



Análisis de complejidad de algoritmos

Análisis de complejidad de algoritmos

- Complejidad del peor caso: es la complejidad de resolver el problema para la peor entrada de tamaño n . Proporciona el límite superior del algoritmo. Este es el análisis más utilizado.
- Complejidad promedio del caso: es la complejidad de resolver el problema en promedio. Calculamos el tiempo para todas las entradas posibles y luego tomar un promedio.
- Complejidad del mejor caso: es la complejidad de resolver el problema para la mejor entrada de tamaño n .

Tiempo Constante $O(1)$

Se dice que un algoritmo se ejecuta en tiempo constante si la salida se produce en tiempo constante independientemente del tamaño de entrada.

Ejemplos:

1. Accediendo al n -ésimo elemento de una matriz
2. Colocar y retirar nodos a una pila.
3. Agregar y quitar nodos a una cola.
4. Accediendo a un elemento de Hash-Table.



Tiempo Lineal $O(n)$

Se dice que un algoritmo se ejecuta en tiempo lineal si el tiempo de ejecución del algoritmo es directamente proporcional al tamaño de entrada.

Ejemplos:

1. Operaciones de matriz como elemento de búsqueda, encontrar mínimo, encontrar máximo, etc.
2. Operaciones de listas vinculadas como recorrido, encontrar mínimo, encontrar máximo, etc.

Nota: cuando necesitamos ver/atravesar todos los nodos de una estructura de datos para alguna tarea, entonces la complejidad no es menor que $O(n)$.

Tiempo N-LogN $O(n \log(n))$

Se dice que un algoritmo se ejecuta en tiempo $n \log n$ si el tiempo de ejecución de un algoritmo es proporcional al producto del tamaño de entrada por el logaritmo del tamaño de entrada n . En estos algoritmos, cada vez la entrada se divide a la mitad y cada porción se procesa de forma independiente.

Ejemplo:

1. Combinar-Ordenar, Quicksort
2. Clasificación rápida (caso promedio)
3. Clasificación en montón

Tiempo cuadrático $O(n^2)$

Se dice que un algoritmo se ejecuta en tiempo cuadrático si el tiempo de ejecución de un algoritmo es proporcional al cuadrado del tamaño de entrada. En estos algoritmos cada elemento se compara con todos los demás elementos.

Ejemplos:

1. Clasificación de burbujas
2. Selección-Ordenar
3. Ordenación por inserción



Tiempo Exponencial $O(2^n)$

En estos algoritmos, todos los posibles subconjuntos de elementos de entrada son generados.

Tiempo Factorial $O(n!)$

En estos algoritmos, todas las posibles permutaciones de elementos de de entrada son generados.