

03 Ordenamiento Adaptativo

① Algoritmos Clásicos

Promedio Peor

Peor

- Merge

$O(n \lg n)$

$O(n \lg n)$

$O(n \lg n)$

- Bubble

$O(n^2)$

$O(n^2)$

- Quick

$O(n \lg n)$

$O(n \lg n)$

$O(n^2)$

- Insertion (Average)

$O(n^2)$

$O(n^2)$

- Selection

$O(n^2)$

$O(n^2)$

- Heap Sort

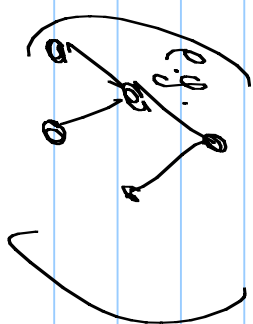
$O(n \lg n)$

$O(n \lg n)$

Recuerdos

HeapSort

MaxHeap = Arbol completo



HeapSort

Construye

un Heap $O(n)$

extraer el max,
poner el fin

$O(n \lg n)$

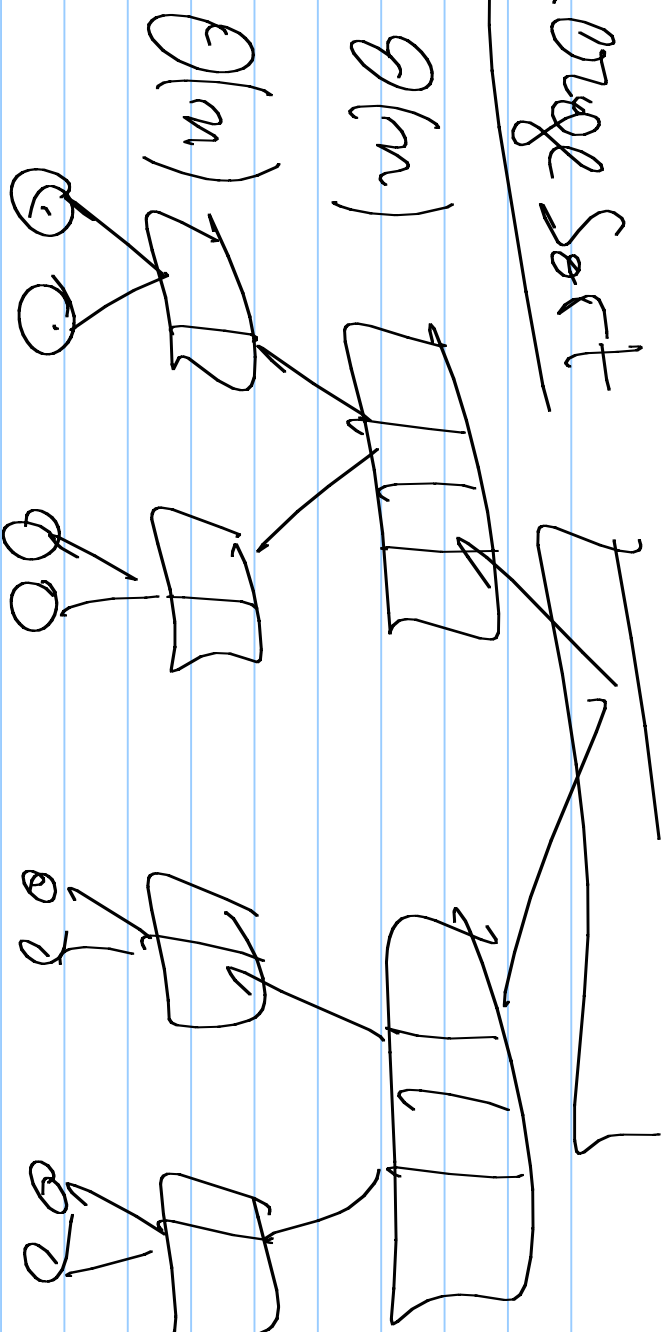
que permite

① extraer en tiempo $O(\lg n)$

el max, y corregir su estructura

② que se puede construir
en tiempo $O(n)$

Merge Sort



$$\Rightarrow O(n \log n)$$

② Alopata tivida

③ Ingeniero

Es super facil de obtener un
algoritmo $O(n)$ mejor caso

$O(n \log n)$ peor :

Verificar orden en tiempo $O(n)$

↓

⑥ Knuth Merge Sort

Algo

- Busca para "Step Down" $O(n)$

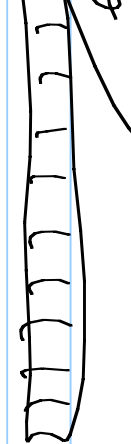
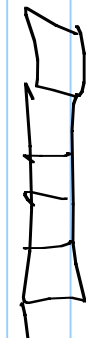
- Si encuentras Step Down
Fusiona los en tiempo $O(n \log p)$

Extension

$$\Rightarrow \Theta(n(1+H)) \quad O(n(1+\log p))$$

donde $H = \text{entropie}$

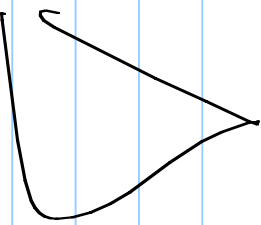
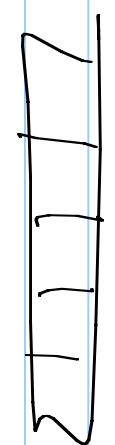
② Insertion Sort con Arbol

~~Insertar~~  /  buscar $O(\lg n)$

$O(I_{NV} + n)$ - colocar $O(n)$

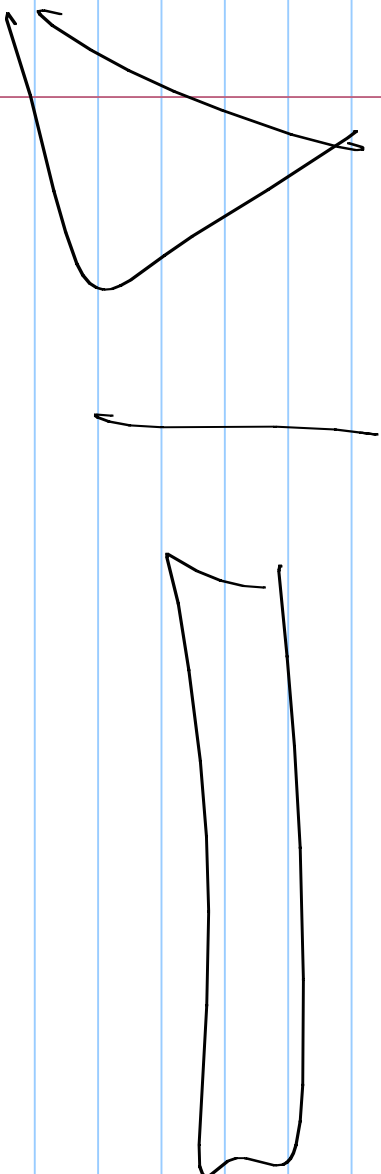
~~Insertar~~ colocar en tiempo $O(\lg n)$?

$\Rightarrow$ Arbol AVL?

  $O(n \lg n)$

$I_{NV} = \# \{ (i, j) \mid i < j \wedge x_i > x_j \}$

~~→~~ Finger Search Tree



$O(n)$ worst case

(e.g. $1, 2, \dots, n$
 $0, n, n-1, \dots, 1$)

$O(n \lg n)$ worst case

Analysis

$$O_f = \# \{ i \leq f \mid \log X_i > X_f \}$$

$$T_{nv} = \sum_{f=1}^n d_f$$

$$O(n \log n)$$

$$\text{complicated as } \sum (\log(d_f + 1)) + 1$$

$$\approx 2 \sum_{f=1}^n \log d_f \leq 2n \log \frac{\sum d_f}{n} = 2n \log \left(\frac{T_{nv}}{n} \right)$$

Reducciones

2 Medibles: P & Inv

2 Algoritmos Alternativos

Cual elegir?

P Se puede calcular en $O(n)$

Se puede comparar a Inv ?

① Idea Ingenious (No Practice)

Queremos un Algoritmo

$$O(\min(n \lg p, n \lg \frac{I_{nv}}{n}))$$

Puede tener

$$2 \min(n \lg p, n \lg \frac{I_{nv}}{n})$$

② May an algorithm
adaptive para P vs NP

(cf Paper
"Susceptibility of
Adaptive Sorting")

III Cotas Inferiores

① $\Omega(n \lg n)$ por Árboles de

decisões

Proof

$A_q \Rightarrow \text{Árvor}$

Instância $\Rightarrow H_{\text{Jes}}$

$n!$ $\Rightarrow n!$ H_{Jes}

$\Rightarrow \text{Altura } \lg n! \in \Omega(n \lg n)$ Altura em $\text{Pior caso } \Omega(n \lg n)$

2 Figuras 2 a 11

$\Omega(n \lg p) ???$

p instâncias de tamanho n de dificuldade $\Omega(n \lg p)$

