## Bài 1: Khoảng cách

Trước cửa nhà Mr Bill có một cái hồ rất rộng. Giữa hồ có một hòn đảo nhỏ. Một lần Mr Bill nảy ra ý định bắc một cái cầu từ cửa nhà mình đến đảo giữa hồ để kinh doanh du lịch. Một vấn đề khá hóc búa đối với Mr Bill là làm thế nào xác định được khoảng cách từ nhà mình đển đảo giữa hồ?.

Có thể mô tả đảo giữa hồ như là một đa giác lồi còn nhà của Mr Bill như là một điểm nằm ngoài đa giác đó trên mặt phẳng toạ độ. Bạn hãy lập trình giúp Mr Bill tính khoảng cách nhỏ nhất từ nhà mình đến đảo.

**Input: Vào từ file văn bản BILL.INP**

* Dòng đầu tiên ghi N là số đỉnh của đa giác (N≤1000)
* Dòng thứ hai ghi toạ độ của điểm được xem như là nhà của Mr Bill
* Tiếp theo là N dòng, mỗi dòng liệt kê toạ độ của một đỉnh của đa giác. Các đỉnh của đa giác được liệt kê ngược theo chiều kim đồng hồ.

**Output: Ghi ra file BILL.OUT**

Ghi ra một số thực duy nhất là khoảng cách từ nhà của Mr Bill đến đảo giữa hồ (giữ lại 4 chữ số phần thập phân).

|  |  |
| --- | --- |
| BILL.INP | BILL.OUT |
| 3  0 0  2 0  0 2  2 2 | 1.4142 |

***https://www.hackerrank.com/hinh-hoc-2023***

***Hướng dẫn:***

* Gọi Q là nhà của Bill, p[i] là mảng chứa tọa độ đa giác, res là kết quả bài toán.
* Đầu tiên ta tính khoảng cách từ Q đến các đỉnh đa giác. Cập nhật giá trị khoảng cách nhỏ nhất. res:=min(res,khoangcach(Q,p[i]));
* Tuy nhiên khoảng cách từ đỉnh đa giác tới nhà bill chưa phải là khoảng cách ngắn nhất từ bill đến đa giác, vì thế ta phải tìm khoảng cách ngắn nhất từ Bill (điểm Q) đến từng cạnh của đa giác cụ thể như sau:

+ Gọi (d1) là đường thẳng đi qua 2 đỉnh p[i] và p[i+1] của đa giác.

(d1) có dạng: a1x+b1y+c=0

+Gọi (d2) là đường thẳng đi qua điểm Q và vuông góc với (d1).

(d2) nhận vector (a1, b1) làm vector chỉ phương.

(d2): a2x+b2y+c=0 với a2=b1, b2=-a1, c=-b1\*Q.x+a1\*Q.y;

+Tìm điểm M là tọa độ giao điểm của d1 và d2.

Nếu điểm M thuộc cạnh ( p[i], p [i+1]) đang xét thì ta tính khoảng cách từ Q đến M cập nhật lại res.

Xét vị trí tương đối giữa 2 đường thẳng (d1) và (d2)







Nếu D ≠0 thì (d1) và (d2) cắt nhau tại điểm M.x=Dx/D; M.y=Dy/D;

Nếu (D=0) và (Dx≠0) thì (d1) song song (d2)

Nếu (D=Dx=Dy) thì (d1) trùng (d2).

Độ phức tạp của thuật toán: O(n)

**Cài đặt:**

|  |
| --- |
| #include <bits/stdc++.h>  # define ll long long  # define maxn 10000  # define Eps 0.0000001  using namespace std;  ll n;  struct point  { double x,y;};  point p[maxn], Q;  void nhap()  { cin>>n;  cin>>Q.x>>Q.y;  for (int i=1; i<=n;i++)  cin>>p[i].x>>p[i].y;  }  double dist(point p1, point p2)  {  double k=(p2.x-p1.x)\*(p2.x-p1.x)+(p2.y-p1.y)\*(p2.y-p1.y);  return sqrt(k);  }  bool equal (double x, double y)  {  double k=abs(x-y);  if (k<=Eps) return true;  return false;  }  void line(point p1,point p2, double &a, double &b, double &c)  { a=p2.y-p1.y;  b=p1.x-p2.x;  c=-(a\*p1.x+b\*p1.y);  }  int ccw(point a, point b, point c)  { double k=(b.x-a.x)\*(c.y-b.y)-(b.y-a.y)\*(c.x-b.x);  if (abs(k)<=Eps) return 0;  else if (k>0) return 1;  else return -1;  }  bool giaodiem(double a1,double b1, double c1, double a2, double b2, double c2, point &M)  { double D=a1\*b2-a2\*b1;  double Dx=b1\*c2-b2\*c1;  double Dy=a1\*c2-a2\*c1;  if (equal (D,0)==true) return false ;  M.x=Dx/D;  M.y=-Dy/D;  return true;  }  bool kt(point M, point p1, point p2)  { if ((ccw(p1, M, p2)==0) && (M.x>=min(p1.x, p2.x ))&& (M.x<=max(p1.x, p2.x)) && (M.y>=min(p1.y, p2.y )) && (M.y<=max(p1.y, p2.y)))  return true; else return false;  }  void xuly()  { double a1, b1, c1, a2, b2, c2; point M;  double res=dist(Q, p[1]);  for (int i=2; i<=n; i++) res=min(res, dist(Q, p[i]));  p[n+1]=p[1];  for (int i=1; i<=n; i++)  {  line (p[i], p[i+1], a1, b1, c1);  a2=b1; b2=-a1; c2=-b1\*Q.x+a1\*Q.y;//xay dung d2;  if (giaodiem(a1, b1, c1, a2, b2, c2, M)==true)  if (kt(M, p[i], p[i+1])==true) res=min(res, dist(Q,M));  }  cout << setprecision(4)<<fixed << res;  }  int main(){  //freopen("BILL.INP", "r", stdin);  //freopen("BILL.OUT", "w", stdout);  nhap();  xuly();  return 0;} |

***Test****:* [*https://www.spoj.com/THPTCBT/problems/MTKC/*](https://www.spoj.com/THPTCBT/problems/MTKC/)

***Code****:* [*https://drive.google.com/file/d/1g2J-Ac8oeBnjsn-Bxz9enxil\_j7vOQEO/view?usp=sharing*](https://drive.google.com/file/d/1g2J-Ac8oeBnjsn-Bxz9enxil_j7vOQEO/view?usp=sharing)