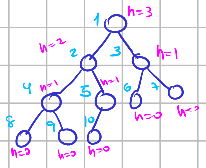
Heaps: Árbol casi lleno

De un nodo i:

Parent(i) → $\lfloor i/2 \rfloor$

left(i) → 2i

right(i) → 2i+1



Se tiene un heap con altura h

$2^h \leq n < 2^{h+1}$

¿Cuál es mínimo y máximo de nodos?

$h \leq \lg n < h+1$

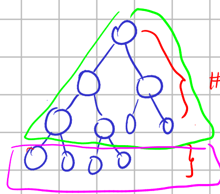
entero, decimal, entero

$h = \lfloor \lg n \rfloor$

Demostración de que los hijos izquierdo y derecho de i son 2i y 2i+1, respectivamente.

Si i=1, los hijos izquierdo y derecho son 2i=2 y 2i+1=3, respectivamente.

Si i>1. Por hipótesis de inducción los hijos izquierdo y derecho de i-1 son 2(i-1) y 2(i-1)+1, respectivamente. Y como sabemos que los hijos i-1 siguen inmediatamente a los hijos de i, tenemos que los hijos de i son 2i y 2i+1



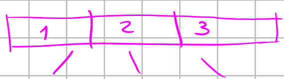
#nodos = 2^d, d<h

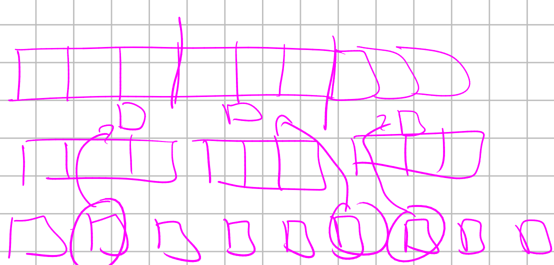
$\sum_{i=0}^{n} x^i = \frac{x^{n+1}-1}{x-1}$

$1 \leq$ #nodos $\leq 2^h$

$\sum_{i=0}^{h-1} 2^i + x = 2^h - 1 + x \in [2^h, 2^{h+1}-1]$

$2^h - 1 + 1 = 2^h$ $\quad 2^h - 1 + 2^h = 2^{h+1} - 1$





a.b, n dígitos

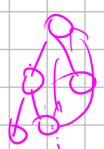
$a = a_1 \times 10^m + a_2$

$b = b_1 \times 10^m + b_2$

$a \cdot b = (a_1 \times 10^m + a_2)(b_1 \times 10^m + b_2)$

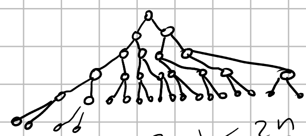
$= a_1 b_1 \times 10^{2m} + (a_1 b_2 + a_2 b_1) 10^m + a_2 b_2$

$(a_1+b_1)(a_2+b_2) - a_1$



Muestre que un heap con n elementos tiene altura $\lfloor \lg n \rfloor$

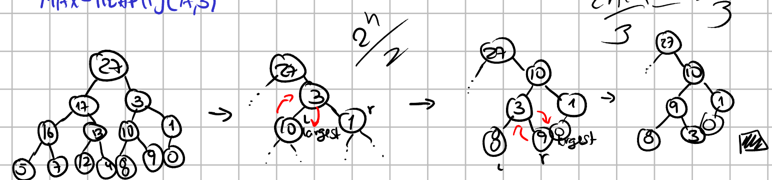
```
MAX-HEAPIFY(A, i)
    l = left(i)
    r = right(i)
    if l ≤ A.heapsize and A[l] > A[i]
        largest = l
    else
        largest = i
    if r ≤ A.heapsize and A[r] > A[largest]
        largest = r
    if largest ≠ i
        exchange A[i] and A[largest]
        MAX-HEAPIFY(A, largest)
```



$A = [27, 17, 3, 16, 13, 10, 1, 5, 7, 12, 4, 8, 9, 0]$

MAX-HEAPIFY(A,3)

$\frac{2n-1}{3} \leq \frac{2n}{3}$



Demostrando que Build-max-heap es O(n)

$T(n) = \sum_{h=1}^{\lfloor \lg n \rfloor} |\{i : \text{altura de } i \text{ es } h\}| \cdot O(h)$

$\leq \sum_{h=1}^{\lfloor \lg n \rfloor} (\lceil n/2^{h+1} \rceil k \cdot h)$

$\leq \sum_{h=1}^{\lfloor \lg n \rfloor} \left(\left(\frac{n}{2^{h+1}} + 1 \right) \cdot kh \right)$

$= \sum_{h=1}^{\lfloor \lg n \rfloor} \frac{nkh}{2^{h+1}} + \sum_{h=1}^{\lfloor \lg n \rfloor} kh$

$= \frac{kn}{2} \sum_{h=1}^{\lfloor \lg n \rfloor} h/2^h + k \sum_{h=1}^{\lfloor \lg n \rfloor} h$

$\leq \frac{kn}{2} \cdot \sum_{h=0}^{\infty} h/2^h + k \sum_{h=1}^{\lfloor \lg n \rfloor} h$

$= \frac{kn}{2} \cdot \frac{(1/2)}{(1/2)^2} + k \sum_{h=1}^{\lfloor \lg n \rfloor} h$

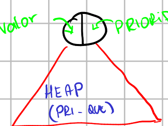
$= kn +$

$\sum_{i=0}^{\lfloor \lg n \rfloor} i = \frac{\lfloor \lg n \rfloor (\lfloor \lg n \rfloor + 1)}{2} \leq \frac{\lg n^2}{2} + \frac{\lg n}{2} \leq \frac{n}{2}$

$\frac{\lg^2 n}{2} = O(n)$

PRIORITY-QUEUE / HEAPSORT

23/09



* Para simplificar un solo valor

generalización con delete(A,i)

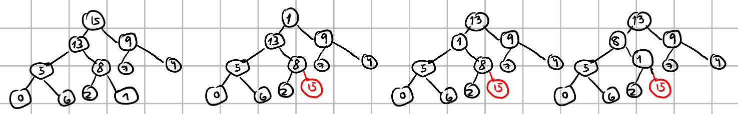
→ EXTRACT-MAX(A)
Toma el máximo y elimina el elemento del heap.

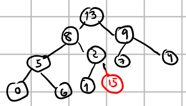
→ HEAP-INCREASE-KEY(A, i, key)

→ INSERT(A, v)
inserta al final y realiza heapify-up.

ejercicio:

$A = [15, 13, 9, 5, 8, 7, 4, 0, 6, 2, 1]$





INSERT (A,10)