最小生成树 入门题

Kruskal

#include <cstdio>

#include <cstring>

#include<iostream>

#include<cmath>

#include<algorithm>

using namespace std;

int fa[1010];

int x[1010],y[1010];

struct node

{

int x,y;

double dis;

} edge[1000010];

int Find(int x)

{

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

return fa[x]=Find(fa[x]);//是否需要带路径压缩还要考虑一下,有时带了会超时

}

double getdis(int i,int j)

{

return sqrt((x[i]-x[j])\*(x[i]-x[j])+(y[i]-y[j])\*(y[i]-y[j])+0.0);

}

void Merge(int x,int y)//不按秩合并

{

x=Find(x);

y=Find(y);

if(x!=y)

{

fa[y]=x;

}

}

bool cmp(node a,node b)

{

return a.dis<b.dis;

}

int main()

{

//freopen("input.txt","r",stdin);

int n;

cin>>n;

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

fa[i]=i;

int k=0;

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

cin>>x[i]>>y[i];

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

for(int j=i+1; j<=n; j++)

{

edge[k].x=i;

edge[k].y=j;

edge[k].dis=getdis(i,j);

k++;

}

sort(edge,edge+k,cmp);

int t=0;

int m,u,v;

cin>>m;

while(m--)

{

cin>>u>>v;

int du=Find(u);

int dv=Find(v);

if(du!=dv)

{

fa[du]=dv;

t++;

}

}

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

{

int dx=Find(edge[i].x);

int dy=Find(edge[i].y);

if(dx!=dy)

{

t++;

cout<<edge[i].x<<' '<<edge[i].y<<endl;

fa[dy]=dx;

}

if(t==n-1)break;

}

return 0;

}