

```
#include<cstdio>
#include<cstring>
#include<iostream>
#include<algorithm>
#include<vector>
using namespace std;
const int N=1e5+5,M=2e5+5;
int n,m;
int read()
{
    int x=0,f=1;char c=getchar();
    while(c<'0' || c>'9'){if(c=='-')f=-1;c=getchar();}
    while(c>='0'&&c<='9'){x=x*10+c-'0';c=getchar();}
    return x*f;
}
struct Node1{int a,b;}bomb[N];
struct Node2{int l,r,id;}cord[M];
struct Node3{int nxt,to,id;}Edges[M*2];
int num[N],cnt=0,w[N],head[N],tot=0;
bool vis[N],legal=true;
vector<int>c;
int getw(int x){return lower_bound(num+1,num+1+cnt,x)-num;}
void addedge(int u,int v,int id)
{
    Edges[++tot].nxt=head[u];
    head[u]=tot;
    Edges[tot].id=id,Edges[tot].to=v;
}
void dfs(int u,int f,int fEdge)
{
    vis[u]=1;
    for(int i=head[u];~i;i=Edges[i].nxt)
    {
        int v=Edges[i].to;
        if(!vis[v])dfs(v,u,Edges[i].id);
    }
    if(w[u]==1)
    {
        if(!f){legal=0;return;}
        w[u]^=1,w[f]^=1,c.push_back(fEdge);
    }
}
int main()
{
    memset(head,-1,sizeof(head));
    n=read(),m=read();
    for(int i=1;i<=n;i++)
    {
        bomb[i].a=read(),bomb[i].b=read();
        num[++cnt]=bomb[i].a;
    }
}
```

```
sort(num+1,num+1+cnt);
cnt=unique(num+1,num+1+cnt)-num-1;
for(int i=1;i<=n;i++)w[getw(bomb[i].a)]=bomb[i].b;
for(int i=cnt+1;i;i--)w[i]^=w[i-1];
for(int i=1;i<=m;i++)
{
    cord[i].l=read(),cord[i].r=read(),cord[i].id=i;
    int x=getw(cord[i].l),y=getw(cord[i].r+1);
    addedge(x,y,i),addedge(y,x,i);
}
for(int i=1;i<=cnt+1;i++)
{
    if(!vis[i])dfs(i,0,0);
    if(!legal)break;
}
if(!legal)puts("-1");
else
{
    printf("%d\n",c.size());
    sort(c.begin(),c.end());
    for(int i=0;i<c.size();i++)printf("%d ",c[i]);
}
return 0;
}
```

From:
<https://wiki.cvbbacm.com/> - CVBB ACM Team

Permanent link:
https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:wangzai_milk:f._perils_in_parallel

Last update: 2020/05/18 11:19

