Geometría Computacional
Laboratorio Práctica 3
Semestre 2023-2

Profesora: Adriana Ramírez Vigueras[*]
Ayudantes:
Fhernanda Montserrat Romo Olea [**] Marco Antonio Velasco Flores[***]

Fecha de entrega: 9 de mayo de 2023

El objetivo de esta práctica será implementar el algoritmo de **Jarvis March** o también conocido como **Gift Wrapping** para calcular el cierre convexo de un conjunto de puntos.

1. Intrucciones

Las instrucciones son las mismas que la práctica 1, usarán la herramienta de visualización para verificar que su solución sea correcta con el mismo flujo de trabajo que utilizaron.

Se mantiene el mismo formato de entrada para los puntos, y se espera el mismo formato de salida para su solución.

[*] adriana.rv@ciencias.unam.mx
[**] fher@ciencias.unam.mx
[***] mutska@ciencias.unam.mx

2. Jarvis March

Para implementar el algoritmo de **Jarvis March** que consiste en tomar un punto extremo y de ahí ir construyendo el cierre convexo al tomar el siguiente punto más cercano en orden angular, ocuparemos como subrutina el test de orientación que implementaron en la práctica 1 que es fundamental para este algoritmo, y una función auxiliar para encontrar el primer punto extremo.

2.1. Jarvis March

El algoritmo de **Jarvis March** tiene una complejidad de $O(n^2)$:v

Esta es una manera de implementar este algoritmo, lo pueden hacer de otra manera si así lo desean, pero deben respetar la complejidad :v.

Tomen las consideraciones necesarias para los índices, recuerden que normalmente se ocupan índices que empiezan en cero y no en uno.

Entrada: p, q (Dos puntos)
Salida: *Boolean* ($True$ si p es menor que q en coordenada en y, $False$ en otro caso)
Function `comparePoints(`p, q`)`**:**
    si $p.y < q.y$ **entonces**
        devolver $True$
    fin
    si $p.y == q.y$ *and* $p.x < q.x$ **entonces**
        devolver $True$
    fin
    devolver $False$
End Function

Algoritmo 1: Compare Two Points

Entrada: $points, n$ (Lista de puntos, Tamaño de la lista de puntos)
Salida: $convexHull$ (Cierre convexo de la lista de puntos)
Function `Jarvis March`$(points, n)$**:**
    $convexHull \longleftarrow \emptyset$
    (Encontrar un punto extremo)
    $minPointIndex = 0$
    para $i \leftarrow 1$ *to* n **hacer**
        $p \longleftarrow points[minPointIndex]$
        $q \longleftarrow points[i]$
        si $ComparePoints(p, q) == False$ **entonces**
            $minPointIndex = i$
        fin
    fin
    $currentIndex = minPointIndex$
    $p \longleftarrow points[currentIndex]$
    $convexHull.append(p)$
    (Encontrar los puntos restantes)
    mientras $True$ **hacer**
        $nextPointIndex \longleftarrow (currentIndex + 1)\ \%\ n$
        para $i \leftarrow 1$ *to* n **hacer**
            si $i == currentIndex$ ***or*** $i == nextPointIndex$ **entonces**
                continue
            fin
            $p \longleftarrow points[currentIndex]$
            $q \longleftarrow points[i]$
            $r \longleftarrow points[nextPointIndex]$
            $currentOrientation \longleftarrow orientation(p, q, r)$
            si $currentOrientation == COUNTERCLOCKWISE$ **entonces**
                $nextPointIndex = i$
            fin
        fin
        $currentIndex = nextPointIndex$
        si $currentIndex == minPointIndex$ **entonces**
            break
        fin
        $p \longleftarrow points[currentIndex]$
        $convexHull.append(p)$
    fin
    devolver $convexHull$
End Function

Algoritmo 2: Jarvis March

3. Consideraciones Y Entregables

- Esta práctica cuenta sobre calificación final 0.25, en total cuatro prácticas, es decir todas las prácticas de laboratorio cuentan como 1 punto sobre calificación final.
- Todo su código debe estar bien documentado.
- Deben dar instrucciones claras y precisas para ejecutar su código en el lenguaje que eligieron.
- No hay prórroga para cambiar la fecha de entrega.

El entregable consiste de un archivo comprimido en zip con su código de solución al problema, instrucciones para ejecutar su código en el lenguaje escogido y un archivo de texto con su nombre completo y número de cuenta.