

Estructura de Datos y Algoritmos (Python)



Autoevaluación - Tema 3

1. El análisis de algoritmos
 - a) nos permite conocer los errores de sintaxis de un programa.
 - b) nos dice si un problema es fácil o difícil de programar.
 - c) sirve para estimar cómo de eficiente es un algoritmo.

2. En el análisis teórico de algoritmos,
 - a) debemos implementar el algoritmo y medir el tiempo que tarda para resolver el problema con distintos tamaños para los datos de entrada.
 - b) únicamente medimos la complejidad temporal, mientras que la complejidad espacial se mide por medio del análisis empírico.
 - c) es independiente tanto del entorno hardware como del software, porque se calcula una cota superior de la función temporal del algoritmo.

3. La complejidad de un algoritmo:
 - a) es la complejidad espacial, es decir, se refiere a la cantidad de memoria que el algoritmo necesita para ejecutarse.
 - b) se refiere a los recursos (espaciales, temporales o de otro tipo) que el algoritmo requiere para ser ejecutado.
 - c) tiene la propiedad de que a mayor complejidad temporal, más eficiente es el algoritmo.

4. Mejor y peor caso:
 - a) cuando hablamos sobre la complejidad temporal de un algoritmo, siempre nos debemos referir a su mejor caso.
 - b) cuando hablamos sobre la complejidad temporal de un algoritmo, siempre nos debemos referir a su peor caso.
 - c) la complejidad de un algoritmo será la media entre su peor y su mejor caso.

5. Respecto a los órdenes de complejidad, por favor, ordena las funciones de menor complejidad a mayor complejidad
 - a) $1 < \log n < n < n \log n < n^2 < n^3 < 2^n$
 - b) $1 < \log n < n \log n < n < n^2 < n^3 < 2^n$
 - c) $1 < n < \log n < n \log n < n^2 < n^3 < 2^n$