## 实现图的交、并、差、对称差、判断是否为空

### 代码：

/\*计算图的交、并、差、对称差、是否为空 \*/

#include<bits/stdc++.h>

using namespace std;//node小于10个

class Graph{

public:

int matrix[10][10];//邻接表存图

int node;

Graph(){

memset(matrix,0,sizeof(matrix));

node=0;

}

int get\_data(string name){

fstream io(name.c\_str(),ios::in);//读取data

char ch=0;//作为读取过渡

for(int i=0;;i++){

for(int j=0;;j++){

io>>ch;

matrix[i][j]=ch-'0';

if(ch=='#'||ch=='\*')//# 和 \* 作为结束的标志

break;

}

node++;

if(ch=='\*')

break;

}

my\_print();

io.close();//关闭

}

void my\_print()

{

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

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

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

cout<<endl;

}

cout<<endl;

}

};

Graph G\_con(const Graph& A,const Graph& B)//合集

{

Graph C;//for return con

C.node=max(A.node,B.node);

for(int i=0;i<max(A.node,B.node);i++)

for(int j=0;j<max(A.node,B.node);j++)

C.matrix[i][j]=max(A.matrix[i][j],B.matrix[i][j]);//取图中较大的边数，有就行

return C;

}

Graph G\_inter(const Graph& A,const Graph& B)//交集

{

Graph C;//for return con

C.node=max(A.node,B.node);//节点加入

for(int i=0;i<max(A.node,B.node);i++)

for(int j=0;j<max(A.node,B.node);j++)

C.matrix[i][j]=min(A.matrix[i][j],B.matrix[i][j]);//取图中较小的边数，有就行

return C;

}

Graph G\_diff(const Graph& A,const Graph& B)//差集

{

Graph C;//for return con

C.node=max(A.node,B.node);

for(int i=0;i<max(A.node,B.node);i++)

for(int j=0;j<max(A.node,B.node);j++)

C.matrix[i][j]=(A.matrix[i][j]>0&&B.matrix[i][j]==0)?A.matrix[i][j]:0;//A有边B无边

return C;

}

Graph G\_opp\_diff(const Graph& A,const Graph& B)//对称差集

{

Graph C;//for return con

C.node=max(A.node,B.node);

C=G\_con(G\_diff(A,B),G\_diff(B,A));

return C;

}

bool is\_empty(const Graph& A)//判断是否为零图

{

int ans=0;//ans 为边的个数

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

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

ans+=A.matrix[i][j];

return ans==0;//等于零为零图

}

int main()

{

Graph G1,G2;

G1.get\_data("graph\_1.txt");

G2.get\_data("graph\_2.txt");

cout<<"合集："<<endl;

G\_con(G1,G2).my\_print();

cout<<"交集："<<endl;

G\_inter(G1,G2).my\_print();

cout<<"差集："<<endl;

G\_diff(G1,G2).my\_print();

cout<<"对称差集："<<endl;

G\_opp\_diff(G1,G2).my\_print();

if(is\_empty(G1))cout<<"G1 is empty"<<endl;

else cout<<"not empty"<<endl;

}

### 运行图：

