

Heap

Un heap è una struttura dati basata sugli alberi che soddisfa la "proprietà di heap": se A è un genitore di B, allora la chiave di A è ordinata rispetto alla chiave di B conformemente alla relazione d'ordine applicata all'intero heap.

Max Heap

Il nodo padre deve essere maggiore del nodo figlio. *L'heap non garantisce la relazione tra i 2 fratelli.*

Il valore del nodo N è sicuramente maggiore di N-→R

$$\text{Val}[N] > \text{Val}[N-R]$$

Il valore del nodo N è sicuramente maggiore di N-→L

$$\text{Val}[N] > \text{Val}[N-L]$$

La radice dell'albero conterrà il valore massimo. Guardando la radice saprò di già il valore massimo in $O(1)$.

Struttura che risponde alla domanda:

(Max) Heap ->

- Chi è il massimo $O(1)$
- Inserisco nodo $O(\log n)$
- Elimino nodo $O(\log n)$

Algoritmi

Ho un Maxheap e ho la richiesta di inserimento.

Inserisco(V)

Posso metterlo come foglia dell'ultimo elemento. È un heap?

- V: figlio
- P: padre

P--->V

Se $P > V$ OK!

SE $P < V$ Li scambio e non devo andare a controllare l'altro sottoalbero per la proprietà transitiva.

Posso iterare questo algoritmo fino a quando non trovo $P > V$ (anche fino alla radice).

Quindi l'inserimento ha come valore medio $O(\text{altezza dell'heap}) = O(\log n)$.

Elimino il massimo (la radice) a basso costo.

Elimino la radice e prendo una foglia per sostituirla. Non è un heap, come posso risolvere?

V: foglia sostituita alla radice

Deve essere:

- $V < L$!!
- $V < R$!!

Max(L,R) ?

Discesa verso la fine dell'heap finchè non si viene a ricreare la struttura.

Costo massimo per eliminare la radice è di $O(\log n)$

[Riferimento a Priority Queue e Binary Heap](#), un po' di pseudo codice.