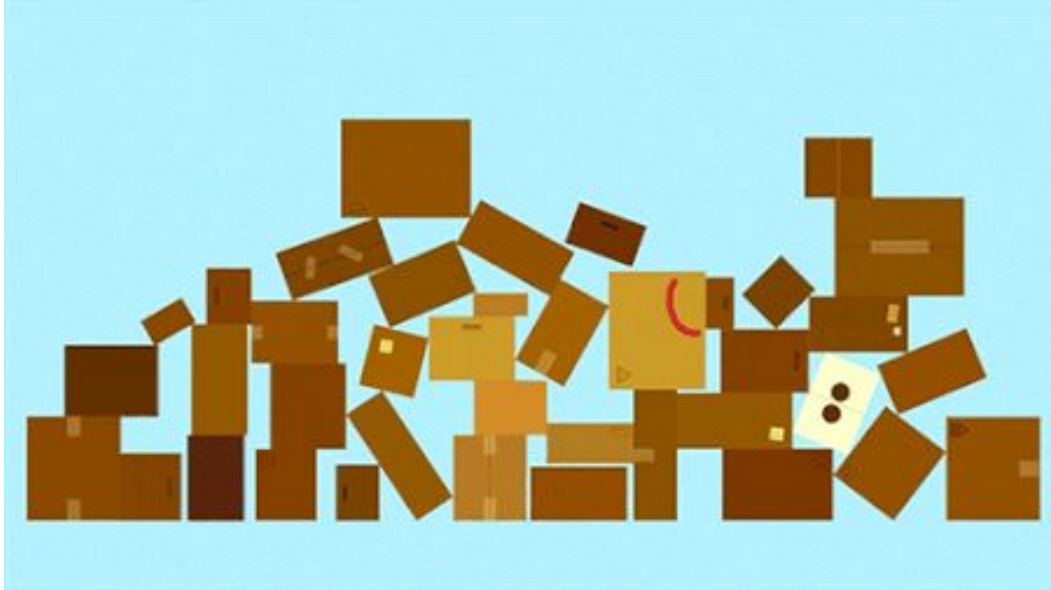


Algoritmos de Ordenamiento

DEV.F
DESARROLLAMOS(PERSONAS);

dev



Algoritmos de Ordenamiento

Están diseñados para colocar elementos de una lista en un orden específico.

DEV.F
DESARROLLAMOS(PERSONAS);

dev

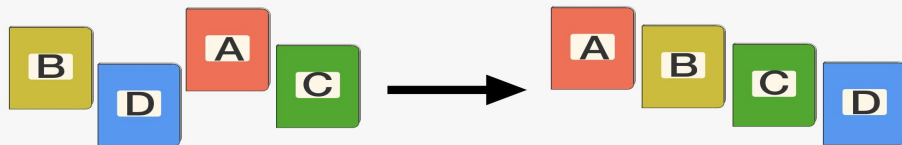


¿Qué podemos ordenar?

Cualquier estructura de datos con elementos que sean ordenables

- Podemos **ordenar números**, porque unos son mayores que otros.
- Podemos **ordenar meses**, porque unos vienen antes que otros
- Podemos **ordenar palabras**, por el orden en el alfabeto.

Sorting Algorithms



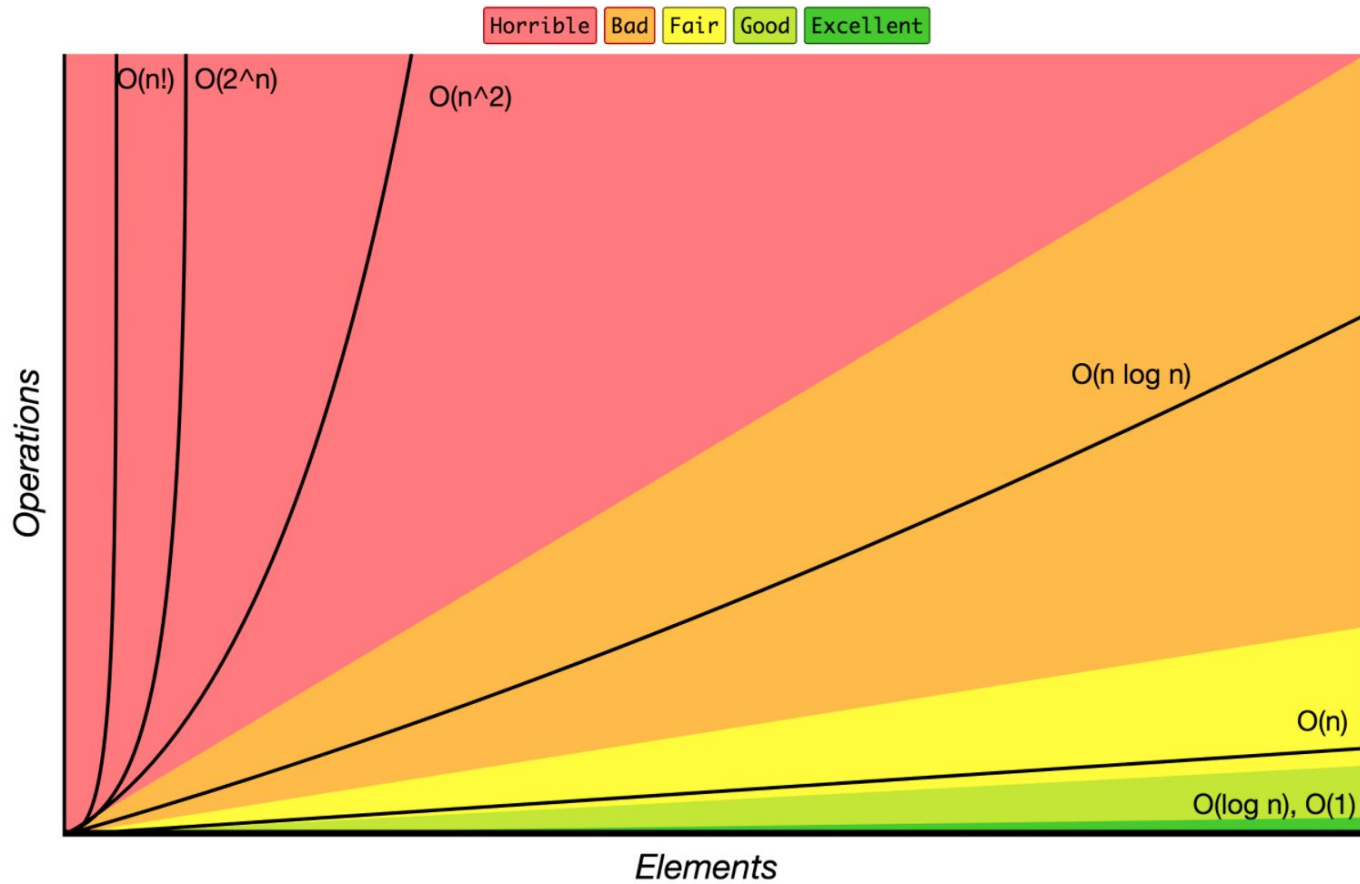
Algoritmos de ordenación

Algunos ejemplos:

- **Ordenamiento de burbuja (Bubble sort)**
- **Ordenamiento por selección (Selection sort)**
- **Ordenamiento por inserción (insertion sort)**
- **Ordenamiento por unión (Merge sort)**
- **Ordenamiento rápido (Quick sort)**
- **Ordenamiento por montones (Heap Sort)**

La eficiencia del ordenamiento es importante ya que **impacta en la eficiencia de otros algoritmos**, como los algoritmos de búsqueda

Algorithm	Best Time Complexity	Average Time Complexity	Worst Time Complexity	Worst Space Complexity
Linear Search	$O(1)$	$O(n)$	$O(n)$	$O(1)$
Binary Search	$O(1)$	$O(\log n)$	$O(\log n)$	$O(1)$
Bubble Sort	$O(n)$	$O(n^2)$	$O(n^2)$	$O(1)$
Selection Sort	$O(n^2)$	$O(n^2)$	$O(n^2)$	$O(1)$
Insertion Sort	$O(n)$	$O(n^2)$	$O(n^2)$	$O(1)$
Merge Sort	$O(n \log n)$	$O(n \log n)$	$O(n \log n)$	$O(n)$
Quick Sort	$O(n \log n)$	$O(n \log n)$	$O(n^2)$	$O(\log n)$
Heap Sort	$O(n \log n)$	$O(n \log n)$	$O(n \log n)$	$O(n)$



Bubble Sort



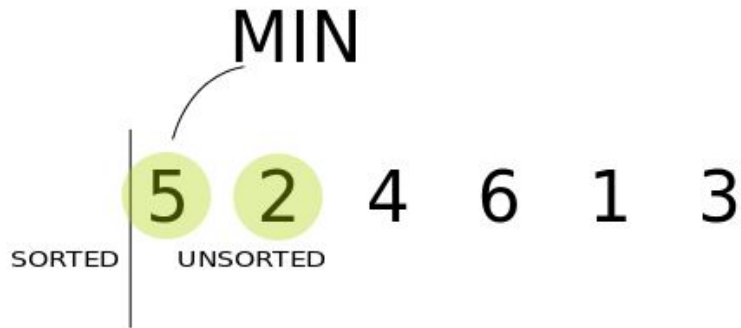
Bubble Sort

6 5 3 1 8 7 2 4

Selection Sort



Selection Sort



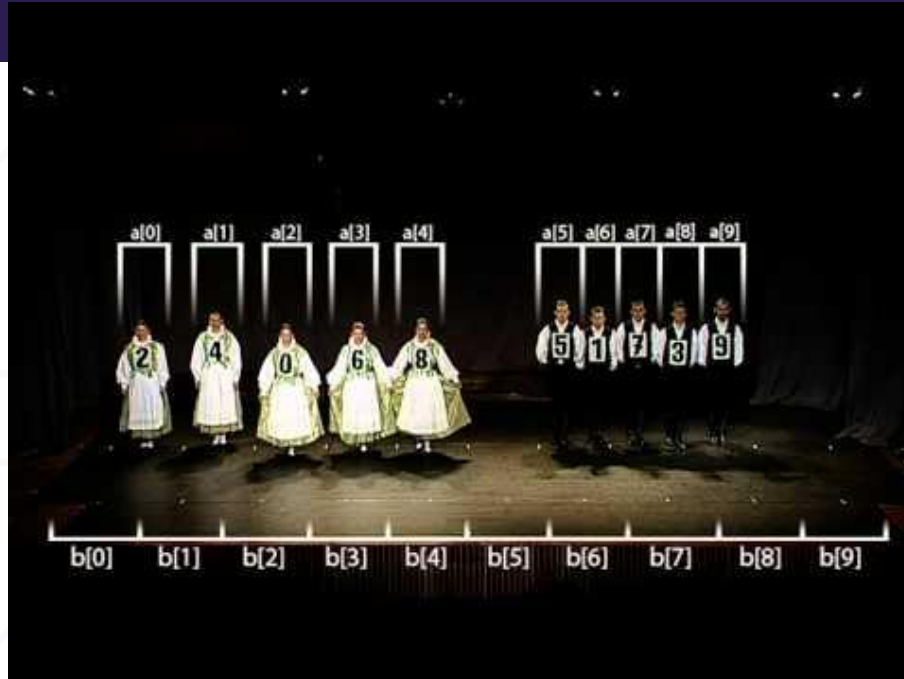
INSERTION SORT



Insertion Sort



Merge Sort

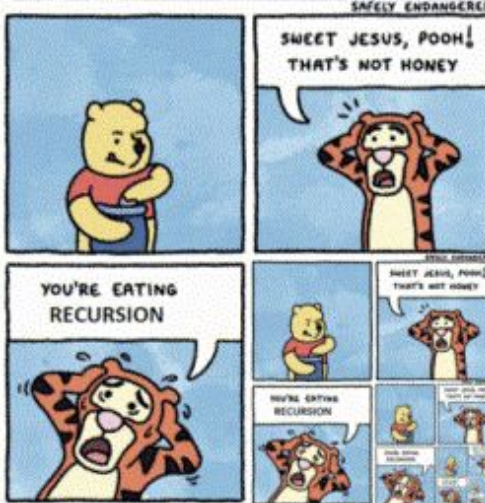
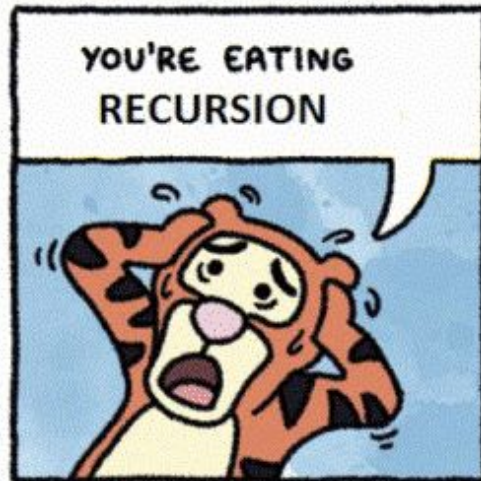
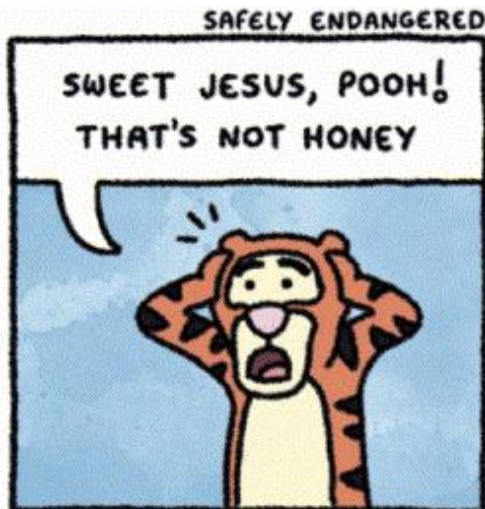


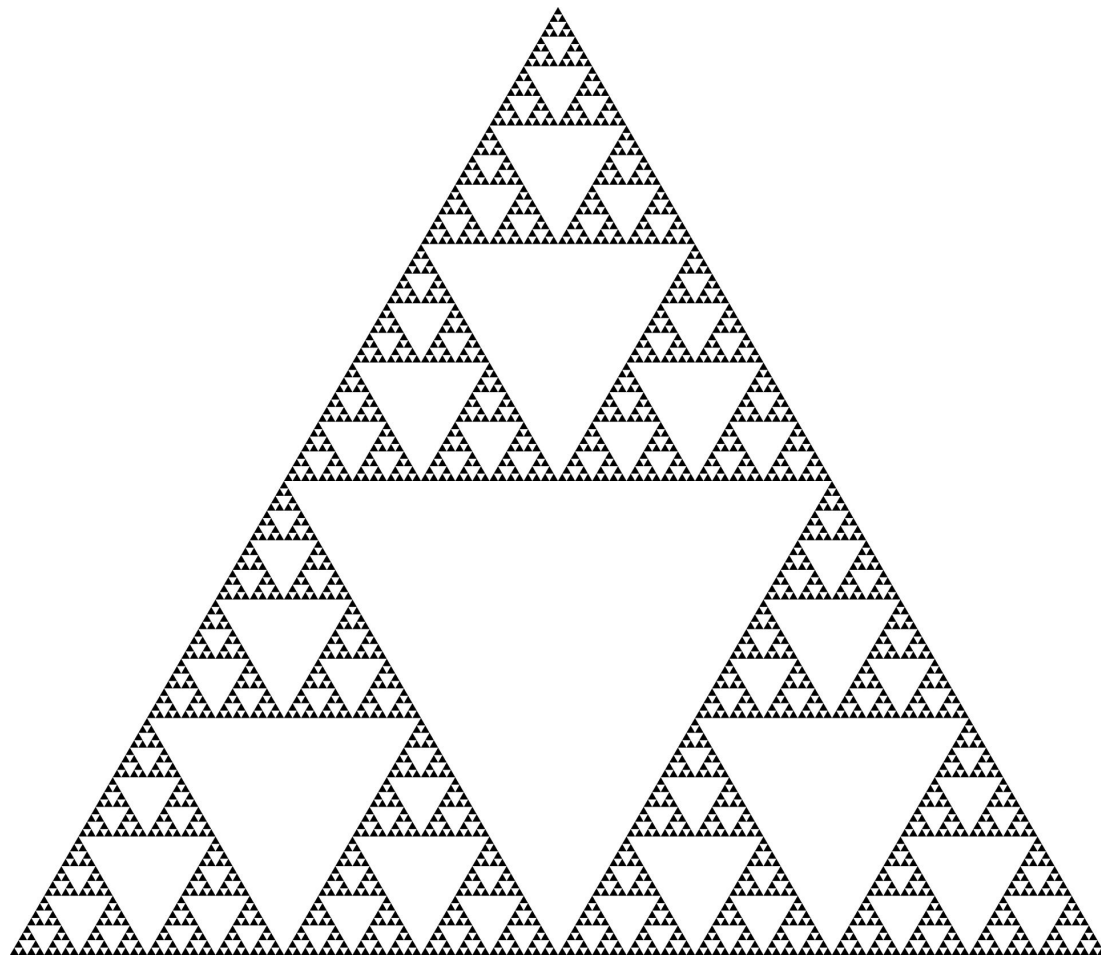
Merge Sort

6 5 3 1 8 7 2 4

Recursión

Es la forma en la cual se especifica un **proceso basado en su propia definición**







Quick Sort



3 8 2 1 5 4 6 7

A horizontal array of eight numbers, each enclosed in a rounded square box with a dark teal border. The numbers are 3, 8, 2, 1, 5, 4, 6, and 7, arranged from left to right. The boxes are set against a light gray background with a faint circuit-like pattern.