**题目大意:**

给出两只青蛙的坐标A、B，和其他的n-2个坐标，任一两个坐标点间都是双向连通的。显然从A到B存在至少一条的通路，每一条通路的元素都是这条通路中前后两个点的距离，这些距离中又有一个最大距离。

现在要求求出所有通路的最大距离，并把这些最大距离作比较，把最小的一个最大距离作为青蛙的最小跳远距离。

**Floyd**[**算法**](http://lib.csdn.net/base/datastructure)

用Floyd算法求出两两最短路，再求出从每个点开始的最长路，最后从这n个最长路中求出最小的那个即为所求。

#include<cstdio>

#include<cmath>

#include<cstring>

#include<algorithm>

#include<iostream>

#define maxn 1000001

using namespace std;

double d[210][210];

int x[210],y[210],n;

int main()

{

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

int ji=0;

while(cin>>n&&n)

{

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

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

if(i==j)d[i][j]=0;else d[i][j]=maxn;

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

{

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

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

{

int tx=x[i]-x[j],ty=y[i]-y[j];

d[i][j]=d[j][i]=sqrt(tx\*tx+ty\*ty);

}

}

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

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

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

{

d[i][j]=min(d[i][j],max(d[i][k],d[k][j]));

// cout<<i<<' '<<j<<"----"<<d[i][j]<<' '<<d[i][k]<<' '<<d[k][j]<<endl;

}

printf("Scenario #%d\nFrog Distance = %.3f\n\n",++ji,d[1][2]);

}

return 0;

}

这么写是因为，让求的最大跳是指这条路径中两点之间的最大距离

如果选择要走i-k-j了，那么就要看是i-k距离大还是kj距离大