¿Cuántas hojas tiene un heap con n nodos?

$1 \quad 2 \quad 3 \quad \cdots \quad \left\lfloor \frac{n}{2} \right\rfloor \quad \cdots \quad n$

$\left\lfloor \frac{n}{2} \right\rfloor$

$\left\lfloor \frac{n}{2} \right\rfloor + 1$

nodos hoja

$\left.\begin{array}{l} 2(\;) \\ 2(\;)+1 \end{array}\right\} \leq n$

n

$\# \text{hojas} = \left\lceil \frac{n}{2} \right\rceil$

$\frac{n}{2} \cdot 2 = n$

Recordar:

* $x-1 < \lfloor x \rfloor \leq x$

* $\boxed{\left\lceil \frac{n}{2} \right\rceil + \left\lfloor \frac{n}{2} \right\rfloor = n}$

Ejercicio 1. Muestre que un heap con n elementos tiene altura $\lfloor \lg n \rfloor$.

$n=1$ $\quad \lfloor \lg 1 \rfloor = 0$, $\quad h=0$

* $n > \varnothing$

heap con n nodos tiene $\lceil \frac{n}{2} \rceil$ hojas (g anterior)

Sea un heap con $\lfloor \frac{n}{2} \rfloor$ nodos luego de borrar todas las hojas del heap original.

Por hipótesis de inducción, la altura del heap original es $\lfloor \lg \lfloor \frac{n}{2} \rfloor \rfloor + 1$

Mostraremos que este número es igual a $\lfloor \lg n \rfloor$

$$\boxed{\lfloor \lg n \rfloor \le \lg n < \lfloor \lg n \rfloor + 1}$$

$$2^{\lfloor \lg n \rfloor} \le n < 2^{\lfloor \lg n \rfloor + 1}$$

$$\mathbb{Z} \; 2^{\lfloor \lg n \rfloor - 1} \le \frac{n}{2} < 2^{\lfloor \lg n \rfloor} \; \mathbb{Z}$$

$$2^{\lfloor \lg n \rfloor - 1} \le \left\lfloor \frac{n}{2} \right\rfloor < 2^{\lfloor \lg n \rfloor}$$

$$\mathbb{Z} \; \underbrace{\lfloor \lg n \rfloor - 1} \le \lg \left\lfloor \frac{n}{2} \right\rfloor < \lfloor \lg n \rfloor \; \mathbb{Z}$$

$$\lfloor \lg n \rfloor - 1 \le \left\lfloor \lg \left\lfloor \frac{n}{2} \right\rfloor \right\rfloor < \lfloor \lg n \rfloor$$

$$\boxed{\lfloor \lg n \rfloor \le \underbrace{\left\lfloor \lg \left\lfloor \frac{n}{2} \right\rfloor \right\rfloor + 1} < \lfloor \lg n \rfloor + 1}$$

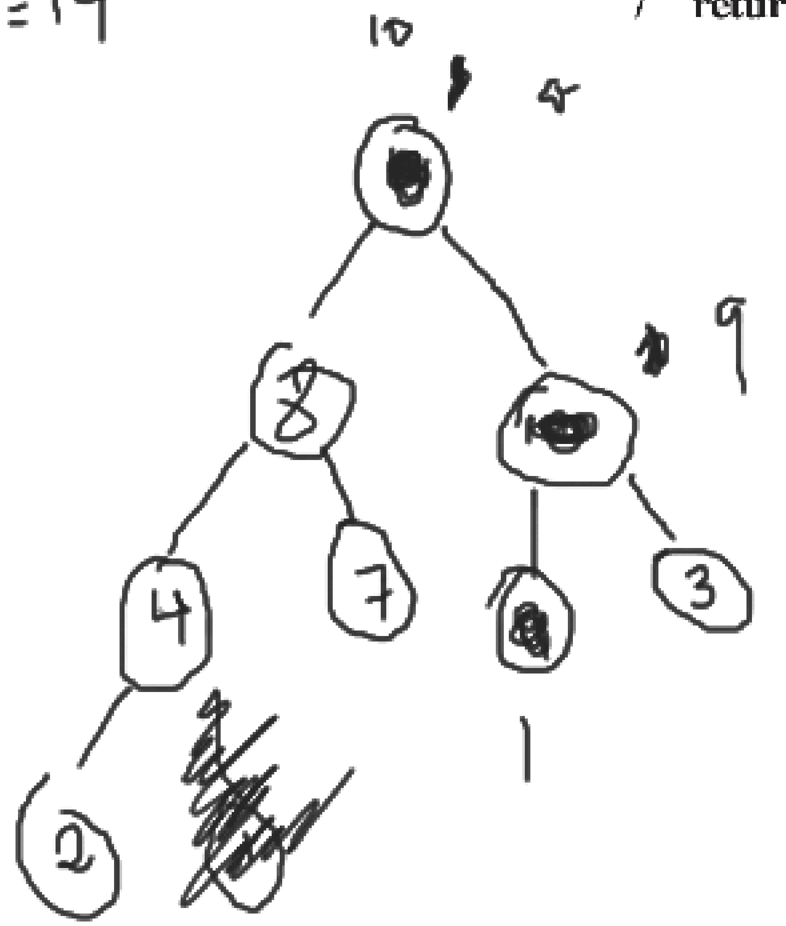
Ejercicio 8. Escriba en seudocódigo cada uno de los siguientes procedimientos, que implementan una fila de prioridades con un min-heap: HEAP-MINIMO, HEAP-EXTRAE-MINIMO, HEAP-DISMINUYE-LLAVE y HEAP-INSERTA.

(A, A.heap.size, 15)

HEAP-MAXIMUM(A)

```
return A[1]
```

may = 14

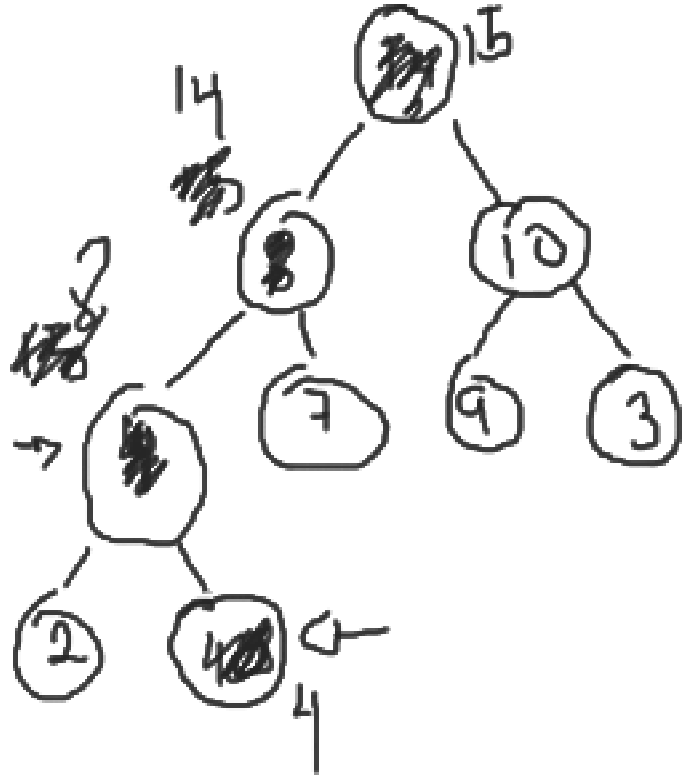


HEAP-EXTRACT-MAX(A)

```
if A.heap-size < 1
    error "heap underflow"
max = A[1]
A[1] = A[A.heap-size]
A.heap-size = A.heap-size - 1
MAX-HEAPIFY(A, 1)
return max
```

HEAP-INCREASE-KEY(A, i, key)

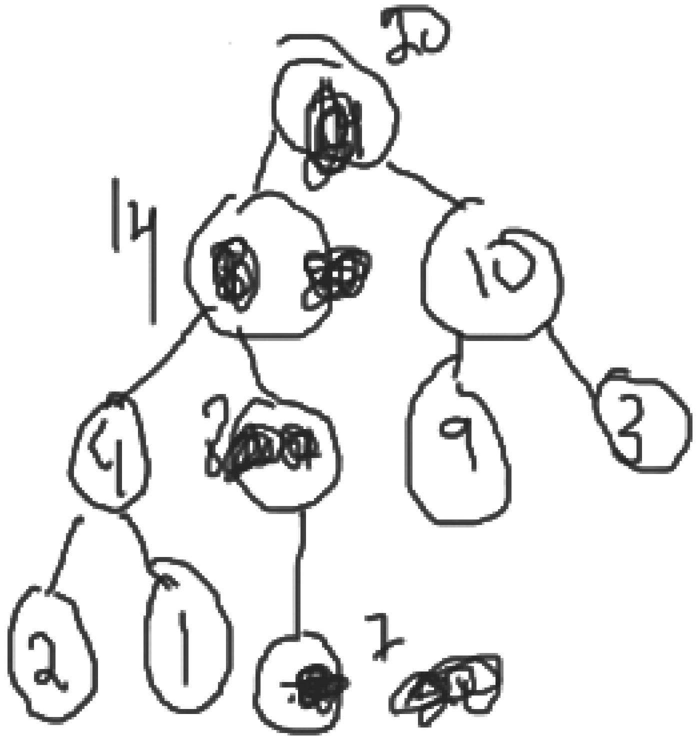
```
if key < A[i]
    error "new key is smaller than current key"
A[i] = key
while i > 1 and A[PARENT(i)] < A[i]      Heapify-Up
    exchange A[i] with A[PARENT(i)]
    i = PARENT(i)
```



MAX-HEAP-INSERT(A, key)

```
A.heap-size = A.heap-size + 1
A[A.heap-size] = -∞
HEAP-INCREASE-KEY(A, A.heap-size, key)
```

Insert(A, 20)



Ejercicio 7. La operación HEAP-DELETE elimina el item en el nodo i del heap A. Dé una implementación de HEAP-DELETE que corre en $O(\lg n)$ para un max-heap con n elementos.

indexado desde 1

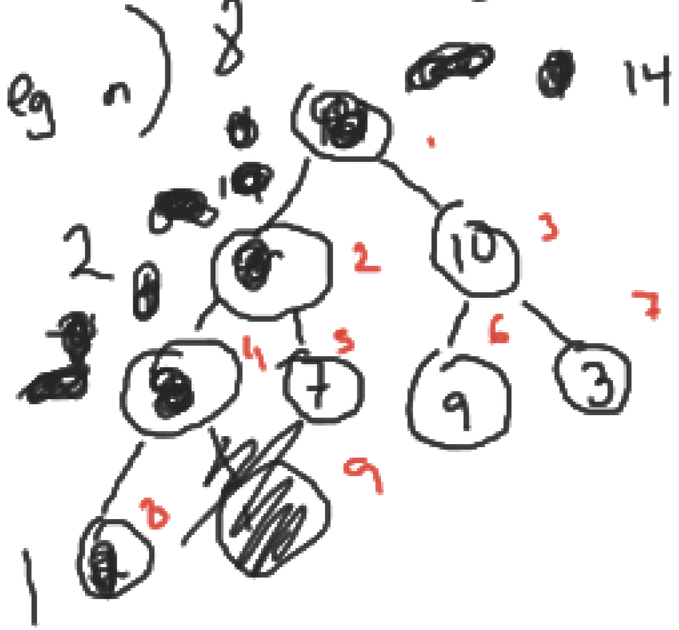
Increase Key a $+\infty$ $O(\lg n)$

Extract Max $O(\lg n)$

Delete (4)

left$(i) = 2i$

right$(i) = 2i + 1$



Heap Delete(A, i)

if i not in bounds:

return

Heap Increase Key (A, i, ∞)

Extract Max (A)