

Divide y vencerás en Busqueda



DEFINICIONES

Divide y Vencerás

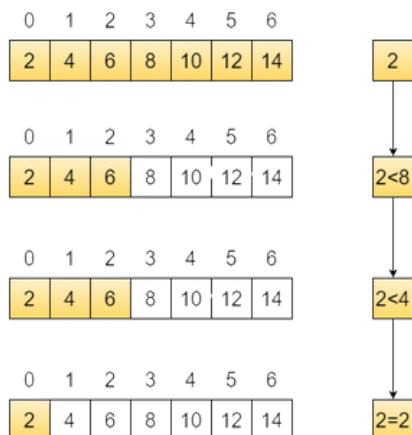
- **Qué es:** Estrategia que divide un problema grande en subproblemas más pequeños, los resuelve por separado y combina sus soluciones.



ALGORITMOS -DEFINICION

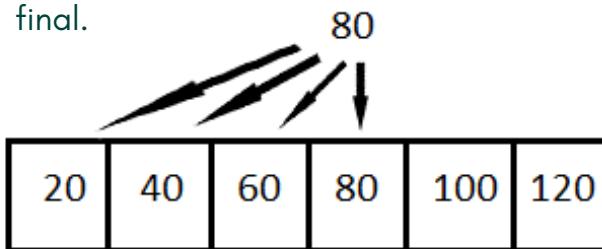
Busqueda Binaria

- **Qué es:** Un algoritmo de búsqueda que busca en lista ordenada, corta por la mitad. Mira el centro y si no es, descarta la mitad donde no está y repite hasta encontrarlo.



Busqueda Lineal

- **Qué es:** Un algoritmo de búsqueda que recorre la lista elemento por elemento. Compara cada uno con el objetivo. Si coincide, está encontrado. Si no, pasa al siguiente y repite hasta llegar al final.



CUANDO USARLOS

- Busqueda binaria:
 - Listas ORDENADAS (obligatorio)
 - Listas grandes
- Busqueda Lineal:
 - Listas pequeñas
 - Listas DESORDENADAS

Complejidad

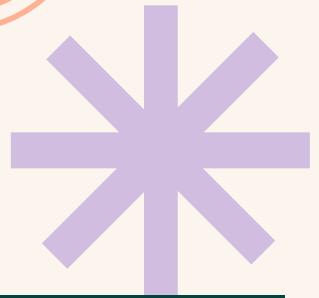
Busqueda binaria:

- $O(\log n) \rightarrow$ Super rápida en listas grandes.
- Crece logarítmicamente con el tamaño de la lista

Busqueda Lineal:

- $O(n) \rightarrow$ Lenta en listas grandes.
- Crece linealmente con el tamaño de entrada

Recursividad



DEFINICIONES

Que es?

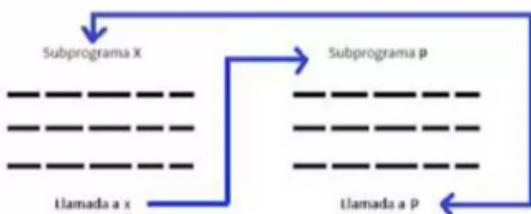
Una función que se llama a sí misma para resolver un problema.



DEFINICION

RECURSIVIDAD INDIRECTA

- La función A llama a la función B, y B llama de nuevo a A.



COMPONENTES

Caso base

- Condición que detiene la recursión.
- Sin él, la función se llamaría infinitamente.

Llamada recursiva

- La función se llama a sí misma con un problema más pequeño.

PILA DE LLAMADAS

- Cada llamada recursiva se apila en memoria.
- Se resuelve en orden LIFO (Last In, First Out).
- Si la pila se llena → Stack Overflow.

Recursion vs Iteracion

Recursion

- Mas clara para problemas repetitivos
- Usa pila de llamadas
- Riesgo de stack overflow

Iteracion

- Más eficiente en memoria
- Usa bucles (for, while)
- Control directo de memoria