**★★☆☆☆**

**題組：Problem Set Archive** **with Online Judge**

**題號：534 : Frogger**

**解題者：鮑邦元**

**解題日期：2019年4月18日**

**題意：**

湖面上有n顆石頭(2≤n≤200)。一隻青蛙想從腳下的石頭跳到指定的另外一顆石頭，但因為湖水太髒又怕自己跳躍力不夠，所以想要藉由其它石頭跳到終點。此題即找出在兩顆石頭間的所有路徑中，最遠蛙跳距離為最小值(minmax distance)的路徑。

**題意範例：**

6 // 石頭數量 n

0 0 // 起點石頭位置

5 0 // 終點石頭位置

1 1 // 其餘石頭位置

6 3 // 其餘石頭位置

2 0 // 其餘石頭位置

3 -2 // 其餘石頭位置

0 // 結束程式

Output:

Scenario #1

Frog Distance = 2.828

**解法：**

使用Floyd-Warshall演算法、動態規劃實作。

1. 輸入完測資後，會先初始化distance[i][j]，表示第i顆到第j顆石頭的直線距離。

2. 演算法中，distance[i][j]表示當前從第i顆到第j顆石頭的過程中的最遠蛙跳距離。

3. 算出從i經由k再到j的過程中的最遠蛙跳距離，再與當前i到j的最遠蛙跳距離去做比較。

4. 最終答案為 distance[0][1]

**解法範例：(如下圖)**

1. 初始化:

distance[0][1] = distance[1][0] = 3

distance[0][2] = distance[2][0] = 1.414

distance[1][2] = distance[2][1] = 2.236

2. ★開始判斷distance[0][1]之值。

先決定max，即找出第0顆石頭到第2顆石頭再由第2顆到第1顆石頭的這2段的過程中的最遠蛙跳距離。故max=2.236

3. 因max小於distance[0][1]=3，所以distance[0][1]更新為2.236。

4. 輸出結果distance[0][1]至小數點後第三位。

**一張含有 文字 的圖片

自動產生的描述**

**討論：**

此題可以用多種演算法實作，例如: Floyd-Warshall 、MST、SPFA、 Dijkstra。以上都是用於求最短路徑的方法，但因為此題的圖形為不含負權的無向圖，用Floyd-Warshall 或 Dijkstra 較直觀。

★比較Dijkstra算法和Floyd-Warshall :

Floyd-Warshall→所有點到所有點。時間複雜度:O()。UVA實測:0.010s。

Dijkstra →單點到所有點。所以Dijkstra跑N次也和Floyd-Warshall一樣效果。

根據使用的資料結構不同，Dijkstra演算法時間複雜度也有所不同。如果使用上課可所教的概念以陣列實作，時間複雜度為O()。UVA實測:0.000s。

**程式：**

#include <cstdio>

#include <cmath>

double x[201], y[201];

double distance[201][201];

int main(){

int n,cas = 0;

while(scanf("%d",&n)&&n){

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

scanf("%lf%lf",&x[i],&y[i]);

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

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

distance[i][j]=distance[j][i]=sqrt((x[i]-x[j])\*(x[i]-x[j])+(y[i]-y[j])\*(y[i]-y[j]));

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

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

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

double max = distance[i][k]>distance[k][j]?distance[i][k]:distance[k][j];

if (distance[i][j]>max)

distance[i][j]=max;

}

printf("Scenario #%d\nFrog Distance = %.3lf\n\n",++cas,distance[0][1]);

}

return 0;

}