

Cotas Inferiores

UnSorted Search

ADT: n Elementos Comparables
Extract Busqueda (x)

PS: Elegir una permutación $[1..n]$
Checkear por cada uno en orden

Peor caso n comparaciones

Mejor caso 1

Si $\exists n$ copias de x ???

↳ Problema → Encontrar todos?

↳ Encontrar 1

↳ conociendo n o no.

Cota inferior.

Se puede hacer mejor que n ?

① Adversario (Estrategia de)

Algoritmo que verifica un
elemento a la vez
se demora n
en el peor caso.

II Ordered Search

ADT n elementos comparables
y/o ordenados

Busqueda

$O(\lg n)$

DS Busqueda Binaria ~~$O(n \lg n)$~~

~~Quick Search~~

Busqueda Secuencial

$O(n)$

Busqueda Exponencial (Doubling Search)

$O(\lg p) \leq O(\lg n)$

Lower Bound

Modelo de comparaciones

↳ Datos solamente comparables

→ Estrategia de adversario

~ Manteniendo pero consistente

Algoritmo en el modelo de comparación
performs $\lg n$ comparaciones
en el peor caso

→ (Combinatorial)

III Sorting

ADT n elements comparable
(non ordered)

ORDENAR()

- DS
- Quick Sort $O(n^2)$ ~~$n \log n$~~
 - Bubble Sort $O(n^2)$
 - Mergesort $O(n \log n)$
 - Insertion Sort $O(n^2)$
 - Selection Sort $O(n^2)$
 - Heap Sort $O(n \log n)$
 - Bucket Sort
 - Radix Sort
 - Counting Sort
- Discrete Values Model
- Tim Sort
- Adaptive Merge Sort
 - Huffman Merge Sort

Lower Bound

Arboles de Decision