

# Algoritmos y Complejidad

## Tarea #3

### “Puntos más cercanos”

Algorithm Knights

3 de diciembre de 2015

#### Problema

Un problema recurrente en muchas áreas es dada una colección de  $n$  puntos en el plano, hallar los dos puntos más cercanos entre sí. Un algoritmo para resolver este problema se basa en dividir y conquistar. Puede describirse como sigue:

1. Ordene los puntos por coordenada  $x$ .
2. Divida los puntos en dos mitades por coordenada  $x$  mediante una recta vertical en  $x = x_0$ . Asigne los puntos en la recta arbitrariamente a la mitad izquierda o derecha, de forma de obtener una división lo más equitativa posible.
3. Recursivamente halle las distancias mínimas en ambas partes, digamos que son los pares  $(z_{l1}, z_{l2})$  a distancia  $\delta_l$  y  $(z_{r1}, z_{r2})$  a distancia  $\delta_r$ .
4. Sea  $\delta$  el mínimo de  $\delta_l, \delta_r$ . Considere los puntos en la franja  $x_0 - \delta$  a  $x_0 + \delta$  para ubicar los más cercanos en mitades distintas.
5. Retorne el par más cercano entre los tres así elegidos.

Basta ordenar los puntos por coordenada  $x$  una vez al comienzo; para acelerar el paso (4) conviene tener los puntos ordenados por coordenada  $y$  (por ejemplo un arreglo paralelo que contiene los índices de los puntos ordenados por coordenada  $y$ ).

#### Preguntas

1. Suponiendo que ordenar un arreglo de  $n$  elementos toma tiempo  $O(n \log n)$ , derive el orden de la complejidad del algoritmo descrito.  
(40 puntos)
2. Escriba un programa en C o C++ que resuelva el problema. Debe leer los datos de la entrada estándar (el número de puntos, luego las coordenadas  $x$  e  $y$  para cada punto). Debe escribir en la salida estándar la distancia mínima, los índices de los puntos y sus coordenadas. Los datos ingresados son correctos, no es necesario validar.  
(60 puntos)

#### Condiciones Generales

- La tarea se realizará *individualmente* (esto es grupos de una persona), sin excepciones.
- La tarea debe ser entregada impresa o manuscrita en la Secretaría Docente de Informática (Piso 1, edificio F3) el día indicado en [Moodle](#).

- Opcionalmente, puede desarrollar la tarea en  $\text{\LaTeX}$ , lo cual tiene una bonificación de 10 puntos. Para obtener la bonificación, junto con entregar la tarea impresa en hojas tamaño carta deberá depositar copia de los fuentes  $\text{\LaTeX}$  de su solución en un *tarball* en el área designada al efecto en [Moodle](#) bajo el formato `tarea3-rol.tar.gz`. El archivo debe contener el directorio `tarea3-rol`, en el cual están los archivos de su solución (al menos `tarea3.tex`). Tiene derecho a la bonificación sólo si el *tarball* tiene el nombre y contenido correctos, y los fuentes  $\text{\LaTeX}$  (y posibles otros archivos anexos) se procesan correctamente en el ambiente que ofrece el Laboratorio de Computación del Departamento de Informática, y están escritos en forma legible.

Si la entrega es en manuscrito, está afecta a descuento de hasta 20 puntos por desorden o ilegibilidad.

- Por cada día de atraso se descontarán 20 puntos. A partir del tercer día de atraso no se reciben más tareas y la nota es automáticamente cero.
- La nota de la tarea puede ser según lo entregado, o (en el caso de algunos estudiantes elegidos al azar) el resultado de una interrogación en que deberá explicar lo entregado. No presentarse a la interrogación significa automáticamente nota cero.

Sobre la nota de la interrogación se aplican los descuentos por atraso si proceden, y la bonificación por entrega en  $\text{\LaTeX}$  o los descuentos por desorden.