Costruzione di un max-heap

La seguente procedura trasforma l'array A in un max-heap applicando in modo bottom-up MAX-HEAPIFY (a partire da metà array perché la seconda metà contiene soltanto foglie):

BUILD-MAX-HEAP(A)

$A.\mathit{heap\text{-}size} = A.\mathit{length}$
for $i = \lfloor A.\mathit{length}/2 \rfloor$ **downto** 1
    MAX-HEAPIFY(A, i)

Invariante di ciclo: all'inizio di ogni iterazione ogni nodo di indice maggiore di i è la radice di un max-heap.

Complessità

Dalla definizione dell'algoritmo si può ricavare immediatamente un limite superiore di $\mathrm{O}(n \log n)$ ($\mathrm{O}(n)$ chiamate a una funzione di complessità $\mathrm{O}(\log n)$), ma si può trovare un limite più stretto: MAX-HEAPIFY ha costo $\mathrm{O}(h)$ se il nodo i ha altezza h, quindi:

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

Visto che la sommatoria vale 2 per $\lfloor \log n \rfloor \to \infty$ (o comunque converge per il criterio della radice),

$$T(n) = \mathrm{O}(2n) = \mathrm{O}(n).$$