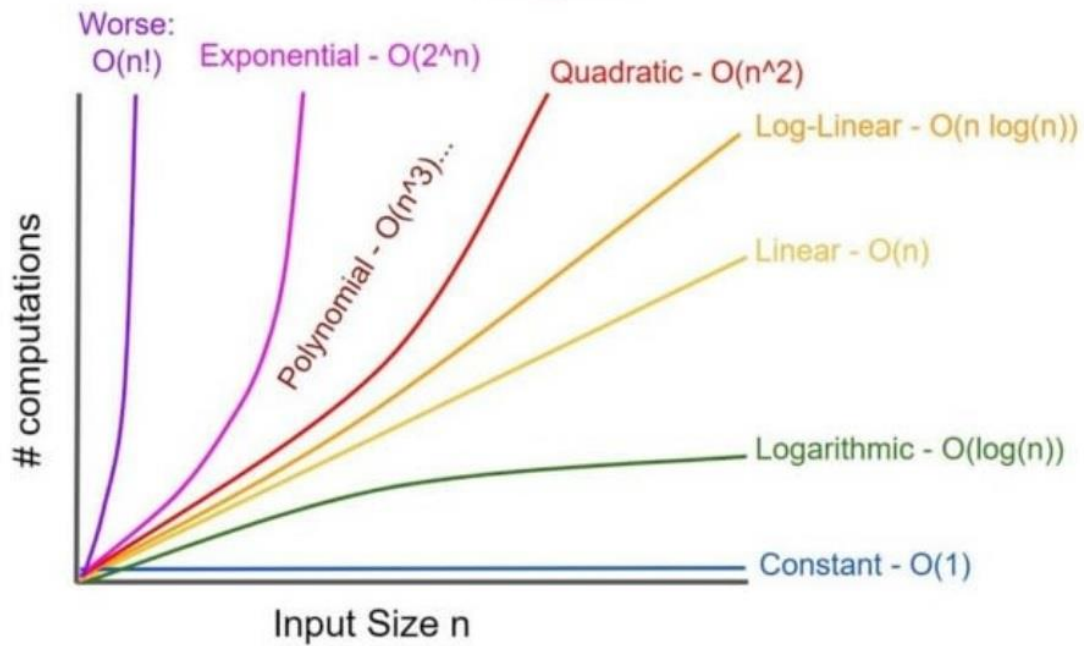


Lista de Ejercicios

Búsqueda Linear y Binaria

-
1. Dada una array de números enteros $A[1 \dots n]$ tal que, para todo i , $1 \leq i \leq n$, tenemos $|A[i] - A[i+1]| < 1$. Con $A[1]=x$ y $A[n]=y$, tal que $x < y$.
Diseñe un algoritmo de búsqueda eficiente para encontrar j tal que $A[j] = z$ para un valor dado z , $x \leq z \leq y$.
Cuál es el número máximo de comparaciones a z que realiza su algoritmo?
 2. Reciba como entrada un **conjunto S de n números reales** y un **número real x** . Diseñe un algoritmo para determinar si **existen dos elementos adyacentes en S , cuya suma es x** .
 - Suponga que S está ordenado, El algoritmo debe rodar en un tiempo máximo **$O(n)$**
-
3. Reciba como entrada un **conjunto S de n números reales** y un **número real x** . Diseñe un algoritmo para determinar si **existen dos elementos en S , cuya suma es x** .
 - Suponga que S está ordenado, El algoritmo debe rodar en un tiempo máximo de **$O(n \log n)$**



$O(1)$: Complejidad constante. Cuando las instrucciones se ejecutan una vez.

$O(\log n)$: Complejidad logarítmica. Esta suele aparecer en determinados algoritmos con iteración o recursión no estructural, por ejemplo, la búsqueda binaria.

$O(n)$: Complejidad lineal. Aparece en la evaluación de bucles simples siempre que la complejidad de las instrucciones interiores sea constante.

$O(n \log n)$: Complejidad cuasi-lineal. Se encuentra en algoritmos de tipo divide y vencerás como por ejemplo en el método de ordenación quicksort y se considera una buena complejidad. Si n se duplica, el tiempo de ejecución es ligeramente mayor del doble.

$O(n^2)$: Complejidad cuadrática. Aparece en bucles o ciclos doblemente anidados. Si n se duplica, el tiempo de ejecución aumenta cuatro veces.