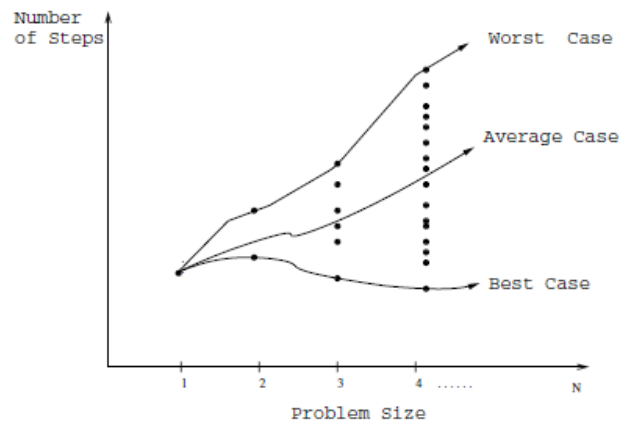


**Actividad: Investigar el análisis de casos de Notación Big O (mejor, peor y promedio estricto).**

Para comprender las nociones de la mejor, la peor y la complejidad del caso promedio, piense en ejecutar un algoritmo sobre todas las instancias posibles de datos que puedan alimentarse.



- **Mejor caso:** Mínimo número de pasos necesarios para finalizar la tarea.
- **Caso promedio:** Esto es ambiguo.
- **Peor caso:** La cantidad máxima de pasos que un algoritmo tomará, en cualquier caso. Simplemente no puede ser peor que eso para cualquier entrada.

***La complejidad del peor de los casos resulta ser la más útil de estas tres medidas en la práctica.***

***La notación de Big O y el peor de los casos son herramientas que simplifican enormemente nuestra capacidad para comparar la eficacia de los algoritmos.***

**Ejemplo 1:**

$$T(n) = 12754n^2 + 4353n + 834 \lg 2 n + 13546$$

Es un trabajo muy difícil, pero nos proporciona poca información adicional a la observación de que "el tiempo crece cuadráticamente con n." Ahora podemos ver que expresar tiempos algorítmicos en el mejor, medio y peor caso posible puede ser extremadamente tedioso y complicado.

**Ejemplo 2:**

Imagina que eres un maestro y tienes una estudiante de nombre Jane. Quieres encontrar sus registros, así que usas un algoritmo de búsqueda simple para recorrer la base de datos de tu distrito escolar.

Sabes que a la búsqueda simple le toma  $O(n)$  veces ejecutarse. Esto significa que, en el peor de los casos, tendrás que buscar en cada uno de los registros (representados por  $n$ ) para encontrar el de Jane.

Pero cuando ejecutas la búsqueda simple, encuentras que los registros de Jane son la primera entrada en la base de datos. No tienes que mirar cada entrada – la encontraste en tu primer intento.

¿Este algoritmo tardó  $O(n)$  tiempo? ¿O tardó  $O(1)$  porque encontraste los registros de Jane en el primer intento?

En este caso,  $O(1)$  es el mejor de los casos: tuviste suerte de que los registros de Jane estuvieran al principio. Pero la notación Big O se enfoca en el peor de los casos, el cual es  $O(n)$  para la búsqueda simple. Es una garantía de que la búsqueda simple nunca será más lenta que el tiempo  $O(n)$ .

### Referencias:

- <https://www.freecodecamp.org/espanol/news/explicacion-de-la-notacion-big-o-con-ejemplo/>
- <https://es.linkedin.com/pulse/qu%C3%A9-es-la-notaci%C3%B3n-de-big-oh-en-algoritmos-gurupratap-matharu>