

# Heaps

Andoni Latorre Galarraga

## 1

*Heaps.py*

Siendo  $n = \text{len}(\text{heap})$  antes de insertar y  $n = \text{len}(\text{heap}) - 1$  despues de insertar, lo de dentro del *while* tiene coste fijo y se ejecuta  $\lfloor \log_2 n \rfloor$  veces como máximo.

$$t_{sup}(n) = 1 + \sum_{i=1}^{\lfloor \log_2 n \rfloor} i = \lfloor \log_2 n \rfloor + 1$$

Por lo que  $t(n) \in O(\log n)$  y  $t(n) \in \Omega(1)$  cuando  $n = 1$

## 2

*Heaps.py*

Siendo  $n = \text{len}(\text{heap}) - 1$ , lo de dentro del *while* tiene coste fijo y se ejecuta  $\lfloor \log_2 n \rfloor$  veces como máximo.

$$t_{sup}(n) = 1 + \sum_{i=1}^{\lfloor \log_2 n \rfloor} i = \lfloor \log_2 n \rfloor + 1$$

Por lo que  $t(n) \in O(\log n)$

## 3

*Heaps.py*

La funcion de prioridad tiene coste fijo. Las funciones *enqueue* y *dequeue* tienen coste  $\lfloor \log_2 n \rfloor$  como hemos visto en los ejercicios 1 y 2.

$$t_{sup}(n) = 1 + 2 \cdot \sum_{i=1}^n \lfloor \log_2 n \rfloor = 2n \cdot \lfloor \log_2 n \rfloor + 1$$

Por lo tanto,  $t(n) \in O(n \cdot \log n)$ .