

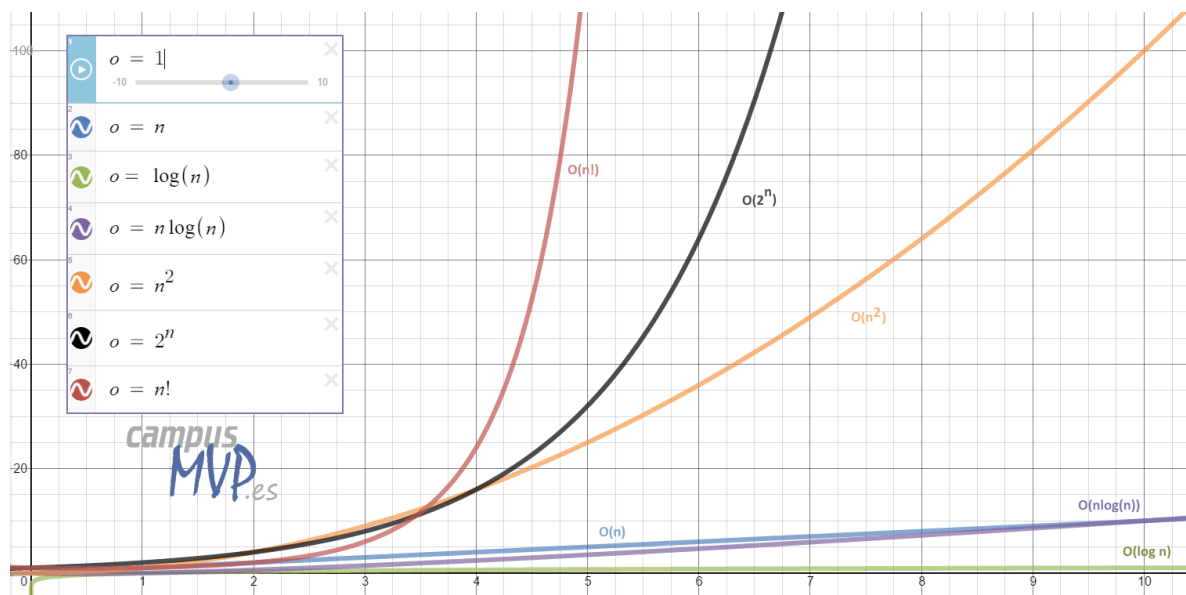
Notación Big-O (Resumen Clase Semana 6)

La notación Big-O provee una cota superior de complejidad en el peor caso y, con ello ayuda a cuantificar el rendimiento a medida que el tamaño de entrada se vuelve arbitrariamente grande.

La notación Big-O nos proporciona una manera de saber cómo se va a comportar un algoritmo en función de los argumentos que le pasemos y la escala de los mismos.

N – el tamaño de la entrada:

- Tiempo constante: $O(1)$
- Tiempo logarítmico: $O(\log(n))$
- Tiempo lineal: $O(n)$
- Tiempo linealitmico: $O(n\log(n))$
- Tiempo cuadrático: $O(n^2)$
- Tiempo cúbico: $O(n^3)$
- Tiempo exponencial: $O(b^n)$, $b > 1$
- Tiempo factorial: $O(n!)$



En el eje horizontal tenemos el número de elementos, y en el vertical el tiempo.

Como podemos observar, a medida que aumenta la complejidad, el tiempo necesario para completar la tarea crece mucho, pudiendo llegar a aumentar enormemente en algunos casos en cuanto hay más datos a manejar.

Referencias.

José Manuel Alarcón. (2016). Rendimiento de algoritmos y notación Big-O. 14/07/2020, de campusmvp.es Sitio web: <https://www.campusmvp.es/recursos/post/Rendimiento-de-algoritmos-y-notacion-Big-O.aspx>