|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | | |  |
|  | | |
|  | | |
|  | | | |
|  | **金融风控的资金流水分析**  **决赛赛题说明** | |  |
|  | |
| **发布日期** | **2020-01-17** |
|  | |

目 录

[1 题目说明 3](#_Toc30181630)

[1.1 输入信息 3](#_Toc30181631)

[1.2 输出信息 4](#_Toc30181632)

[1.3 限制条件 4](#_Toc30181633)

[1.4 接口说明 5](#_Toc30181634)

[1.5 排分规则 5](#_Toc30181635)

[1.6 其它说明 6](#_Toc30181636)

[2 运行环境 7](#_Toc30181637)

# 题目说明

通过金融风控的资金流水分析，可有效识别每个账号的位置关键中心性，辅助公安挖掘洗钱组织，帮助银行预防信用卡诈骗。基于给定的资金流水，精确计算每个账号的位置关键中心性，并输出TOP 100的账户信息。结果准确，用时最短者胜。

[1.1 输入信息](#_ZH-CN_TOPIC_0220760553)

[1.2 输出信息](#_ZH-CN_TOPIC_0220760554)

[1.3 限制条件](#_ZH-CN_TOPIC_0220760555)

[1.4 接口说明](#_ZH-CN_TOPIC_0220760556)

[1.5 排分规则](#_ZH-CN_TOPIC_0220760557)

[1.6 其它说明](#_ZH-CN_TOPIC_0220760558)

## 输入信息

输入为包含资金流水的文本文件，每一行代表一次资金交易记录，包含本端账号ID, 对端账号ID, 转账金额，用逗号隔开。

* 本端账号ID和对端账号ID为一个32位的无符号整数
* 转账金额为一个32位的无符号整数
* 转账记录最多为250万条
* 账号A给账号B最多转账一次

举例如下，其中第一行[1,2,100]表示ID为1的账户给ID为2的账户转账100元：

1,2,100

1,3,100

2,4,90

3,4,50

4,1,95

2,5,95

5,4,90

4,6,30

6,7,29

7,4,28

## 输出信息

输出信息为一个文件，包含TOP 100的账户信息:

从第一行开始，每一行的信息为账户ID和位置关键中心性（四舍五入，有效位数保留小数点后3位），用逗号分开。总体按照位置关键中心性的降序输出(先比较位置关键中心性再精确到小数点后3位)；如果位置关键中心性相同，则按照账户ID字典序（ID转为无符号整数后）升序输出。请注意：位置关键中心性为0，且能排到TOP100时，也需要输出。

举例如下：

4,25.000  
1,14.000  
2,5.000  
6,5.000  
7,5.000  
3,3.000  
5,0.000

## 限制条件

位置关键中心性的定义如下：



为账号 i的位置关键中心性。



: s🡪t 最短加权有向图路径的条数。



: s🡪t 最短加权有向图路径里面通过账号 i的条数。



|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | t= 1 | t = 2 | t = 3 | t = 4 | t = 5 | t = 6 | t = 7 |
| s = 1 | - | 0/1 | 0/1 | - | 0/1 | 1/1 | 1/1 |
| s = 2 | 1/1 | - | 1/1 | - | 0/1 | 1/1 | 1/1 |
| s = 3 | 1/1 | 1/1 | - | - | 1/1 | 1/1 | 1/1 |
| s = 4 | - | - | - | - | - | - | - |
| s = 5 | 1/1 | 1/1 | 1/1 | - | - | 1/1 | 1/1 |
| s = 6 | 1/1 | 1/1 | 1/1 | - | 1/1 | - | 0/1 |
| s = 7 | 1/1 | 1/1 | 1/1 | - | 1/1 | 1/1 | - |



|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | t = 1 | t = 2 | t = 3 | t = 4 | t = 5 | t = 6 | t = 7 |
| s = 1 | - | 0/1 | 0/1 | 0/1 | 0/1 | - | 1/1 |
| s = 2 | 0/1 | - | 0/1 | 0/1 | 0/1 | - | 1/1 |
| s = 3 | 0/1 | 0/1 | - | 0/1 | 0/1 | - | 1/1 |
| s = 4 | 0/1 | 0/1 | 0/1 | - | 0/1 | - | 1/1 |
| s = 5 | 0/1 | 0/1 | 0/1 | 0/1 | - | - | 1/1 |
| s = 6 | - | - | - | - | - | - | - |
| s = 7 | 0/1 | 0/1 | 0/1 | 0/1 | 0/1 | - | - |



## 接口说明

* 选手需要在代码中读取输入文件/data/test\_data.txt，分析后将结果输出到/projects/student/result.txt
* C语言的编译命令为：gcc -O3 main.c -o test -lpthread -fpic
* C语言的运行命令为：./test
* C++语言的编译命令为：g++ -O3 main.cpp -o test -lpthread -fpic
* C++语言的运行命令为：./test
* Python语言的运行命令为： python3 ./Main.py
* JAVA语言的编译命令为：javac Main.java
* JAVA语言的运行命令为：java Main

## 排分规则

* 结果准确，用时最短者胜出。
* 如果时间一样，先提交的选手排名靠前。
* 选手成绩取个人多次提交里面的最好成绩。

## 其它说明

* 只能使用Python标准库和numpy，其中判题程序使用的是numpy 1.17.2版本。

numpy 1.17.2在Euler OS上的安装方法为：

1. 安装python3-devel

wget <https://developer.huawei.com/ict/site-euleros/euleros/repo/yum/2.8/os/aarch64/updates/python3-devel-3.7.0-9.h15.eulerosv2r8.aarch64.rpm>

rpm -ivh python3-devel-3.7.0-9.h15.eulerosv2r8.aarch64.rpm --nodeps

1. 修改pip的安装源

创建如下的文件~/.pip/pip.conf，并在文件中添加如下配置，保存并退出

[global]

index-url = <https://repo.huaweicloud.com/repository/pypi/simple>

trusted-host = repo.huaweicloud.com

timeout = 120

1. 安装numpy 1.17.2

pip3 install numpy==1.17.2

* JAVA代码请使用UTF-8编码。
* ID为32位的无符号整数，转账金额为32位的无符号整数，且都小于2的31次方，选手可以不用考虑范围之外的输入。
* 不能使用网络相关的接口调用，如Python里面的multiprocessing.Manager接口，否者会出现运行错误。
* 打开输入文件时，要使用只读方式打开文件，否者打开文件会失败。
* 通过open创建文件时需要指定文件权限，将open(fileName, O\_RDWR | O\_CREAT)改为open(fileName, O\_RDWR | O\_CREAT，0666)，否则会导致编译失败。

# 运行环境

* 选手使用的练习资源：8U16G
* 判题系统使用的判题资源：8U16G
* 操作系统：Euler OS
* 服务器：TaiShan服务器
* 芯片：鲲鹏920