**图论大作业——豆瓣电影分析**

**说明文档**

软件72 何冠德 2017013566

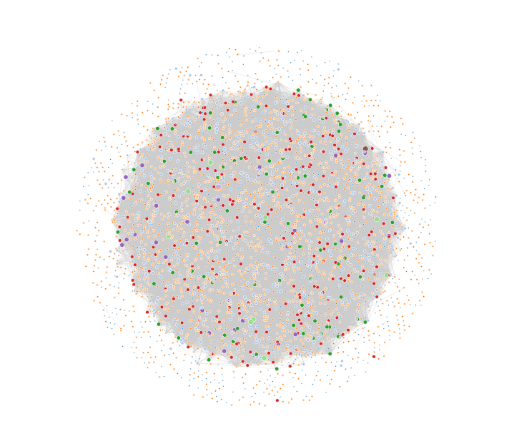
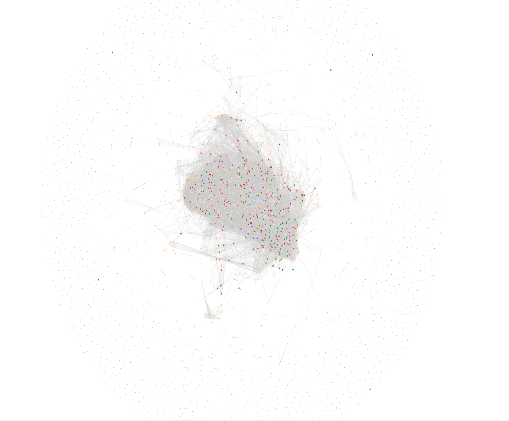
本次大作业主要分为两个部分完成，建图和算法可视化及交互。其中建图部分使用VC++完成，算法、可视化及交互均使用javascript完成，最终成品为一个简洁的网页，上面有简单的ui元素。本说明文档将从建图、算法实现两个主要方面，并在其中加入对具体问题的分析来介绍本项目。

**研究问题：电影与电影之间的相关性，用来进行电影的推荐。**

1. 建图

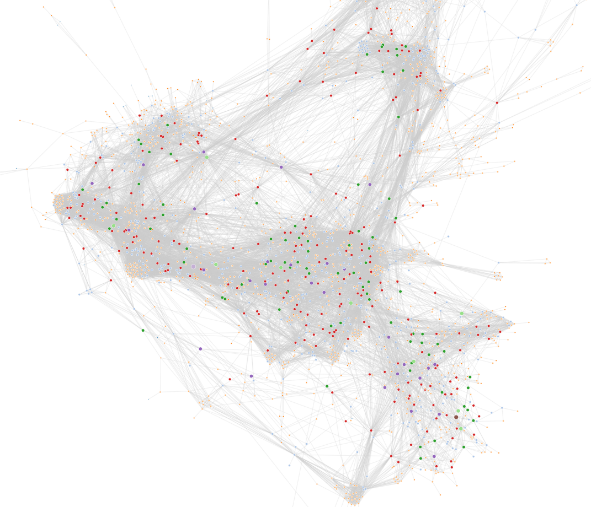
建图部分在C++中完成，根据助教所提供的豆瓣电影信息为数据源，其中以不同电影为节点，边的设置通过自己的设定规则，可以动态调整边权大小及图的稀疏程度，简要地说，一个电影结点有多种指标，例如类别、导演、主演、国家等，这些因素需要我们综合考量，在反复的迭代与重构之后，我选择将拥有3个共同因素的电影结点之间建成边，并考量一个电影的不同指标的权重（类别优先，指标数量多、权重小等规则），建成了一个无向图。并使用d3.js的力导向图布局进行了图的可视化，从图中我们可以获得一些信息。

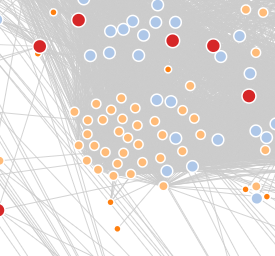
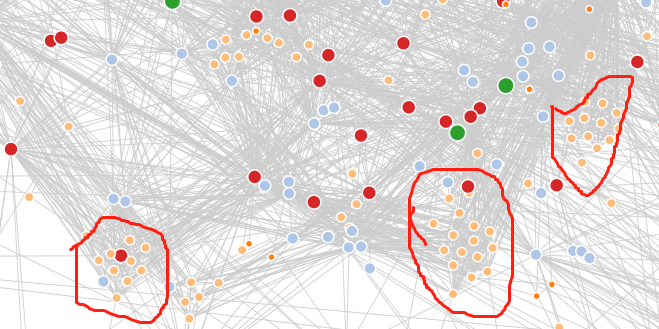
演示效果(4300结点、20000边)：



由于力导向图的作用，非连通结点渐渐向外扩散，最大连通分支开始伸展

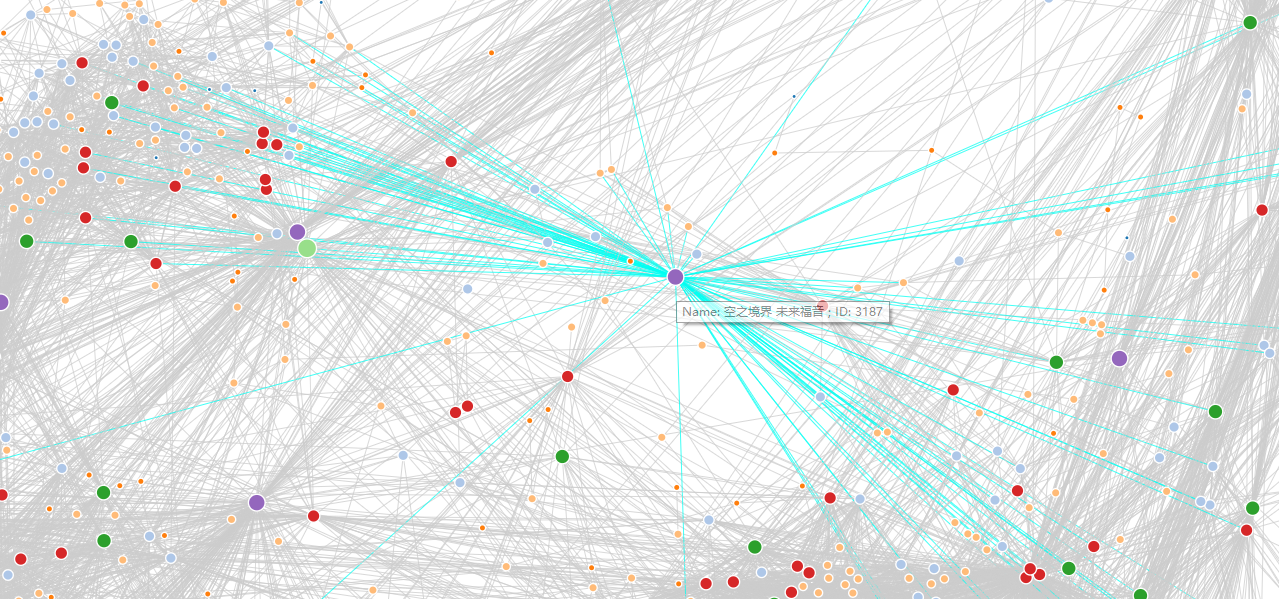
该图为原始图，中间的圆形点簇是最大连通分支所在，四周是离散点和联通分支。



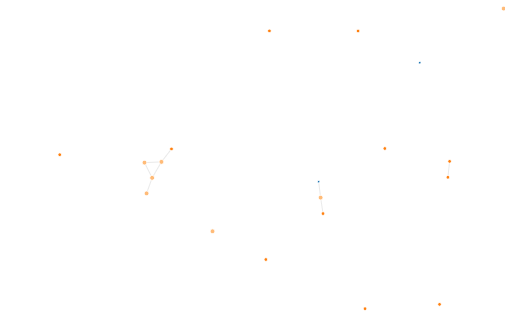


上图是已经大致伸展开的图，从中我们可以得到一些比较有用的信息。

在图的周围，会有许许多多的小节点簇存在，这些节点簇都是性质及其相似的电影，比如左图的结点簇中就包含了众多美国的系列电影（饥饿游戏、星球大战、星际迷航、分歧者），而右图中有恐怖电影簇，以及某些题材的…



在建图阶段，我也将一些结点附上了权重，这是较初步的判断，根据的是结点的所有指标的个数，个数越多，理论上说会更容易与其他结点建成边，我们可以看见，上图呈三角形的三个较大的紫色结点都连着很多的边，其中中间的电影《空之境界》所关联的边已经用浅蓝色标识出，是很庞大的数量，在之后的算法部分，结点的权重还会发生变化，变得和图本身的性质更加有关。



上图为被甩出去的较小连通分支，可以看出，它们的半径也较小，说明权重较小，不易与其他电影发生关联，这是本次建图作业中很难顾及到的部分。

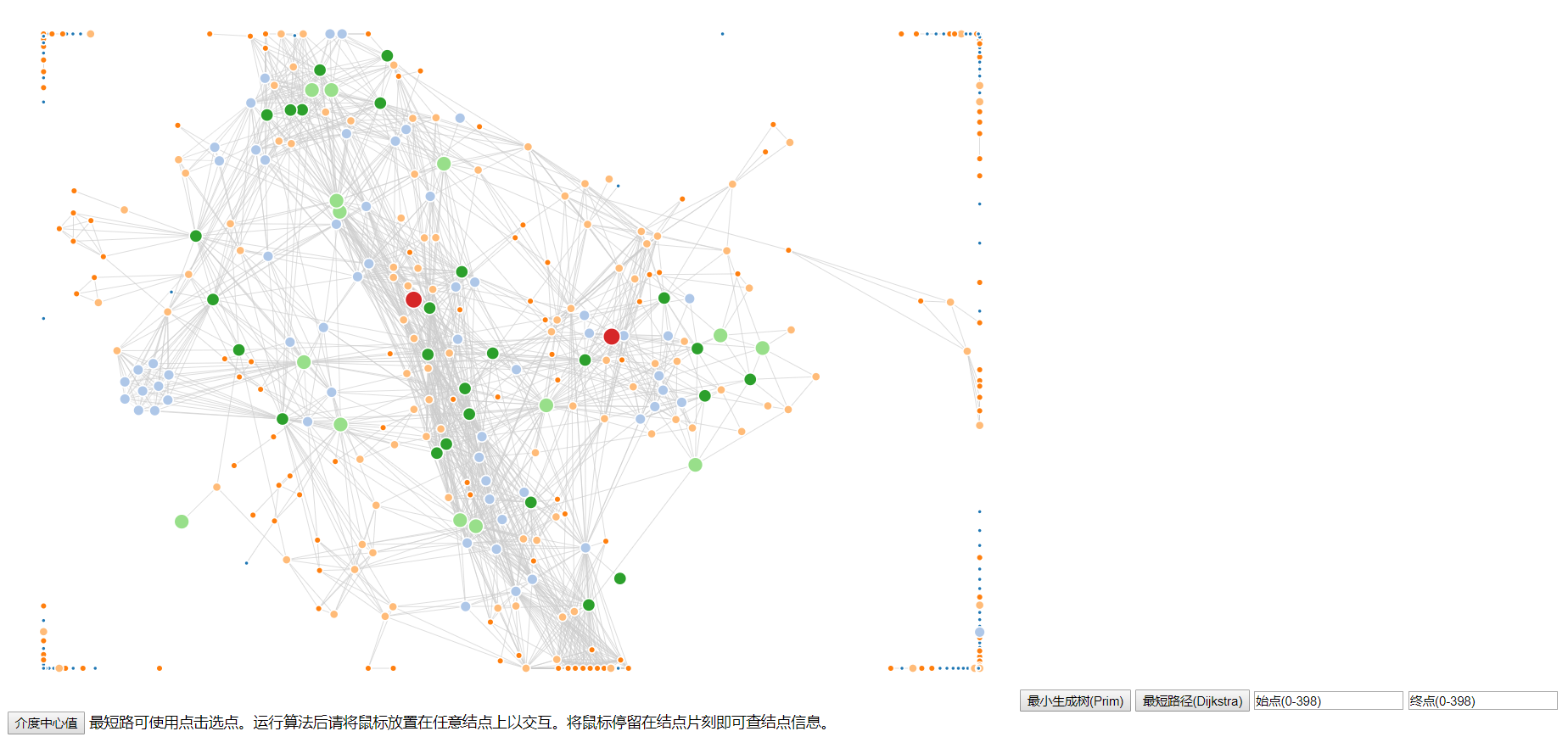
在边权的设置上，我也使用电影的指标来进行细化。性质相似的点之间边权就越短，这也就是为什么会形成结点簇的原因，它们之间的距离高度相似，在力导向图中反映出一簇，这也通常是系列电影会有的特征。

总结：在建图部分，通过对不同电影特征的提取和数值化，使得一连串数据能够形象的展示在图中，并让数据之间的相关性变得非常直观。在对这次问题的研究中，建图是最为重要的环节，它为之后的算法提供了一个平台，也是对于实际问题研究最为重要的一个步骤。

1. 算法

本次在算法层面上，我一共实现了三个算法，分别是最小生成树的Prim算法、最短路径的Dijkstra算法，和基于Folyd-Warshall并加入自己在时间复杂度优化的计算结点的介值中心度的算法。（测试环境：mac/win下visual studio code，使用live server插件加chrome调试运行）

具体说明（为了交互性和演示性，这里使用结点数为400、边为2000的数据规模）



（丑出天际的UI界面……）

算法1：Prim算法

算法描述：

1).输入：一个加权连通图，其中顶点集合为V，边集合为E；

2).初始化：Vnew = {x}，其中x为集合V中的任一节点（起始点），Enew = {},为空；

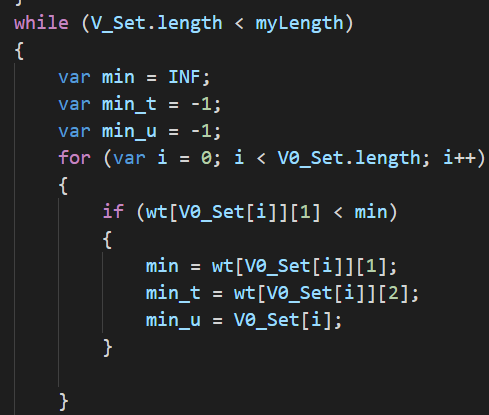
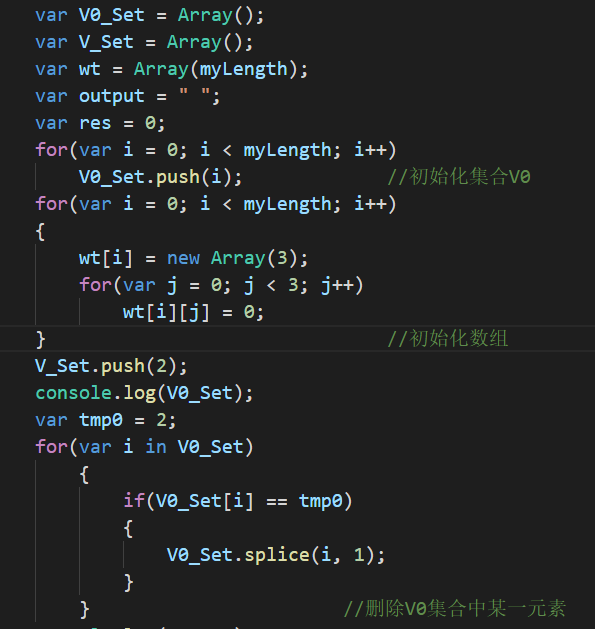
3).重复下列操作，直到Vnew = V：

a.在集合E中选取权值最小的边<u, v>，其中u为集合Vnew中的元素，而v不在Vnew集合当中，并且v∈V（如果存在有多条满足前述条件即具有相同权值的边，则可任意选取其中之一）；

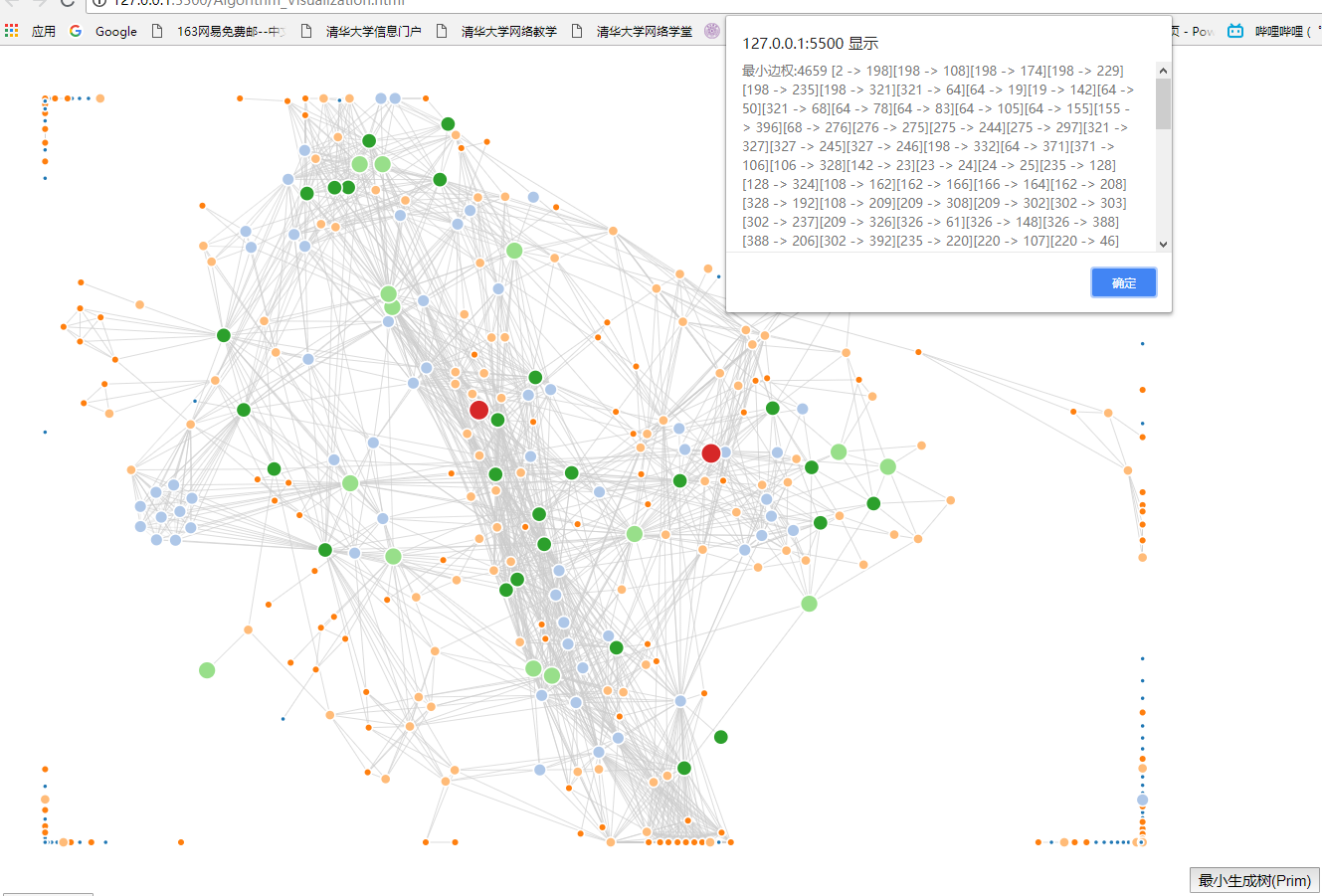
b.将v加入集合Vnew中，将<u, v>边加入集合Enew中；

4).输出：使用集合Vnew和Enew来描述所得到的最小生成树。

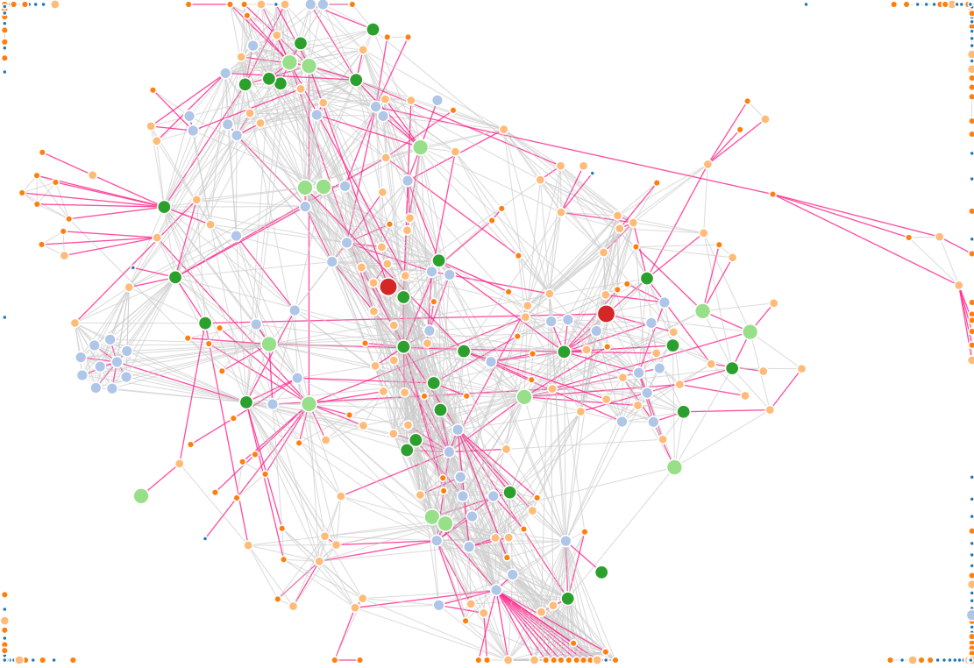
实现很简单，基本按照描述来，设置两个javascript中数组作为两个集合，

以上即为算法核心部分，完成算法后，使用鼠标放置或点击任意结点处触发刷新事件，从而对结果进行可视化。

具体演示：

点击后



可视化

实际研究意义：最小支撑树提供了整体的一个架构，它是从整体上的边权最小值来考虑，也就是说，沿着最小生成树去寻找电影，整体的相关性是很好的，这一点在寻找强相关性电影中具有整体的指导意义。当有时最短路仅仅包含一条边时，可以通过最小支撑树来去保证整体的强相关性。

算法2：Dijkstra算法

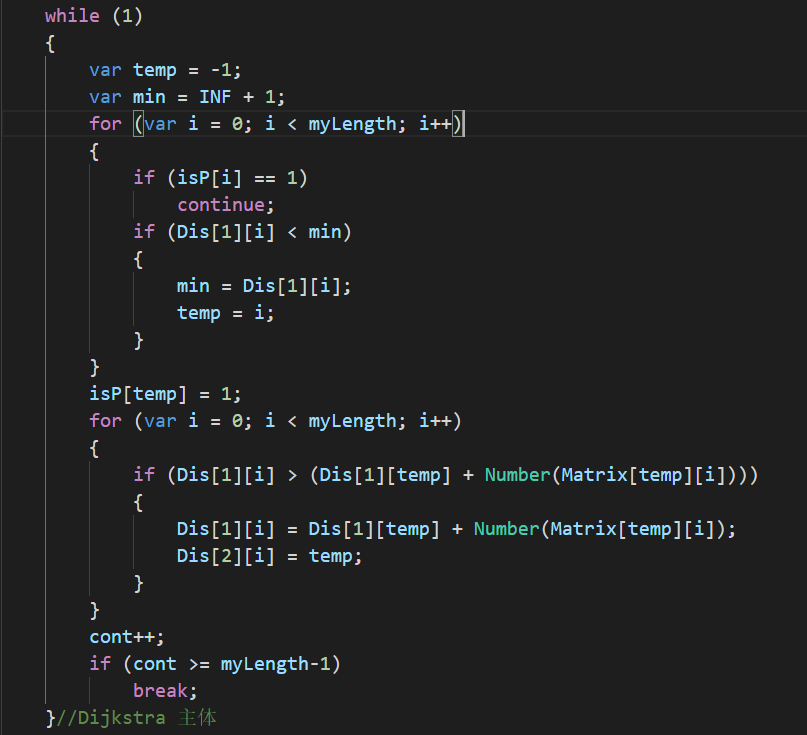
算法描述：

Dijkstra算法采用的是一种贪心的策略，声明一个数组dis来保存源点到各个顶点的最短距离和一个保存已经找到了最短路径的顶点的集合：T，初始时，原点 s 的路径权重被赋为 0 （dis[s] = 0）。若对于顶点 s 存在能直接到达的边（s,m），则把dis[m]设为w（s, m）,同时把所有其他（s不能直接到达的）顶点的路径长度设为无穷大。初始时，集合T只有顶点s。

然后，从dis数组选择最小值，则该值就是源点s到该值对应的顶点的最短路径，并且把该点加入到T中，OK，此时完成一个顶点，

然后，我们需要看看新加入的顶点是否可以到达其他顶点并且看看通过该顶点到达其他点的路径长度是否比源点直接到达短，如果是，那么就替换这些顶点在dis中的值。

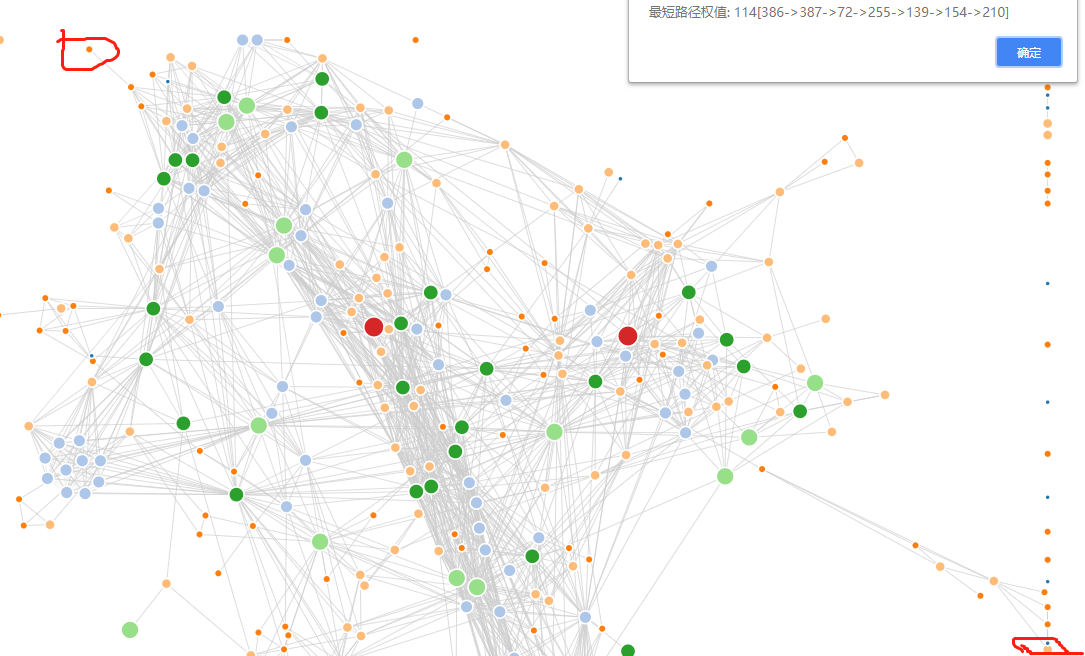
然后，又从dis中找出最小值，重复上述动作，直到T中包含了图的所有顶点。



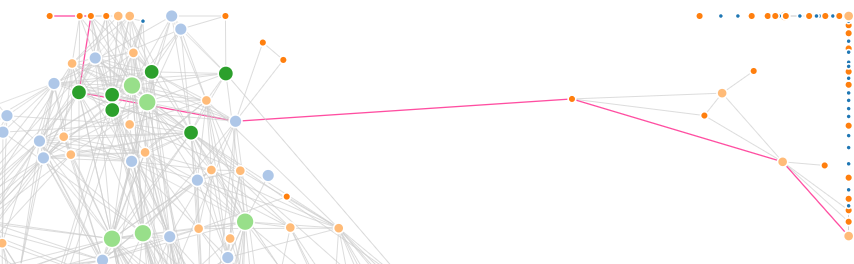
本算法可以通过点击或者输入来获取起点和终点，和Prim算法一样，在执行算法之后可以通过点击结点或者放置鼠标来刷新事件，实现可视化。

具体演示：

（执行算法后）



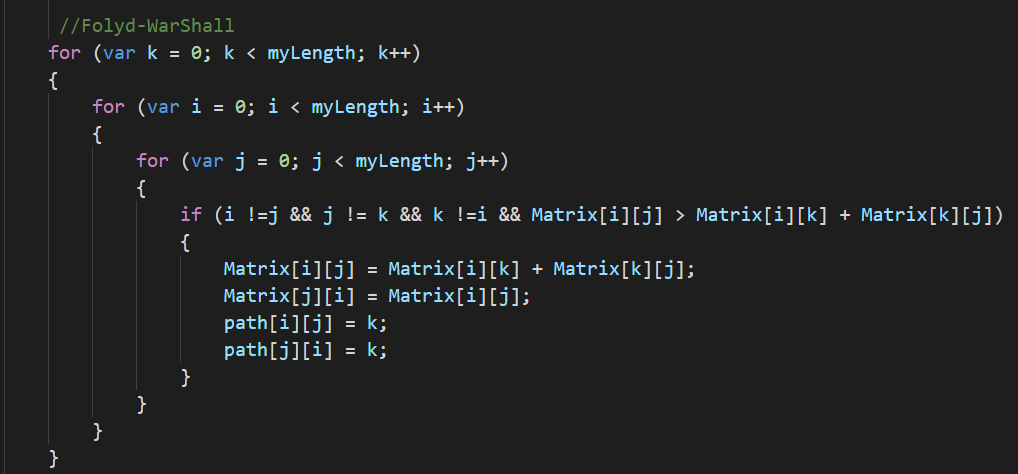
（可视化）



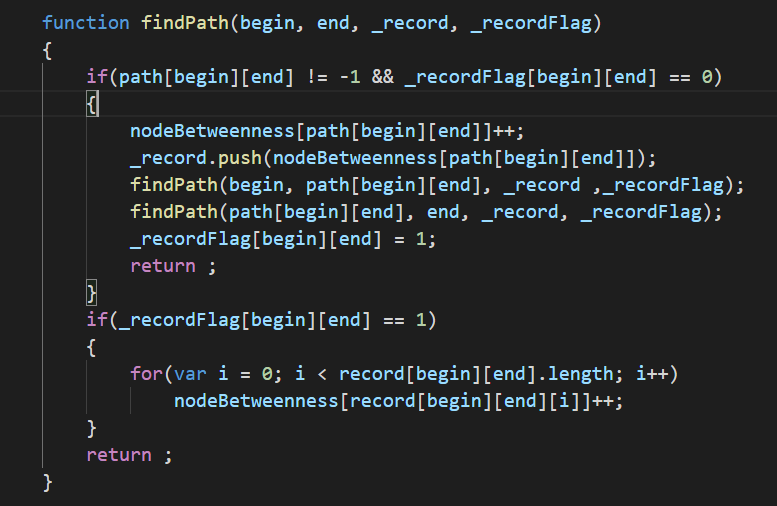
实际意义：最短路在研究电影之间的相关度之间有着十分重要的意义，由于边权越短，结点之间的相关度越高，比如当用户搜索过两个电影时，这时通过大数据，将两个结点间最短路径的电影作为广告或者推荐的权重增加，这就有助于提高广告投放的准确性，以及电影网站推荐系统的体验性，虽然简单，但是很有意义。

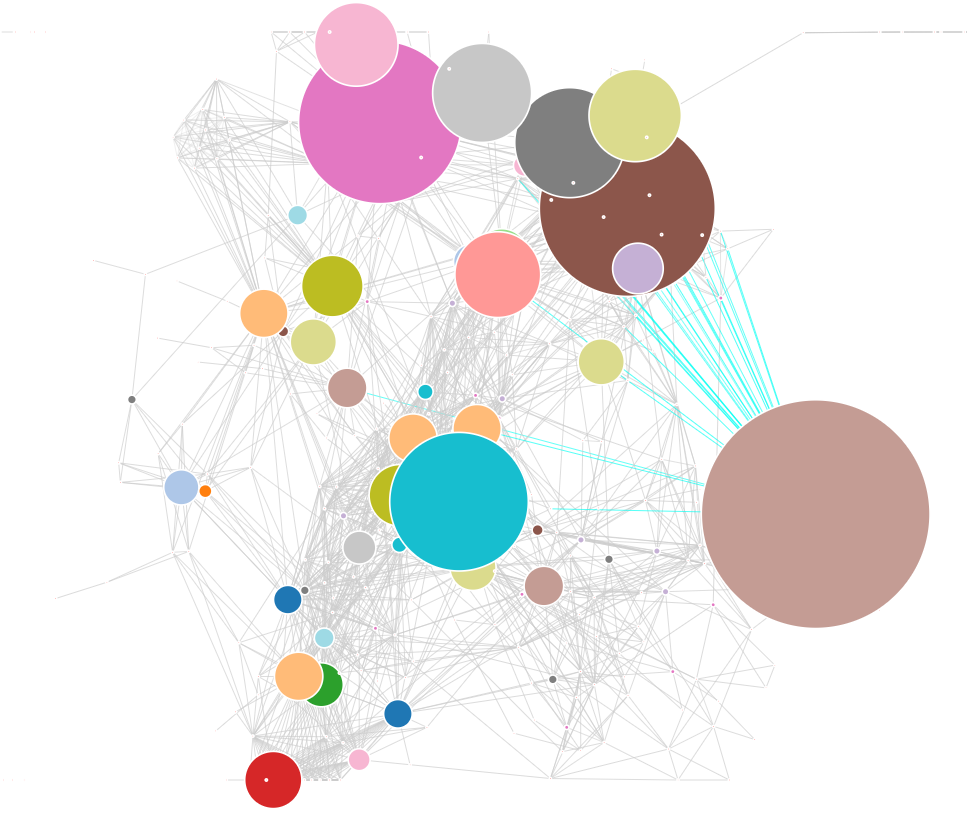
算法3：介值中心度

介值中心度(Betweenness centrality)对于我来说是一个全新的概念,它描述了一个结点位于所有结点之间最短路径间的次数，这是对于一个点的重要程度的一种描述方式。在本次作业中，我的算法主题是Floyd-Warshall算法，在时间复杂度上采用计划搜索，来提高效率，能更适应大规模图的计算。

（算法主体）

（计划搜索优化）



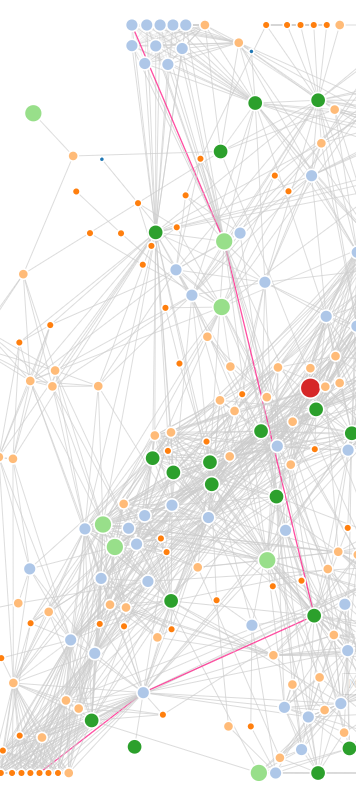


可视化（颜色偏棕，半径越大说明中心度越高）

实际意义：中心度作为衡量结点重要程度的另一个指标，可以辅助我们了解各个结点的性质，在建图时，我一开始使用的是结点的指标数量，但是有时数量并不代表一切，这就需要我们对结点其他性质进行了解，并进一步将结点的重要程度准确化，这就是求介值中心度的实际意义，它可以辅助我们，将图的建立进一步完善，将点的重要程度更加准确化，从而辅助我们分析整张图

1. 总结

在这次的问题研究中，我通过了图的建立，形象的得到了数据的一些特性，并可以将其进行合理的解释，在算法部分，最短路、支撑树可以直接应用于寻找相关电影。



例如：该情景中，是侠僧探案-》黄飞鸿-》间谍-》人皮脸

可以看出相似之处。

而我的不足之处在于算法还不够深入，没有深层挖掘进一步的信息，需要再深化图论、数据结构与算法的学习。

何冠德