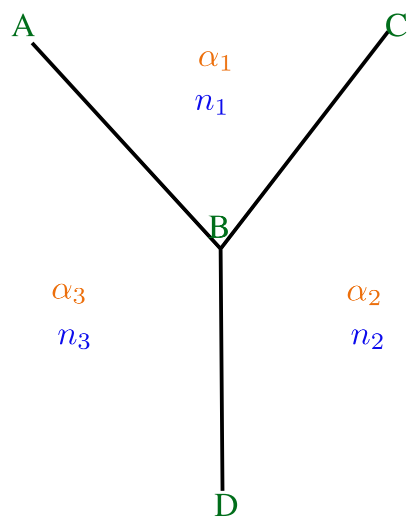
Algoritmo de la *Pata de Gallo* en tiempo $O(n\ logn)$

Análisis de Algoritmos II, Dr. Jorge Urrutia Galicia

Miguel Concha Vázquez

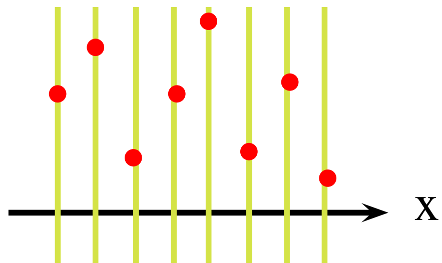
Facultad de Ciencias, UNAM. 23 de marzo de 2018.

En clase se motivó la *pata de gallo*, consistente en trazar tres rectas que separaran un conjunto de puntos S en el plano en tres regiones α_1, α_2 y α_3 de forma tal que en cada región quedaran n_1, n_2 y n_3 puntos respectivamente, con $n_1 + n_2 + n_3 = |S|$. El algoritmo discutido tiene una complejidad en tiempo del orden $O(n\ log^2 n)$. Veamos ahora cómo obtener un algoritmo para trazar dicha *pata de gallo* en $O(n\ logn)$-tiempo.

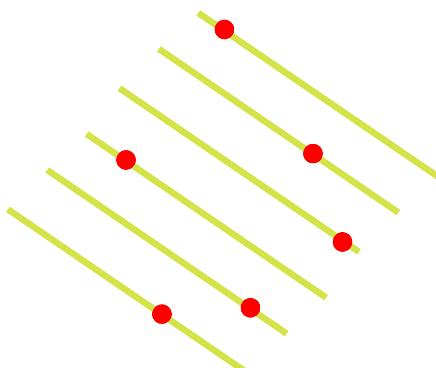


Pata de gallo y regiones en que separa al conjunto de puntos S.

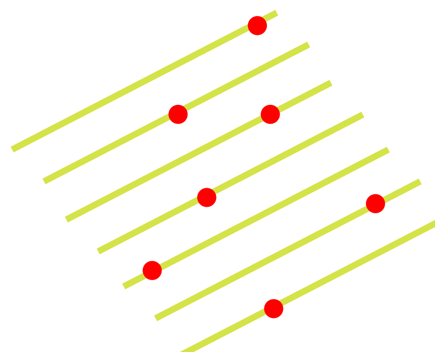
Primeramente se tienen que llevar a cabo tres ordenamientos de los puntos: con respecto al eje de abscisas y posteriormente con respecto a las rectas $\overline{AB}$ y $\overline{BC}$ de la *pata de gallo*. El problema de ordenamiento basado en comparaciones tiene una complejidad en tiempo del orden $\Theta(n\ logn)$ como ya fue demostrado en el curso pasado; por ende, la complejidad de estas operaciones sería del orden $3 \cdot \Theta(n\ logn) = \Theta(n\ logn)$.



Ordenamiento de los puntos con respecto al eje X.



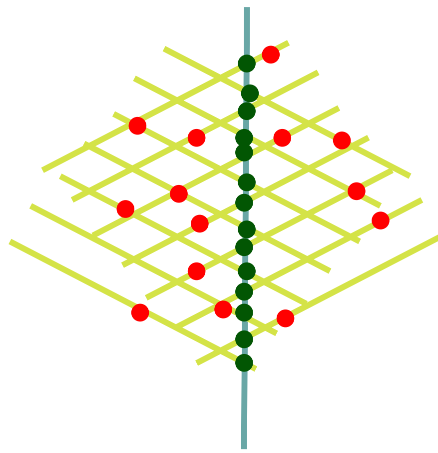
Ordenamiento de los puntos con respecto a $\overline{AB}$.



Ordenamiento de los puntos con respecto a $\overline{BC}$.

A partir del ordenamiento de los puntos con respecto al eje X se realizará una búsqueda binaria con tal de determinar en qué parte del plano deberá ser colocada la recta $\overline{BD}$ de la *pata de gallo* para que queden exactamente n_2 puntos a su derecha y n_3 a su izquierda. Para este fin se toma el punto que está a la mitad de los puntos ordenados y se traza la recta perpendicular al eje X que pasa por dicho punto.

Ahora, una vez fijada la recta vertical en la iteración de la búsqueda binaria, se procede a encontrar la altura a la que deberá ser colocada la *copa* de la *pata de gallo* para que haya n_1 en α_1 en tiempo lineal $O(n)$. Para ésto, tomamos la recta vertical e intersecamos con las rectas que corresponden a los puntos ordenados con respecto a $\overline{AB}$ y $\overline{BC}$, lo que genera $2n$ puntos sobre la vertical y pueden ser mezclados en tiempo lineal $O(n)$.



Resultado de intersecar las rectas de ambos ordenamientos con la vertical; produce $2n$ puntos sobre la vertical (color verde), n por correspondientes a la pendiente de $\overline{AB}$ y los otros n a la pendiente de $\overline{BC}$.

Comenzando desde el punto sobre la vertical ubicado en la parte superior, colocamos sobre el mismo el punto B correspondiente a la *pata de gallo* para formar su *copa* y se mantiene un contador que indica el número de puntos que se encuentran en cada iteración en la región interna de la copa, es decir, en α_1. La copa irá descendiendo sobre la recta y ubicando su punto B sobre los diversos puntos que se encontraron a raíz de intersecar las rectas hasta que el contador sea igual al valor deseado n_1.

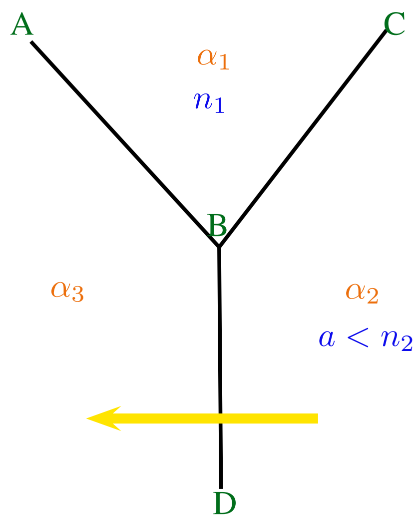
Eventualmente se tienen que considerar todos los puntos a pesar de que no estén como tal ubicados sobre la recta vertical. En el peor de los casos se tiene que repetir este proceso por cada uno de los $2n$ puntos de la recta. Entonces, la complejidad de esta parte del algoritmo (costo por parada de la búsqueda) es lineal $O(n)$.

El proceso de la búsqueda binaria, que incluye en cada iteración el proceso lineal antes descrito se lleva a cabo de la siguiente manera:

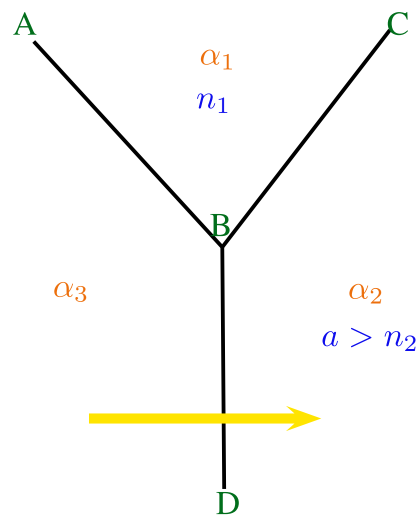
- Si el número de puntos en la región α_2 es un a con $a < n_2$, entonces quiere decir que debemos lograr que dicha región tome más puntos y por ende la solución de dónde ubicar a la recta vertical de la *pata de gallo* en el plano se encuentra más a la izquierda y entonces nos tomamos el punto que está a la mitad de los puntos ordenados con respecto al eje X desde el primero a la izquierda y que le sigue a la izquierda al punto actual.

- El caso en el que el número de puntos en la región α_2 sea a con $a > n_2$, es análogo al anterior, teniéndonos que mover a la derecha para considerar menos puntos de dicho lado.

- Si $a = n_2$, hemos terminado con la búsqueda binaria.



Escenario correspondiente al primer caso: se recursa en el lado izquierdo.



Escenario correspondiente al segundo caso: se recursa en el lado derecho.

Como hay un total de $logn$ iteraciones por ser una búsqueda binaria al ir descartando la mitad de los puntos en cada paso y en cada parada se lleva a cabo un proceso de tiempo lineal $O(n)$ para encontrar la altura a la que debe ser colocada la *copa* de la *para de gallo* para que queden n_1 puntos en la región α_1, entonces la complejidad de todo el algoritmo es del orden en tiempo $O(n\ logn)$ como se buscaba.