CCF 第四题

问题描述

　　目前在一个很大的平面房间里有 n 个无线路由器,每个无线路由器都固定在某个点上。任何两个无线路由器只要距离不超过 r 就能互相建立网络连接。  
　　除此以外,另有 m 个可以摆放无线路由器的位置。你可以在这些位置中选择至多 k 个增设新的路由器。  
　　你的目标是使得第 1 个路由器和第 2 个路由器之间的网络连接经过尽量少的中转路由器。请问在最优方案下中转路由器的最少个数是多少?

输入格式

　　第一行包含四个正整数 n,m,k,r。(2 ≤ n ≤ 100,1 ≤ k ≤ m ≤ 100, 1 ≤ r ≤ 108)。  
　　接下来 n 行,每行包含两个整数 xi 和 yi,表示一个已经放置好的无线路由器在 (xi, yi) 点处。输入数据保证第 1 和第 2 个路由器在仅有这 n 个路由器的情况下已经可以互相连接(经过一系列的中转路由器)。  
　　接下来 m 行,每行包含两个整数 xi 和 yi,表示 (xi, yi) 点处可以增设一个路由器。  
　　输入中所有的坐标的绝对值不超过 108,保证输入中的坐标各不相同。

输出格式

　　输出只有一个数,即在指定的位置中增设 k 个路由器后,从第 1 个路由器到第 2 个路由器最少经过的中转路由器的个数。

样例输入

5 3 1 3  
0 0  
5 5  
0 3  
0 5  
3 5  
3 3  
4 4  
3 0

样例输出

2

本题问题模型建立是关键，把问题转化成图，同时也结合了动态规划思想

建图：

两个路由器可以通讯则二者连一条无向边。

用d[i][j]表示从起点开始经过增设的j个路由器到达i的最短路径，vis[i][j]表示是否可以从起点经过增设的j个路由器到达i。

然后就和普通的最短路一样了。这里选用spfa求解。

最后d[1][0]~d[1][k]的最小值就是答案

//CCF无线网络二维最短路spfa动规

#include<algorithm>

#include<cstdio>

#include<cstring>

#include<iostream>

#include<queue>

#define N 205

#define INF 0x3f3f3f3f

typedef long long LL;

using namespace std;

struct P{

int x,y;

}p[205];

int n,m,k,r;

int d[205][205];

bool vis[205][205],Map[205][205];

void spfa()

{

queue<P> q;

memset(vis,0,sizeof(vis));

memset(d,INF,sizeof(d));

d[0][0]=0;

vis[0][0]=1;

P s,tem;

s.x=s.y=0;

q.push(s);

while(!q.empty())

{

s=q.front();

q.pop();

vis[s.x][s.y]=0;

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

if(Map[s.x][i])

{

tem.x=i;

tem.y=s.y;

if(i>=n) ++tem.y;

if(tem.y<=k&&d[tem.x][tem.y]>d[s.x][s.y]+1)

{

d[tem.x][tem.y]=d[s.x][s.y]+1;

if(!vis[tem.x][tem.y])

{

vis[tem.x][tem.y]=1;

q.push(tem);

}

}

}

}

int ans=INF;

for(int i=0;i<=k;i++) ans=min(ans,d[1][i]);

printf("%d\n",ans-1);

}

int main()

{

int i,j;

cin>>n>>m>>k>>r;

for(i=0;i<n+m;++i) scanf("%d%d",&p[i].x,&p[i].y);

memset(Map,0,sizeof(Map));

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

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

if(LL(p[i].x-p[j].x)\*(p[i].x-p[j].x)+LL(p[i].y-p[j].y)\*(p[i].y-p[j].y)<=LL(r)\*r) Map[i][j]=Map[j][i]=1;

spfa();

return 0;

}