**邮局**

问题描述

　　C村住着n户村民，由于交通闭塞，C村的村民只能通过信件与外界交流。为了方便村民们发信，C村打算在C村建设k个邮局，这样每户村民可以去离自己家最近的邮局发信。  
  
　　现在给出了m个备选的邮局，请从中选出k个来，使得村民到自己家最近的邮局的距离和最小。其中两点之间的距离定义为两点之间的直线距离。

输入格式

　　输入的第一行包含三个整数n, m, k，分别表示村民的户数、备选的邮局数和要建的邮局数。  
　　接下来n行，每行两个整数x, y，依次表示每户村民家的坐标。  
　　接下来m行，每行包含两个整数x, y，依次表示每个备选邮局的坐标。  
　　在输入中，村民和村民、村民和邮局、邮局和邮局的坐标可能相同，但你应把它们看成不同的村民或邮局。

输出格式

　　输出一行，包含k个整数，从小到大依次表示你选择的备选邮局编号。（备选邮局按输入顺序由1到m编号）

样例输入

5 4 2  
0 0  
2 0  
3 1  
3 3  
1 1  
0 1  
1 0  
2 1  
3 2

样例输出

2 4

数据规模和约定

　　对于30%的数据，1<=n<=10，1<=m<=10，1<=k<=5；  
　　对于60%的数据，1<=m<=20；  
　　对于100%的数据，1<=n<=50，1<=m<=25，1<=k<=10。

本题的C++参考代码如下：

#include <iostream>

#include <stdio.h>

#include <math.h>

using namespace std;

struct Point {

int no;

int x, y;

double getDist(Point p) {

return sqrt((x - p.x) \* (x - p.x) + (y - p.y) \* (y - p.y));

}

};

Point ans[10];

int n, m, k;

Point person[50];

Point fire[25];

Point result[10];

bool repeat[50], ban[50];

double ansDist = 1000000000;

double minDist[50], sum = 0;

double G[50][25];

void dfs(int deep, int index)

{

if(deep == k) {

if(sum < ansDist) {

ansDist = sum;

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

ans[i] = result[i];

}

}

} else {

double tmpDist[50] = {0}, tsum = sum;

bool flag2 = false;

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

tmpDist[i] = minDist[i];

for(int i = m-k+deep; i >= index; i--) {

if(repeat[i]) continue;

if(deep > 0 && ban[i]) continue;

bool flag = false;

if(deep == 0) {

sum = 0;

for(int j = 0; j < n; j++) {

minDist[j] = G[j][i];

sum += G[j][i];

}

flag = true;

}

else {

sum = tsum;

for(int j = 0; j < n; j++) {

if(G[j][i] < tmpDist[j]) {

sum -= tmpDist[j] - G[j][i];

minDist[j] = G[j][i];

flag = true;

} else minDist[j] = tmpDist[j];

}

}

if(flag) {

flag2 = true;

result[deep] = fire[i];

dfs(deep+1, result[deep].no + 1);

} else {

ban[i] = true;

}

}

if(flag2 == false) {

result[deep] = fire[result[deep-1].no+1];

dfs(deep+1, m-k+deep+1);

} else {

sum = tsum;

for(int j = 0; j < n; j++) {

minDist[j] = tmpDist[j];

}

}

}

}

int main(int argc, char \*argv[]) {

scanf("%d%d%d", &n, &m, &k);

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

scanf("%d%d", &person[i].x, &person[i].y);

minDist[i] = ansDist;

}

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

scanf("%d%d", &fire[i].x, &fire[i].y);

fire[i].no = i;

}

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

if(!repeat[i]) {

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

if(fire[i].x == fire[j].x && fire[i].y == fire[j].y)

repeat[j] = true;

}

}

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

for(int j = 0; j < m; j++)

G[i][j] = person[i].getDist(fire[j]);

dfs(0, 0);

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

printf("%d ", ans[i].no+1);

}

return 0;

}

本题的C参考代码如下：

#include<stdio.h>

#include<math.h>

struct

{

int x, y;

}pos[50];

float dis[25][50];

int res[10];

float minn = 100000000;

int n, m, k, i, j, f[50];

void dfs(int step,int cnt,float sum,int temp[],float las\_dis[])

{

if(step == m || cnt == k)

{

if(cnt == k && minn > sum)

{

minn = sum;

for(i = 0;i < k;i ++)

res[i] = temp[i];

}

return;

}

else if(cnt == 0)

{

float w[50];

dfs(step+1,cnt,sum,temp,w);

temp[cnt] = step+1;

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

{

sum += dis[step][i];

w[i] = dis[step][i];

}

dfs(step+1,cnt+1,sum,temp,w);

}

else if(m - step == k - cnt)

{

float w[50];

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

w[i] = las\_dis[i];

temp[cnt] = step+1;

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

{

if(w[i] > dis[step][i])

{

sum = sum - w[i] + dis[step][i];

w[i] = dis[step][i];

}

}

dfs(step+1,cnt+1,sum,temp,w);

}

else

{

float w[50];

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

w[i] = las\_dis[i];

dfs(step+1,cnt,sum,temp,w);

if(!f[step])

{

temp[cnt] = step+1;

int flag = 0;

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

{

if(w[i] > dis[step][i])

{

sum = sum - w[i] + dis[step][i];

w[i] = dis[step][i];

flag = 1;

}

}

if(flag)

dfs(step+1,cnt+1,sum,temp,w);

else

f[step] = 1;

}

}

}

int main()

{

int x, y;

scanf("%d%d%d", &n, &m, &k);

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

{

scanf("%d%d",&pos[i].x,&pos[i].y);

}

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

{

scanf("%d%d", &x, &y);

for(j = 0;j < n;j ++)

{

dis[i][j] = sqrt(pow(x - pos[j].x,2) + pow(y - pos[j].y,2));

}

}

int temp[25];

float las\_dis[50];

dfs(0,0,0,temp,las\_dis);

printf("%d",res[0]);

for(i = 1;i < k;i ++)

{

printf(" %d",res[i]);

}

printf("\n");

return 0;

}

本题的Java参考代码如下：

import java.util.Scanner;

public class Main {

static int n,m,k,j,f1,f2;

static int [][]c=new int [55][2];

static int [][]y=new int [27][2];

static int []d=new int [12];

static int []f=new int [55];

static float [][]yc=new float[27][55];

static float s=1000000000;

public static void main(String[] args) {

Scanner input=new Scanner(System.in);

int i,j;

int []o=new int[12];

float []w=new float[55];

n=input.nextInt();

m=input.nextInt();

k=input.nextInt();

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

{

c[i][0]=input.nextInt();

c[i][1]=input.nextInt();;

}

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

{

y[i][0]=input.nextInt();

y[i][1]=input.nextInt();

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

yc[i][j]=(float) Math.sqrt((c[j][0]-y[i][0])\*(c[j][0]-y[i][0])+(c[j][1]-y[i][1])\*(c[j][1]-y[i][1]));

}

dfs(0,1,o,w,0);

for(i=0;i<k;i++)

System.out.print(d[i]+" ");

}

private static void dfs(int t,int i,int o[],float w[],float sum) {

if(i<=m+1)

{

if(t==k)

{

if(sum<s)

{

s=sum;

for(j=0;j<k;j++)

d[j]=o[j];

}

}

else if(i<=m&&t<k)

{

float []ww=new float[55];

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

ww[j]=w[j];

dfs(t,i+1,o,w,sum);

f1=1;

f2=0;

if(f[i]==0)

{

o[t]=i;

if(t>0)

{

f2=1;

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

{

if(ww[j]>yc[i][j])

{

sum=sum-ww[j]+yc[i][j];

ww[j]=yc[i][j];

f1=0;

}

}

}

else

{

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

{

sum+=yc[i][j];

ww[j]=w[j]=yc[i][j];

}

}

if(f1==1&&f2==1)

{

f[i]=1;

dfs(t,i+1,o,w,sum);

}

else

dfs(t+1,i+1,o,ww,sum);

}

}

}

}

}