```
/*
朱刘算法是干什么的呢
给定一个有向图以及点X，求以X为根节点的最小生成树（有向）
*/
#include<iostream>
#include<stdio.h>
#include<string.h>
#include<algorithm>
#include<math.h>
#define maxn 10005
using namespace std;
struct nod
{
    int x,y;
    double w;
    nod(){}
}edge[maxn];
int n,m,root;
double x[maxn],y[maxn];
int id[maxn],vis[maxn],pre[maxn];
double in[maxn];
double zhuliu()
{
    double ans=0;
    int sz=n;
    while(1)
    {
        for(int i=1;i<=sz;i++)
            in[i]=123456789.0,vis[i]=id[i]=0;
        for(int i=1;i<=m;i++)
        {
            if( edge[i].w<in[edge[i].y] && edge[i].x!=edge[i].y )
                pre[edge[i].y]=edge[i].x,in[edge[i].y]=edge[i].w;
        }

        in[root]=0.0;

        for(int i=1;i<=sz;i++)
        {
            if(in[i]==123456789.0 && i!=root)
                return -1.0;
        }
        int cnt=0;
        for(int i=1;i<=sz;i++)
        {
            ans+=in[i];
            int now=i;
            while(vis[now]!=i && !id[now] && now!=root)
            {
                vis[now]=i;
                now=pre[now];
```

```
            }
            if(now!=root && !id[now])
            {
                cnt++;
                id[now]=cnt;
                for(int j=pre[now];j!=now;j=pre[j])
                    id[j]=cnt;
            }
        }
        if(!cnt)
            break;
        for(int i=1;i<=sz;i++)
            if(!id[i])
                id[i]=++cnt;
        for(int i=1;i<=m;i++)
        {
            int temp=edge[i].y;
            edge[i].x=id[edge[i].x];
            edge[i].y=id[edge[i].y];
            if(edge[i].x!=edge[i].y)
                edge[i].w-=in[temp];
        }
        sz=cnt;
        root=id[root];
    }
    return ans;
}
int main()
{
    while(~scanf("%d%d",&n,&m))
    {
        for(int i=1;i<=n;i++)
            scanf("%lf%lf",&x[i],&y[i]);
        for(int i=1;i<=m;i++)
        {
            int a,b;
            scanf("%d%d",&a,&b);
            edge[i].x=a; edge[i].y=b;
            if(a!=b)
                edge[i].w=sqrt((x[a]-x[b])*(x[a]-x[b])+(y[a]-y[b])*(y[a]-
y[b]));
            else
                edge[i].w=123456789;
        }
        root=1;
        double res=zhuliu();
        if(res==-1.0)
            printf("poor snoopy\n");
        else
            printf("%.2f\n",res);
```

```
    }
}
```