NOTACION BIG-O

Ulises Isaac Rocha Huerta

Unidad 1

Tarea 3: Notación Big-O

Resumen escrito de los videos de Notación Big-O

21/marzo/2019

La notación Big O presentada en los videos describe las siguientes características de un algoritmo:

1. Rendimiento
2. Complejidad
3. Eficiencia

La notación Big-O es un tipo de análisis asintótico y ayuda a la eficiencia de un algoritmo.

La notación Big-O nos dice que tan positivo se crea un algoritmo en algunos casos lo hacen con alguna complejidad y en el video con la notación describe y evalúa la complejidad que se está usando en algún algoritmo, también explica cómo trabajan los algoritmos creados ya sean con los datos, la memoria almacenada o el disco duro.

Las expresiones que se usan para evaluar con la notación Big-O son las siguientes:

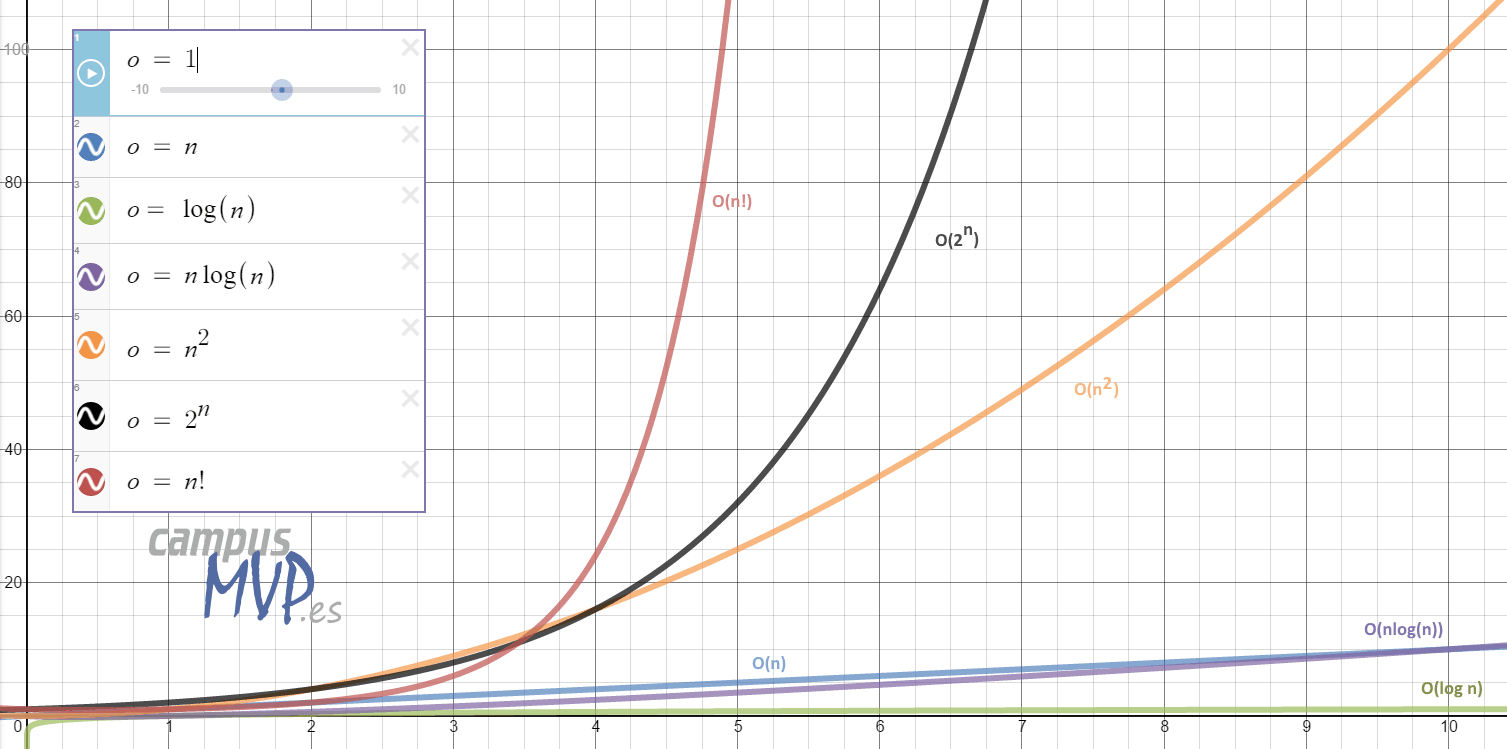
1. O (1)
2. O (log N)
3. O (N log N)
4. O(N)
5. O(N^2)
6. O(2^N)

A continuación, intentare explicar a lo que entendí, el significado de cada expresión anterior:

Descripción de cada expresión

**O (1):** Indica un tiempo constante, significa que el algoritmo tendrá el mismo rendimiento sin importar las entradas.

**O(N):** Describe en una gráfica el crecimiento lineal y la complejidad se basa en las proporciones de la entrada.



**O(2^N):** Vi que es como las funciones recursivas de un código fuente, y al utilizarse esta notación en un algoritmo pues es poco eficiente.

**O(N^2):** Muestra la complejidad basada en las entradas de datos.

**O(NlogN):** Es una notación casi como a log N, pero este tipo de expresión corta el problema para cada problema.

**O(logN):** esta notación es como un algoritmo que divide el mismo algoritmo, para encontrar la respuesta al problema

Conclusión:

La notación Big O ayuda a los programadores a analizar algoritmos de una manera más eficiente y saber cuándo usar un método de programación, y así resolver problemáticas y solucionar rápidamente los problemas de programación.