Geometria

Juan Pablo Cabaña

Universidad Tecnológica Nacional - Facultad Regional Santa Fe

Training Camp 2025

Gracias Sponsors!

Organizador

Diamond Plus

Gracias Sponsors!

Platinum

FOLDER IT

INTERNATIONAL
SOFTWARE COMPANY

Gold

NeuralSoft

JS JERÁRQUICOS

Gracias Sponsors!

Aliado

Santa Fe
PROVINCIA

Presentacion

Temario

1. Punto/Vector
2. Polar Sort - Convex Hull
3. Línea
4. Segmento

Temario

1. Punto/Vector
2. Polar Sort - Convex Hull
3. Línea
4. Segmento

Punto/Vector: Definiciones

Vector de N componentes

Conjunto ordenado de n numeros:

$$(a_1, ..., a_N)$$

R^N

Conjunto de todos los vectores de dimension N $(a_1, ..., a_N)$, donde a_i es un numero real. Nos vamos a centrar en R^2, es decir los vectores de pares ordenados (a_1, a_2).

Punto/Vector: Definiciones

Interpretacion geometrica

- Como los puntos del plano xy estan descritos por un par ordenado (x, y) podemos considerarlos equivalentes a los vectores de R^2.
- Tambien pueden considerarse a los vectores de R^2 como una entidad con magnitud y direccion.

Punto/Vector: Operaciones

¿Que podemos hacer con vectores?

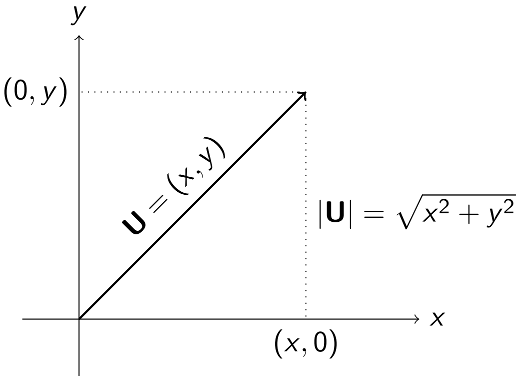
- Norma de un vector U.
- Suma/Resta de dos vectores U y V.
- Multiplicar/Dividir un vector U por un numero K.
- Producto punto entre dos vectores U y V.
- Producto cruz entre dos vectores U y V.
- Proyectar un vector U sobre un vector V.

Punto/Vector: Operaciones

Norma de un vector

Largo o extension de un vector U en el espacio:

$$|U| = \sqrt{x^2 + y^2}$$



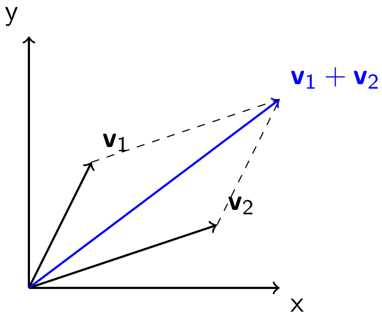
Punto/Vector: Operaciones

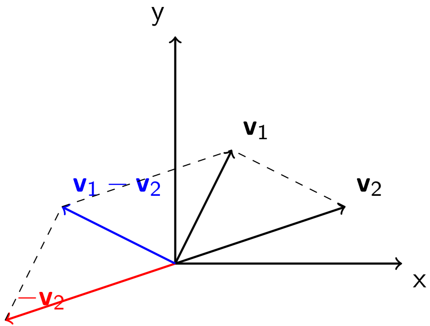
Suma de dos vectores

$$\begin{pmatrix} x_1 \\ y_1 \end{pmatrix} + \begin{pmatrix} x_2 \\ y_2 \end{pmatrix} = \begin{pmatrix} x_1 + x_2 \\ y_1 + y_2 \end{pmatrix}$$

Resta de dos vectores

$$\begin{pmatrix} x_1 \\ y_1 \end{pmatrix} - \begin{pmatrix} x_2 \\ y_2 \end{pmatrix} = \begin{pmatrix} x_1 - x_2 \\ y_1 - y_2 \end{pmatrix}$$





Punto/Vector: Operaciones

Distancia entre dos puntos

Punto/Vector: Operaciones

Distancia entre dos puntos

- Dado dos puntos U y V, el vector que va de V a U es U-V.

Punto/Vector: Operaciones

Distancia entre dos puntos

- Dado dos puntos U y V, el vector que va de V a U es U-V.
- Dado un vector, ya sabemos calcular su longitud.

Punto/Vector: Operaciones

Distancia entre dos puntos

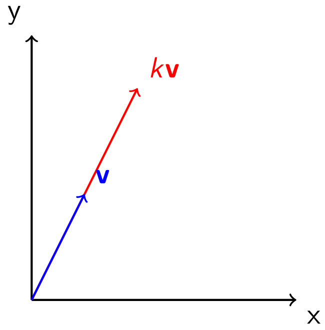
- Dado dos puntos U y V, el vector que va de V a U es U-V.
- Dado un vector, ya sabemos calcular su longitud.

$$|U - V|$$

Punto/Vector: Operaciones

Multiplicar/Dividir por un escalar

$$k * \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} k * x \\ k * y \end{pmatrix}$$



Punto/Vector: Operaciones

Producto punto de dos vectores

$$\begin{pmatrix} x_1 \\ y_1 \end{pmatrix} * \begin{pmatrix} x_2 \\ y_2 \end{pmatrix} = x_1 * x_2 + y_1 * y_2$$

Teorema

Sean U y V vectores en R^2 diferentes del vector cero, y sea α el angulo formado entre U y V, entonces:

$$U * V = |U||V| * cos(\alpha)$$

Punto/Vector: Operaciones

Interpretacion geometrica

Dado un vector U, que conjunto de puntos tienen igual producto punto con U?

Punto/Vector: Operaciones

Interpretacion geometrica

Dado un vector U, que conjunto de puntos tienen igual producto punto con U?

$$\begin{pmatrix} x \\ y \end{pmatrix} * \begin{pmatrix} x_u \\ y_u \end{pmatrix} = k$$

$$x * x_u + y * y_u = k$$

Punto/Vector: Operaciones

Interpretacion geometrica

Dado un vector U, que conjunto de puntos tienen igual producto punto con U?

$$\begin{pmatrix} x \\ y \end{pmatrix} * \begin{pmatrix} x_u \\ y_u \end{pmatrix} = k$$

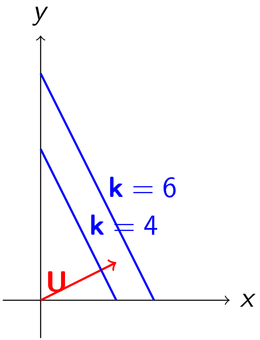
$$x * x_u + y * y_u = k$$

Al fijar el valor del producto punto (K), obtenemos una linea perpendicular al vector U.

Punto/Vector: Operaciones

Interpretacion geometrica

Podemos interpretar el producto punto entre una direccion U y un punto V, como una medida de "que tan lejos esta el punto V en la direccion U".



Punto/Vector: Operaciones

Algunas observaciones

Dados dos vectores de norma conocida:

- Que valores puede tomar el producto punto de ellos?

Punto/Vector: Operaciones

Algunas observaciones

Dados dos vectores de norma conocida:

- Que valores puede tomar el producto punto de ellos?
  $U * V = |U||V| * cos(\alpha) \rightarrow -|U||V| \leq U * V \leq |U||V|$

Punto/Vector: Operaciones

Algunas observaciones

Dados dos vectores de norma conocida:

- Que valores puede tomar el producto punto de ellos?
  $U * V = |U||V| * cos(\alpha) \rightarrow -|U||V| \leq U * V \leq |U||V|$
- En que direccion el producto punto es maximo y minimo?

Algunas observaciones

Dados dos vectores de norma conocida:

- Que valores puede tomar el producto punto de ellos?
  $U * V = |U||V| * cos(\alpha) \rightarrow -|U||V| \leq U * V \leq |U||V|$
- En que direccion el producto punto es maximo y minimo?
  Alcanza un maximo cuando U y V tienen igual direccion ($cos(\alpha) = 1$) y un minimo cuando tienen direccion opuesta ($cos(\alpha) = -1$)

Punto/Vector: Operaciones

Aplicaciones del producto punto

- Determinar si dos vectores son perpendiculares.

Punto/Vector: Operaciones

Aplicaciones del producto punto

- Determinar si dos vectores son perpendiculares.
- Ordenar un conjunto de puntos en una cierta direccion U.

Punto/Vector: Operaciones

Aplicaciones del producto punto

- Determinar si dos vectores son perpendiculares.
- Ordenar un conjunto de puntos en una cierta direccion U.
- Encontrar el punto mas cercano/lejano en una direccion U.

Punto/Vector: Operaciones

Aplicaciones del producto punto

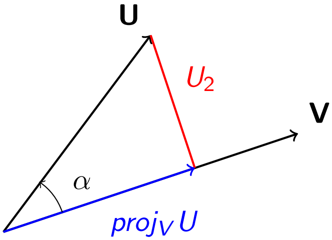
- Determinar si dos vectores son perpendiculares.
- Ordenar un conjunto de puntos en una cierta direccion U.
- Encontrar el punto mas cercano/lejano en una direccion U.
- Resolver problemas de optimizacion.

Proyeccion de un vector U sobre un vector V

$$proj_V U = V * \frac{U * V}{V * V}$$

Interpretacion geometrica

Es la proyeccion ortogonal del vector U sobre la linea que contiene al vector V.



Punto/Vector: Operaciones

Aplicaciones de la proyeccion de vectores

- Dado un punto P y una linea/segmento L, encontrar el punto en L mas cercano a P.

Punto/Vector: Operaciones

Producto cruz

$$\begin{pmatrix} x_1 \\ y_1 \end{pmatrix} x \begin{pmatrix} x_2 \\ y_2 \end{pmatrix} = x_1 * y_2 - x_2 * y_1$$

Teorema

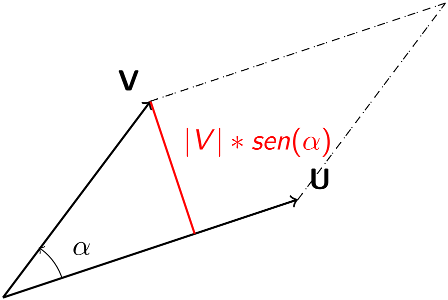
Sean U y V vectores en R^2 diferentes del vector cero, y sea α el angulo que va de U a V, entonces:

$$UxV = |U||V| * sen(\alpha)$$

Punto/Vector: Operaciones

Interpretacion geometrica

Podemos interpretar el producto cruz entre dos vectores U y V como el "area con signo" del paralelogramo que tiene como lados a U y V.



Punto/Vector: Operaciones

Aplicaciones del producto cruz

- Determinar si dos vectores son paralelos.

Punto/Vector: Operaciones

Aplicaciones del producto cruz

- Determinar si dos vectores son paralelos.
- Dado un vector U y un punto P, saber si P esta a la izquierda de U.

Punto/Vector: Operaciones

Aplicaciones del producto cruz

- Determinar si dos vectores son paralelos.
- Dado un vector U y un punto P, saber si P esta a la izquierda de U.
- Dados dos vectores, calcular el area del paralelogramo que los tiene como lados.
- Dados tres puntos, calcular el area del triangulo formado.

Punto/Vector: Implementacion

```
typedef long long T;
typedef double ld;
struct pto {
    T x, y;
    pto() : x(0), y(0) {}
    pto(T _x, T _y) : x(_x), y(_y) {}
    pto operator+(pto b) { return pto(x+b.x, y+b.y); }
    pto operator-(pto b) { return pto(x-b.x, y-b.y); }
    pto operator+(T k) { return pto(x+k, y+k); }
    pto operator*(T k) { return pto(x*k, y*k); }
    pto operator/(T k) { return pto(x/k, y/k); }
    T operator*(pto b) { return x*b.x + y*b.y; }
    pto proj(pto b) { return b*((*this)*b) / (b*b); }
    T operator^(pto b) { return x*b.y - y*b.x; }
    ld norm() { return sqrt(x*x + y*y); }
    ld dist(pto b) { return (b - (*this)).norm(); }
    bool left(pto a, pto b) { return ((a-*this) ^ (b-*this)) > 0; }
};
```

Punto/Vector: Geometry Game

Enunciado

Dado cuatro puntos, determinar que figura forman:

- Cuadrado: tiene cuatro angulos rectos y cuatro lados iguales.

Punto/Vector: Geometry Game

Enunciado

Dado cuatro puntos, determinar que figura forman:

- Cuadrado: tiene cuatro angulos rectos y cuatro lados iguales.
- Rectangulo: tiene cuatro angulos rectos.

Punto/Vector: Geometry Game

Enunciado

Dado cuatro puntos, determinar que figura forman:

- Cuadrado: tiene cuatro angulos rectos y cuatro lados iguales.
- Rectangulo: tiene cuatro angulos rectos.
- Rombo: tiene cuatro lados de igual largo.

Punto/Vector: Geometry Game

Enunciado

Dado cuatro puntos, determinar que figura forman:

- Cuadrado: tiene cuatro angulos rectos y cuatro lados iguales.
- Rectangulo: tiene cuatro angulos rectos.
- Rombo: tiene cuatro lados de igual largo.
- Paralelogramo: tiene dos pares de lados paralelos.

Punto/Vector: Geometry Game

Enunciado

Dado cuatro puntos, determinar que figura forman:

- Cuadrado: tiene cuatro angulos rectos y cuatro lados iguales.
- Rectangulo: tiene cuatro angulos rectos.
- Rombo: tiene cuatro lados de igual largo.
- Paralelogramo: tiene dos pares de lados paralelos.
- Trapecio: tiene un par de lados paralelos.

Punto/Vector: Geometry Game

Enunciado

Dado cuatro puntos, determinar que figura forman:

- Cuadrado: tiene cuatro angulos rectos y cuatro lados iguales.
- Rectangulo: tiene cuatro angulos rectos.
- Rombo: tiene cuatro lados de igual largo.
- Paralelogramo: tiene dos pares de lados paralelos.
- Trapecio: tiene un par de lados paralelos.
- Romboide: tiene simetría de reflexión a través de una diagonal.

Punto/Vector: Geometry Game

Enunciado

Dado cuatro puntos, determinar que figura forman:

- Cuadrado: tiene cuatro angulos rectos y cuatro lados iguales.
- Rectangulo: tiene cuatro angulos rectos.
- Rombo: tiene cuatro lados de igual largo.
- Paralelogramo: tiene dos pares de lados paralelos.
- Trapecio: tiene un par de lados paralelos.
- Romboide: tiene simetría de reflexión a través de una diagonal.
- Ninguno.

Temario

1. Punto/Vector
2. Polar Sort - Convex Hull
3. Línea
4. Segmento

Polar Sort: Definicion

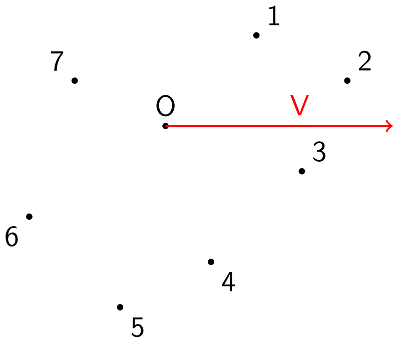
Polar Sort

Dado un punto O y un vector V, el polar sort de un conjunto de puntos S es el ordenamiento de los puntos de S, en sentido antihorario, tomando a O como centro y V como direccion inicial.

Polar Sort: Definicion

Polar Sort

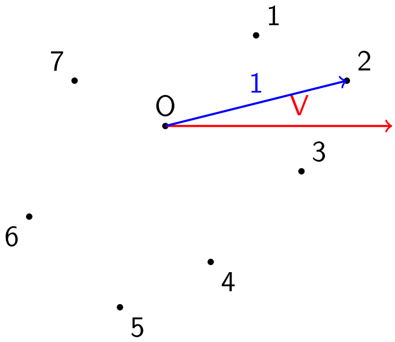
Dado un punto O y un vector V, el polar sort de un conjunto de puntos S es el ordenamiento de los puntos de S, en sentido antihorario, tomando a O como centro y V como direccion inicial.



Polar Sort: Definicion

Polar Sort

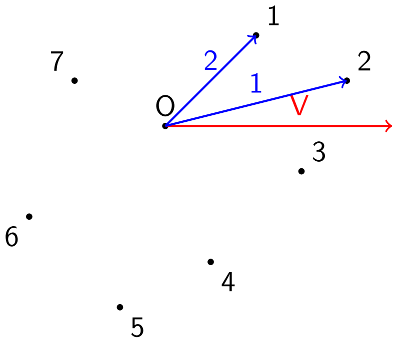
Dado un punto O y un vector V, el polar sort de un conjunto de puntos S es el ordenamiento de los puntos de S, en sentido antihorario, tomando a O como centro y V como direccion inicial.



Polar Sort: Definicion

Polar Sort

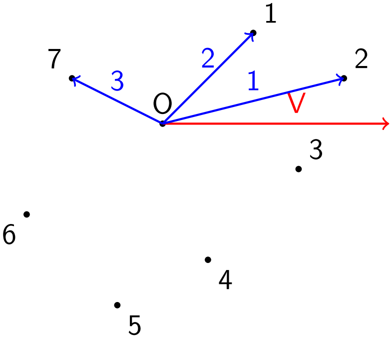
Dado un punto O y un vector V, el polar sort de un conjunto de puntos S es el ordenamiento de los puntos de S, en sentido antihorario, tomando a O como centro y V como direccion inicial.



Polar Sort: Definicion

Polar Sort

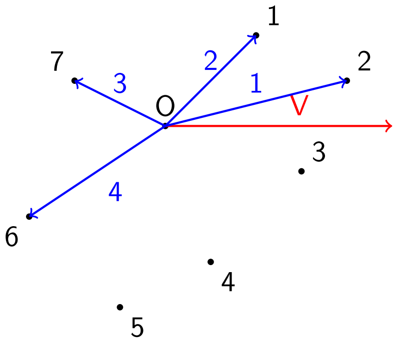
Dado un punto O y un vector V, el polar sort de un conjunto de puntos S es el ordenamiento de los puntos de S, en sentido antihorario, tomando a O como centro y V como direccion inicial.



Polar Sort: Definicion

Polar Sort

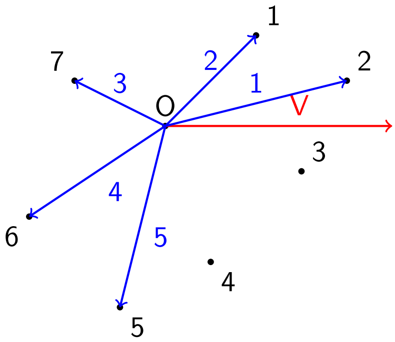
Dado un punto O y un vector V, el polar sort de un conjunto de puntos S es el ordenamiento de los puntos de S, en sentido antihorario, tomando a O como centro y V como direccion inicial.



Polar Sort: Definicion

Polar Sort

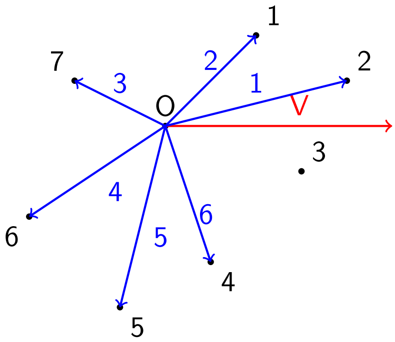
Dado un punto O y un vector V, el polar sort de un conjunto de puntos S es el ordenamiento de los puntos de S, en sentido antihorario, tomando a O como centro y V como direccion inicial.



Polar Sort: Definicion

Polar Sort

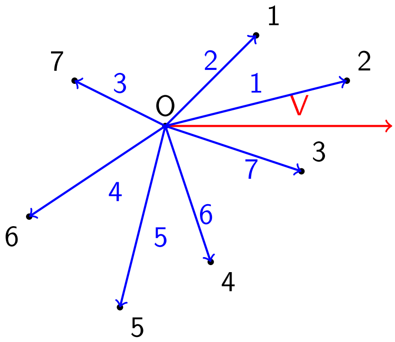
Dado un punto O y un vector V, el polar sort de un conjunto de puntos S es el ordenamiento de los puntos de S, en sentido antihorario, tomando a O como centro y V como direccion inicial.



Polar Sort: Definicion

Polar Sort

Dado un punto O y un vector V, el polar sort de un conjunto de puntos S es el ordenamiento de los puntos de S, en sentido antihorario, tomando a O como centro y V como direccion inicial.



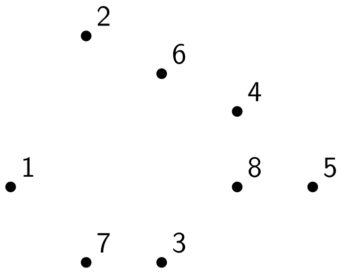
Polar Sort: Implementacion

```
struct cmp {
    pto o, v;
    cmp(pto no, pto nv) : o(no), v(nv) {}
    bool half(pto p) {
        assert(!(p.x == 0 && p.y == 0));  // (0,0) isn't well defined
        return (v ^ p) < 0 || ((v ^ p) == 0 && (v * p) < 0);
    }
    bool operator()(pto& p1, pto& p2) {
        return mp(half(p1 - o), T(0)) < mp(half(p2 - o), ((p1 - o) ^ (p2 - o)))
    }
};
```

Convex Hull: Definicion

Convex Hull

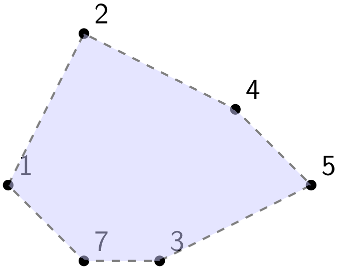
- Un conjunto de puntos es convexo si contiene todos los segmentos entre todo par de puntos del conjunto.
- El convex hull de un conjunto de puntos S es el conjunto convexo minimo que contiene a S.



Convex Hull: Definicion

Convex Hull

- Un conjunto de puntos es convexo si contiene todos los segmentos entre todo par de puntos del conjunto.
- El convex hull de un conjunto de puntos S es el conjunto convexo minimo que contiene a S.



Convex Hull: Algoritmo

Algoritmo

Hallar el upper hull y el lower hull por separado, luego unirlos.

Convex Hull: Algoritmo

Algoritmo

Hallar el upper hull y el lower hull por separado, luego unirlos.
Veamos como hallar el upper hull (UH):

- Ordenar los puntos por x, luego por y.

Algoritmo

Hallar el upper hull y el lower hull por separado, luego unirlos.
Veamos como hallar el upper hull (UH):

- Ordenar los puntos por x, luego por y.
- Recorrer los puntos ordenados:

Convex Hull: Algoritmo

Algoritmo

Hallar el upper hull y el lower hull por separado, luego unirlos.
Veamos como hallar el upper hull (UH):

- Ordenar los puntos por x, luego por y.
- Recorrer los puntos ordenados:
  - Si $|UH| \geq 2$, sean U y V los dos ultimos puntos agregados. Mientras $\overrightarrow{UV} x \overrightarrow{UP} \geq 0$ borrar V de UH.

Convex Hull: Algoritmo

Algoritmo

Hallar el upper hull y el lower hull por separado, luego unirlos.
Veamos como hallar el upper hull (UH):

- Ordenar los puntos por x, luego por y.
- Recorrer los puntos ordenados:
    - Si $|UH| \geq 2$, sean U y V los dos ultimos puntos agregados. Mientras $\overrightarrow{UV} x \overrightarrow{UP} \geq 0$ borrar V de UH.
    - Agregar P a UH.

Convex Hull: Algoritmo

Algoritmo

Hallar el upper hull y el lower hull por separado, luego unirlos.
Veamos como hallar el upper hull (UH):

- Ordenar los puntos por x, luego por y.
- Recorrer los puntos ordenados:
  - Si $|UH| \geq 2$, sean U y V los dos ultimos puntos agregados. Mientras $\overrightarrow{UV} x \overrightarrow{UP} \geq 0$ borrar V de UH.
  - Agregar P a UH.

Para hallar el lower hull solo hay que reemplazar $\overrightarrow{UV} x \overrightarrow{UP} \geq 0$ por $\overrightarrow{UV} x \overrightarrow{UP} \leq 0$.

Convex Hull: Algoritmo

Algoritmo

Hallar el upper hull y el lower hull por separado, luego unirlos.
Veamos como hallar el upper hull (UH):

- Ordenar los puntos por x, luego por y.
- Recorrer los puntos ordenados:
  - Si $|UH| \geq 2$, sean U y V los dos ultimos puntos agregados. Mientras $\overrightarrow{UV} x \overrightarrow{UP} \geq 0$ borrar V de UH.
  - Agregar P a UH.

Para hallar el lower hull solo hay que reemplazar $\overrightarrow{UV} x \overrightarrow{UP} \geq 0$ por $\overrightarrow{UV} x \overrightarrow{UP} \leq 0$.

Complejidad

Ordenar los puntos lleva $O(N * log(N))$, recorrer todos los puntos $O(N)$, entonces $O(N * log(N) + N) = O(N * log(N))$.

Convex Hull: Implementacion

```
vector<pto> upper_hull(vector<pto>& p) {
    vector<pto> uh;
    sort(p.begin(), p.end());
    forn(i, sz(p)) {
        while(sz(uh)>=2 && (uh[sz(uh)-1]-uh[sz(uh)-2])^(p[i]-uh[sz(uh)-2])>=0)
            uh.pop_back();
        uh.pb(p[i]);
    }
    return uh;
}
```

Polar Sort - Convex Hull: Almost Convex

Enunciado

Dado un conjunto de puntos S, un poligono P es casi convexo si:

- P es simple: sus vertices son diferentes y ningun lado se intersecta con otro.
- Los vertices de P pertenecen a S, y todos los puntos de S estan dentro o en el borde de P.

Sea R el poligono casi convexo de tamaño minimo, ¿Cuantos poligonos casi convexos Q existen tal que $|Q| <= |R| + 1$?.
Cotas: $(3 <= |S| <= 2000)$

Cotas

- $3 <= |S| <= 2000$

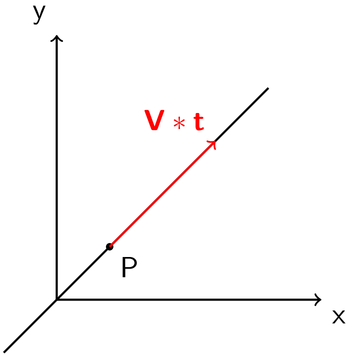
Temario

Línea: Definicion

Linea

Sea P un punto en la linea L, y V un vector director de L, la linea L es el conjunto de puntos Q dados por:

$$Q = P + V * t, t \in R$$



Línea: Operaciones

Lineas Paralelas

Dos lineas L1 y L2 son paralelas si sus vectores directores V_1 y V_2 cumplen:

$$V_1 x V_2 = 0$$

Línea: Operaciones

Lineas Paralelas

Dos lineas L1 y L2 son paralelas si sus vectores directores V_1 y V_2 cumplen:

$$V_1 x V_2 = 0$$

Lado de un punto

Dada una linea L que contiene al punto P con vector director V, y un punto cualquiera Q:

- Q esta a la izquierda de L si: $\overrightarrow{V} x \overrightarrow{PQ} > 0$

Línea: Operaciones

Lineas Paralelas

Dos lineas L1 y L2 son paralelas si sus vectores directores V_1 y V_2 cumplen:

$$V_1 x V_2 = 0$$

Lado de un punto

Dada una linea L que contiene al punto P con vector director V, y un punto cualquiera Q:

- Q esta a la izquierda de L si: $\overrightarrow{V} x \overrightarrow{PQ} > 0$
- Q esta a la derecha de L si: $\overrightarrow{V} x \overrightarrow{PQ} < 0$

Línea: Operaciones

Lineas Paralelas

Dos lineas L1 y L2 son paralelas si sus vectores directores V_1 y V_2 cumplen:

$$V_1 x V_2 = 0$$

Lado de un punto

Dada una linea L que contiene al punto P con vector director V, y un punto cualquiera Q:

- Q esta a la izquierda de L si: $\overrightarrow{V} x \overrightarrow{PQ} > 0$
- Q esta a la derecha de L si: $\overrightarrow{V} x \overrightarrow{PQ} < 0$
- Q esta sobre L si: $\overrightarrow{V} x \overrightarrow{PQ} = 0$

Línea: Operaciones

Punto mas cercano

Dada una linea L que contiene al punto P con vector director V, el punto de L mas cercano a un punto cualquiera Q es:

$$P + proj_V \overrightarrow{PQ}$$

Línea: Operaciones

Interseccion de lineas

Dadas las lineas:

$$P_1 + V_1 * \alpha$$

$$P_2 + V_2 * \beta$$

Igualando:

$$P_1 + V_1 * \alpha = P_2 + V_2 * \beta$$

Multiplicando ambos lados por V_2:

$$P_1 x V_2 + V_1 x V_2 * \alpha = P_2 x V_2 + V_2 x V_2 * \beta$$

$$P_1 x V_2 + V_1 x V_2 * \alpha = P_2 x V_2$$

$$\alpha = \frac{(P_2 - P_1) x V_2}{V_1 x V_2}$$

Línea: Implementacion

```
struct line{
    pto p, v;
    line(){}
    line(pto p_, pto v_): p(p_), v(v_) {}

    bool inside(pto q) { return abs(v^(q-p)) <= EPS; }
    bool left(pto q) { return v^(q-p)>EPS; }
    pto closest(pto q) { return p + (q-p).proj(v); }
    pto inter(line l) {
        if(v^l.v<=EPS) return pto(INF,INF);
        return p+v*((l.p-p)^l.v)/(v^l.v);
    }
};
```

Línea: Endless Turning

Enunciado

Dadas C calles (representadas como lineas) y la posicion inicial de un auto (representada como un punto), este va a moverse por las calles y en cada interseccion va a realizar un giro a la derecha. Si el auto empieza a moverse hacia el este ¿en que calle se encuentra el auto despues de N giros?

Cotas

- $1 <= C <= 100$
- $1 <= N <= 10^{10}$

Temario

Segmento: Definicion

Segmento

Sean U y V los puntos extremos del segmento, el mismo queda definido por el conjunto de puntos P:

$$P = \overrightarrow{UV} * t + U, t \in [0, 1]$$

Segmento: Implementacion

```
struct segm {
    pto s, e;
    segm(pto s_, pto e_) : s(s_), e(e_) {}
    pto closest(pto b) {
        pto bs = b - s, es = e - s;
        ld l = es * es;
        if (abs(l) <= EPS) return s;
        ld t = (bs * es) / l;
        if (t < 0.) return s; // comment for lines
        else if (t > 1.) return e; // comment for lines
        return s + (es * t);
    }
    bool inside(pto b) {
        return abs(s.dist(b) + e.dist(b) - s.dist(e)) < EPS;
    }
    pto inter(segm b) { // if a and b are collinear, returns one point
        if ((*this).inside(b.s)) return b.s;
        if ((*this).inside(b.e)) return b.e;
        pto in = line(s, e).inter(line(b.s, b.e));
        if ((*this).inside(in) && b.inside(in)) return in;
        return pto(INF, INF);
    }
};
```

Segmento: Regional Integration

Enunciado

Dado un conjunto de edificios de forma triangular (T), cuadrada (Q) y circular (C), una persona debe caminar de un edificio triangular a uno cuadrado. Seleccionar un edificio triangular y otro cuadrado de forma que se minimice la distancia caminada al sol (sin estar debajo de un edificio).

Cotas

- $(T + C) * (Q + C) <= 10^6$

Problema Extra: Large Triangle

Enunciado

Dado N puntos y un entero S, encontrar tres puntos tal que el area del triangulo resultante sea igual a S.

Cotas

- $3 <= N <= 2000$

Problemas tematicos

- Geometry Game
- Almost Convex
- Endless Turning
- Regional Integration
- Large Triangle

Consultas

Pueden preguntarme durante el tc o escribirme por:

- jp.cabana@icloud.com
- Aristides