# Input Order Adaptivity

Note Title 3/31/2009

Output Sensitivity
& Instance Optimality

(1) Ordenar Permutacion

Verificacion $O(n)$

## ② Ordenar Multiconjuntos

$[h] = \{1, \ldots, h\}$

$I \in [h]^n$ hay $h^n$ instancias

Peor Caso

↳ $n$ fijado $O(n \lg n)$ $(h = n)$

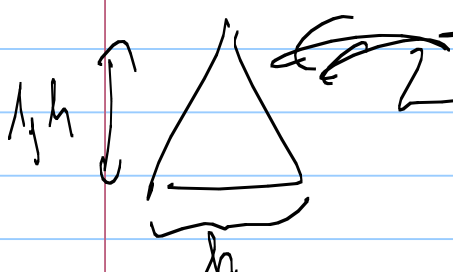
↳ $n, h$ fijado → $h$ conocido $O(n + h)$ con valores

→ $h$ desconocido $O(n \lg h)$ con $< > =$

**Cota inferior** $\Omega(\lg n^n) = \Omega(n \lg n)$

Puede ser **Más** Adaptativo?

**Ejemplo** Insertion Sort con FingerSearchTree




$$D_i = \#\{ j < i,\ X_i < X_j \}$$

$$Inv = \sum_{i=1}^{n} D_i$$

Permutaciones

Pb Instancia 3333312

Tiene grande Inv, pero es facil

$D_i' = \#\{ j<i, X_i < \underline{\leq} X_j \ \nexists k<j \ X_k = X_j \} \leq h$

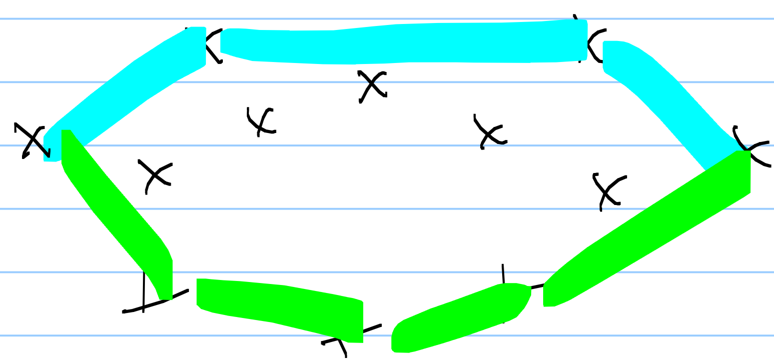
$Inv' = \sum_{i=1}^{n} D_i \leq nh$

$$\text{Complejidad} \leq \sum_{i=1}^{n} \left( 2\lg(D_i'+1) + 1 \right)$$

$$\leq 2n \lg\left( \frac{\sum D_i'}{n} + 1 \right) + n$$

$$\in O\left(n \lg \frac{Inv'}{n}\right) \subset O(n \lg h)$$

③ "Upper Hull" en Plano



## Algoritmo

- Ordenar Puntos en x: $O(n\lg n)$
- Construir de izquierda a derecha $\Rightarrow O(n)$

$$O(n\lg n)$$

## Cota inferior?

Reduccion a Ordenar

Areglo de n valores $x \Rightarrow$ Conjunto de puntos

$A[i] \Rightarrow \Omega(n\lg n)$ $(A[i], \quad )$

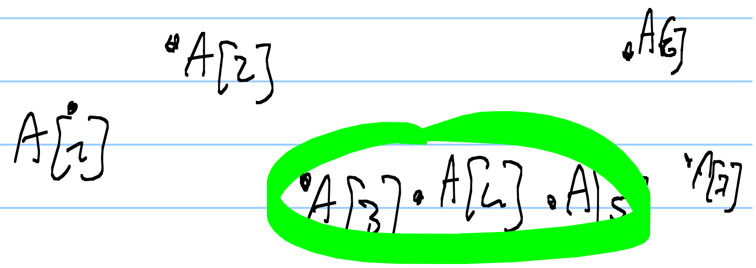
No es muy adaptivo!

Que podemos hacer?

(1) Proposiciones Ingenuo

(a) Todas las medidas de dificultades para ordenar sobre las coordinadas $x$ de los puntos.

Ej.:



## ② "Merge Sort" y Runs

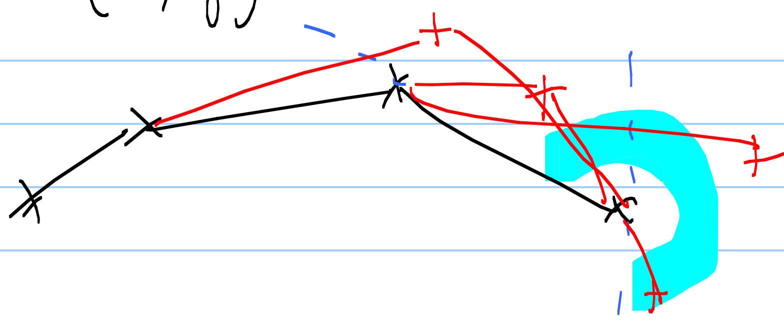
- Kalcular la Union de 2 Coverturas Superiores $O(n)$

- Hay un algoritmo $O(n)$ mejor caso

$\Longrightarrow$ Hay un Algoritmo en $O(n(\lg \rho + 1))$ Donde $\rho$ = # "Runs"

"Runs"

- x, y creciente
- x creciente
- (x, y) hice un "bueno" angulo



# II Output Sensitive Complejidad

## ① Median

| | | | | | | |
|---|---|---|---|---|---|---|
| 5 | 4 | 3 | 6 | 2 | 1 | 7 |

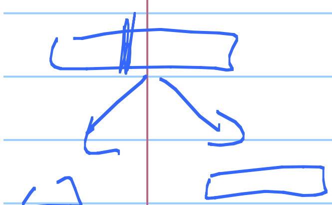
Se puede calcular en

$O(n \lg n)$ $\Omega(n)$

$\Theta(n)$

# Algoritmo 1 $O(n^2)$ peor caso

Quick select $(A, l, r, k)$



select un pivot $x \in A[l, \ldots, r]$

particionar $A[l, \ldots, r]$ en $\underbrace{A[l, p-1]}_{A<x}, \underbrace{A[p+1, r]}_{A>x}$

Si $|A_<| > k$

Quickselect$(A, l, p-1, k)$

Otra Quickselect$(A, p+1, r, k)$

(b) Si se podría dividir el areglo de manera equilibrada, Cual será la complejidad?

$O(\lg n)$ $\Omega(n)$

$n + nq + nq^2 + nq^3 + \dots + \quad = ?$

$\leq n\left(1 + q + q^2 + \dots + q^\infty\right) \in O(n)$

$\frac{1-q^\infty}{1-q} = \frac{1}{1-q}$

(c) Como se puede?

Selection de pivot $(A, l, r)$

$O(n)$ { - Dividir el areglo en grupos de tamaño 5
- Para cada grupo, calcular su median

$T(n/5)$ - Calcular la mediana de las medianas (recursivamente)
- return eso como el pivot $x$

# Analisis

$n/5$ grupos

$n/10$ grupos de median $\leq x$

$3n/10$ elementos $\leq x$

$\Rightarrow q \leq \frac{7n}{10}$

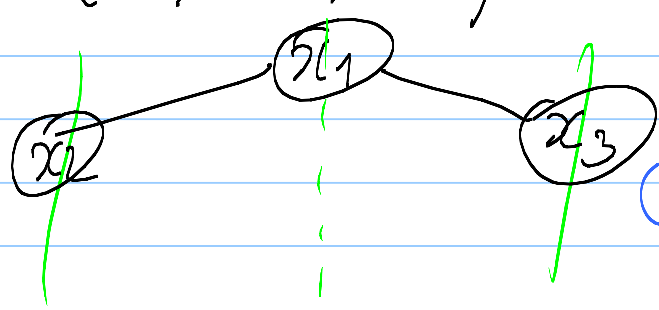
$\Rightarrow \Theta(n)$ algo para calcular la mediana

## (2) Multiconjuntos Ordenamiento

$O(n \lg h)$ con Finger Search tree

pero tambien con Median:

$\begin{pmatrix} x_1 - - - x_h \\ n_1, - - -, n_h \end{pmatrix}$ occurencias




(1) A la etapa $i$ descubri todos los $n_j \geq \frac{1}{2^i}$

(2) $x_i$ es contribuendo a $\lg \frac{n}{n_i}$ particiones

$\Longrightarrow$ Precio total | para $x_i$ es $n_i \lg \frac{n}{n_i}$

| es $\sum_{i=1}^{h} n_i \lg \frac{n}{n_i}$

Dos maneras de reescribir eso:

(1) $\sum n_i (\lg n - \lg n_i) = n \lg n - \sum n_i \lg n_i$

[Munro et al.]

(2) $n \sum \frac{n_i}{n} \lg \frac{n}{n_i} = n H(n_1, \ldots, n_h) \leq n \lg h$

Eso se llama Instance Optimalidad

Order Oblivious

porque $\forall A, I \; \max_{\pi} C(A, \pi(I)) \geq n(H(I)+1)$

Mas Fuerte En Promedio Orden Instance Optimalidad

$$\forall A \quad E_{\pi}\left(C(A,\pi(I))\right) \geq n(H(I)+1)$$

## 3 Maximo en el Plan

$(x, y) < (x', y') \iff x' > x$ y $y' > y$

Maximo(S) =
{puntos no dominados}