

Esercizi

Esercizio 1

Il minimo in uno heap di massimo risiede in una foglia. Perché? Sfruttando questa proprietà, progettare un algoritmo che restituisca l'elemento minimo di uno heap di massimo, e calcolarne la complessità.

Esercizio 2

Progettare un algoritmo che verifichi se gli elementi memorizzati in un array soddisfano la proprietà di heap di minimo.

Esercizio 3

Sia H uno heap di massimo di dimensione n , e Δ un numero reale positivo o negativo. Progettare un algoritmo $\text{Cambia}(H, i, \Delta)$ che modifica la chiave contenuta in $H[i]$ come $H[i] = H[i] + \Delta$.

Esercizio 4

Si consideri un max-heap **ternario** di n nodi e altezza h definito con le seguenti proprietà:

- (1) per $0 \leq i < h$ ci sono 3^i nodi al livello i ;
 - (2) tutte le foglie nell'ultimo livello sono ammassate a sinistra;
 - (3) la chiave memorizzata in ogni nodo è maggiore di (o uguale a) quella dei suoi figli.
- Descrivere un'implementazione efficiente dell'heap ternario tramite un array A , indicando le formule per calcolare la posizione del padre, del primo, del secondo e del terzo figlio di un nodo i .
 - Descrivere e analizzare le implementazioni delle operazioni $\text{Heap-Extract-Max}(A)$ e $\text{Max-Heap-Insert}(A, \text{key})$ per il max-heap ternario.

Esercizio 5

Progettare un algoritmo che, ricevuto in input un intero k e un array A di n elementi distinti, restituisce il k -esimo elemento più piccolo in A .

- Progettare una soluzione di costo in tempo $O(n \log n)$.
- Progettare una soluzione di costo in tempo $O(n + k \log n)$.
- Progettare una soluzione di costo in tempo $O(n \log k)$ che usi uno heap su k elementi.