//prims algo

#include<bits/stdc++.h>

using namespace std;

int parent[100005];

int weight[100009];

bool visited[100009];

int n,e;

class edge

{

int wt;

int node;

public:

edge(int a,int b)

{

wt=a;node=b;

}

int retwt()

{

return(wt);

}

int retnode()

{

return(node);

}

};

vector<list<edge>>v;

int minpick()

{int x=INT\_MAX,y=0;

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

{

if(visited[i]==false)

{

if(weight[i]<x)

{

x=weight[i];

y=i;

}

}

}

return(y);

}

void prims()

{

int tobevisited=minpick();

visited[tobevisited]=true;

list<edge>temp=v[tobevisited];

list<edge>::iterator it;

for(it=temp.begin();it!=temp.end();it++)

{

if(visited[(\*it).retnode()]==false)

{

if((\*it).retwt()<weight[(\*it).retnode()])

{

weight[(\*it).retnode()]=(\*it).retwt();

parent[(\*it).retnode()]=tobevisited;

}

}

}

}

int main()

{

cin>>n>>e;

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

{

weight[i]=INT\_MAX;

}

weight[0]=0;

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

{

visited[i]=false;

}

v.assign(n+1, list<edge>());

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

{

int a,b;

cin>>a>>b;

int weight;

cin>>weight;

v[a].push\_back(edge(weight,b));

v[b].push\_back(edge(weight,a));

}

for(int k=0;k<n;k++)

prims();

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

{

cout<<i<<" "<<parent[i]<<" "<<weight[i]<<"\n";

}

return 0;

}