

Dividir para Reinar

① Exemplos

$$\textcircled{a} \text{ MergeSort } \begin{cases} \text{divide } O(n) \\ \text{ordena} + 2T(n/2) \\ \text{merge } + O(n) \end{cases} \quad T(n) = O(n \lg n)$$

② Binary Search

$$T(n) =$$

$$\textcircled{c} \text{ QuickSort } = \begin{cases} \text{divide } O(n) \\ \text{ordena} + T(p) + T(n-p) \\ \text{merge } + O(n) \end{cases} \quad \left. \begin{matrix} O(n^2) \text{ peor caso} \\ O(n \lg n) \text{ promedio} \end{matrix} \right\}$$

③ OPEP

Organization of the
Petroleum
Exporting
Countries

2) Buscar k -ésimo x ^{en} ~~de~~ ~~ordenado~~ A

a) Solución ingenua $O(n \lg n) \Rightarrow$ no es optimal

b) Usar un heap $O(n + k \lg n)$

c) QuickSelect (A, k) // buscar el k -ésimo mínimo
elige un pivote x
elemento en A

2) partition A en $A[l_i] < x, A[p] = x, A[j] \geq x$
 $\forall i < p < j$

3) Si $p = k$ return x
Sino si $k < p$ return QuickSelect($A[l_1, \dots, p-1], k$)
Sino (si $k > p$) return QuickSelect($A[p+1, n], k-p$)

③ Complejidad de Quick Select

- peor caso $\Theta(n^2)$

- promedio

$$T(n) = \sum_{p=1}^n \frac{1}{n} (n + T(\max(p-1, n-p)))$$

Teorema

Por una cota superior

del resultado de la Recurrencia.

