Analisi e Dimostrazione Heap

Heapify viene descritta dall'equazione di ricorrenza

$$T(n) \le T\left(\frac{2n}{3}\right) + \Theta(1)$$

$$f(n) = 1$$

CASO 2 TEOREMA MASTER $\Rightarrow$ $\omega(n) = \log_{\frac{3}{2}} 1 = n^0 = 1$

$$f(n) = \Theta(n^0 \cdot \log^0 n) \longrightarrow T(n) = O(\log n)$$

Build-Heap

Potremmo calcolare velocemente $O(n \log n)$ ma non sarebbe asintoticamente stretto

Il tempo di MAX-HEAPIFY varia con l'altezza dei nodi e le altezze della maggior parte sono piccole

Un heap di n elementi ha altezza $h = \lfloor \log n \rfloor$
e per ogni h ha al massimo: $\left\lceil \frac{n}{2^{h+1}} \right\rceil$ nodi di altezza h

Essendo che HEAPIFY $= O(h)$ abbiamo

$$\sum_{h=0}^{\lfloor \log n \rfloor} \left\lceil \frac{n}{2^{h+1}} \right\rceil \cdot O(h) = O\left(n \sum_{h=0}^{\lfloor \log n \rfloor} \frac{h}{2^h}\right) = O\left(n \sum_{h=0}^{\infty} \frac{h}{2^h}\right) = O(n)$$

In breve ci riconduciamo a una serie che limita superiormente la nostra sommatoria
questo perché:

$$\left\lceil \frac{n}{2^{h+1}} \right\rceil \leq \frac{n}{2^h}$$

AVENDO

$$\sum_{h=0}^{\lfloor \lg n \rfloor} \frac{n}{2^h} hc \implies cn \sum_{h=0}^{\lfloor \lg n \rfloor} \frac{h}{2^h}$$

ricondotti alla dimostrazione