## 题目描述

城市C是一个非常繁忙的大都市，城市中的道路十分的拥挤，于是市长决定对其中的道路进行改造。城市C的道路是这样分布的：城市中有n个交叉路口，有些交叉路口之间有道路相连，两个交叉路口之间最多有一条道路相连接。这些道路是双向的，且把所有的交叉路口直接或间接的连接起来了。每条道路都有一个分值，分值越小表示这个道路越繁忙，越需要进行改造。但是市政府的资金有限，市长希望进行改造的道路越少越好，于是他提出下面的要求：

1．改造的那些道路能够把所有的交叉路口直接或间接的连通起来。 2．在满足要求1的情况下，改造的道路尽量少。 3．在满足要求1、2的情况下，改造的那些道路中分值最大的道路分值尽量小。

任务：作为市规划局的你，应当作出最佳的决策，选择那些道路应当被修建。

## 输入格式

第一行有两个整数n,m表示城市有n个交叉路口，m条道路。

接下来m行是对每条道路的描述，u, v, c表示交叉路口u和v之间有道路相连，分值为c。(1≤n≤300，1≤c≤10000，1≤m≤100000)

## 输出格式

两个整数s, max，表示你选出了几条道路，分值最大的那条道路的分值是多少。

## 输入输出样例

**输入 #1**

4 5

1 2 3

1 4 5

2 4 7

2 3 6

3 4 8

**输出 #1**

3 6

#include<iostream>

#include<cstdio>

#include<algorithm>

#include<cstring>

using namespace std;

int read()

{

int x=0;int f=1;char ch=getchar();

while(ch>'9'||ch<'0'){if(ch=='-'){f=-1;}ch=getchar();}

while(ch>='0'&&ch<='9'){x=x\*10+ch-'0';ch=getchar();}

return x\*f;

}

int f[10006];

int n,m;

struct node

{

int u,v,w;

}e[200003];

int cmp(node a,node b)

{

return a.w<b.w;

}

int find1(int x)

{

int r=x;

while(r!=f[r])

{

r=f[r];

}

int i=x,j;

while(i!=r)

{

j=f[i];

f[x]=r;

i=j;

}

return r;

}

void merge(int x,int y)

{

int t1=find1(x);

int t2=find1(y);

if(t1!=t2)

{

f[t1]=t2;

}

}

int main()

{

n=read();m=read();

int u,v,w;

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

{

scanf("%d %d %d",&u,&v,&w);

e[i].u=u;

e[i].v=v;

e[i].w=w;

}

for(int i=0;i<=n;i++) f[i]=i;

sort(e+1,e+1+m,cmp);

long long ans=0;int ct=0;

int i;

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

{

if(find1(e[i].u)!=find1(e[i].v))

{

merge(e[i].u,e[i].v);

ct++; ans+=e[i].w;

if(ct==n-1) break;

}

}

cout<<ct<<" "<<e[i].w;

return 0;

}