METODO MAESTRO

El teorema maestro proporciona una solución sencilla en términos asintóticos (usando una Cota superior asintótica) para ecuaciones de recurrencia que ocurren en muchos algoritmos **recursivos** y trabaja de la siguiente forma, para un n grande:

$$\text{Si } T(n) <= a\, T(n/b) + O(n^d)$$

Donde T(n) representa el número de operaciones que el algoritmo necesita para resolver un problema

Entonces:

$$\begin{array}{ll} T(n) \text{ es de } O(n^d \log n) & \text{si } a=b^d \\ T(n) \text{ es de } O(n^d) & \text{si } a<b^d \\ T(n) \text{ es de } O(n^{\log_b a}) & \text{si } a>b^d \end{array}$$

Por ejemplo la recurrencia que representa al algoritmo MERGE SORT:

$$\begin{cases} T_{MS}(n) = 2\, T(n/2) + O(n) \\ T_{MS}(1) = 2 \end{cases}$$

Como en este caso $a=b^d$, por que se tiene $2=2^1$

El tiempo $T_{MS}(n)$ es de $O(n \log n)$

La otra opción es analizar la recurrencia y luego demostrar por inducción.

Casos Irresolubles

Los siguientes casos no pueden ser resueltos a través de la utilización del teorema maestro:

- $T(n) = 2^n T\left(\frac{n}{2}\right) + n^n$

  a no es una constante; el numero de subproblemas debe ser fijo.

- $T(n) = 2T\left(\frac{n}{2}\right) + \frac{n}{\log n}$

  f(n) debe ser polinomial.

- $T(n) = 0.5T\left(\frac{n}{2}\right) + n$

  a<1 no puede darse el caso de que haya menos de un subproblema.

- $T(n) = 64T\left(\frac{n}{8}\right) - n^2 \log n$

  f(n) que es el tiempo de trabajo, no puede ser negativo.

- $T(n) = T\left(\frac{n}{2}\right) + n(2 - \cos n)$

  Caso 3 pero hay una violación de regularidad.