1034. Head of a Gang (30)

时间限制

100 ms

内存限制

65536 kB

代码长度限制

16000 B

判题程序

Standard

作者

CHEN, Yue

One way that the police finds the head of a gang is to check people's phone calls. If there is a phone call between A and B, we say that A and B is related. The weight of a relation is defined to be the total time length of all the phone calls made between the two persons. A "Gang" is a cluster of more than 2 persons who are related to each other with total relation weight being greater than a given threshold K. In each gang, the one with maximum total weight is the head. Now given a list of phone calls, you are supposed to find the gangs and the heads.

**Input Specification:**

Each input file contains one test case. For each case, the first line contains two positive numbers N and K (both less than or equal to 1000), the number of phone calls and the weight threthold, respectively. Then N lines follow, each in the following format:

Name1 Name2 Time

where Name1 and Name2 are the names of people at the two ends of the call, and Time is the length of the call. A name is a string of three capital letters chosen from A-Z. A time length is a positive integer which is no more than 1000 minutes.

**Output Specification:**

For each test case, first print in a line the total number of gangs. Then for each gang, print in a line the name of the head and the total number of the members. It is guaranteed that the head is unique for each gang. The output must be sorted according to the alphabetical order of the names of the heads.

**Sample Input 1:**

8 59

AAA BBB 10

BBB AAA 20

AAA CCC 40

DDD EEE 5

EEE DDD 70

FFF GGG 30

GGG HHH 20

HHH FFF 10

**Sample Output 1:**

2

AAA 3

GGG 3

**Sample Input 2:**

8 70

AAA BBB 10

BBB AAA 20

AAA CCC 40

DDD EEE 5

EEE DDD 70

FFF GGG 30

GGG HHH 20

HHH FFF 10

**Sample Output 2:**

0

[提交代码](https://www.patest.cn/contests/pat-a-practise/1034)

再次见到图论题目，但是这次是完完全全自己做完的，好像在几个地方需要特殊讨论一下，以及网上的做法是并查集，并不知道，看来要学习一个。

主要思路是在输入阶段根据握手定理统计出每个点的权值，这个权值在判断head和dfs所有点的时候都很有用（因为如果不是对于点进行统计的话那就要遍历边了，这个又是另一回事了。）

然后DFS出一个联通分支，对于这个联通分支进行判断，如果其和大于标准且成员数大于2则判断为一个gang.

#include<iostream>

#include<string>

#include<algorithm>

#include<queue>

#include<vector>

#include<sstream>

#include<stack>

#include<map>

#include<cstring>

#include<climits>

#define MAX 2005

using namespace std;

struct verticle {

int head;

int num=0;

};

map<string, int>indi\_code;

map<int, string>code\_indi;

map<int, vector<pair<int, int>>>edges;

verticle target;

long long int total\_amount;

int max\_amount;

int each\_num;

vector<verticle>result;

int vst[MAX] = { 0 }, total[MAX] = { 0 };

void dfs(int index)

{

vst[index] = 1;

if (total[index] > max\_amount)

{

target.head = index;

max\_amount = total[index];

}

each\_num++;

total\_amount += total[index];

for (int i = 0; i < edges[index].size(); i++)

{

if (vst[edges[index][i].first])

continue;

dfs(edges[index][i].first);

}

return ;

}

int main()

{

int num, stand;

cin >> num >> stand;

int order = 0;

map<string, int>::iterator it;

for (int i = 0; i < num; i++)

{

string a, b;

int temp;

cin >> a >> b >> temp;

it = indi\_code.find(a);

if (it == indi\_code.end())

{

indi\_code.insert(make\_pair(a, order));

code\_indi.insert(make\_pair(order++, a));

}

it = indi\_code.find(b);

if (it == indi\_code.end())

{

indi\_code.insert(make\_pair(b, order));

code\_indi.insert(make\_pair(order++, b));

}

total[indi\_code[a]] += temp;

total[indi\_code[b]] += temp;

edges[indi\_code[a]].push\_back(make\_pair(indi\_code[b], temp));

edges[indi\_code[b]].push\_back(make\_pair(indi\_code[a], temp));

}

for (int i = 0; i < num; i++)

{

if (vst[i])

continue;

dfs(i);

target.num = each\_num;

if(total\_amount/2>stand&&each\_num>2)

result.push\_back(target);

target.num = 0;

total\_amount = 0;

max\_amount = 0;

each\_num = 0;

}

if (result.size() == 0)

cout << "0";

else

{

cout << result.size()<<endl;

for (int i = 0; i < result.size(); i++)

{

cout << code\_indi[result[i].head] << " " << result[i].num << endl;

}

}

return 0;

}