Análisis y Diseño de Algoritmos II

Trabajo práctico especial 3 . Algoritmos Geométricos

---

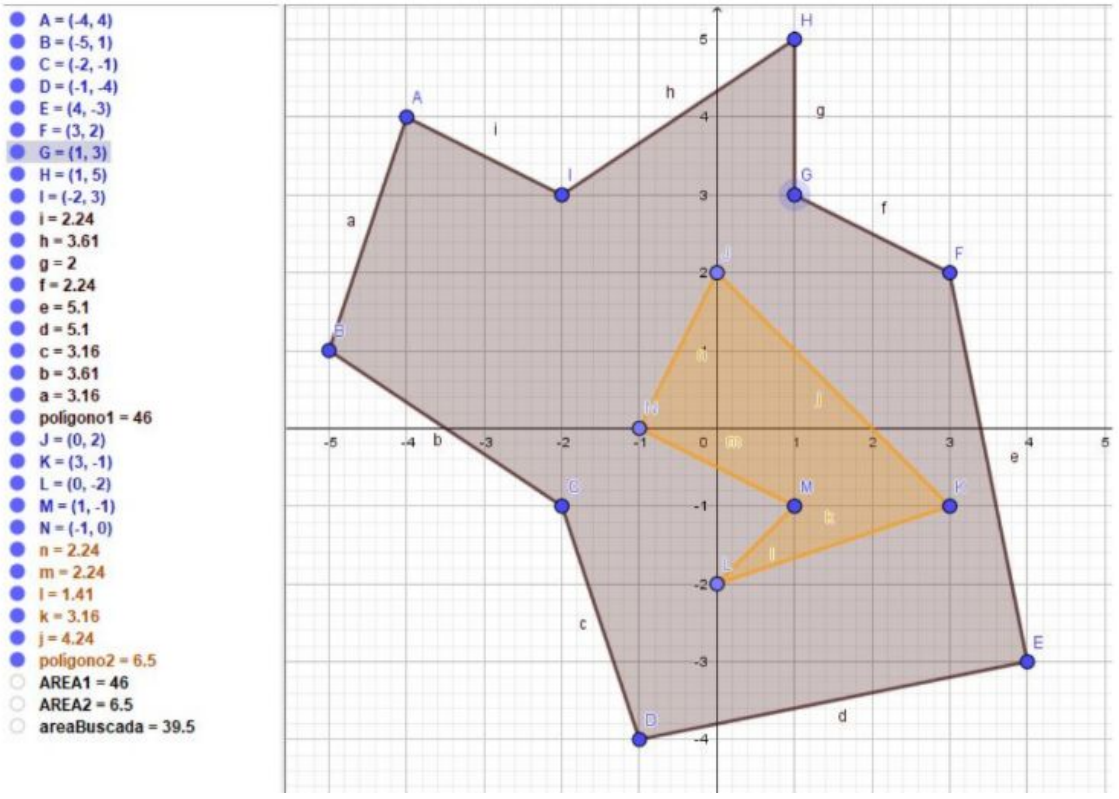
1 - Se desea calcular el área de un polígono exterior sin el área de un polígono interior. En el ejemplo área pintada de gris (suponga que la entrada es correcta, es decir, los polígonos ingresados ya fueron verificados por algún otro programa y el polígono interior está completamente contenido dentro del exterior).

Escriba un programa que calcule el área sombreada de gris utilizando el producto cruzado. Pruebe utilizando tanto un punto interior como un punto exterior (ingresado por teclado).

Debe entregar el programa funcionando para los siguientes polígonos, leyéndolos desde un archivo.

punto PolígonoExterior [ ]= { (-4, 4), (-5, 1), (-2, -1), (-1, -4), (4, -3), (3, 2), (1, 3), (1, 5), (-2, 3) };

punto Polígono1 [ ] = { (0, 2), (3, -1), (0, -2), (1, -1), (-1, 0) };



Análisis y Diseño de Algoritmos II

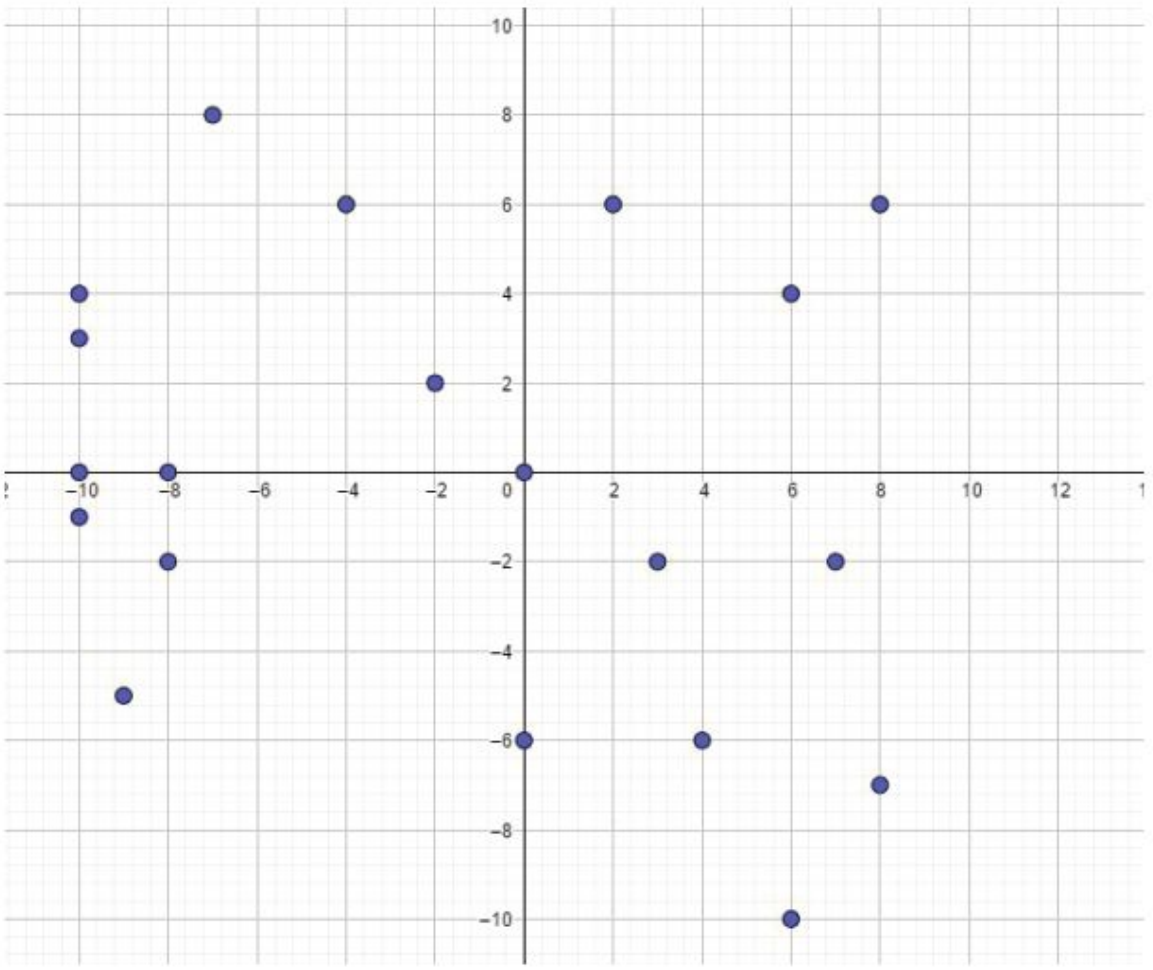
Trabajo práctico especial 3 . Algoritmos Geométricos

---

2. Dada una nube de puntos, obtener el **camino simple cerrado** que los une.

Debe entregar el programa funcionando para la siguiente nube de puntos, leyéndola desde un archivo.

punto nube[ ] = { punto (-10,-1), punto (-10,0), punto(-7,8), punto(-4,6), punto(2,6), punto(6,4),
punto(8,6), punto(7,-2), punto(4,-6), punto(8,-7), punto(0,0), punto(3,-2),
punto(6,-10), punto(0,-6), punto(-9,-5), punto(-8,-2), punto(-8,0), punto(-10,3),
punto(-2,2), punto(-10,4) };



Análisis y Diseño de Algoritmos II

Trabajo práctico especial 3 . Algoritmos Geométricos

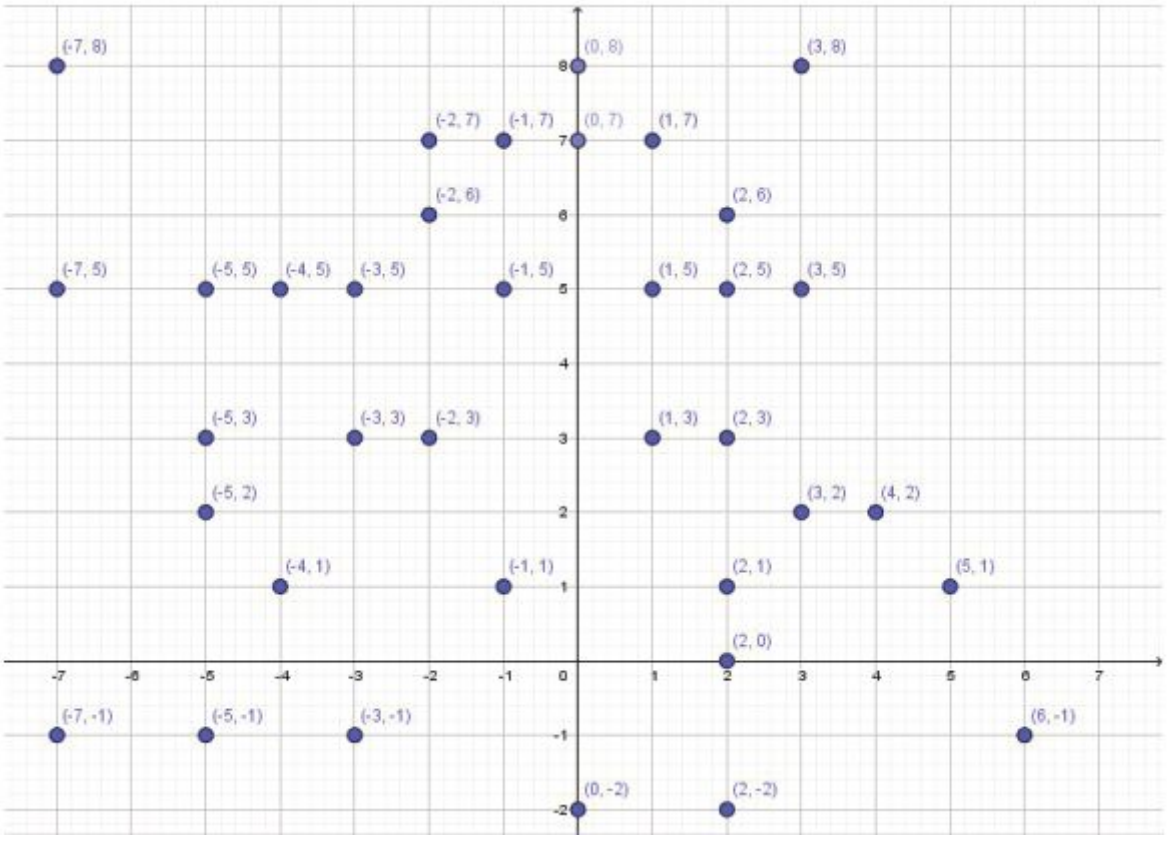
---

3. a) Implemente el algoritmo de **Graham's scan** para encontrar el convex hull de una nube de puntos.

b) Implemente la eliminación interior.

Debe entregar el programa funcionando para la siguiente nube de puntos leyéndola desde un archivo, permitiendo al usuario correr el programa con o sin eliminación interior.

Punto nube[ ] = { Punto(-2, 3), Punto(2, 5), Punto(2, 1), Punto(-3, 5), Punto(-4, 1), Punto(3, 2),
Punto(2, 0), Punto(6, -1), Punto(5, 1), Punto(-3, -1), Punto(-1, 5), Punto(2, 6),
Punto(1, 5), Punto(1, 7), Punto(0, -2), Punto(1, 3), Punto(-1, 1), Punto(3, 5),
Punto(4, 2), Punto(2,-2), Punto(-4, 5), Punto(-5, 5), Punto(-5, 3), Punto(-5, 2),
Punto(-5, -1), Punto(2, 3), Punto(0, 7), Punto(-1, 7), Punto(-2, 7), Punto(-2, 6),
Punto(-3, 3) , Punto(0, 8), Punto(-7, 5), Punto(3, 8), Punto(-7, 8), Punto(-7, -1) };



Análisis y Diseño de Algoritmos II

Trabajo práctico especial 3 . Algoritmos Geométricos

---

4. a) Implemente el algoritmo de **Jarvis's march** para encontrar el convex hull de una nube de puntos.

b) Implemente la eliminación interior.

Debe entregar el programa funcionando para la siguiente nube de puntos leyéndola desde un archivo, permitiendo al usuario correr el programa con o sin eliminación interior.

Punto nube[ ] = { Punto(-2, 3), Punto(2, 5), Punto(2, 1), Punto(-3, 5), Punto(-4, 1), Punto(3, 2),
Punto(2, 0), Punto(6, -1), Punto(5, 1), Punto(-3, -1), Punto(-1, 5), Punto(2, 6),
Punto(1, 5), Punto(1, 7), Punto(0, -2), Punto(1, 3), Punto(-1, 1), Punto(3, 5),
Punto(4, 2), Punto(2,-2), Punto(-4, 5), Punto(-5, 5), Punto(-5, 3), Punto(-5, 2),
Punto(-5, -1), Punto(2, 3), Punto(0, 7), Punto(-1, 7), Punto(-2, 7), Punto(-2, 6),
Punto(-3, 3) , Punto(0, 8), Punto(-7, 5), Punto(3, 8), Punto(-7, 8), Punto(-7, -1) };