

Geometría para ICPC

Training Camp Argentina 2013

Fidel I. Schaposnik (UNLP) - fidel.s@gmail.com
19 de julio de 2013

- Introducción
- Representación de objetos fundamentales
 - Puntos y vectores
 - Líneas y segmentos
- Problemas lineales en geometría computacional
 - Ecuaciones implícitas
 - Intersección de planos en 3D
- Técnicas de barrido
 - *A Safe Bet*
 - *November Rain*
- Problemas para practicar

¿En qué consiste un problema de geometría computacional?

Generalmente, los problemas de geometría en el ICPC requieren calcular alguna cantidad (e.g. distancia, área, parámetros óptimos para una configuración determinada, etc) relacionada con elementos geométricos como puntos, líneas, círculos, y demás. La solución de un problema de geometría involucra dos pasos:

- 1 Representar los objetos geométricos involucrados en el problema, de modo de poder operar con ellos.
- 2 Desarrollar un algoritmo utilizando lo anterior para calcular la respuesta buscada.

Para poder dedicarnos a la parte “interesante”, debemos primero asegurarnos de que el primer punto no es un impedimento.

Introducción (cont.)

A FAVOR: Podemos *visualizar* los algoritmos fácilmente.

EN CONTRA: Debemos evitar nuevos *tipos* de problemas (errores numéricos, casos degenerados, código más largo, etc).

A FAVOR: Podemos *visualizar* los algoritmos fácilmente.

EN CONTRA: Debemos evitar nuevos *tipos* de problemas (errores numéricos, casos degenerados, código más largo, etc).

Antes de empezar, algunas consideraciones generales

- A lo largo de esta clase nos vamos a concentrar casi exclusivamente en problemas de geometría en 2D.
- La **práctica** y las **buenas costumbres** son todavía más importantes que en los problemas “convencionales”.
- ¡Los problemas de geometría muchas veces sirven para distinguir a los buenos equipos de los excelentes!

Puntos y vectores

Un punto en el plano (o un vector desde el origen hasta dicho punto) se puede representar con un par ordenado de coordenadas en un sistema cartesiano:

$$\vec{P} = (x, y) \quad \text{con} \quad x, y \in \mathbb{R}$$

Algunas operaciones entre vectores

- La suma y resta de vectores se realiza componente a componente:

$$\vec{P} = \vec{P}_1 \pm \vec{P}_2 \quad \iff \quad (x, y) = (x_1 \pm x_2, y_1 \pm y_2)$$

- La longitud o *norma* de un vector es $|\vec{P}| = \sqrt{x^2 + y^2}$.
- La distancia euclídea entre dos puntos \vec{P}_1 y \vec{P}_2 es $|\vec{P}_1 - \vec{P}_2|$
- El producto escalar de dos vectores es un número, y se define como

$$\vec{P}_1 \cdot \vec{P}_2 := x_1x_2 + y_1y_2 = |\vec{P}_1||\vec{P}_2| \cos \theta$$

Observar que es la proyección de un vector sobre otro, y en particular si $\theta = 90^\circ$ el producto escalar se anula.

- El producto vectorial de dos vectores es un vector en la dirección normal al plano generado por ellos. En dos dimensiones nos puede interesar su única componente no nula,

$$\left(\vec{P}_1 \times \vec{P}_2\right)_z = x_1y_2 - x_2y_1 \equiv \vec{P}_1 \wedge \vec{P}_2$$

Observar que

$$|\vec{P}_1 \times \vec{P}_2| = |\vec{P}_1||\vec{P}_2| \sin \theta,$$

es decir el área del paralelogramo formado por los vectores y sus traslaciones paralelas.

En particular si $\theta = 0^\circ$ ó $\theta = 180^\circ$ el producto vectorial se anula.

Puntos y vectores (código.)

```
1 struct pt {
2     double x, y;
3     pt(double xx=0.0, double yy=0.0) { x=xx; y=yy; }
4 };
5
6 pt operator+(const pt &p1, const pt &p2) {
7     return pt(p1.x+p2.x, p1.y+p2.y); }
8
9 pt operator-(const pt &p1, const pt &p2) {
10    return pt(p1.x-p2.x, p1.y-p2.y); }
11
12 double operator*(const pt &p1, const pt &p2) {
13     return p1.x*p2.x + p1.y*p2.y; }
14
15 double operator^(const pt &p1, const pt &p2) {
16     return p1.x*p2.y - p1.y*p2.x; }
17
18 double norm(const pt &p) { return sqrt(p*p); }
19
20 double dist(const pt &p1, const pt &p2) {
21     return norm(p1-p2); }
```

Estructura y operaciones elementales con puntos

Líneas y segmentos

Una línea o un segmento se pueden representar de varias formas distintas:

- Con dos puntos \vec{P}_1 y \vec{P}_2 sobre la línea (los extremos del segmento)
- Con un punto \vec{P}_0 sobre la línea y un vector director \vec{V}
- Mediante una ecuación implícita $ax + by + c = 0$.

Se puede pasar de una a otra representación trivialmente, y la elección de una u otra depende en definitiva del uso que se vaya a darle.

La representación $\{\vec{P}_0, \vec{V}\}$ se conoce como paramétrica, porque nos permite recorrer todos los puntos de la línea o segmento con

$$\vec{P}(t) = \vec{P}_0 + t\vec{V} \quad \text{con} \quad \begin{cases} t \in \mathbb{R} & \text{línea} \\ t \in [0, 1] & \text{segmento} \end{cases}$$

Líneas y segmentos (cont.)

```
1 struct line {
2     double a, b, c;
3     line(double aa=0.0, double bb=0.0, double cc=0.0) {
4         a=aa; b=bb; c=cc;
5     }
6 };
7
8 double dist(const pt &p, const line &l) {
9     return ABS(l.a*p.x+l.b*p.y+l.c)/sqrt(SQ(l.a)+SQ(l.b)); }
10
11 line line_pp(const pt &p1, const pt &p2) {
12     return line(p2.y-p1.y, p1.x-p2.x, p2^p1); }
13
14 line line_perp_p(const line &l, const pt &p) {
15     return line(-l.b, l.a, l.b*p.x - l.a*p.y); }
16
17 line mediatriz(const pt &p1, const pt &p2) {
18     return line_perp_p(line_pp(p1, p2), (p1+p2)/2.0); }
```

Ejemplo de uso de la representación implícita

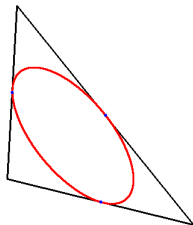
Líneas y segmentos (cont.)

```
1 line bisectriz(const pt &p1, const pt &pc, const pt &p2) {
2     pt pc1 = p1-pc, pc2 = p2-pc;
3
4     if (ABS(pc1^pc2) < EPS) {
5         if (pc1*pc2 > ZERO) return line_pp(pc, p1);
6         else return line_perp_p(line_pp(p1, p2), pc);
7     }
8     return line_pp(pc, pc+(pc1/norm(pc1)+pc2/norm(pc2))/2.0);
9 }
10
11 int inter_ll(const line &l1, const line &l2, pt &p) {
12     double det = l1.a*l2.b - l1.b*l2.a;
13
14     if (ABS(det) < EPS) {
15         if (ABS(l1.a*l2.c - l1.c*l2.a) < EPS) return -1;
16         else return 0;
17     }
18     p.x = (l1.b*l2.c - l2.b*l1.c)/det;
19     p.y = (l2.a*l1.c - l1.a*l2.c)/det;
20     return 1;
21 }
```

Ejemplo de uso de la representación implícita (cont.)

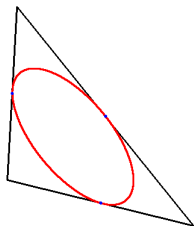
Problemas lineales en geometría computacional

A veces aparecen sistemas de ecuaciones en problemas de geometría computacional: *Joe's Triangular Gardens* (NA-GNY'08) pide hallar la elipse tangente a un triángulo en los puntos medios de sus lados:



Problemas lineales en geometría computacional

A veces aparecen sistemas de ecuaciones en problemas de geometría computacional: *Joe's Triangular Gardens* (NA-GNY'08) pide hallar la elipse tangente a un triángulo en los puntos medios de sus lados:



Una elipse queda definida por $ax^2 + bxy + cy^2 + dx + ey + f = 0$ con $b^2 - 4ac < 0$, de modo que tenemos 5 parámetros que definen la elipse (a, b, c, d, e y f , a menos de una normalización).

Problemas lineales en geometría computacional (cont.)

Si $\vec{P}_i^m = (x_i^m, y_i^m)$ con $i = 1, 2, 3$ son los puntos medios de los lados del triángulo, tenemos 3 ecuaciones de intersección

$$a(x_i^m)^2 + b x_i^m y_i^m + c(y_i^m)^2 + d x_i^m + e y_i^m + f = 0$$

y 2 ecuaciones de tangencia (derivando implícitamente para dos lados no verticales)

$$2a x_i^m + b(y_i^m + x_i^m y'(x_i^m, y_i^m)) + 2c y_i^m y'(x_i^m, y_i^m) + d + e y'(x_i^m, y_i^m) = 0$$

Los valores de las derivadas son simplemente las pendientes de los correspondientes lados del triángulo: $y'(x_i^m, y_i^m) = \frac{\Delta y_i}{\Delta x_i}$.

Problemas lineales en geometría computacional (cont.)

Si $\vec{P}_i^m = (x_i^m, y_i^m)$ con $i = 1, 2, 3$ son los puntos medios de los lados del triángulo, tenemos 3 ecuaciones de intersección

$$a(x_i^m)^2 + b x_i^m y_i^m + c(y_i^m)^2 + d x_i^m + e y_i^m + f = 0$$

y 2 ecuaciones de tangencia (derivando implícitamente para dos lados no verticales)

$$2a x_i^m + b(y_i^m + x_i^m y'(x_i^m, y_i^m)) + 2c y_i^m y'(x_i^m, y_i^m) + d + e y'(x_i^m, y_i^m) = 0$$

Los valores de las derivadas son simplemente las pendientes de los correspondientes lados del triángulo: $y'(x_i^m, y_i^m) = \frac{\Delta y_i}{\Delta x_i}$.

Si resolvemos el sistema de 5 ecuaciones con 5 incógnitas, la solución al problema está prácticamente dada.

Problemas lineales en geometría computacional (cont.)

Otro ejemplo: consideremos la intersección de dos planos en tres dimensiones.

Problemas lineales en geometría computacional (cont.)

Otro ejemplo: consideremos la intersección de dos planos en tres dimensiones.

Un plano se define con un punto \vec{P}_0 contenido en el plano y un vector \vec{N}_0 ortogonal al mismo. Todos los otros puntos satisfacen

$$(\vec{P} - \vec{P}_0) \cdot \vec{N}_0 = 0$$

lo cual conduce a la ecuación más familiar

$$ax + by + cz + d = 0 \quad \text{para} \quad \vec{N}_0 = (a, b, c) \quad \text{y} \quad d = -\vec{P}_0 \cdot \vec{N}_0$$

En general, dos planos definidos por $\{\vec{P}_1, \vec{N}_1\}$ y $\{\vec{P}_2, \vec{N}_2\}$ se intersecan para dar una línea. **¿Cómo la hallamos?**

Otros problemas lineales (cont.)

Usamos la representación paramétrica de la línea, de modo que buscamos un punto \vec{P} sobre ella y un vector director \vec{V} .

Otros problemas lineales (cont.)

Usamos la representación paramétrica de la línea, de modo que buscamos un punto \vec{P} sobre ella y un vector director \vec{V} .

Nuestra línea debe pertenecer a ambos planos, luego es ortogonal a los vectores \vec{N}_1 y \vec{N}_2 . Entonces

$$\vec{V} = \vec{N}_1 \times \vec{N}_2$$

Si $\vec{V} = \vec{0}$, los planos son paralelos y no hay intersección entre ellos.
¿Cómo hallamos \vec{P} ?

Otros problemas lineales (cont.)

Usamos la representación paramétrica de la línea, de modo que buscamos un punto \vec{P} sobre ella y un vector director \vec{V} .

Nuestra línea debe pertenecer a ambos planos, luego es ortogonal a los vectores \vec{N}_1 y \vec{N}_2 . Entonces

$$\vec{V} = \vec{N}_1 \times \vec{N}_2$$

Si $\vec{V} = \vec{0}$, los planos son paralelos y no hay intersección entre ellos.

¿Cómo hallamos \vec{P} ?

Sea \vec{P} el punto sobre la línea que está más cerca de cierto punto fijo y arbitrario (e.g. el origen). Entonces $\vec{P} = (x, y, z)$ minimiza

$$D^2 = x^2 + y^2 + z^2$$

mientras está en ambos planos, *i.e.* satisfaciendo

$$(\vec{P} - \vec{P}_1) \cdot \vec{N}_1 = 0 \quad \text{y} \quad (\vec{P} - \vec{P}_2) \cdot \vec{N}_2 = 0$$

Otros problemas lineales (cont.)

Podemos hallar el punto \vec{P} minimizando D^2 bajo estas restricciones, usando multiplicadores de Lagrange. Definimos

$$f(x, y, z, \mu_1, \mu_2) = x^2 + y^2 + z^2 + \mu_1 (\vec{P} - \vec{P}_1) \cdot \vec{N}_1 + \mu_2 (\vec{P} - \vec{P}_2) \cdot \vec{N}_2$$

Luego debemos exigir

$$\frac{\partial f}{\partial x} = 2x + \mu_1 x_{N_1} = 0 \quad \frac{\partial f}{\partial y} = 2y + \mu_1 y_{N_1} = 0 \quad \frac{\partial f}{\partial z} = 2z + \mu_1 z_{N_1} = 0$$

$$\frac{\partial f}{\partial \mu_1} = (\vec{P} - \vec{P}_1) \cdot \vec{N}_1 = 0 \quad \frac{\partial f}{\partial \mu_2} = (\vec{P} - \vec{P}_2) \cdot \vec{N}_2 = 0$$

Otros problemas lineales (cont.)

Podemos hallar el punto \vec{P} minimizando D^2 bajo estas restricciones, usando multiplicadores de Lagrange. Definimos

$$f(x, y, z, \mu_1, \mu_2) = x^2 + y^2 + z^2 + \mu_1 (\vec{P} - \vec{P}_1) \cdot \vec{N}_1 + \mu_2 (\vec{P} - \vec{P}_2) \cdot \vec{N}_2$$

Luego debemos exigir

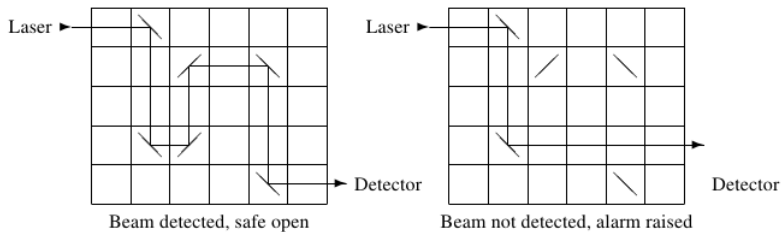
$$\frac{\partial f}{\partial x} = 2x + \mu_1 x_{N_1} = 0 \quad \frac{\partial f}{\partial y} = 2y + \mu_1 y_{N_1} = 0 \quad \frac{\partial f}{\partial z} = 2z + \mu_1 z_{N_1} = 0$$

$$\frac{\partial f}{\partial \mu_1} = (\vec{P} - \vec{P}_1) \cdot \vec{N}_1 = 0 \quad \frac{\partial f}{\partial \mu_2} = (\vec{P} - \vec{P}_2) \cdot \vec{N}_2 = 0$$

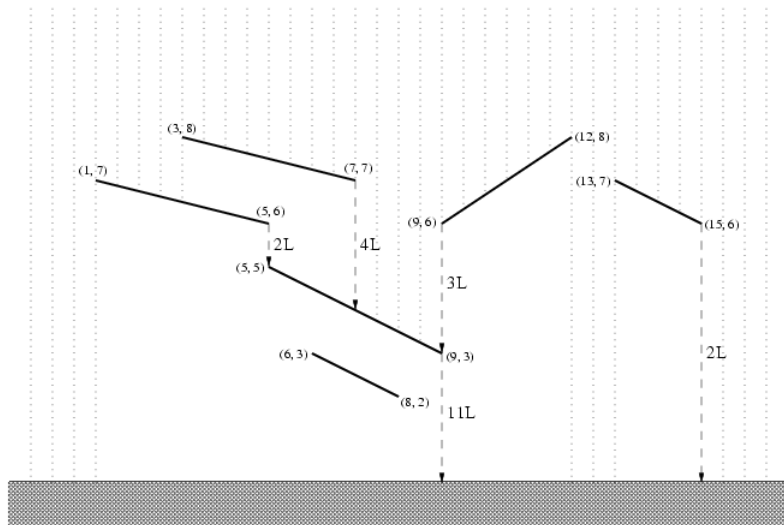
O, en notación matricial

$$\begin{pmatrix} 2 & 0 & 0 & x_{N_1} & x_{N_2} \\ 0 & 2 & 0 & y_{N_1} & y_{N_2} \\ 0 & 0 & 2 & z_{N_1} & z_{N_2} \\ x_{N_1} & y_{N_1} & z_{N_1} & 0 & 0 \\ x_{N_2} & y_{N_2} & z_{N_2} & 0 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ \mu_1 \\ \mu_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \vec{P}_1 \cdot \vec{N}_1 \\ \vec{P}_2 \cdot \vec{N}_2 \end{pmatrix}$$

Técnicas de barrido: *A Safe Bet*



Técnicas de barrido: *November Rain*



Problemas para practicar

- *In-circles Again* - LA 4714
- *Shortest Flight Path* - LA 6035
- *Coverage* - LA 4562
- *Malfatti Circles* - LA 4642
- *Onion Layers* - LA 3655
- *Deer-Proof Fence* - LA 4450
- *Watering Plants* - GCJ 2009, Round 2
- *High Mountains* - SPOJ TAP2012H
- *Garden Fence* - LA 5795

¡Gracias!