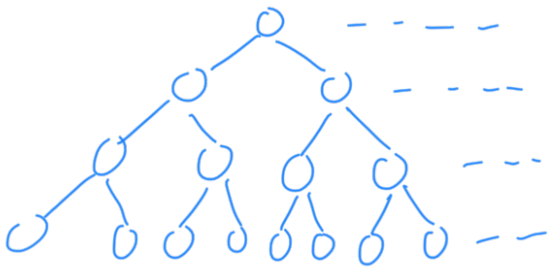
Build-Heap - Complexidade

1.

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

Assume-se, por simplicidade, que nº nós é uma potência inteira de 2 menos um:

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



nível 0

" 1

" 2

" 3

nº níveis

$n = 4$

$N = 2^n - 1 =$

$= 2^4 - 1 = 15$

$\lceil \log_2(2^n - 1) \rceil = 4$

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

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

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

$$\Leftrightarrow \boxed{h = \log_2(N+1)}$$

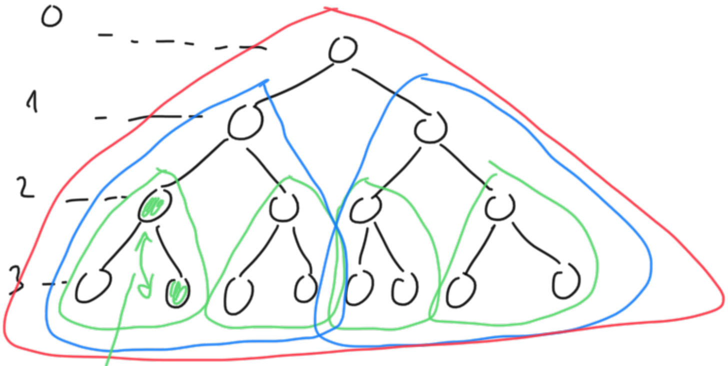
Build_heap

nível 0

" 1

" 2

" 3



15 elementos $= 2^4 - 1$

4 x heaps dimensão 3 + 2 x heaps dim. 7

+ 1 heap dim. 15

$\Rightarrow 4 \times 1 + 2 \times 2 + 1 \times 3 = 11$ promoções/max-heapify no pior caso

nº de max_heapify

(2x mais comparações)

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

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

é um upper bound

ex: $n=4$

$$\sum_{\substack{1 \leq k < 4 \\ = \\ n}} k \cdot 2^{4-k-1} = \underbrace{1}_{\text{nº de max-heapify}} \cdot \underbrace{2^2}_{\text{nº heaps dim. 3}} + 2 \cdot \underbrace{2^1}_{\text{nº heaps dim. 7}} + 3 \cdot \underbrace{2^0}_{\text{nº heaps dim. 15}} = 4+4+3 = 11$$

Complexidade Temporal de build-heap: $O(N)$