



Definición de algoritmo y órdenes de complejidad

En esta presentación, explicaremos la definición de algoritmo, la recursividad y los órdenes de complejidad, conceptos fundamentales en el desarrollo de software. Aprenderemos a identificar y analizar la eficiencia de los algoritmos utilizando la notación Big O.

¿Qué es un algoritmo?

1 Definición

Un algoritmo es una secuencia finita de instrucciones bien definidas y ordenadas, diseñadas para resolver un problema o realizar una tarea específica.

2 Características

Los algoritmos deben ser precisos, no ambíguos, efectivos y finitos.

3 Aplicaciones

Los algoritmos se utilizan en una amplia gama de campos, desde la programación hasta la toma de decisiones.

Recursividad

Definición

La recursividad es una técnica de programación en la que una función se llama a sí misma para resolver un problema.

Ejemplo

Un ejemplo clásico de algoritmo recursivo es el cálculo del factorial de un número.

1

2

Caso base

El caso base es la condición que detiene la recursividad, evitando que el algoritmo se vuelva infinito.

3

Algoritmos recursivos

Ventajas

Los algoritmos recursivos son elegantes, compactos y a menudo más fáciles de entender que las soluciones iterativas.

Permiten resolver problemas complejos de una manera natural y concisa.

Desventajas

Pueden consumir más recursos de memoria y tiempo de ejecución que los algoritmos iterativos, especialmente para problemas grandes.

Es importante definir adecuadamente el caso base para evitar bucles infinitos.

Ejemplos

Cálculo de factorial, generación de secuencias de Fibonacci, búsqueda en árboles, entre otros.

Órdenes de complejidad

1 Definición

El orden de complejidad de un algoritmo es una medida de cuánto tiempo o espacio de memoria se requiere para ejecutar el algoritmo en función del tamaño de la entrada.

2 Tipos

Los órdenes de complejidad más comunes son:

constante ($O(1)$), lineal ($O(n)$),
logarítmica ($O(\log n)$),
cuadrática ($O(n^2)$) y
exponencial ($O(2^n)$).

3 Importancia

Analizar el orden de complejidad de un algoritmo