Heap - Build max heap

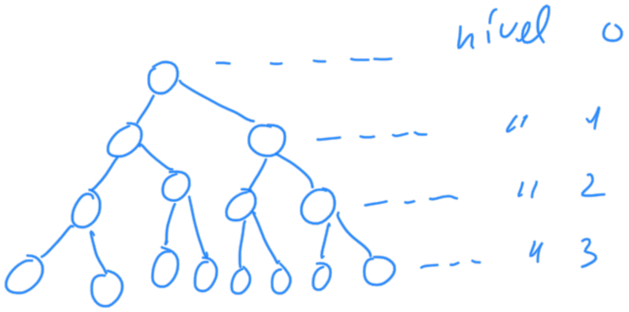
1.

Uma árvore binária balanceada com N nós tem altura igual a $\log_2(N+1)$.

Por simplicidade, assume-se que nº de nós é uma potência inteira de 2 menos um.

$$N = 2^n - 1$$

$$N = 2^4 - 1 = 15$$



$$\Rightarrow \lceil \log_2(2^n - 1) \rceil = n$$

- nível k tem 2^k nós
- h níveis $\Rightarrow$ N nós $= 2^n - 1$ nós $\Rightarrow \underbrace{2^n - 1}_{N} = \underbrace{\sum_{k=0}^{h-1} 2^k}_{\text{nós dos níveis}}$

$$\Leftrightarrow 2^n - 1 = \underbrace{a_0 \cdot \frac{r^n - 1}{r - 1}}_{\text{progressão geométrica}} = 1 \cdot \frac{2^h - 1}{2 - 1}$$

$$\Leftrightarrow 2^n \cancel{-1} = 2^h \cancel{-1} \Leftrightarrow h = n \quad \text{ou} \quad N = 2^h - 1 \Leftrightarrow \boxed{h = \log_2(N+1)}$$

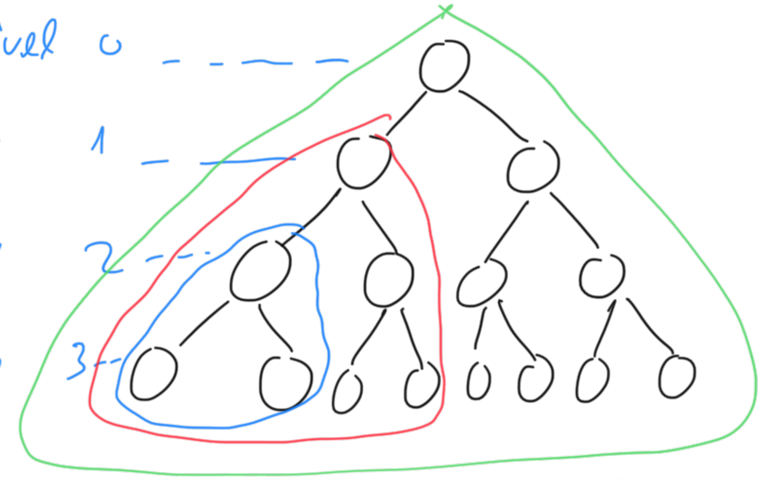
Complexidade do build max-heap

nível 0

" 1

" 2

" 3



15 elementos $= 2^4 - 1$

4 × heaps dimensão 3 + 2 × heaps dim. 7 + 1 × heap dim. 15

$4 \times 1 + 2 \times 2 + 1 \times 3 = 11$ promoções no pior caso

max-heapify

(2× mais comparações)

$\boxed{\text{Considerando } N = 2^n - 1}$

$$\boxed{\sum_{1 \le k < n} k \cdot 2^{n-k-1}} = \underbrace{2^n - n - 1}_{\text{não vou provar}} < N$$

→ é um upper bound

3.

$$\sum_{\substack{1 \le k < 4 \\ \shortparallel \\ n}} k \cdot 2^{4-k-1} = 1 \cdot 2^2 + 2 \cdot 2^1 + 3 \cdot 2^0 = 4 + 4 + 3 = 11$$

nº de heaps

nº de promoções/max-heapify

$\Downarrow$

logo, complexidade temporal de build max-heap é $O(N)$