Geometria Computacional (contínua)

Notas de aula da disciplina TE: Técnicas de Construção de Algoritmos

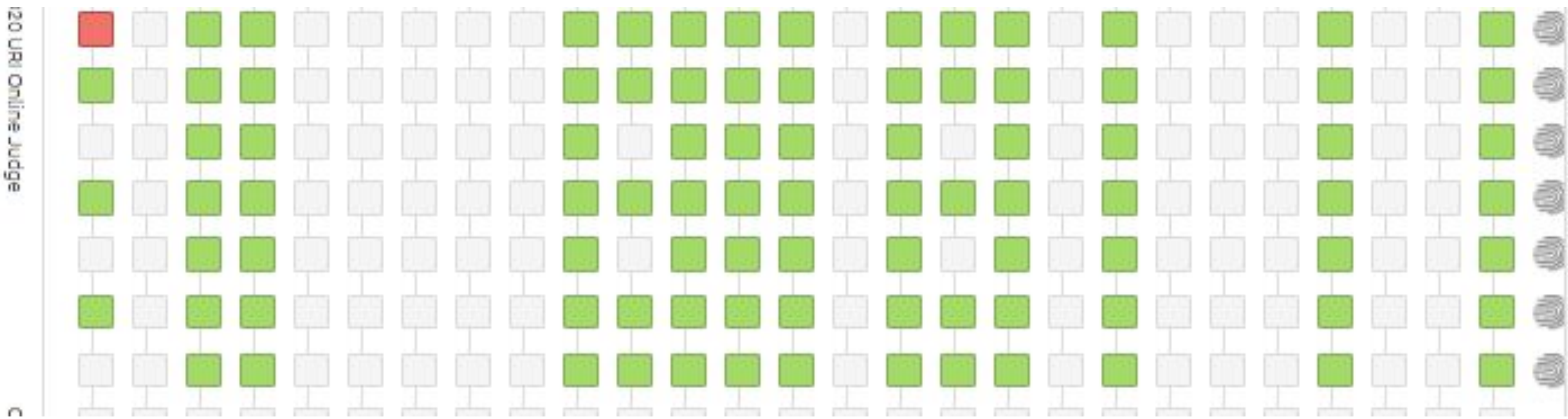
Fabiano de Souza Oliveira
(fabiano.oliveira@ime.uerj.br)

Paulo Eustáquio Duarte Pinto
(pauloedp arroba ime.uerj.br)

setembro/2020

TE: Técnicas de Construção de Algoritmos
Geometria Computacional (contínua)

Desempenho da semana passada



TE: Técnicas de Construção de Algoritmos
Geometria Computacional (contínua)

Problemas de 26/09/2020:

1015 - Distância entre dois pontos
2499 - Triângulo interno
1137 - Pontos Cocirculares
1223 - Tobogan de Bolinhas
1631 - O Fantástico Bolo de Bobby

TE: Técnicas de Construção de Algoritmos Geometria Computacional (contínua)

Dicas Gerais para C/C++:

a) Definição de PI em C/C++:

```
const double PI = acos(-1);
```

b) Usar constantes com notação de ponto flutuante em expressões desse tipo. Ex:

```
V = 4.0/3.0*PI*R*R*R;
```

c) Transformar inteiros para notação de ponto flutuante em expressões desse tipo. Ex:

```
D = r;    A = PI*D*D;
```
(D declarado como double)

TE: Técnicas de Construção de Algoritmos
Geometria Computacional (contínua)

d) Para imprimir parte decimal formatada arredondada (C++):

```
#include<iomanip>

cout<<fixed<<setprecision(3)<<x<<endl;
```

e) Comparação segura em ponto flutuante:

```
const double Z = 0.000000001;

bool MaiorIgual(double a, double b){
     return a > b || fabs(a-b) <= Z;
}

bool Igual(double a, double b){
     return fabs(a-b) <= Z;
}
```

1015 - Distância entre dois pontos

Contexto: São dados dois pontos e quer-se saber a distância entre eles.

Entrada: Um único caso de teste contendo dois números reais na primeira linha, as coordenadas do primeiro ponto e dois outros na segunda, as coordenadas do segundo.

Saída: Para cada caso de teste, imprima a distância entre os pontos, com 4 casas decimais.

Exemplo de entrada:

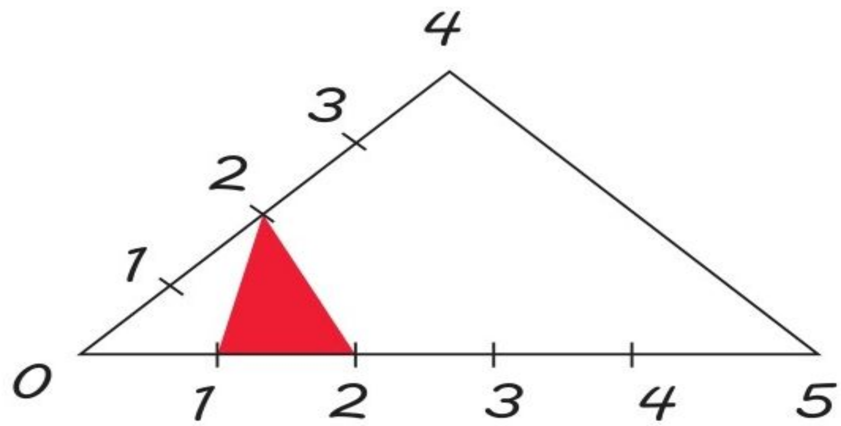
```
1.0   7.0
5.0   9.0
```

Exemplo de saída:

```
4.4721
```

2499 - Triângulo interno

Contexto: Dado um triângulo ABC, sua área S, N pontos equidistantes no lado AB e M pontos equidistantes no lado BC. Calcular a área de um triângulo que tem um vértice em AB e os dois outros em BC.



Entrada: Vários casos de teste terminados por 0 0 0. Cada caso vem em 2 linhas. Na primeira, 3 inteiros S ($1 \leq S \leq 10^6$), N ($0 \leq N \leq 1000$) e M ($0 \leq M \leq 1000$). Na segunda, o vértice C_1 em AB ($0 \leq C_1 \leq N+1$) e os vértices C_2 e C_3 ($0 \leq C_2, C_3 \leq M+1$).

Saída: Para cada teste, imprimir a área (inteira) do triângulo C1C2C3.

Exemplo de entrada:

```
4112   3   3
3  1  2
```

Exemplo de saída:

```
771
```

1137 - Pontos Cocirculares

Contexto: Dado um conjunto de pontos, determinar o maior subconjunto de pontos cocirculares.

Entrada: Vários casos de teste, terminados por uma linha com 0. Cada teste começa com uma linha contendo um inteiro N ($1 \leq N \leq 100$), o número de pontos. Depois vêm N linhas com dois inteiros X, Y ($-10^4 \leq X, Y \leq 10^4$), as coordenadas de cada ponto.

Saída: Para cada caso imprimir o número máximo de pontos cocirculares..

Exemplo de entrada:

```
7
-10   0
0   -10
10    0
0    10
-20  10
-10  20
-2    4
```

Exemplo de saída:

```
5
```

1137 - Pontos Cocirculares

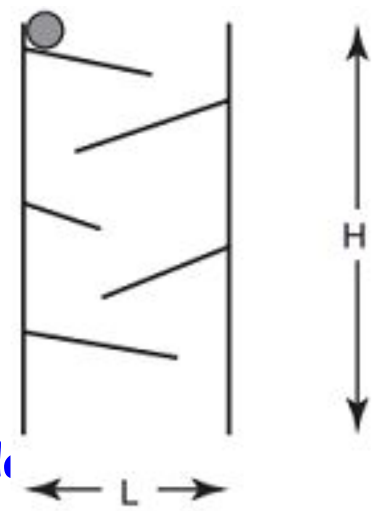
1. Para cada tripla de pontos, se os pontos não forem colineares, calcular o centro do círculo e testar quais dos demais estão no perímetro desse círculo.

2. Como testar se os pontos não são colineares: verificando se formam um triângulo de área > 0.

3. Como encontrar o centro do círculo para os pontos p1, p2, p3:

a) Formar as retas (p1, p2) e (p1, p3).
b) Traçar perpendiculares pelo meio de cada segmento e determinar a interseção pc, dessas perpendiculares.
c) O raio do círculo r = distância (p1, pc).
d) Para testar se os demais pontos pk estão no perímetro, determinar se a distância (pk, pr) = r.

1223 - Tobogan de Bolinhas



Contexto: É dado um brinquedo em forma de um tobogã pelo qual uma bolinha vai descer. Quer-se saber o diâmetro máximo da bolinha para que ela não fique presa no meio do tobogã.

Entrada: Vários casos de teste, terminados por EOF. Cad com uma linha com o inteiro N (1 ≤ N ≤ 1000). Depois vem uma linha contendo L e H (1 ≤ L, H ≤ 1000). Depois vêm N linhas com 3 inteiros: Yi, Xf, Yf, a coordenada Y do primeiro ponto da haste e as coordenadas do ponto final da haste.

Saída: O diâmetro máximo da bolinha para não ficar presa (2 decimais).

Exemplo de entrada:

3
6 10
9 2 8
6 2 5
4 3 1

Exemplo de saída:

2.00

1223 - Tobogan de Bolinhas

1. Para todas as hastes i, determinar $d1_i$ = distância mínima à parede oposta.

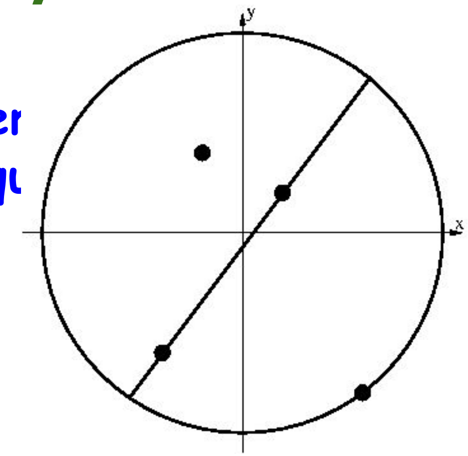
2. Para as hastes i de 1 a N-1, determinar $d2_i$ = distância do ponto final da haste i ao segmento dado pelo haste i+1.

3. Como encontrar a distância entre um ponto p1 e um segmento (p2, p3)?, sendo p2 o ponto do segmento mais perto de p1:
a) determina-se a distância de p1 à reta formada por p2 e p3
b) se esse ponto estiver no segmento ele é a distância procurada;
c) senão, a distância procurada é a distância entre p1 e p2.

O diâmetro mínimo procurado é a menor das distâncias calculadas.

1631 - O Fantástico Bolo de Bobby

Contexto: É dado um bolo com várias cerejas. Quer cortar o bolo passando por duas cerejas, de forma qu soma das distâncias das cerejas de um lado seja a próxima possível da soma das cerejas do outro.



Entrada: Vários casos de teste, terminados por 0.
Cada teste começa com uma linha com o inteiro N ($1 \leq N \leq 100$). Depois vem N linhas com as coordenadas das cerejas.

Saída: A diferença das somas para a situação de diferença mínima (3 decimais).

Exemplo de entrada:

```
3
10 10
-10 20
-20 -30
30 -40
0
```

Exemplo de saída:

```
24.000
```

1631 - O Fantástico Bolo de Bobby

1. Tentar todos os pares de cerejas.

2. Para cada par de pontos p_i, p_j, formar a reta $r(p_i, p_j)$. Para todos os demais pontos p_k verificar de que lado da reta ele está e somar convenientemente sua distância a r.

3. Como saber de que lado p_k está?

a) verificar a área sinalizada do triângulo (p_i, p_j, p_k). se for positiva está de um lado; senão do outro.

A diferença procurada é o mínimo encontrado.

FIM