

## 实验报告

### 代码整体思路

主函数就是先对 data.txt 的数据利用 ifstream 进行读取，然后利用读取的数据构建用户关系存储到我定义的一个类 SocialAnalysisNetworkSystem 中，构建关系的同时统计超级连接者相关情况，然后展示查询用户的朋友数量、查询两个用户之间的最短社交距离、查询网络中的“超级连接者”这三个基本功能。然后展示查询用户推荐朋友高级要求 b 的功能。

### 已实现的功能思路

类 SocialAnalysisNetworkSystem 中 private 里有一个存储朋友关系的 map。

成员函数 addUser 就是初始化添加对象。

成员函数 addFriendship 就是添加用户关系，就是直接互相 push\_back。

成员函数 getDirectFriendCount 就是获取直接朋友数量，就是直接获取朋友关系 map 的 size。

成员函数 getGapFriend 就是获取朋友的朋友如果不是自己的朋友就放到 set 中，利用 set 中无重复元素的性质获取间接朋友。

成员函数 getShortestDistance 就是利用广度优先算法找到二者间的最短路径，从用户开始，逐层扩展到用户的朋友，然后再扩展到这些朋友的朋友，以此类推，直到找到目标用户或者无法继续扩展为止。

成员函数 recommand 就是先找到和用户不是朋友的用户，然后算出不是朋友的用户和用户的共同好友数量从而得出最推荐的好友，遍历过程利用了数据都是按顺序输入的规律。

### 测试用例及结果

11	用户5的直接朋友数量为13，间接朋友数量为334。
5	用户777的直接朋友数量为18，间接朋友数量为153。
777	用户85的直接朋友数量为14，间接朋友数量为333。
85	用户47的直接朋友数量为2，间接朋友数量为345。
47	用户96的直接朋友数量为9，间接朋友数量为338。
96	用户120的直接朋友数量为4，间接朋友数量为454。
120	用户919的直接朋友数量为29，间接朋友数量为1080。
919	用户1485的直接朋友数量为47，间接朋友数量为1125。
1485	用户9的直接朋友数量为57，间接朋友数量为290。
9	用户4038的直接朋友数量为9，间接朋友数量为50。
4038	用户1024的直接朋友数量为85，间接朋友数量为960。
1024	#####
12	用户5和用户12的最短社交距离为2。
5 12	用户25和用户73的最短社交距离为1。
25 73	用户45和用户88的最短社交距离为2。
45 88	用户58和用户45的最短社交距离为2。
58 45	用户332和用户156的最短社交距离为2。
332 156	用户818和用户77的最短社交距离为7。
818 77	用户321和用户64的最短社交距离为2。
321 64	用户1245和用户2125的最短社交距离为3。
1245 212	用户2426和用户1658的最短社交距离为4。
2426 165	用户4034和用户1069的最短社交距离为5。
4034 106	用户5和用户777的最短社交距离为7。
5 777	用户33和用户221的最短社交距离为2。
33 221	#####
10	网络中的“超级连接者”为用户107
5	#####
365	向用户5推荐用户67，136，322。
87	向用户365推荐用户373，387，395。
54	向用户87推荐用户277，304，56。
17	向用户54推荐用户315，322，332。
211	向用户17推荐用户343，14，28。
985	向用户211推荐用户252，67，26。
2	向用户985推荐用户1703，1725，1236。
777	向用户2推荐用户14，41，144。
648	向用户777推荐用户828，805，823。
	向用户648推荐用户422，573，575。
	#####