

# Algoritmia

## Objetivo de la Cheat Sheet

Que es un algoritmo y la algoritmia.

Que es una estructura de datos.

Complejidad temporal y espacial, O de Landau.

Un problema usual: Ordenación

Estructuras de datos y complejidad.

## Algoritmos y algoritmia

Un algoritmo es una secuencia finita de instrucciones rigurosas diseñados para resolver una tarea. Son las ideas detrás de los programas. La algoritmia es el estudio de los algoritmos.

## Estructuras de datos

Una estructura de datos es una forma de organizar un conjunto de datos. Una organización puede facilitar ciertas operaciones frente a otras. Conocer las diferentes estructuras de datos y sus características facilita la creación de buenos algoritmos.

## Complejidad

No todos los algoritmos que solucionan un problema son iguales. Algunos requieren realizar más instrucciones (usan más tiempo) o necesitan más memoria (usan más espacio). Este uso de recursos (tiempo y espacio) suele depender del tamaño del problema que resuelve el algoritmo. Para comparar el gasto de recursos entre algoritmos se usa la notación de la O de Landau. Si un problema tiene tamaño  $n$  (cual es el tamaño puede depender del problema concreto) la complejidad se expresa como  $\mathcal{O}(f(n))$  lo que indica que cuando  $n$  tiende a infinito el consumo de ese recurso es inferior a  $f(n)$ .

## Ordenación

Un problema usual es el de la ordenación de un vector de números. En este problema el tamaño es la longitud del vector que denominaremos  $n$ . Además no es igual de costoso ordenar un vector casi ordenado que uno especialmente preparado para que tu algoritmo tarde mucho. Por ello a veces la complejidad de los algoritmos se estudia en función de esto. En el caso de la ordenación en el mejor de los casos el vector ya está ordenado, en un caso medio el vector está colocado al azar y en el peor la entrada está diseñada para perjudicar a nuestro algoritmo. Con eso en mente algunos algoritmos conocidos tienen las siguientes complejidades:

Algoritmo	Complejidad temporal		
	mejor	medio	peor
Quicksort	$\mathcal{O}(n \log(n))$	$\mathcal{O}(n \log(n))$	$\mathcal{O}(n^2)$
Mergesort	$\mathcal{O}(n \log(n))$	$\mathcal{O}(n \log(n))$	$\mathcal{O}(n \log(n))$
Bubble Sort	$\mathcal{O}(n)$	$\mathcal{O}(n^2)$	$\mathcal{O}(n^2)$

## Estructuras de datos y complejidad

Existen muchas operaciones que se pueden hacer en una estructura de datos pero algunas son muy comunes. Podemos añadir un dato (inserción), podemos acceder a un dato (acceso), podemos buscar un dato (búsqueda) o podemos borrar un dato (borrado). Diferentes estructuras de datos tienen diferente coste para estas operaciones respecto a la cantidad de datos que contienen ( $n$ ):

Estructura	Complejidad temporal media			
	Inserción	Acceso	Búsqueda	Borrado
Vector	$\mathcal{O}(n)$	$\mathcal{O}(1)$	$\mathcal{O}(n)$	$\mathcal{O}(n)$
Cola	$\mathcal{O}(1)$	$\mathcal{O}(n)$	$\mathcal{O}(n)$	$\mathcal{O}(1)$
Pila	$\mathcal{O}(1)$	$\mathcal{O}(n)$	$\mathcal{O}(n)$	$\mathcal{O}(1)$

## Aprender más

Si te interesa más este tema Steven Skiena ofrece gratuitamente su [curso de introducción a la algoritmia](#) en youtube