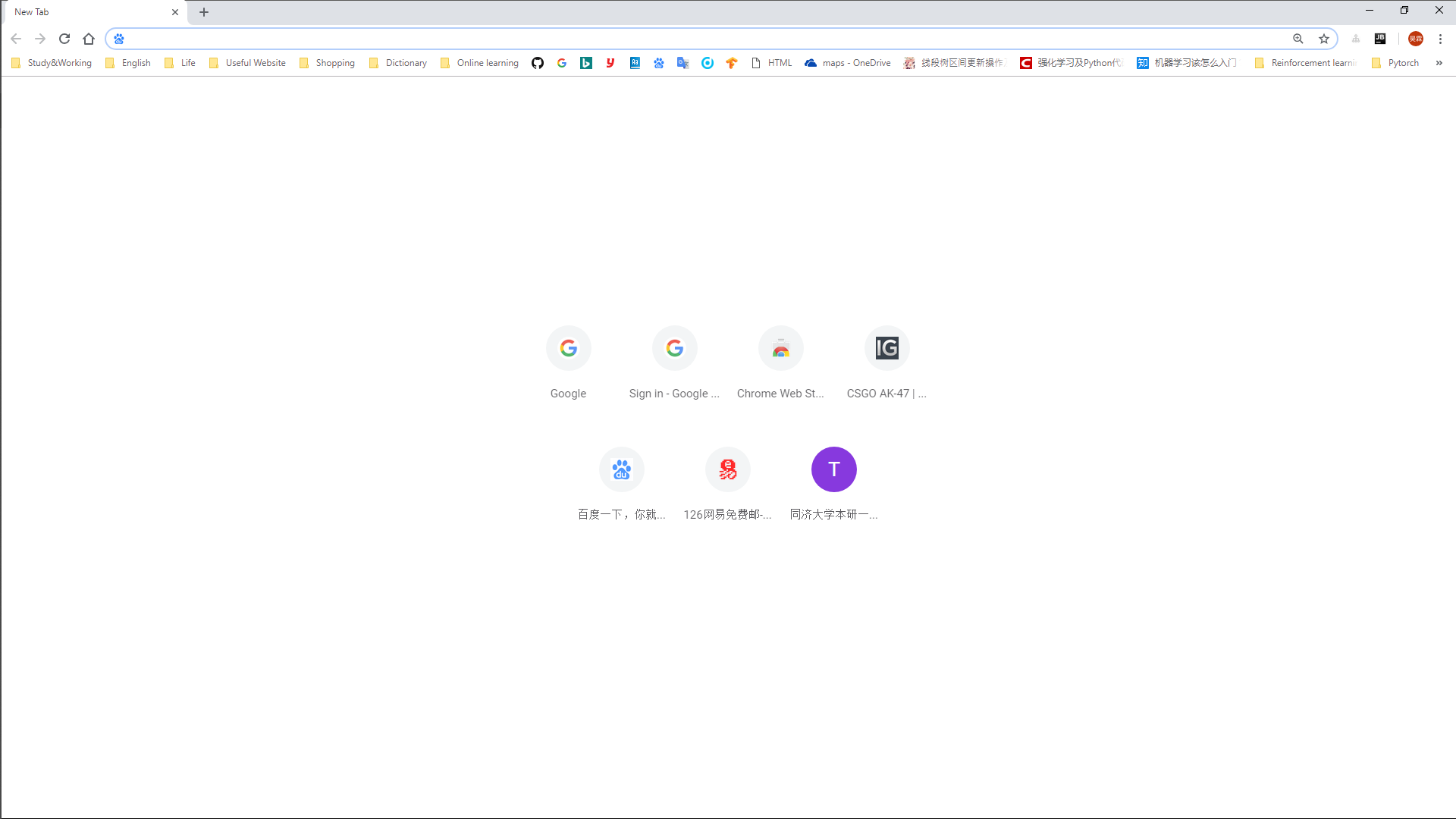
## 2.4 开发平台

Webstorm 2018.2



Chrome

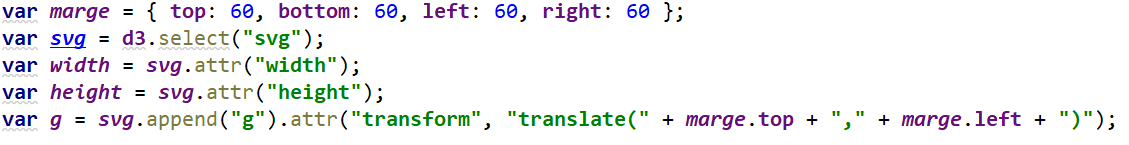




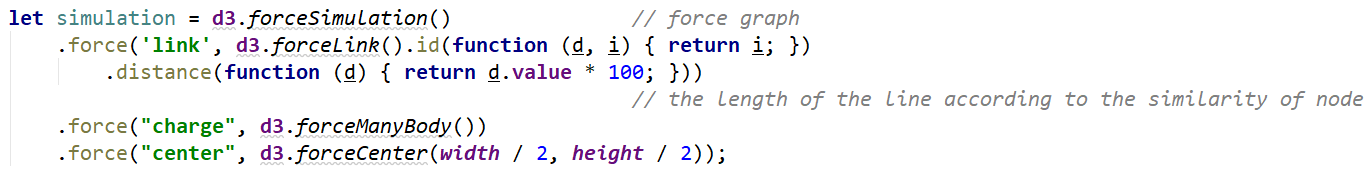
**仍旧与前一个相同**

## 2.5 系统的运行结果分析说明

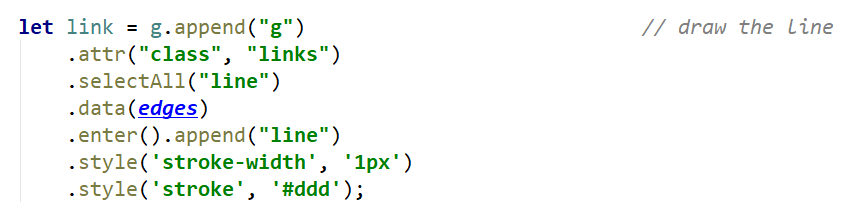
### 2.5.1 SocialNetWork.js:



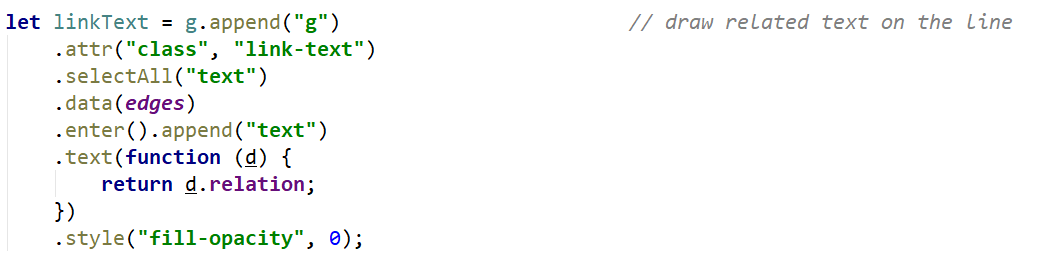
此为svg默认参数变量设置



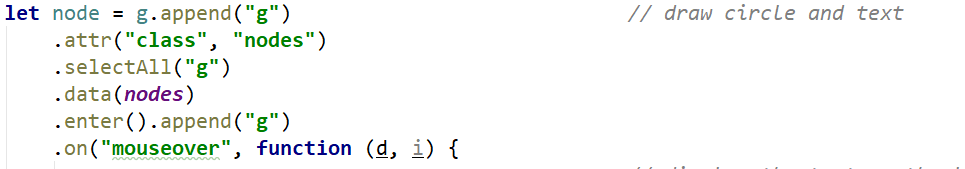
与d3\_v3不同，v4及其以上版本选用simulation 变量产生力导向图



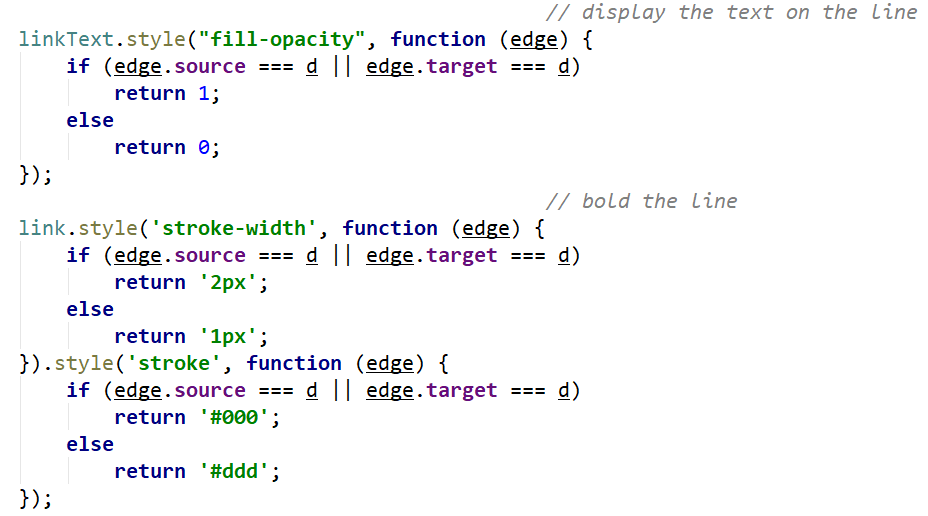
画边操作



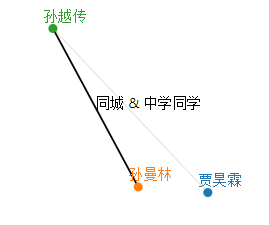
标出边上的文字操作



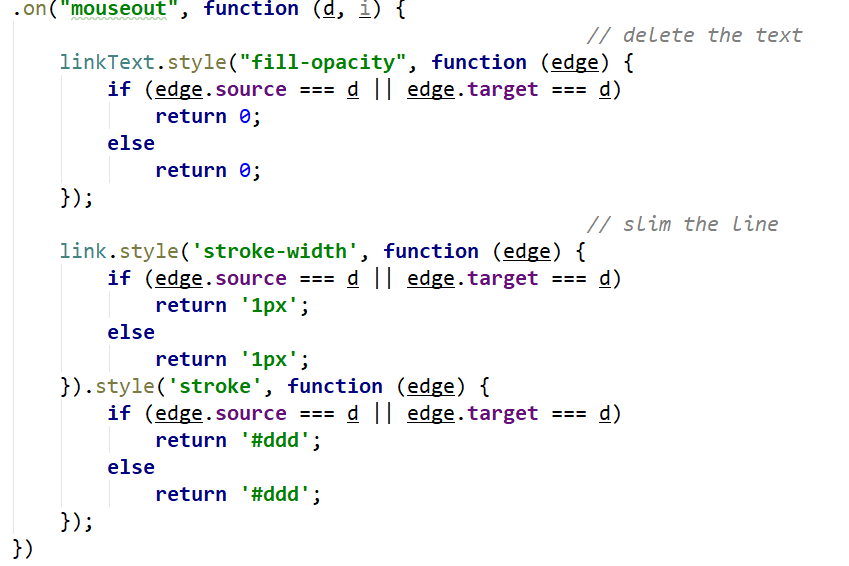
创建点操作，其中on函数为javascript常见的高阶函数，具体函数如下：



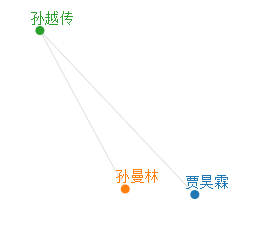
函数中由调用了三个html设置属性函数，分别为连线文字显示设置，即当前鼠标悬停节点显示边上文字、加粗相关连线，以及设置直线颜色，如下图所示：

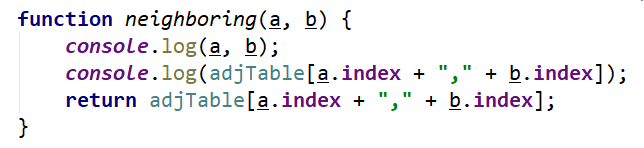


而当鼠标结束悬停，离开当前节点位置时：

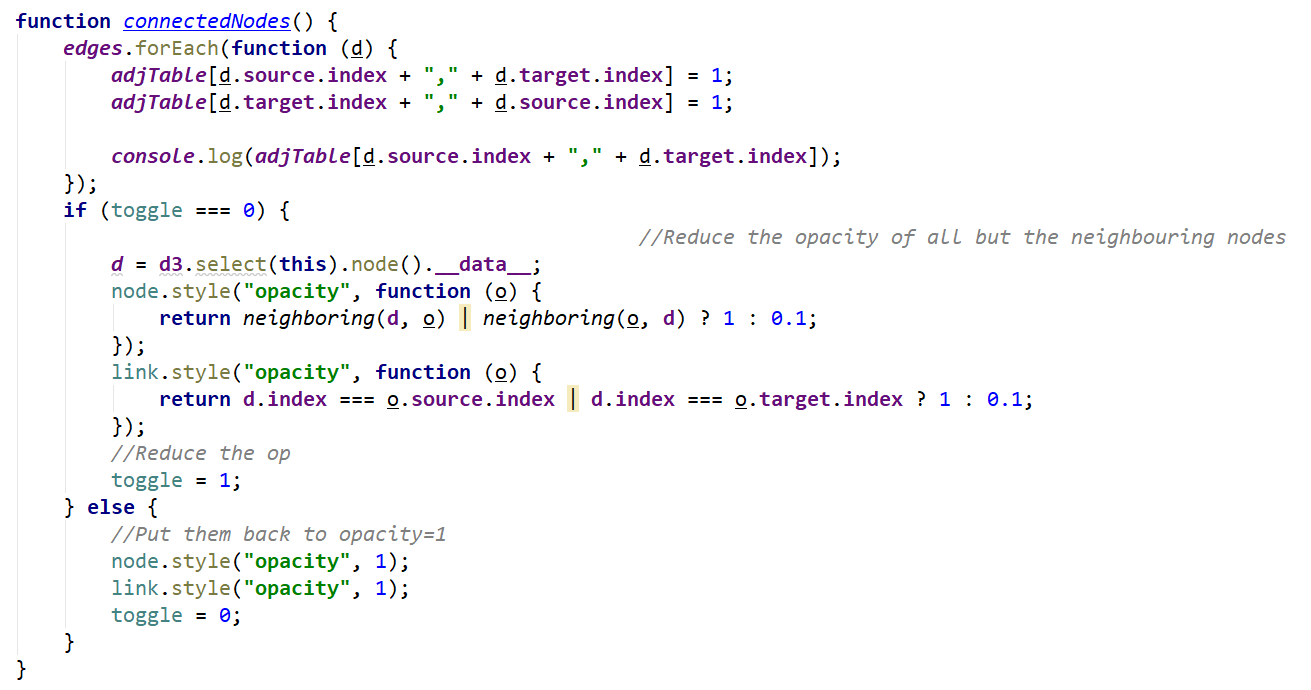


取消显示文字、取消连线加粗、取消连线颜色设置，如下图所示：

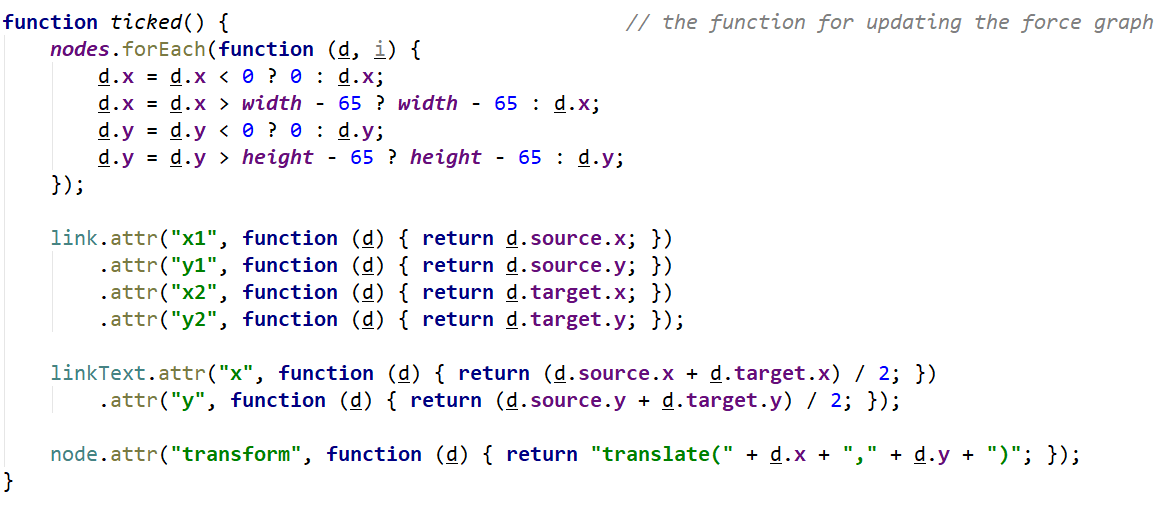




利用一个临界矩阵来表示两个点的关联，每个index表示点的序数，相应地值为边权值，即关联系数

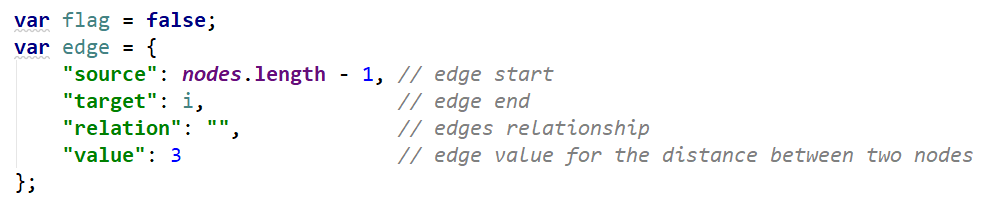


节点项链函数为关键，通过临界矩阵中两两节点所对应的下标值，即关联系数的大小，返回这两个点是否链接，或者链接线的长度。其中开关toggle变量表示透明值是否被设置为开或关

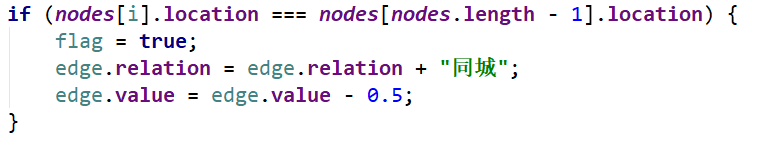


ticked 函数相同前一项目的刷新函数，故在此不过多叙述！

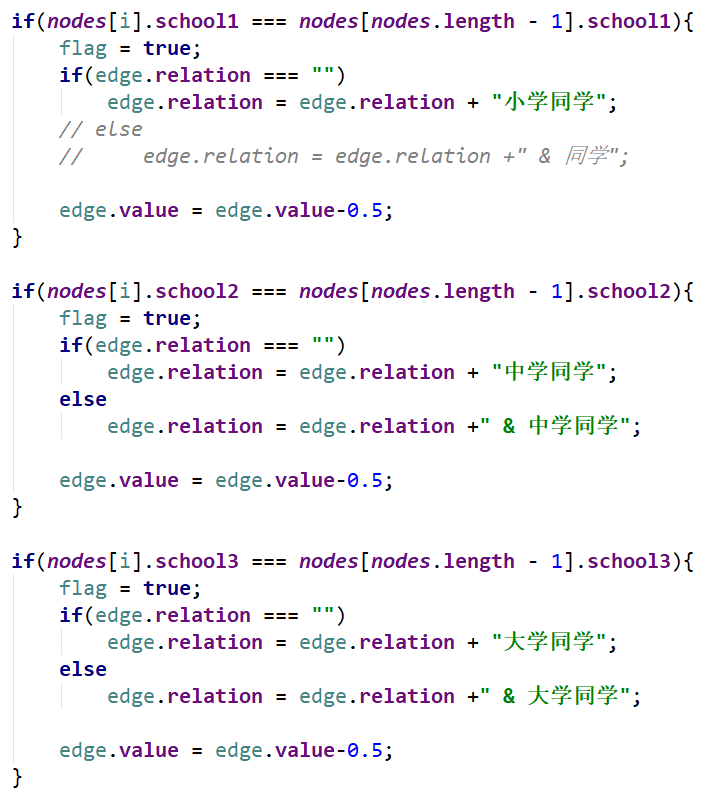
### 2.5.2 AddEdge.js:



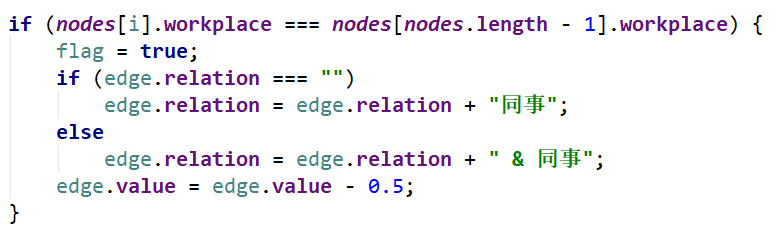
一个边的属性值如上图所示



根据是否同城在relation字符串中是否添加“同城”



根据输入的学校情况，分别依此判断是否同学

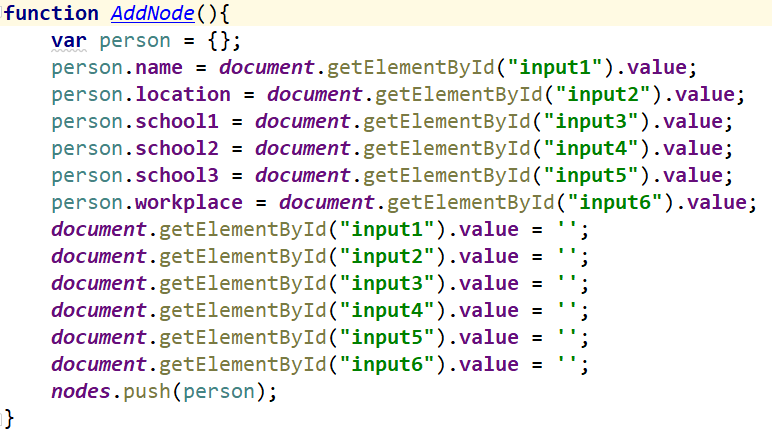


根据输入的就职公司情况，判断两人是否为同事关系



根据是否为有关系的，即relationship不为空串，加入flag变量

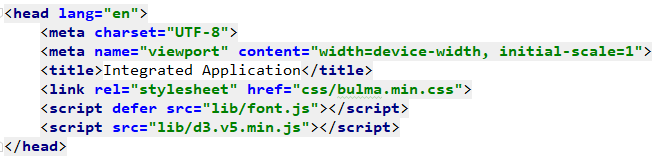
### 2.5.3 AddNode.js:



AddNode函数很简单，也很暴力，利用朴素javascript语句，由document得到节点input框中的值，从而得到value，添加到临时变量person中，之后再push到nodes数组中

### 2.5.4 index.html

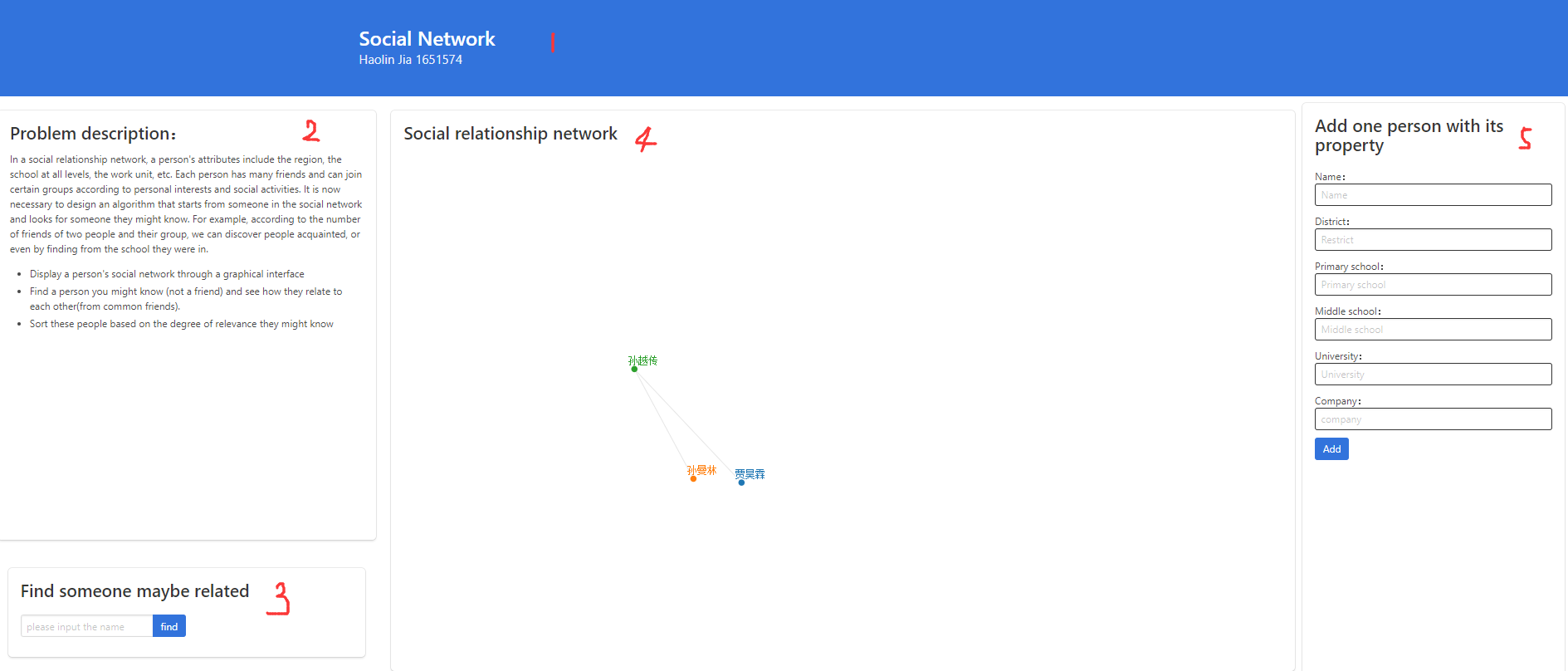
相当于C++中的主函数（我是这样理解的）



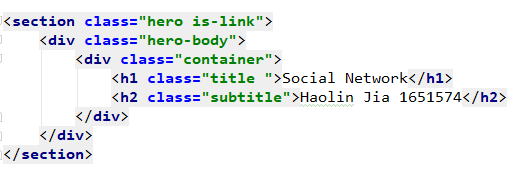
头中加入了一些引用，如css，jQeury，以及d3大数据可视化库等



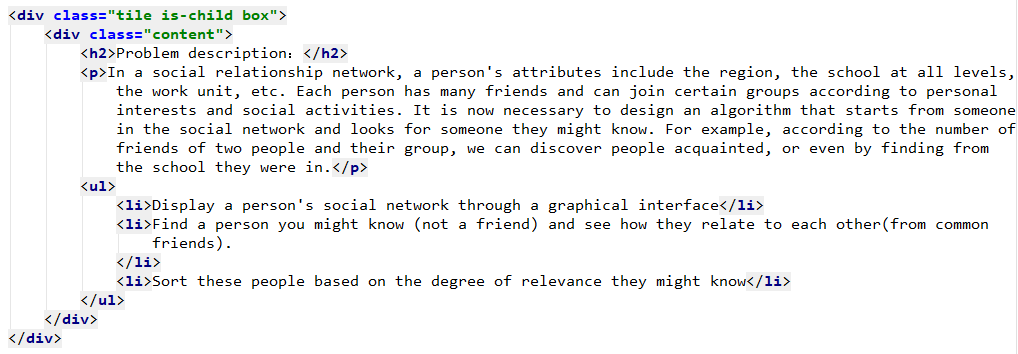
我将代码折叠后的html类标签如上图所示，由5个模块组成，分别为头、问题描述模块、输入模块、社会关系可视化模块、以及输入模块



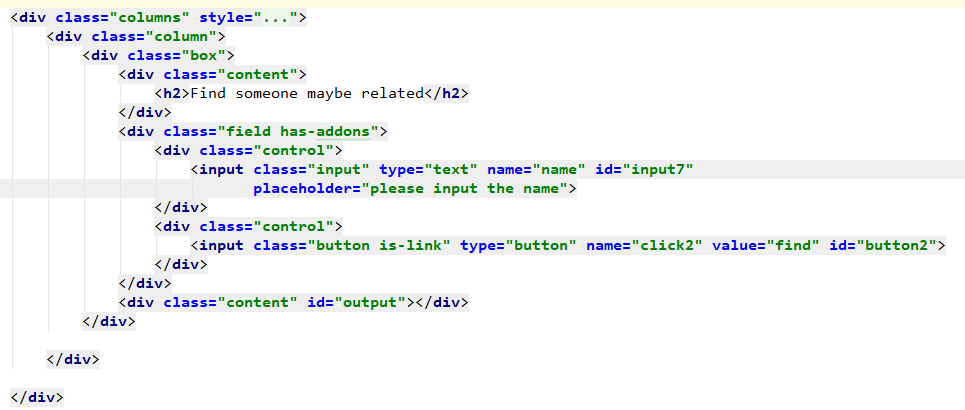
如上图所表示，根据学到的class 进行定位、排版，对比上一个工程中只是简单地罗列，这个工程我运用到了css排版的知识，因此更加美观



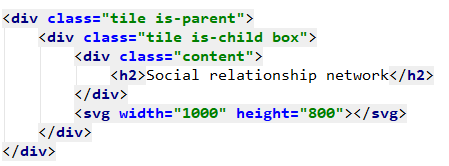
此为头类标签



此为问题描述类标签



此为寻找相似度输入类标签



此为大数据可视化类标签，注意此处的svg很好的限制了d3生成的力导向图的显示区域！

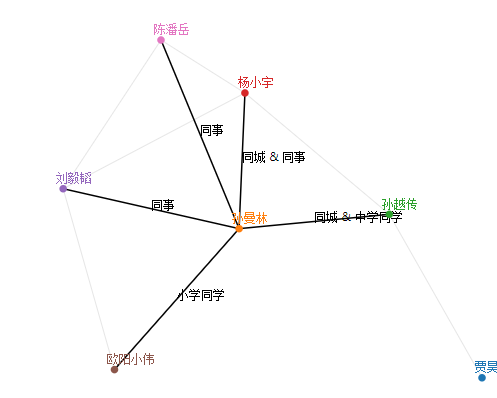
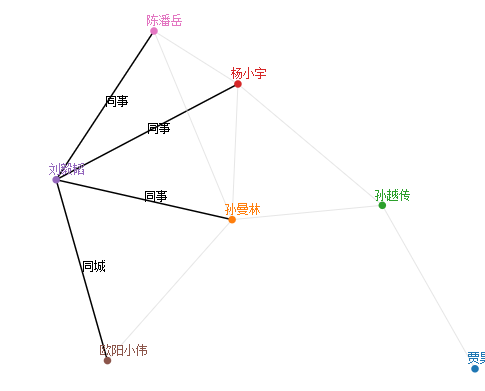


以及最后令人眼花缭乱但是非常容易实现的输入类标签组

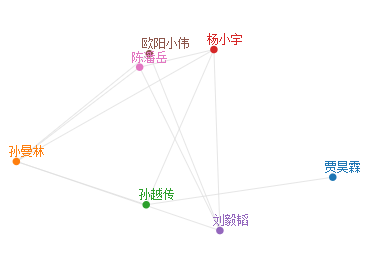
## 2.7 操作说明



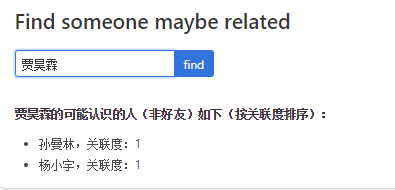
在右侧输入栏中输入相关信息，并点击Add按钮，即可添加一个节点的条目



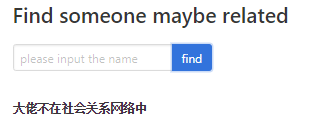
鼠标悬停在“刘毅韬”上，会显示其与周围节点的关系



也可随意拖动力导向图



输入“贾昊霖”可以返回其节点与周围节点的关系排序



在寻找框中键入“大佬”发现没有输入的人，即在节点组中不存在，会提示报错

# 第三部分 实践总结

## 3.1所做的工作

完成所有代码的编写，以及自学包括html,CSS,Javascript,JQuery，以及d3大数据可视化库的学习，还有bulma样式库的调用

## 3.2 总结及感悟

作为一个计算机科学与技术的学生，写底层、或搞科研本在我心中是第一要紧的，学习算法，运用高级的数据结构乃是所有程序员需要花大量时间的事情，然而我忽然觉得，发展方向不能太单一，不会写前端，看不懂html，不会javascript（网友们号称最好用的语言）毕业出去以后，也许有点丢人，因此我挤出了时间，在非常非常短的时间里学习了javascript，html，jquery，以及css样式库的语法规范等。更重要的是，在一个大数据的时代，数据可视化是关键，因此我又踏上了学习D3的不归路... 学习过程可谓十分艰辛！调试代码也遇到了各式各样的问题，通过同学尝试解决，但是有的直到现在仍没能解决，算作这两个工程留下的遗憾吧。

不过回首这几天，我学到了真的不少东西！我倍感欣慰，熬夜，值了！！！

# 第四部分 参考文献

[1] 廖雪峰. javascript教学[EB/OL]. https://www.liaoxuefeng.com/wiki/00143444 6689867b27157e896e74d51a89c25cc8b43bdb3000.

[2] 严蔚敏，吴伟民. 数据结构(C语言版)[M]. 北京：清华大学出版社，2007.

[3] 李春葆 等. 数据结构教程(第3版)上机实验指导[M]. 北京：清华大学出版社，2009.

[4] 王国钧，唐国民 等. 数据结构实验教程(C语言版)[M]. 北京：清华大学出版社，2009.

[5] cyber. D3力导向图实现[EB/OL]. https://www.cnblogs.com/koto/p/5983693.html

[6] 菜鸟教程. HTML5教程[EB/OL]. www.runoob.com/html/html-tutorial.html.

[7] 未知. Bulma官方文档[EB/OL]. https://bulma.io/.

[8] 未知. D3官方文档[EB/OL]. https://d3js.org/.

[9] 刘畅. 社会关系网络C#[EB/OL]. https://github.com/LiuChangFreeman/SocialR

elationshipNet..

参考说明：

[1]：普及了javascript的语法知识

[2]: 数据结构理论与算法伪码 参考

[3]、[4]：程序中涉及相关数据结构（循环链表、循环队列、串、内部排序）的算法实现 参考

[5]：学习掌握 d3的构建过程，以及用svg来绘图

[6]：学习HTML标签以及各个前端类极其属性

[7]：为了美化前端，选用朴素简单上手的css库

[8]: d3官方文档，权威且十分有参考价值！

[9]: 参考了学长的社会关系网络开源代码，学长是用C#写的，我吸取了一下经验，加入了我的想法实现了我的代码

注：所有代码、文档以及学习笔记均以push到我的github上：<https://github.com/Harrypotterrrr/Data-structure-course-project>