Problema 1

Sean $\mathbf{X} = \{x_1, \ldots, x_n\}$ y $\mathbf{Y} = \{y_1, \ldots, y_m\}$ dos conjuntos de puntos en el plano. Escriba una función que calcule la función de distancia de Hausdorff $H(\mathbf{X}, \mathbf{Y}) = \text{máx}(h(\mathbf{X}, \mathbf{Y}), h(\mathbf{Y}, \mathbf{X}))$ en donde $h(\mathbf{X}, \mathbf{Y}) = \text{máx}_{x \in \mathbf{X}} \{\text{mín}_{y \in \mathbf{Y}} \ d(x, y)\}$ y $d(x, y)$ es la distancia Euclidiana entre x y y.

Ejemplo: ante la ejecución de la función con los siguientes conjuntos :

$$\mathbf{X} = \{(1, 1), (0, 0)\}, \mathbf{Y} = \{(1, -1), (0, 0)\},$$

se debe devolver $\sqrt{2}$.

Problema 2

Sea D una matriz $N \times N$ de coeficientes enteros (positivos o negativos). Escribir una función que determine **puntos silla** en la matriz, es decir elementos de la matriz cuyo valor es mayor o igual a los valores de sus vecinos de izquierda y derecha e inferior o igual a los valores de sus vecinos de arriba y abajo (o viceversa). Se imprimirá la lista de las coordenadas de los puntos silla de la matriz, por sus coordenadas i (fila) y j (columna).

Ejemplo: ante la ejecución de la función con

$$D = \begin{pmatrix} 1 & -10 & 3 & 2 & 5 \\ -1 & 1 & 4 & 3 & 7 \\ 0 & 2 & 3 & 6 & 3 \\ 1 & 8 & 4 & 6 & 2 \\ 0 & 4 & -1 & 5 & 1 \end{pmatrix}$$

se imprimirá

4 3

porque el único punto silla corresponde al elemento en 4a fila, 3a columna.

Problema 3

Sea A un arreglo de n datos enteros. Escribir una función "eficiente" que calcule el valor mínimo a^* de los valores absolutos de diferencias entre datos del arreglo, es decir

$$a^* = \min_{1 \leq i,j \leq n, i \neq j} |A[i] - A[j]|.$$

Por "eficiente", queremos decir que esta función no tiene que explorar de manera exhaustiva todos los pares i, j posibles.

Ejemplo: ante la ejecución de la función con

$$A = \{1, -1, 3, 0, 8\},$$

se debe devolver 1.

Problema 4

Escribir una función que determine si un punto del plano está adentro o afuera de un polígono con N vértices. Esta función **bool testPointInPolygon(int pol_ver_x[N], int pol_ver_y[N], int point_x, int point_y)** recibe los N vértices consecutivos del polígono, a través de los arreglos **int pol_ver_x[N]** y **int pol_ver_y[N]**. Cada uno de estos arreglos contiene las coordenadas x y las coordenadas y de los vértices, respectivamente. Suponemos que las únicas intersecciones que existen entre las aristas del polígono son sus vértices. El punto que queremos probar es (point_x, point_y) y suponemos que nunca estará ubicado sobre la frontera del polígono.

Ejemplo: si N = 3, pol_ver_x = {0, 1, 2}, pol_ver_y = {0, 2, 0}, point_x = 1 y point_y = 1, entonces el método debería regresar true. La figura siguiente muestra para este caso la configuraciónón del polígono así como la ubicación del punto que probar.