

Algoritmos y Complejidad

Tarea #3

“Puntos más cercanos”

Algorithm Knights

4 de octubre de 2016

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 δ_l y (z_{r1}, z_{r2}) a distancia δ_r .
4. Sea δ el mínimo de δ_l, δ_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)

1. Condiciones de entrega

- La tarea se realizará *individualmente* (esto es grupos de una persona), sin excepciones.
- La entrega debe realizarse vía [Moodle](#) en un *tarball* en el área designada al efecto, bajo el formato `tarea-3-rol.tar.gz` (rol con dígito verificador y sin guión). Dicho *tarball* debe contener dos directorios:
 - Un directorio `tarea`, que contenga las fuentes en LaTeX (al menos `tarea.tex`) de la parte escrita de su entrega, además de un archivo `tarea-3.pdf`, correspondiente a la compilación de esas fuentes.

- Un directorio `programa`, que contenga su código *Python* con al menos un archivo `main.py` como punto de entrada y un `README` indicando la versión de *Python* utilizada y, opcionalmente, información adicional que deba saber el evaluador.
- Además de esto, la parte escrita de la tarea debe en hojas de tamaño carta en Secretaría Docente de Informática (Piso 1, edificio F3).
- Tanto el *tarball* como la entrega física deben realizarse el día indicado en [Moodle](#).
Por cada día de atraso se descontarán 20 puntos y a partir del tercer día de atraso no se reciben más tareas y la nota es automáticamente cero.

2. Revisión del código

Para evaluar el código se considerarán los siguientes aspectos:

Programa

- | | |
|--|-----|
| ■ Uso adecuado de funciones | 10% |
| ■ Uso de estructuras de control | 15% |
| ■ Uso de estructuras de datos | 10% |
| ■ Código claro y simple | 15% |
| • Nombres adecuados. | |
| • Indentación correcta. | |
| • Comentarios suficientes. | |
| • Ausencia de código comentado. | |
| ■ Ejecución correcta | 50% |
| • Ausencia de errores. | |
| • Resultados correctos en casos de prueba. | |