Universidade do Vale do Rio dos Sinos - Unisinos
São Leopoldo - RS, 29/Outubro/2024
Henrique Haag

Assim como pedido, dado o algoritmo das Torres de Hanoi:

```python
def torre_de_hanoi (n, fonte, destino, muleta):
    if n == 1:
        print(f"Mova disco 1 da {fonte} para o {destino}")  #->c
        return
    torre_de_hanoi(n-1, fonte, muleta, destino)
    print(f"Mova disco {n} da {fonte} para o {destino}")    #->d
    torre_de_hanoi(n-1, muleta, destino, fonte)
# n = número de discos -> no caso do exercício, usei 4 discos
n = 4
torre_de_hanoi(n, "A", "B", "C")
```

podemos montar os seguintes cálculos:

$$T(n) = \begin{cases} 1, & n=1 \\ T(n-1)+T(n-1)+1, & n>1 \end{cases}$$

$$\hookrightarrow 2T(n-1)+1,\ n>1$$

$T(n) = 2T(n-1)+1$

$T(n-1) = 2T(n-2)+1$

$T(n-2) = 2T(n-3)+1$

$T(n-3) = 2T(n-4)+1$

$\vdots$

$T(1) = 1$

$T(n) = 2T(n-1)+1 \Rightarrow i=1$

$T(n-1) = 2(2T(n-2)+1)+1$

$= 4T(n-2)+2+1$

$= 4T(n-2)+3 \Rightarrow i=2$

$T(n-2) = 4(2T(n-3)+1)+3$

$= 8T(n-3+4+3$

$= 8T(n-3)+7 \rightarrow i=3$

$T(n-3) = 8(2T(n-4)+1)7$

$16T(n-4)+8+7$

$16T(n-4+15 \rightarrow i=4$

$T(n) = 2^i T(n-i) + (2^i - 1) \qquad i = n-1$

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

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

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

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

$$T(1) = 2^{1-1} + 2^{1-1} - 1$$

$$T(1) = 2^0 + 2^0 - 1$$

$$T(1) = 1 + 1 - 1$$

$$\boxed{T(1) = 1}$$

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

$$= 2^n + 2^n - 1$$

$$T(n+1) = 2T(n-1+1) + 1$$

$$T(n+1) = 2T(n) + 1$$

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

$$= 2^{n-1+1} + 2^{n-1+1} - 2 + 1$$

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

q.e.d.