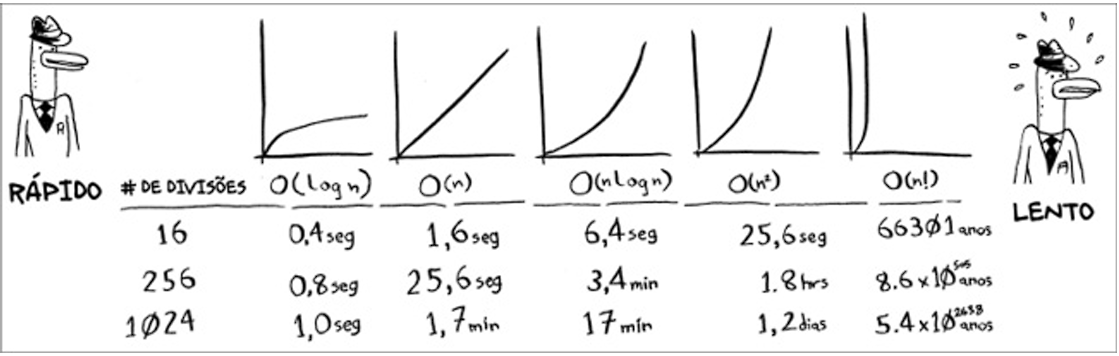
A notação Big O é uma ferramenta matemática usada para descrever a eficiência de algoritmos em termos de tempo de execução ou uso de espaço, conforme o tamanho da entrada aumenta. Ela foca **SEMPRE** no **pior caso**, ou seja, o cenário em que o algoritmo consome mais recursos. Por exemplo:

* **O(1)**: Tempo constante, independente do tamanho da entrada. VERIFICADO/FEITO APENAS 1 VEZ
* **O(n)**: Tempo linear, cresce proporcionalmente ao tamanho da entrada. VERIFICAÇÃO 1 A 1, PONTA A PONTA.
* **O(log n)**: Tempo logarítmico, muito eficiente para grandes entradas. VERIFICAÇAÕ POR PESQUISA BINÁRIA
* **O(n²)**: Tempo quadrático, menos eficiente para entradas maiores**. (dois loops aninhados)** VERIFICAÇÃO E COMPARAÇAO DE CADA ELEMENTO COM **TODO** OUTRO ELEMENTO DO ARRAY. (bubble sort e ordenação por seleção)
* **O(n!):** Exemplo: um algoritmo bastante lento, como o do caixeiro-viajante

Suponha que você esteja desenhando novamente a grade de 16 divisões e você possa escolher cinco algoritmos diferentes para fazer isso.

Se escolher o primeiro algoritmo, levará um tempo de execução de O(log n) para desenhar a grade. **Você pode fazer dez operações por segundo**. Com o tempo de execução O(log n), você levará quatro operações para desenhar uma grade com 16 divisões **(log2 16 é 4).** Logo, levará 0,4 segundos para desenhar a grade. E se tiver que desenhar 1.024 divisões? Levará 1.024 = 210 operações, ou um segundo para desenhar uma grade de 1.024 divisões. Estes números são para o primeiro algoritmo.

O segundo algoritmo é mais lento: ele tem tempo de execução O(n). Levará 16 operações para desenhar 16 divisões e levará 1.024 operações para desenhar 1.024 divisões. Quanto tempo isso leva em segundos? Aqui está quanto tempo levaria para desenhar a grade com os algoritmos restantes, do mais rápido ao mais lento



* A rapidez de um algoritmo não é medida em segundos, mas pelo crescimento do número de operações.
* Em vez disso, discutimos sobre o quão rapidamente o tempo de execução de um algoritmo aumenta conforme o número de elementos aumenta.
* O tempo de execução em algoritmos é expresso na notação Big O.
* **O(log n)** é mais rápido do que **O(n)**,e **O(log n)** fica ainda mais rápido conforme a lista aumenta