- Chave de Resposta -

1) Dada a relação de recorrência da Busca Binária Recursiva apresentada abaixo, calcule a complexidade assintótica de **pior caso**.

Obs.: Não é necessário fazer a verificação, somente resolver (encontrar a SFF).

$$\begin{cases} T(1) = 1, & \text{para} \;\; n = 1 \\ T(n) = T(\frac{n}{2}) + \Theta(1), & \text{para} \;\; n > 1 \end{cases}$$

- Expandindo a Relação de Recorrência, temos:

$$\begin{aligned} T(n) &= T(\frac{n}{2}) + 1 \\ &= T(\frac{n}{4}) + 1 + 1 \\ &= T(\frac{n}{8}) + 1 + 1 + 1 \\ &= T(\frac{n}{16}) + 1 + 1 + 1 + 1 \\ &= T(\frac{n}{2^4}) + 4 \end{aligned}$$

- Assim, após k passos, temos:

$$T(n) = T(\frac{n}{2^k}) + k \tag{1}$$

- Ao final esperamos obter:

$$\begin{aligned} \frac{n}{2^k} &= 1 && (\textit{A execução finaliza em } T(1), \textit{ i.e., quando } \frac{n}{2^k} \textit{ atingir } 1) \\ 2^k &= n \\ \log_2 2^k &= \log_2 n \\ k \log_2 2 &= \log_2 n \\ k &= \log_2 n && (2) \end{aligned}$$

- Substituindo (2) em (1):

$$\begin{aligned} T(n) &= T(\frac{n}{2^{\log_2 n}}) + \log_2 n && (2^{\log_2 n} = n^{\log_2 2} = n) \\ &= T(\frac{n}{n^{\log_2 2}}) + \log_2 n \\ &= T(\frac{n}{n}) + \log_2 n && (T(\frac{n}{n}) = T(1) = 1) \\ &= 1 + \log_2 n && \Rightarrow \quad O(\log_2 n) \end{aligned}$$