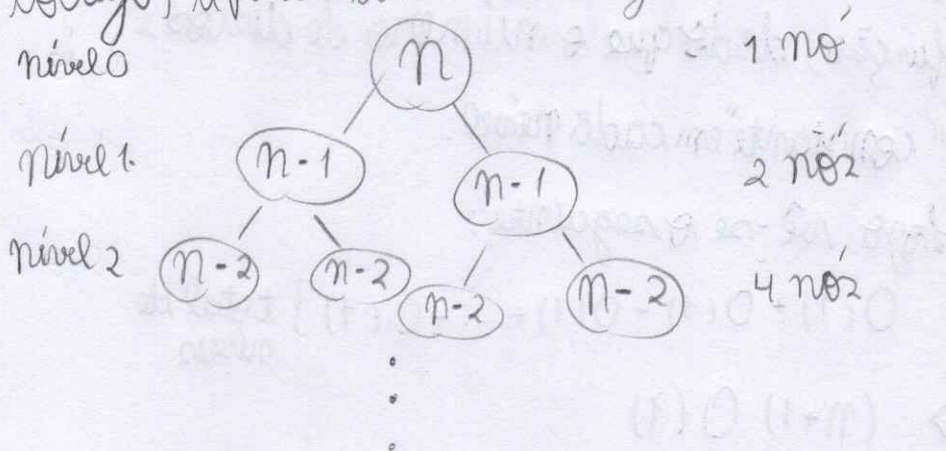


2) Análise assintótica para o tempo:

É possível retratar o tempo de execução do código, a partir da árvore a seguir:



Nível n : 3 nós (rotulados n)

Observação: n é o total de discos declarados na main().

De cada nó tem $O(1)$ como complexidade de execução, então 2^{n+1} nós possui esta complexidade na notação "O":

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

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

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

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

Análise assintótica para a memória:

Observa-se que na árvore cada novo nó possui o mesmo caráter de chamada da função, dado que o número de discos é "constante" em cada nível.

Logo, vê-se o seguinte:

$$O(1) + O(1) + O(1) + \dots + O(1) \} \text{ total de níveis}$$

$$\Rightarrow (n+1) O(1)$$

$$\Rightarrow n \cdot O(1) + O(1)$$

$$\Rightarrow O(n) + O(1)$$

$$\therefore ((n+1) O(1) = O(n))$$