//wrong kruskals algo

#include <bits/stdc++.h>

using namespace std;

int parent[100005];

int j=0;

class edge

{

public:

int wt;

int src;

int dest;

}obj1[100004],obj2[100005];//obj1 holds edges of graph obj2 holds edges of mst

int calcparent(int x)

{

if(parent[x]==x)//condition for topmost parent

return x;

else

return(calcparent(parent[x]));

}

bool checkvalid(edge a)

{

if(calcparent(a.src)==calcparent(a.dest))//calcparent calculates the topmost parent of src nd dest.if same,then edge might form a cycle

return false;

else

return true;

}

bool myfunc(edge a,edge b)

{

return(a.wt<b.wt);

}

void kruskals(int e,int n)

{

sort(obj1,obj1+e,myfunc);

int i=0;//i keeps a trackof obj1 array and j keeps a trackof obj2

while(j<n-1)//only n-1 edges have to be added in obj2

{

bool a=checkvalid(obj1[i]);

if(a)

{

obj2[j]=obj1[i];

parent[obj1[i].dest]=obj1[i].src;

i++;j++;

}

else

{i++;}

}

}

int main()

{

int n,e;

cin>>n>>e;

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

{

parent[i]=i;//initially for unattached edges

}

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

{

int s,d, w;

cin>>s>>d>>w;

obj1[i].wt=w;

obj1[i].src=s;

obj1[i].dest=d;

}

kruskals(e,n);

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

{

cout<<obj2[i].src <<" "<<obj2[i].dest<<" "<<obj2[i].wt<<"\n";

}

return(0);

}