

#include<iostream>

#include<cstring>

using namespace std;

const int N = 2e5+5;

int n,m,s[N],p[N],ans;

void init(){

ans=0;

memset(s,0,sizeof(s));

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

p[i]=i;

}

int fd(int x) { ///此时find不单有查找任务，还有更新距离任务

if(x==p[x]) return x;

int t=p[x];

p[x]=fd(p[x]);

s[x]+=s[t]; ///记录到根节点的距离，一定要有一个思想，根节点是一个区间的一个端点而不是一个区间，输入的区间被合并成了两个点

return p[x];

}

void Union(int a,int b,int num) {

int x=fd(a),y=fd(b);

if(x==y) {

if(s[b]!=s[a]+num) ans++;

}else {

p[y]=x;

s[y]=s[a]+num-s[b]; ///y到x的距离等于a到x的距离+b到a的距离-b到y的距离

}

}

int main(){

while(cin>>n>>m) {

init();

for(int i=0;i<m;i++) {

int a,b,c;

cin>>a>>b>>c;

Union(a-1,b,c);

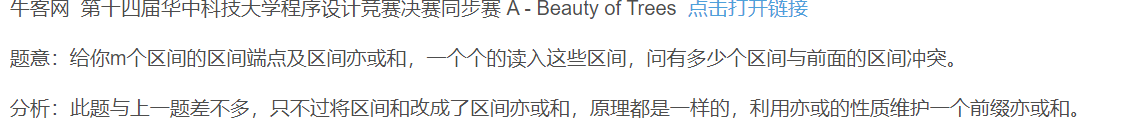
///等价于Union(a,b+1,c);

}

cout<<ans<<endl;

}

}



#include<iostream>

#include<cstring>

using namespace std;

const int N = 1e5+10;

int n,m,s[N],p[N],k,f;

void init(){

f=0;

for(int i=0;i<N;i++) p[i]=i;

memset(s,0,sizeof(s));

}

int fd(int x) {

if(x==p[x]) return x;

int t=p[x];

p[x]=fd(p[x]);

s[x]^=s[t];

return p[x];

}

void Union(int a,int b,int c) {

int x=fd(a),y=fd(b);

if(x==y) {

if(s[b]!=(s[a]^c)) cout<<k<<endl,f=1; ///亦或这里一定要打括号，亦或的优先级低于！=

}else {

p[y]=x;

s[y]=s[a]^s[b]^c;

}

}

int main(){

while(cin>>n>>m) {

init();

for(k=1;k<=m;k++) {

int a,b,c;

cin>>a>>b>>c;

Union(a-1,b,c);

}

if(!f) cout<<-1<<endl;

}

}