



TAREA 1

Reglas

- Es posible trabajar en la tarea en grupos, se debe entregar individualmente.
- Pueden usar su lenguaje de programación favorito.
- La tarea se debe entregar a más tardar el Lunes 26 de Febrero.

Problema 1: La matriz R_θ define una rotación en \mathbb{R}^2 ,

$$R_\theta = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix},$$

con $\mathbf{v}_1 = (x_1, y_2)$, y $\mathbf{v}_2 = (x_2, y_2) \in \mathbb{R}^2$ tenemos $\det(\mathbf{v}_1, \mathbf{v}_2)$,

$$\det(\mathbf{v}_1, \mathbf{v}_2) = \begin{vmatrix} x_1 & y_1 \\ x_2 & y_2 \end{vmatrix} = (x_1 * y_2) - (y_1 * x_2).$$

Mostrar que para cualquier par $\mathbf{v}_1, \mathbf{v}_2 \in \mathbb{R}^2$,

$$\det(\mathbf{v}_1, \mathbf{v}_2) = \det(R_\theta(\mathbf{v}_1), R_\theta(\mathbf{v}_2)).$$

Problema 2: La propiedad mostrada en el punto anterior implica que el determinante de dos vectores es invariante bajo rotaciones por un ángulo θ . Mostrar que para \mathbf{v}_1 y \mathbf{v}_2 , dos vectores que empiezan en el origen, el determinante define la orientación entre ellos.

Pista: Existe una rotación en la que la componente y de $R_\theta(\mathbf{v}_1)$ es cero, con el uso de dicha rotación y la propiedad anterior se puede mostrar el enunciado.

Problema 3: Desarrollar un algoritmo que calcule la envolvente convexa de un conjunto de puntos. Para un Input un conjunto de puntos en \mathbb{R}^2 , regresa como Output una secuencia de puntos, en sentido antihorario que sigue el camino de la frontera de la envolvente convexa. Entregar el código de este algoritmo.

Problema 4: Desarrollar pruebas de desempeño de el algoritmo en el punto anterior. Mostrar el input de estos casos de prueba y el respectivo Output. Realizar esto para 6 casos de prueba de un conjunto de 9 puntos cada uno. El algoritmo diseñado en el punto anterior se desempeña satisfactoriamente?

Problema 5: Desarrollar el algoritmo de búsqueda de intersecciones sobre un conjunto de segmentos usando el concepto de la línea de barrido u otro algoritmo que no sea la búsqueda exhaustiva de las posibles intersecciones de todos los segmentos contra todos los segmentos de complejidad en tiempo de $O(n^2)$.

El algoritmo a desarrollar tendrá como Input un conjunto de segmentos en \mathbb{R}^2 , donde cada segmento se representa como un par de puntos en \mathbb{R}^2 y un identificador único por cada segmento, tendrá como Output los pares de segmentos que se intersectan, representados por sus respectivos identificadores.



Problema 6: Para un half-edge, e , de un doubly connected edge list. Cuáles de las siguientes afirmaciones siempre son ciertas?



$$\text{Twin}(\text{Twin}(e)) = e$$

$$\text{Next}(\text{Prev}(e)) = e$$



$$\text{Twin}(\text{Prev}(\text{Twin}(e))) = \text{Next}(e)$$

$$\text{IncidentFace}(e) = \text{IncidentFace}(\text{Next}(e))$$

