数据结构期末Project

## 简要信息

设计并实现一个基本的**社交网络分析系统**，涉及对社交网络中的用户关系和特性的管理和分析，包括但不限于朋友关系的建立、社交距离的计算以及网络特征的分析。请注意所有代码必须使用C/C++语言编写。

提交截止日期：2023.12.31 23:59

## Project要求

1. 基本要求
2. 实现基础数据结构

* 设计记录用户列表的数据结构
* 设计记录用户关系的数据结构

1. 用户关系分析

* 查询用户的直接和间接朋友数量

用户的直接朋友是最短社交距离为1的用户，间接朋友是最短社交距离为2的用户。

* 计算两个用户之间的最短社交距离

确定任意两个用户之间的最短联系链长度。如果两个用户是直接的朋友，他们之间的社交距离就是1；如果他们不是直接朋友，但有共同的朋友，那么他们之间的社交距离就是2，以此类推。

1. 社交网络特性分析

* 计算网络中的“超级连接者”

“超级连接者”即拥有最多直接朋友的用户。

1. 高级要求
2. 高级网络分析算法

* 分析网络的平均聚集系数（Clustering Coefficient）和网络中朋友三角（即三个人互为朋友）的数量，并进行时间复杂度和空间复杂度分析。

平均聚集系数代表所有节点的局部聚集系数（无向图）的平均值，一个节点的局部聚集系数为该节点的邻居之间实际存在的边数除以可能存在的最大边数。假设有一个社交网络图，包括5个节点，其连接如下：节点1与节点2、节点3相连。节点2与节点1、节点3、节点4相连。节点3与节点1、节点2相连。节点4与节点2、节点5相连。节点5与节点4相连。首先计算节点1的聚集系数。节点1有2个邻居，在这两个邻居之间有1条边（节点2和节点3相连）。节点1的邻居之间可能有1条边。因此，节点1的聚集系数为。依次类推，节点2，3，4，5的聚集系数分别为：1/3，1，0，0。所以该社交网络的平均聚集系数为7/15，约等于0.47。

1. 优化和扩展

* 实现用户推荐功能，并进行时间复杂度和空间复杂度分析。

针对任一用户，基于共同朋友数量，向其推荐三个最有可能的潜在朋友。例如，用户A与用户B拥有最多共同好友，但用户B并不是用户A的直接朋友，则向用户A推荐用户B。

## 实现细节

1. 社交数据

社交数据（data.txt）随Project下发,以无向图的边形式给出,可自行决定如何解析并构建相应结构。

1. 代码结构

代码应直接对数据文件进行操作而非从屏幕进行读入，并且有相应的输出文件。对于基础要求b)/高级要求b)等需要额外输入的，请自行设计测试用例，记录于sample.txt，并随代码文件提交，测试用例在10组左右为宜，但不需要提交下发的社交数据文件。

代码的所有输出应在同一个output.txt文件内，并包含必要的文字说明输出的是什么，输出可以使用中文。

我们将会下发sample.txt和output.txt的样例。请注意sample.txt和output.txt中的数据均为捏造，仅有格式可供参考。

1. 正确示例

我们提供基础要求b）和高级要求b）的正确输入和输出示例，以便测试程序正确性。

1. 查询用户的直接和间接朋友数量

输入：5

输出：用户5的直接朋友数量为13，间接朋友数量为334。

1. 计算两个用户之间的最短社交距离

输入：5 12

输出：用户5和用户12的最短社交距离为2。

1. 用户推荐

输入：5

输出：向用户5推荐用户67，136，322。

## 提交要求

作业通过elearning进行提交，提交文件应为一个文件名为学号，后缀名为rar或者zip的压缩包，压缩包示例结构如下：

student-id.zip

- report.pdf

- code

main.cpp

sample.txt

report.pdf长度不宜超过10页，正文字号行距等格式以清晰易读为佳，内容包括解释整体实现思路、介绍已实现功能的思路、**展示测试用例及结果**、描述实现过程中收获（非必须）。仅有报告，无代码/抄袭代码/乱写代码零分处理。

在code文件夹中，所有代码文件必须包含必要的中文注释，所有文本文件请使用UTF-8编码提交。

在完成PJ过程中若参考了互联网资料或获得了同学的帮助，请在报告的末尾注明，如未注明但在评分时发现有直接抄袭互联网资料或其他同学提交的代码等情况，零分处理。

## 评分标准

完成基础要求可得基础分数，完成基础要求以及**任一高级要求**可得全部分数。除此之外，代码完备性更高者（例如，考虑错误处理）可酌情加分。但在已经完成上述所有要求获得全部分数的情况下，不会有额外加分。

其他评判标准**（请仔细逐条阅读！！！）**：

1. 对于提交代码/报告存在雷同或大部分雷同的，双方均直接零分。
2. 提交文件不全或不符合要求，包括没有代码、没有测试用例、没有报告，没有单独存放代码的code文件夹，报告不是pdf格式等，酌情扣分或零分。请注意，为了保证格式一致，**请不要提交doc或其他格式**，可以使用pandoc等工具进行相应转换。
3. 评分时会在Linux环境下（WSL）使用gcc9在code文件夹下直接编译，编译命令为 g++ main.cpp -o main -Wall -std=c++11，然后将文件data.txt拷贝到code文件夹下后，运行./main，然后查看output.txt文件的输出是否与要求和报告的说明相符，如不能正常编译运行程序，或程序不能正常读入、输出相应文件，均会有较大程度的扣分。另外，在-Wall编译下，如编译器有Warning报出，同样会酌情扣分，请解决编译器报出的所有问题。
4. 在代码能正确编译运行，并按要求正确的前提下，报告能说明整个程序的设计思路即可获得全部分数，请不要在报告中加入冗长的说明或复制大段的代码，一页报告能说清楚，分数和10页报告不会有区别。
5. 不限制使用C++标准库，但禁止使用任何第三方库，使用第三方库的定义包括直接引入第三方库的源代码、调用编译好的二进制文件等。