# 求度数

代码：

#include<bits/stdc++.h>

using namespace std;

const int MAX\_NUM=105;

class openFile { //打开文件的class

private:

public:

fstream io;

openFile(string filename) {//需要指定文件名

io.open(filename.c\_str(),ios::in|ios::app);//以读取和添加的方式打开

}

~openFile() {

io.close();

}

};

class Bin\_fun { //将读取的字符串形式转化为两个元素字符串

public:

string x,y,s;//s为原子符串，x为第一元素，y为第二元素

Bin\_fun(string newS) { //new 为a={(1,2),(2,3)...}

s=newS;

int pos=-1;

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

if(s[i]>='0'&&s[i]<='9') {

pos=i;

break;

}

for(int i=pos; i<s.size(); i+=6) {

x+=s[i];

y+=s[i+2];

}

//cout<<x<<endl;

}

};

struct matrix {//邻接矩阵储存

string node;//节点信息

int arcs[MAX\_NUM][MAX\_NUM];//矩阵值表示i->j的边的个数

int node\_num, edge\_num;//节点数,边数

} Graph;

void read\_graph(string sa) { //s为文件名

openFile No\_vector(sa.c\_str());

string s;

No\_vector.io>>s;

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

if(s[i]>='0'&&s[i]<='9')//节点为数字

Graph.node+=s[i];//读取节点

string s1;

No\_vector.io>>s1;

Bin\_fun E(s1.c\_str());//转换

memset(Graph.arcs,0,sizeof(Graph.arcs));//将矩阵初始化 为0

for(int i=0; i<E.x.size(); i++) {//cout<<"ok"<<endl;

Graph.arcs[E.x[i]-1-'0'][E.y[i]-1-'0']++;//矩阵从零开始图节点从1开始

}

Graph.node\_num=Graph.node.size();

}

void print\_graph(){

for(int i=0;i<Graph.node\_num;i++){

for(int j=0;j<Graph.node.size();j++)

cout<<Graph.arcs[i][j]<<" ";

cout<<endl;

}

}

void Calc\_degree(int degree[][MAX\_NUM])

{

//int (\*degree)[2]=new int[node.size()][2];

for(int i=0;i<2;i++)

memset(degree[i],0,sizeof(degree[i]));

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

{

for(int j=0;j<Graph.node.size();j++)//有向图直接遍历就行

{

//cout<<degree[i][j]<<" ";

degree[0][i]+=Graph.arcs[i][j];

degree[1][j]+=Graph.arcs[i][j];

//cout<<"test:"<<Graph.arcs[i][j]<<endl;

}

//cout<<endl;

}

//return degree;

}

int main() {

read\_graph("Calc\_node\_degree.txt");

string name[3]={"出度","入度","度"} ;//0为出 1为入 2为和

print\_graph();

int degree[2][MAX\_NUM];//节点度数

Calc\_degree(degree);//invoked Calc\_degree fun to calc

int max[2]={0,0},min[2]={MAX\_NUM,MAX\_NUM};//最大出度入度 最小 //0为出 1为入

for(int i=0;i<3;i++)

{

cout<<name[i]<<":";

for(int j=0;j<Graph.node.size();j++)

{

if(i!=2){

cout<<degree[i][j]<<" ";

if(degree[i][j]>max[i])max[i]=degree[i][j];

if(degree[i][j]<min[i])min[i]=degree[i][j];

}

else cout<<degree[0][j]+degree[1][j]<<" ";

}

cout<<endl;

}

cout<<"最大出度:"<<max[0]<<endl;

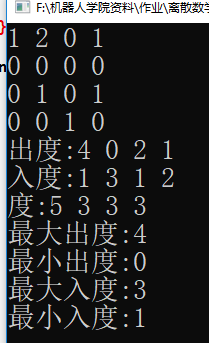
cout<<"最小出度:"<<min[0]<<endl;

cout<<"最大入度:"<<max[1]<<endl;

cout<<"最小入度:"<<min[1]<<endl;

}

运行图：



# 判断竞赛图

代码：

#include<bits/stdc++.h>

using namespace std;

const int MAX\_NUM=105;

class openFile { //打开文件的class

private:

public:

fstream io;

openFile(string filename) {//需要指定文件名

io.open(filename.c\_str(),ios::in|ios::app);//以读取和添加的方式打开

}

~openFile() {

io.close();

}

};

class Bin\_fun { //将读取的字符串形式转化为两个元素字符串

public:

string x,y,s;//s为原子符串，x为第一元素，y为第二元素

Bin\_fun(string newS) { //new 为a={(1,2),(2,3)...}

s=newS;

int pos=-1;

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

if(s[i]>='0'&&s[i]<='9') {

pos=i;

break;

}

for(int i=pos; i<s.size(); i+=6) {

x+=s[i];

y+=s[i+2];

}

//cout<<x<<endl;

}

};

class myGraph {

private:

string node;//节点信息

int arcs[MAX\_NUM][MAX\_NUM];//矩阵值表示i->j的边的个数

int node\_num, edge\_num;//节点数,边数

public:

myGraph(string sa) {

read\_graph(sa);

}

void read\_graph(string sa) { //s为文件名

openFile No\_vector(sa.c\_str());

string s;

No\_vector.io>>s;

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

if(s[i]>='0'&&s[i]<='9')//节点为数字

node+=s[i];//读取节点

string s1;

No\_vector.io>>s1;

Bin\_fun E(s1.c\_str());//转换

memset(arcs,0,sizeof(arcs));//将矩阵初始化 为0

for(int i=0; i<E.x.size(); i++) {//cout<<"ok"<<endl;

arcs[E.x[i]-1-'0'][E.y[i]-1-'0']++;//矩阵从零开始图节点从1开始

}

node\_num=node.size();

}

void print\_graph() {

for(int i=0; i<node\_num; i++) {

for(int j=0; j<node.size(); j++)

cout<<arcs[i][j]<<" ";

cout<<endl;

}

}

bool isTournament() {

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

for(int j=i+1; j<node.size(); j++)

if(arcs[i][j]+arcs[j][i]!=1)

return false;

return true;

}

};

int main() {

myGraph graph1("Tournament\_1.txt");

graph1.print\_graph();

if(graph1.isTournament())cout<<"Yes"<<endl;

else cout<<"No"<<endl;

myGraph graph2("Tournament\_2.txt");

graph2.print\_graph();

if(graph2.isTournament())cout<<"Yes"<<endl;

else cout<<"No"<<endl;

}

运行图：

