



Algoritmos de Ordenamiento

Entendiendo la complejidad de un algoritmo



Clasificación de complejidades

En computación y matemáticas un algoritmo de ordenamiento es un algoritmo que pone elementos de una lista o un vector en una secuencia dada por una relación de orden, es decir, el resultado de salida ha de ser una permutación —o reordenamiento— de la entrada que satisfaga la relación de orden dada.



Clasificación de complejidades

**Algoritmos de ordenamiento interno:
en la memoria del ordenador.**

**Algoritmos de ordenamiento externo:
en un lugar externo como un disco duro.**

**Algoritmos de ordenación natural:
Tarda lo mínimo posible cuando la entrada está ordenada.**

**Algoritmos de ordenación no natural:
Tarda lo mínimo posible cuando la entrada está inversamente ordenada.**



Complejidad computacional

En términos de n , el tamaño de la lista o arreglo. Para esto se usa el concepto de orden de una función y se usa la notación $O(n)$.

El mejor comportamiento para ordenar es $O(n \log n)$. Los algoritmos más simples son cuadráticos, es decir $O(n^2)$.

Los algoritmos que aprovechan la estructura de las claves de ordenamiento tienen un desempeño lineal, es decir $O(n)$.



Complejidad computacional

Ordenamiento de burbuja Bubblesort $O(n^2)$

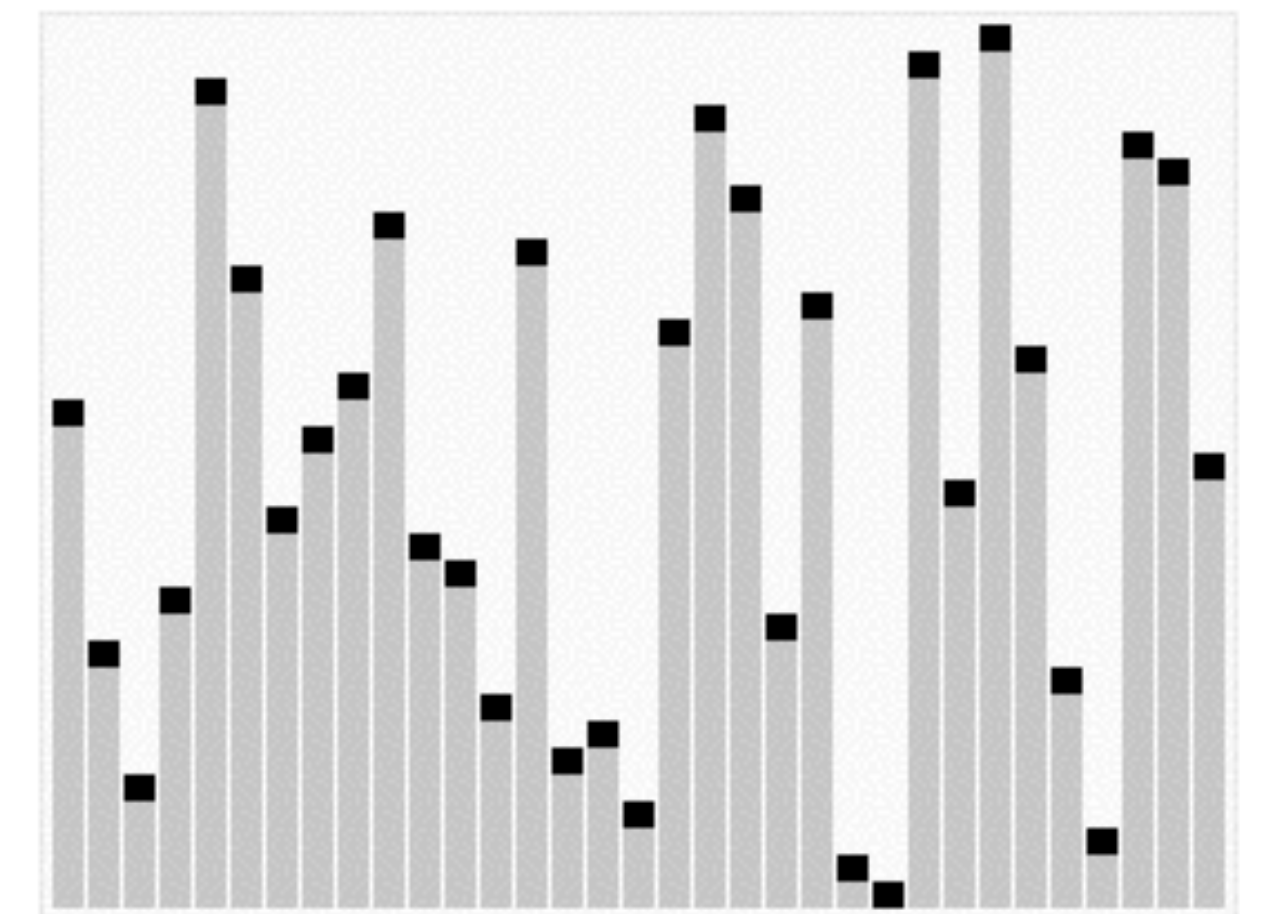
Ordenamiento por inserción Insertion sort $O(n^2)$

Ordenamiento por cuentas Counting sort $O(n+k)$

Ordenamiento por mezcla Merge sort $O(n \log n)$

Ordenamiento con árbol binario Binary tree sort $O(n \log n)$

Ordenamiento rápido Quicksort Promedio: $O(n \log n)$,
peor caso: $O(n^2)$





Lets Try this

Ordenamiento de burbuja Bubblesort $O(n^2)$

6 5 3 1 8 7 2 4



¡Gracias!