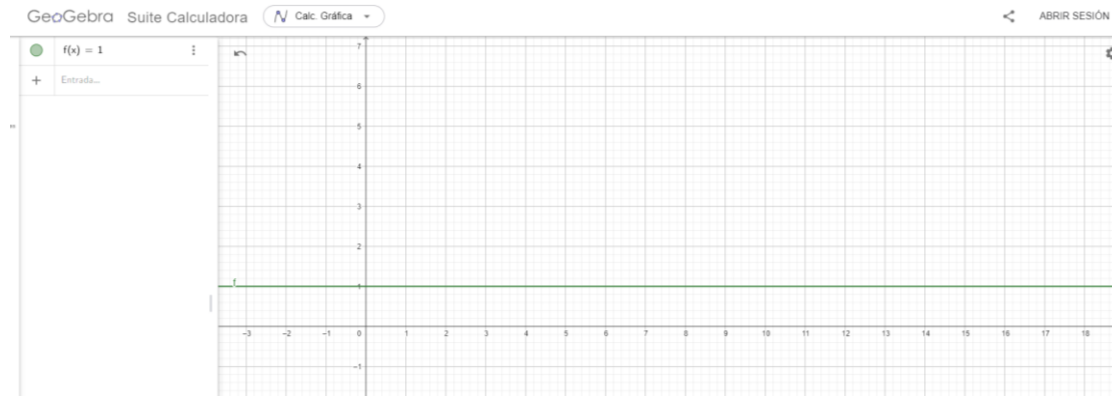


MARIACA VAZQUEZ ENRIQUE

¿Qué significa $O(1)$? Es un tipo de complejidad para los algoritmos de orden constante

dibuja la gráfica de $O(1)$



¿Qué es un algoritmo de tiempo exponencial? Es un algoritmo donde el tiempo de ejecución crece como su nombre, de manera exponencial cuando más crece el número de entradas.

¿qué es un algoritmo? Es un grupo de instrucciones que tienen como función realizar una determinada tarea en determinados pasos.

¿Cuál es la complejidad del algoritmo de búsqueda en un árbol? $O(\log_2 n)$.

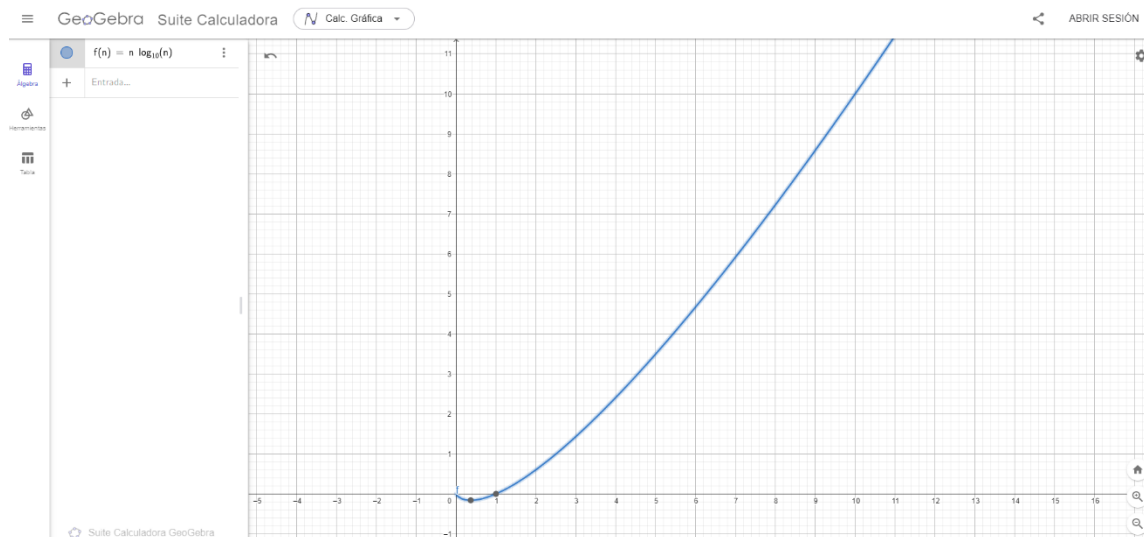
Proporciona un ejemplo de un algoritmo con complejidad $O(n \log n)$. El algoritmo de ordenamiento Merge sort trabaja como $O(n \log n)$.

Proporciona un ejemplo de un algoritmo con complejidad $O(1)$. El algoritmo para acceder a un elemento de una lista usando su índice, trabajando como $O(1)$.

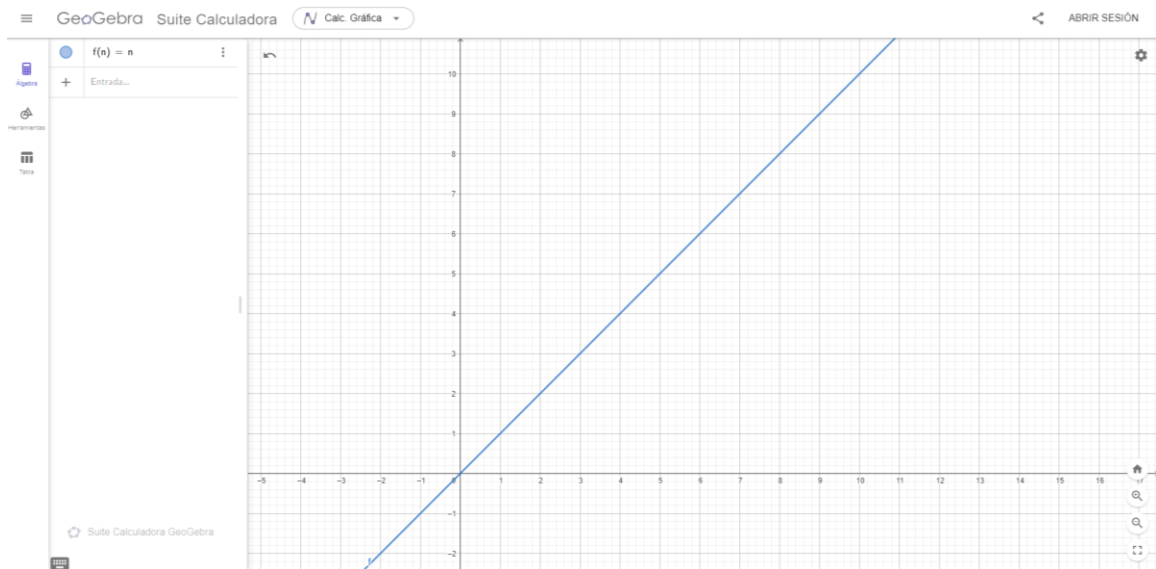
¿Cuál es la complejidad del algoritmo de ordenamiento rápido (quick sort)? $O(n \log n)$.

¿Cuál es la complejidad del algoritmo de búsqueda binaria? $O(\log n)$.

dibuja la gráfica de $O(n \log n)$



dibuja la gráfica de $O(n)$



¿Qué significa que un algoritmo tiene complejidad logarítmica? Significa que el tiempo de ejecución es, como su nombre lo indica, logarítmico, es decir, que aumenta de manera logarítmica con respecto al tamaño que recibe en la entrada.

escribe en orden las diferentes complejidades $O(n)$, etc

- $O(1)$.
- $O(n)$.
- $O(n \log n)$.
- $O(\log_2 n)$.
- $O(n^2)$
- $O(n!)$

Proporciona un ejemplo de un algoritmo con complejidad temporal $O(n)$. La búsqueda de un elemento en una lista no ordenada, utilizando una búsqueda lineal trabajando como $O(n)$.