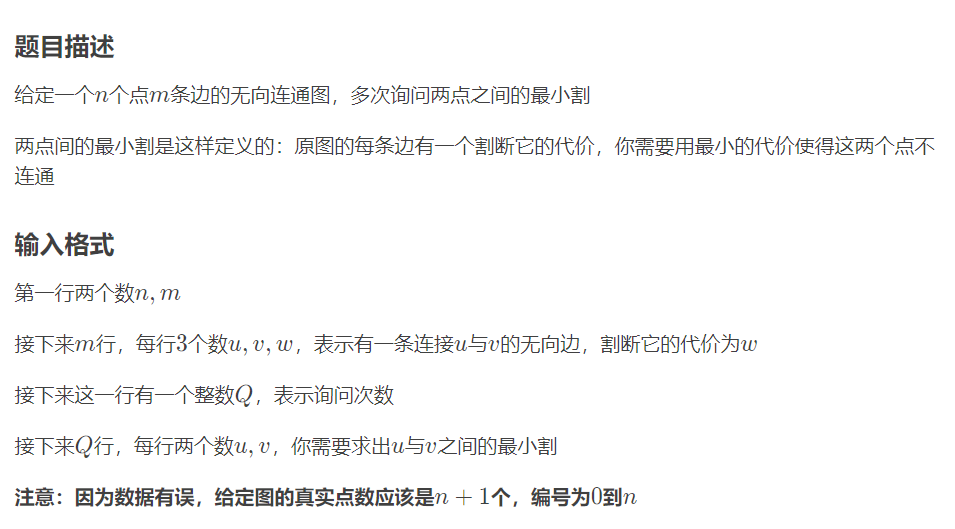
多次求任意点对的之间的最大流（最小割）

可以用最小割树（Gomory-Hu Tree）来解决，抄的别人的板子



#include<iostream>

#include<cstring>

#include<cstdio>

#include<algorithm>

using namespace std;

const int N=505,M=6005,INF=0x3f3f3f3f;

//M要开4倍，因为正反各要2条边

int h[N],e[M],ne[M],w[M],idx;

int cur[N],q[N],d[N],S,T,n,m;

int node[N],ans[N][N],tmp1[N],tmp2[N];

void add(int a,int b,int c)

{

e[idx]=b,ne[idx]=h[a],w[idx]=c,h[a]=idx++;

e[idx]=a,ne[idx]=h[b],w[idx]=0,h[b]=idx++;

}

bool bfs()

{

memset(d,0,sizeof d);

int hh=1,tt=1;

q[1]=S;cur[S]=h[S];d[S]=1;

while (hh<=tt)

{

int x=q[hh++];

for (int i=h[x];~i;i=ne[i])

if (!d[e[i]]&&w[i])

{

d[e[i]]=d[x]+1;

cur[e[i]]=h[e[i]];

if (e[i]==T) return true;

q[++tt]=e[i];

}

}

return false;

}

int findflow(int x,int limit)

{

if (x==T) return limit;

int flow=0;

for (int i=cur[x];~i&&flow<limit;i=ne[i])

{

cur[x]=i;

if (d[e[i]]==d[x]+1&&w[i])

{

int t=findflow(e[i],min(w[i],limit-flow));

if (!t) d[e[i]]=-1;

w[i]-=t,w[i^1]+=t,flow+=t;

}

}

return flow;

}

void init()

{

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

w[i]=(w[i]+w[i^1]),w[i^1]=0;

return;

}//注意退流的方式

int dinic()

{

init();

int r=0,flow;

while (bfs()) while (flow=findflow(S,INF)) r+=flow;

return r;

}//Dinic模板

void work(int l,int r)

{

if (l==r) return ;

S=node[l],T=node[l+1];

int t=dinic(),s=node[l],tt=node[l+1];//将源汇点存下来

ans[T][S]=ans[S][T]=t;

int cnt1=0,cnt2=0;

for (int i=l;i<=r;++i)

if (d[node[i]]) tmp1[++cnt1]=node[i];

else tmp2[++cnt2]=node[i];

for (int i=1;i<=cnt1;++i) node[i+l-1]=tmp1[i];

for (int i=1;i<=cnt2;++i) node[cnt1+l+i-1]=tmp2[i];

work(l,l+cnt1-1);

work(l+cnt1,r);//分治

for (int i=1;i<=cnt1;++i)

for (int j=1;j<=cnt2;++j)

{

int ii=node[i+l-1],jj=node[j+cnt1+l-1];

ans[jj][ii]=ans[ii][jj]=min(min(ans[ii][s],ans[s][tt]),ans[tt][jj]);

}//每个点都要处理

return;

}

int main()

{

memset(h,-1,sizeof h);

memset(ans,0x3f,sizeof ans);

cin>>n>>m;

n++;//下标从0开始

int x,y,z;

while (m--)

{

scanf("%d%d%d",&x,&y,&z);

x++;y++;//下标从0开始

add(x,y,z);add(y,x,z);//双向边

}

for (int i=1;i<=n;++i) node[i]=i;

work(1,n);

int que;cin>>que;

while (que--)

{

scanf("%d%d",&x,&y);

x++;y++;//下标从0开始

printf("%d\n",ans[x][y]);

}

return 0;

}