

Considere o arquivo <http://flaviomoura.info/files/paa-2021-2-pd.pdf>.

1. (4 pontos) Use o método da substituição para mostrar que a recorrência

$$T(n) = 1 + \sum_{j=0}^{n-1} T(j)$$

que representa a complexidade do algoritmo 2 (corte-haste), tem complexidade exponencial.

2. (4 pontos) Use o método da substituição para mostrar que a recorrência

$$\text{Para } n > 1, P(n) = \sum_{k=1}^{n-1} P(k) \cdot P(n-k).$$

tem solução

$$P(n) = \Omega(2^n)$$

$$\textcircled{1} \quad T(n) = 1 + \sum_{j=0}^{n-1} T(j)$$

$$= 1 + T(0) + T(1) + \dots + T(n-1)$$

$$= 1 + T(0) + T(1) + \dots + T(n-2) + 1 + T(0) + T(1) + \dots + T(n-2)$$

$$= 2 \cdot \left(1 + \sum_{j=0}^{n-2} T(j) \right)$$

$$= 2^2 \cdot \left(1 + \sum_{j=0}^{n-3} T(j) \right)$$

$$= 2^3 \cdot \left(1 + \sum_{j=0}^{n-4} T(j) \right)$$

$$= \dots$$

$$= 2^{n-1} \cdot \left(1 + \sum_{j=0}^0 T(j) \right)$$

$$= 2^{n-1} \cdot (1 + 1)$$

$$= 2^n.$$

Agora usamos indução para mostrar que $T(n) = 2^n$ é, de fato, uma solução da recorrência $T(n) = 1 + \sum_{j=0}^{n-1} T(j)$.

Para $n=0$ (base da indução), temos $T(0) = 1 + \sum_{j=0}^{-1} T(j) = 1 = 2^0$. ✓

Para $n > 0$ (passo indutivo), temos $T(n) = 1 + \sum_{j=0}^{n-1} T(j) = 1 + T(0) + \dots + T(n-1)$

$$\stackrel{(h.i)}{=} 1 + 2^0 + 2 + \dots + 2^{n-1}$$

$$= 1 + (2^n - 1) = 2^n. \quad \square$$

$$\textcircled{2} \quad \text{Para } n > 1, P(n) = \sum_{k=1}^{n-1} P(k) \cdot P(n-k).$$

Queremos mostrar que $P(n) = \Omega(2^n)$, ou seja, $P(n) \geq c \cdot 2^n$ para alguma constante $c > 0$.

A prova é por indução em n . Temos $P(n) = \sum_{k=1}^{n-1} P(k) \cdot P(n-k) \geq \sum_{k=1}^{n-1} c \cdot 2^k \cdot c \cdot 2^{n-k} = \sum_{k=1}^{n-1} c^2 \cdot 2^n = c^2 \cdot 2^n \cdot \sum_{k=1}^{n-1} 1 = c^2 \cdot (n-1) \cdot 2^n \geq c \cdot 2^n$ pois $c > 0$ e $n > 1$. □