//圆方树（原题来自bzoj3694）

//dfn和low是tarjan求点双里面的数组

//edge是结构体，存储了边的两个端点和长度

//s是一个栈，存储当前的边

//v是原图的邻接表,vv是圆方树的邻接表

//cir是vector，存了每个环的所有的点和相邻两个点之间的距离

//sum存储了每个环的边权之和,presum记录了每个环的边权前缀和

//pos存储了每个点如果在某个环中作为非根出现，它的位置

inline bool check(int id,Pair E)

{

if (edge[id].x==E.x && edge[id].y==E.y) return true;

if (edge[id].x==E.y && edge[id].y==E.x) return true;

return false;

}

inline void tarjan(int cur,int father)

{

dfn[cur]=low[cur]=++Index;int i,j,x,y,id,u;

for (i=0;i<int(v[cur].size());i++)

{

y=(edge[v[cur][i]].x==cur?edge[v[cur][i]].y:edge[v[cur][i]].x);

if (!dfn[y])

{

s.push(v[cur][i]);tarjan(y,cur);

low[cur]=min(low[cur],low[y]);

if (low[y]>=dfn[cur])

{

If(check(s.top(),mp(cur,y))) {vv[cur].pb(mp(y,edge[v[cur][i]].c));vv[y].pb(mp(cur,edge[v[cur][i]].c));s.pop();continue;}

++ind;type[ind]=true;int pre=cur;

do

{

id=s.top();s.pop();

x=edge[id].x;u=edge[id].y;

if (x==pre) swap(x,u);

cir[ind].pb(mp(x,edge[id].c));

pre=x;

}

while (!check(id,mp(cur,y)));

for (j=0;j<int(cir[ind].size());j++)

{

sum[ind]+=cir[ind][j].y;presum[ind].pb(sum[ind]);

if (j!=int(cir[ind].size())-1) pos[cir[ind][j].x]=j;

}

for (j=0;j<int(cir[ind].size());j++)

if (j!=int(cir[ind].size())-1)

{

int cc=min(presum[ind][j],sum[ind]-presum[ind][j]);

vv[ind].pb(mp(cir[ind][j].x,cc));vv[cir[ind][j].x].pb(mp(ind,cc));

}

else

vv[ind].pb(mp(cir[ind][j].x,0)),vv[cir[ind][j].x].pb(mp(ind,0));

}

}

else if (y!=father && dfn[y]<dfn[cur])

{

s.push(v[cur][i]);

low[cur]=min(low[cur],dfn[y]);

}

}

}