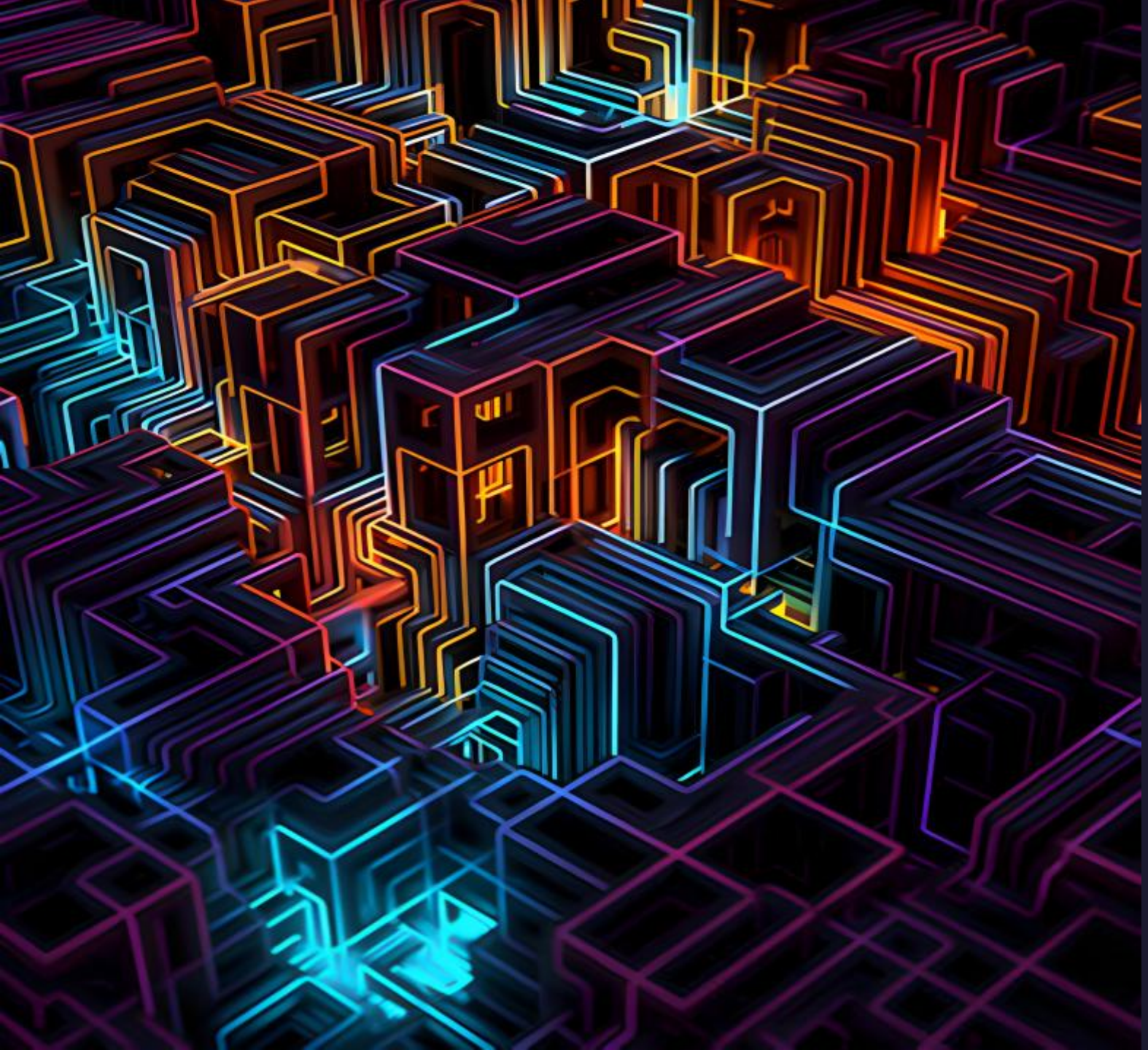


Geometría computacional: convex-hull

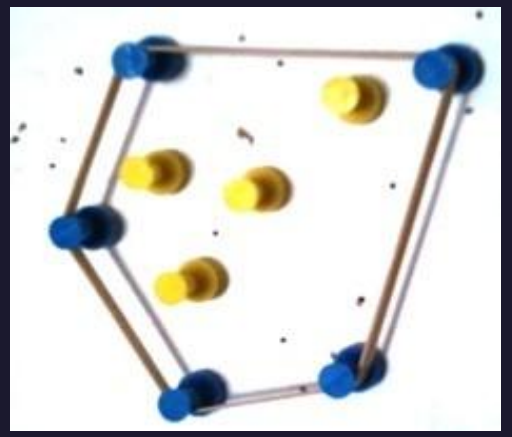
Análisis y diseño de algoritmos avanzados

Dra. Valentina Narváez Terán

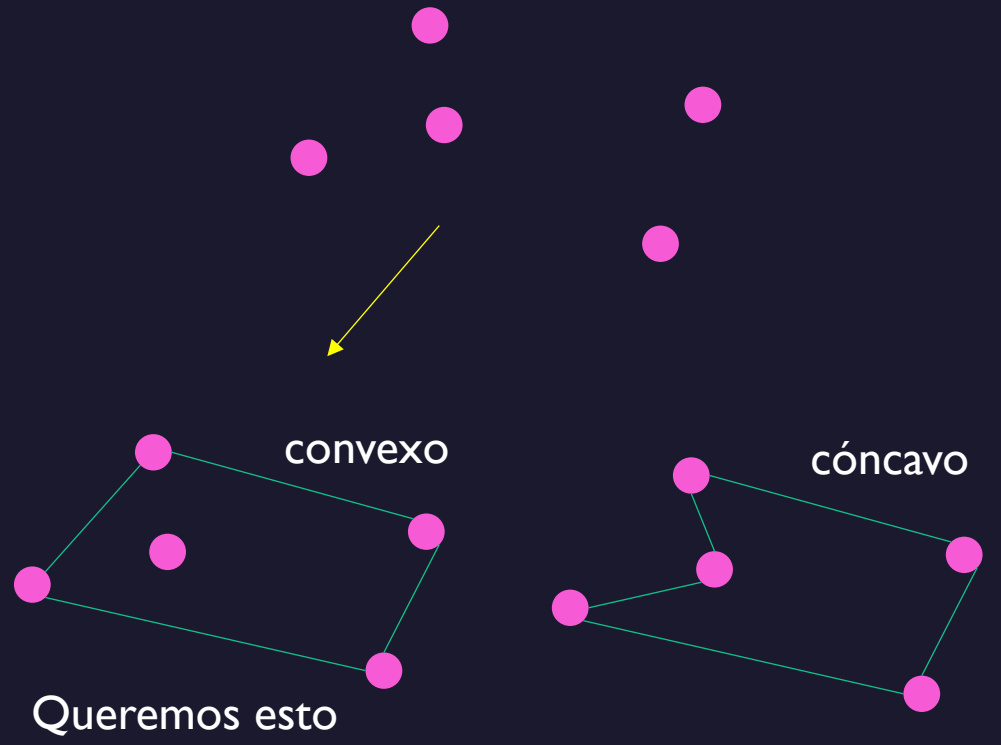
Tecnológico de Monterrey

Problema del convex-hull

El problema:
Crear un polígono convexo, que incluya todos puntos, con algunos de ellos como vértices



La explicación clásica: imaginarlos como tachuelas rodeadas con una liga



Problema del convex-hull

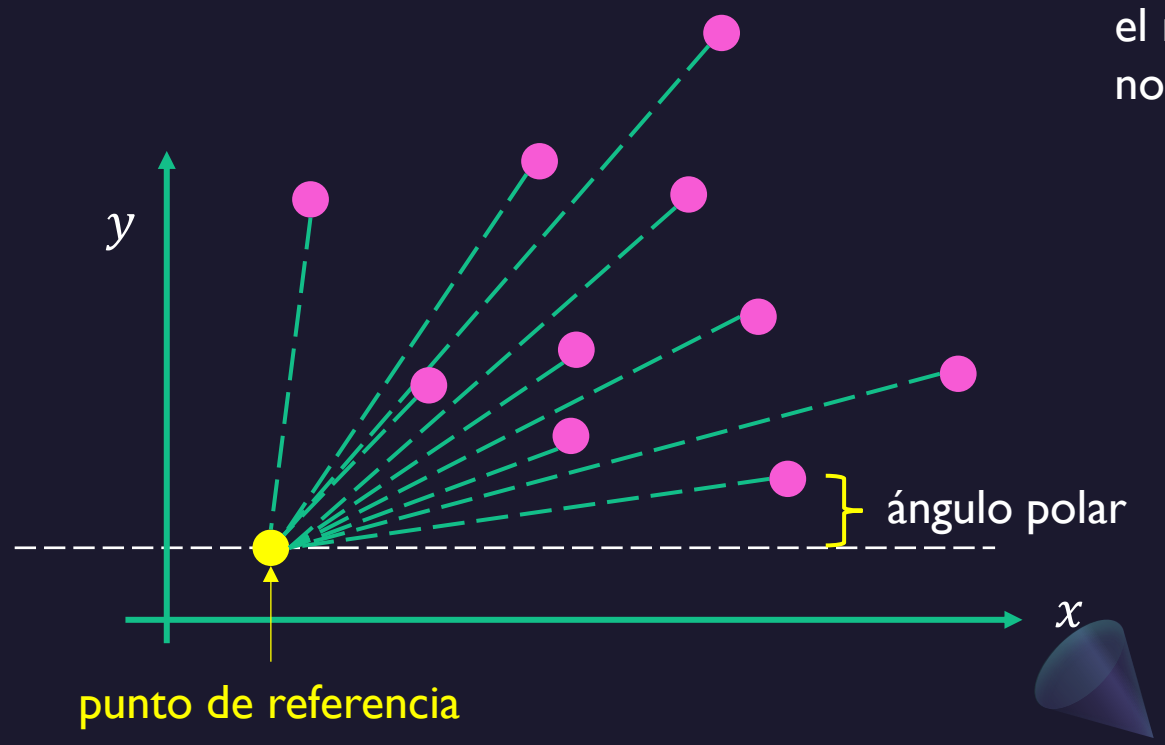
Hay otros algoritmos para este problema: Jarvis/gift wrapping, quick hull, etc.

¿Cómo construir el polígono?

Veremos un algoritmo llamado Graham scan

- El algoritmo usa como referencia el punto que se encuentre más abajo y más a la izquierda
- Su mecánica se parece a la manera intuitiva en que lo resolveríamos visualmente

Problema del convex-hull



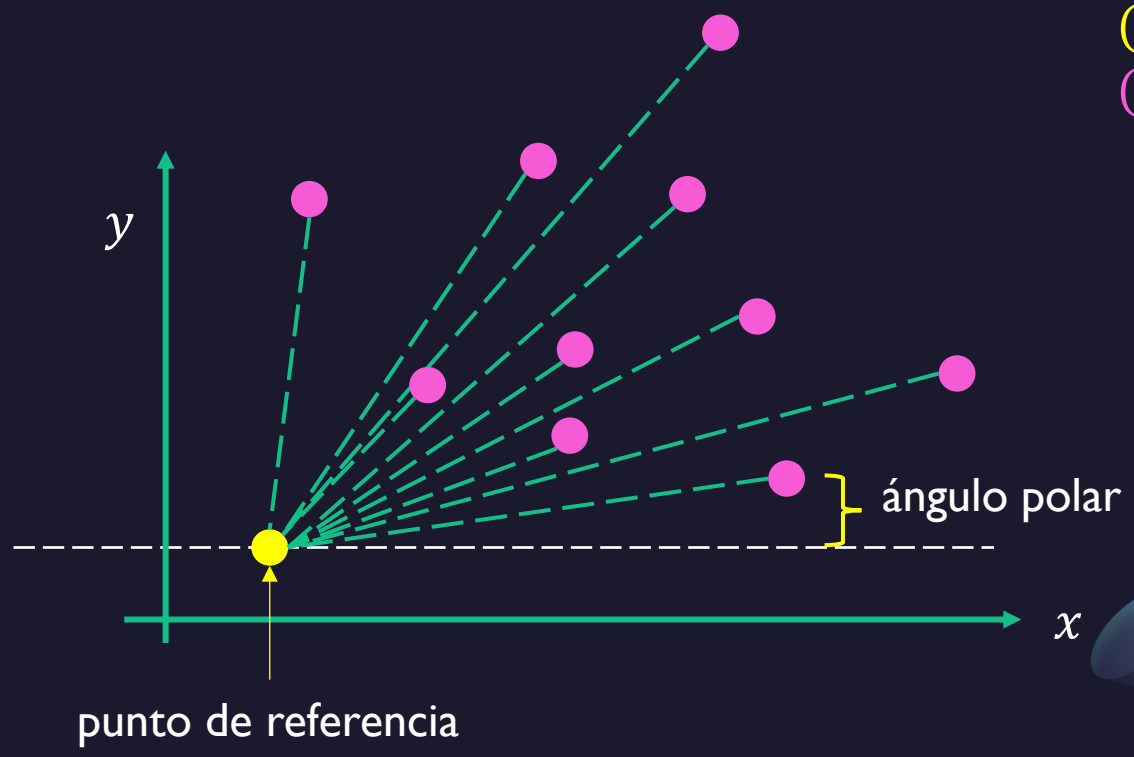
A partir del punto de referencia, procesaremos el resto de los puntos, decidiendo si deberían o no ser vértices del polígono

Los puntos se procesan en un cierto orden: los que formen un ángulo polar más pequeño primero

Es decir, necesitamos obtener el ángulo para cada punto, y luego ordenar los puntos ascendentemente de acuerdo con el ángulo

Ángulo polar: calculado a partir de punto de referencia, y no desde el origen

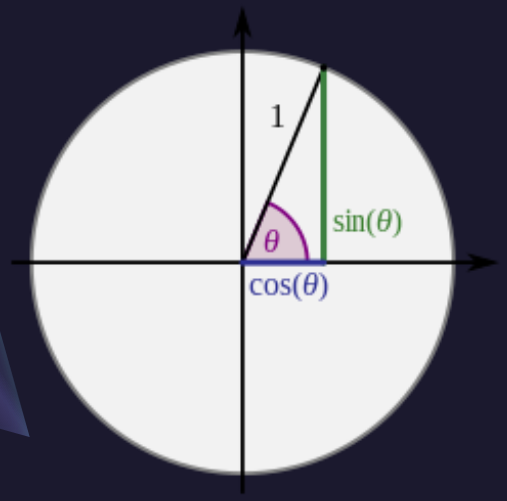
Problema del convex-hull



Para obtener el ángulo polar que forman dos puntos:

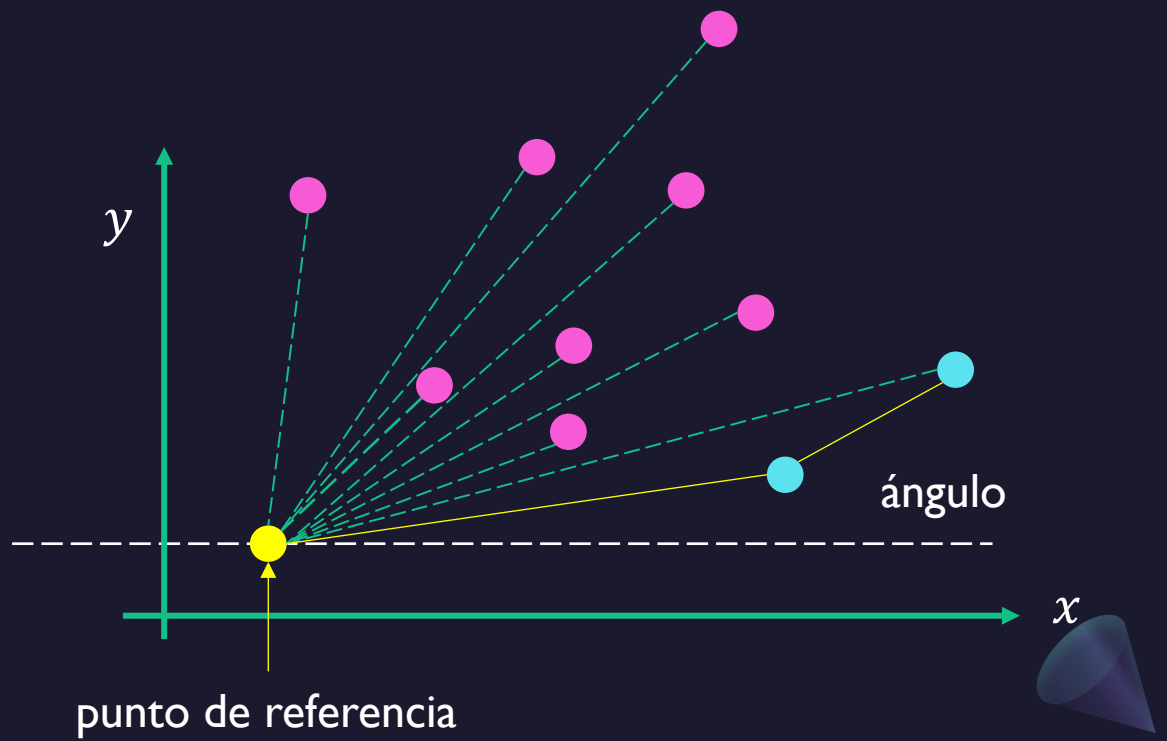
(x_1, y_1) es el punto de referencia
(x_i, y_i) es uno de los otros puntos

$\text{atan2}\ (y_i - y_1\ ,\ x_i - x_1)$ ← el arco tangente (en radianes)



En Python, math.degrees() lo convierte a grados

Problema del convex-hull

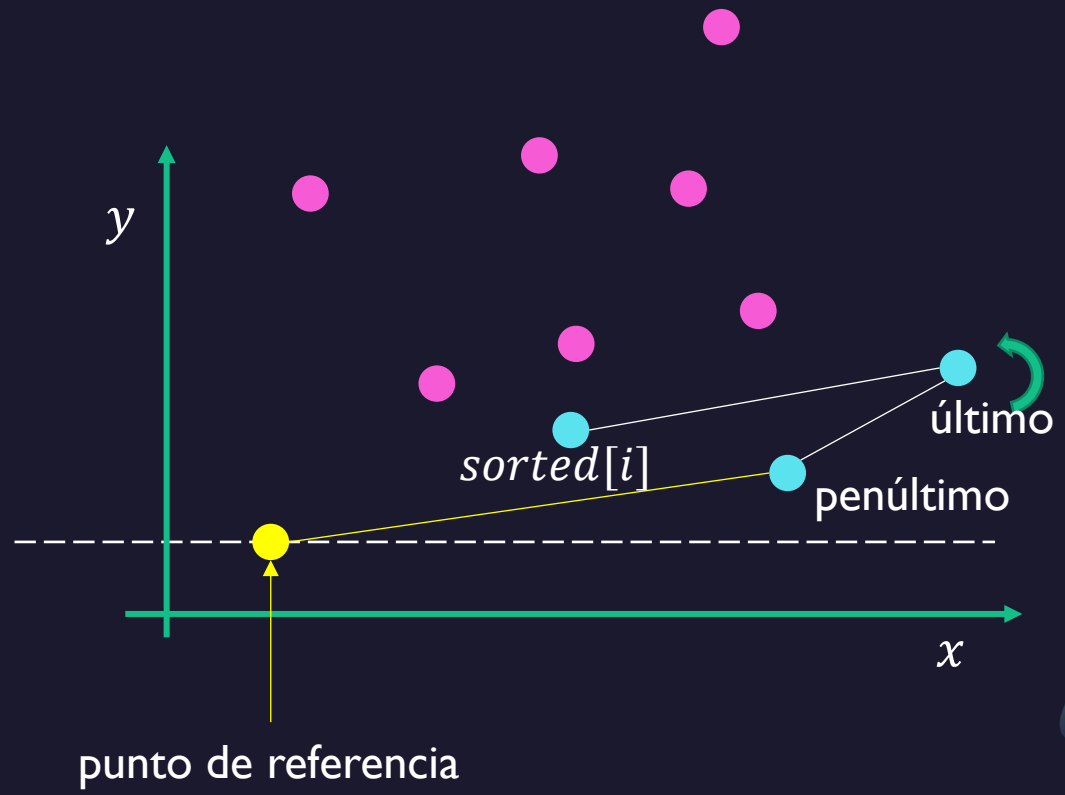


Lo siguiente es agregar los primeros 3 puntos a una pila (stack)

Al final del proceso, los puntos que sigan en la pila serán los vértices del polígono

*No es estrictamente necesario que sea una pila. Puede ser una lista/arreglo

Problema del convex-hull



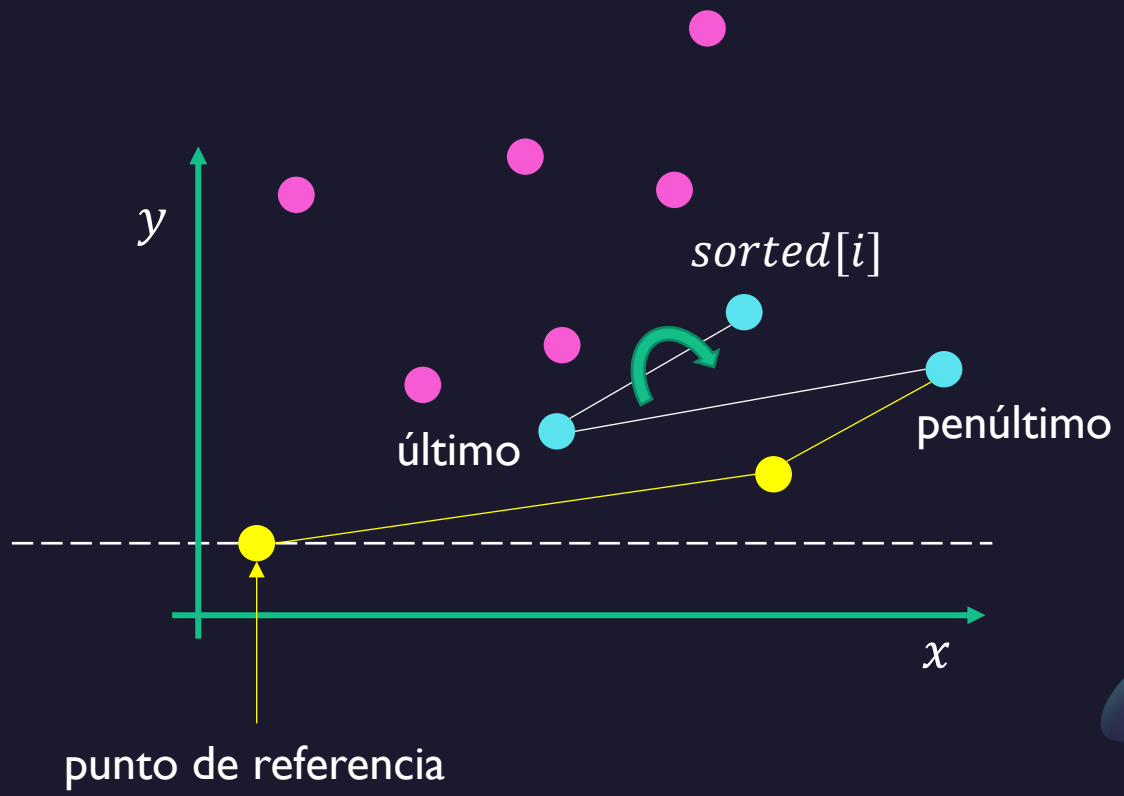
$lowest$ = el punto más abajo (y a la izquierda si hay empates)
$sorted$ = los puntos ordenados según el ángulo que forman con $lowest$

$stack$ = una pila/lista/arreglo
Agregar a $stack$ los primeros 3 puntos

for i de 3 a n:
    while $stack$ tenga 2 o más elementos:
        if los dos últimos elementos de $stack$ y $sorted[i]$ forman un giro en sentido horario :
            eliminar el último elemento de la pila
        else:
            break

    Agregar el punto actual $sorted[i]$ a la pila

Problema del convex-hull



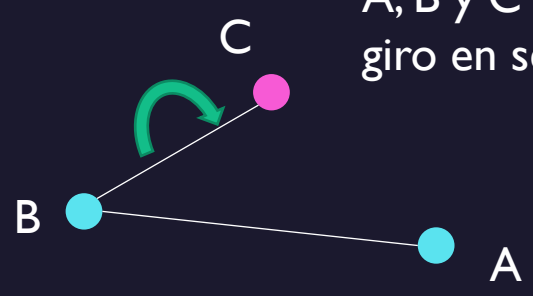
$lowest$ = el punto mas abajo (y a la izquierda si hay empates)
$sorted$ = los puntos ordenados según el ángulo que forman con $lowest$

$stack$ = una pila/lista/arreglo
Agregar a $stack$ los primeros 3 puntos de $sorted$

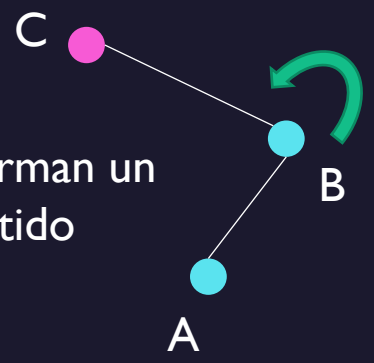
for i de 3 a n:
    while $stack$ tenga 2 o mas elementos:
        if los dos últimos elementos de $stack$ y $sorted[i]$
        forman un giro en sentido horario :
            eliminar el último elemento de la pila
        else:
            break

    Agregar el punto actual $sorted[i]$ a la pila

Giros en sentido horario y antihorario



A, B y C forman un giro en sentido horario



A, B y C forman un giro en sentido antihorario

¿Cómo computarlo? Dadas las coordenadas x y y de cada punto

$$(By - Ay) * (Cx - Bx) - (Bx - Ax) * (Cy - By)$$

Si el valor resultante es

- 0 los 3 puntos son colineales
- > 0 : el giro es en sentido horario
- < 0 : el giro es antihorario