

Algoritmia Elemental (Caso peor y Operación Elemental)

1.4. Caso peor

¿Qué es el caso peor en algoritmos?

El caso peor analiza la situación más desfavorable que puede enfrentar un algoritmo. Es decir, el mayor tiempo o mayor uso de recursos que podría necesitar el algoritmo para resolver un problema.

¿Por qué es importante?

Permite asegurar que el algoritmo no fallará ni tomará un tiempo excesivo, incluso en las peores circunstancias.

Se utiliza en sistemas donde la confiabilidad y tiempos límite son críticos (por ejemplo: sistemas médicos, aviación, seguridad bancaria).

Ejemplo sencillo

Búsqueda lineal en una lista de n elementos:

Si el elemento no está en la lista o está en la última posición, el algoritmo tendrá que revisar todos los elementos.

Entonces, el caso peor es de n operaciones.

Complejidad del peor caso: $O(n)$

Representación en complejidad

En notación Big O, el caso peor se representa como:

- $O(n)$ para búsqueda lineal.
- $O(n^2)$ para ordenamiento burbuja en el peor caso.
- $O(\log n)$ para búsqueda binaria en su peor escenario.

1.5. Operación elemental

¿Qué es una operación elemental?

Una operación elemental es una acción básica en un algoritmo que:

- Se realiza en un solo paso.
- Consume siempre el mismo tiempo (o se considera constante).
- Es independiente del tamaño de los datos.

Ejemplos de operaciones elementales:

- Sumar, restar, multiplicar, dividir dos números.
- Asignar un valor a una variable.
- Comparar dos valores ($a = b$, $a > b$, etc.).
- Acceder a un elemento de un arreglo (por índice).

¿Por qué son importantes?

Al analizar la eficiencia de un algoritmo, contamos cuántas operaciones elementales realiza.

Permiten estimar el tiempo total del algoritmo sumando todas esas operaciones.

Ayudan a comparar algoritmos de forma objetiva.