**5-4最大团问题的迭代回溯（分析题）**

问题描述：设计一个解最大团问题的迭代回溯算法。

(1)首先设最大团为一个空团，往其中加入一个顶点，  
(2)然后依次考虑每个顶点，查看该顶点加入团之后仍然构成一个团，如果可以，考虑将该顶点加入团或者舍弃两种情况，如果不行，直接舍弃，然后递归判断下一顶点,  
(3)可采用剪枝策略来避免无效搜索，  
(4)为了判断当前顶点加入团之后是否仍是一个团，只需要考虑该顶点和团中顶点是否都有连接。

Maxlen:对应各个顶点[最大团](http://www.baidu.com/s?wd=%E6%9C%80%E5%A4%A7%E5%9B%A2&tn=SE_PcZhidaonwhc_ngpagmjz&rsv_dl=gh_pc_zhidao" \t "https://zhidao.baidu.com/question/_blank)的顶点个数

[num](http://www.baidu.com/s?wd=num&tn=SE_PcZhidaonwhc_ngpagmjz&rsv_dl=gh_pc_zhidao" \t "https://zhidao.baidu.com/question/_blank):各顶点的所有团的个数

Vertex：顶点

Edge：边数

程序代码(回溯法)：

#include<iostream.h>

#include<stdlib.h>

#include<stdio.h>

typedef struct

{

int maxlen;//对应各个顶点最大团的顶点个数

int ar[6][10];//最大团方案

int num;//各顶点的所有团的个数

}Arr;

void Turn(Arr m[20],int a1[20][20],int a2[20][20],int i,int vertex)//将无向图矩阵转化为三维数组形式寻找所有最大团

{

int j,k(1),tem,len(1),m1,m2;

int flag=1;

m[i].maxlen=0;

for(j=1;j<=vertex-1;j++)

if(a1[a2[i][0]][a2[i][j]]!=1)

a2[i][j]=0; //更新相邻顶点矩阵数组

for(j=1;j<vertex-1;j++)

for(k=j+1;k<=vertex-1;k++)

if(a1[a2[i][j]][a2[i][k]]!=1)

{

flag=0;

m1=j;

m2=k;

break;

}//判断是否为此顶点的团

if(flag)

{

m[i].num++;

for(j=1;j<=vertex-1;j++)

{

m[i].ar[m[i].num][j]=a2[i][j];

if(a2[i][j]!=0)

a1[a2[i][j]][i]=0;

if(a2[i][j])

len++;

}

if(m[i].maxlen<len)

{

m[i].maxlen=len;

len=0;

}

}

else //递归调用回溯寻找最大团

{

tem=a2[i][m1];

a2[i][m1]=0;

Turn(m,a1,a2,i,vertex);

a2[i][m1]=tem;

tem=a2[i][m2];

a2[i][m2]=0;

Turn(m,a1,a2,i,vertex);

a2[i][m2]=tem;

}

}

void Output(Arr m[20],int a1[20][20],int a2[20][20],int max,int vertex)//输出最大团选取方案

{

int i,j,k,len(1);

cout<<"最大图的选取方案如下:"<<endl;

for(i=1;i<=vertex;i++)

for(j=1;j<=m[i].num;j++)

if(m[i].maxlen==max)//符合最大团条件

{

cout<<len++<<") ";

for(k=0;k<=vertex-1;k++)

if(m[i].ar[j][k]!=0)

cout<<m[i].ar[j][k]<<" ";

cout<<endl;

}

}

void main()

{

int i,j,k,max(0);

int vertex,edge;//顶点和边数

Arr m[20];

int array[20][20];

int a[20][20]={0};//初始化过程

cout<<"有多少个顶点:";

cin>>vertex;

for(i=1;i<=vertex;i++)

for(j=i;j<=vertex;j++)

array[i][j]=array[j][i]=0;

cout<<"初始化顶点名称为: ";

for(i=1;i<=vertex;i++)

cout<<i<<" ";

cout<<"\n有多少条边?:";

cin>>edge;

for(i=0;i<=vertex;i++)

{

array[0][i]=1;

array[i][0]=1;

}

for(i=1;i<=edge;i++)

{

cout<<"第"<<i<<"条边:";

cin>>j>>k;

array[j][k]=array[k][j]=1;

}

for(i=1;i<=vertex;i++)

{

m[i].maxlen=0;

m[i].num=0;

}

for(i=1;i<=vertex;i++)

for(j=1;j<=vertex;j++)

m[i].ar[j][0]=i;

cout<<"初始化无向图矩阵为:"<<endl;

for(i=1;i<=vertex;i++)

{

for(j=1;j<=vertex;j++)

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

cout<<endl;

}

for(i=1;i<=vertex;i++)//建立相邻顶点的邻接矩阵

{

k=0;

a[i][k++]=i;

for(j=1;j<=vertex;j++)

if(array[i][j]==1)

a[i][k++]=j;

}

cout<<"初始化相邻顶点的邻接矩阵为:"<<endl;

for(i=1;i<=vertex;i++)

{

for(j=1;j<=vertex-1;j++)

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

cout<<endl;

}

for(i=1;i<=vertex;i++)//寻找最大团部分

{

Turn(m,array,a,i,vertex);

if(max<m[i].maxlen)

max=m[i].maxlen;

}

Output(m,array,a,max,vertex);

}