

MaxHeapInsert (H, k) {

$O(\log m)$

if (H.heapsize < H.length) {

④ (1)

H.heapsize ← H.heapsize + 1

H[H.heapsize] ← k

i ← H.heapsize

while (H[i] > H[Parent(i)]) {

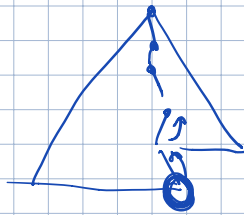
swap (H, i, Parent(i))

i ← Parent(i)

$O(\log m)$

④ (1)

Verificare Ric. ?
Completeness ?

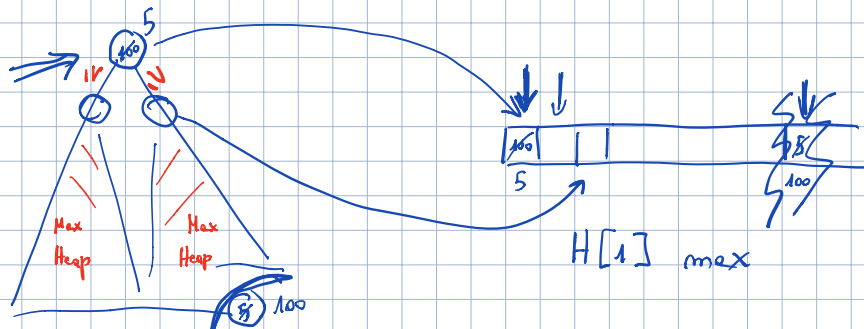


$m = H.heapsize$

~ . ~ . ~ .



Eliminazione della chiave massima



MaxHeapExtract (H) {

$O(\log n)$

⑦(1) {
 if (H.heapsize \geq 1) {
 swap (H, 1, H.heapsize)
 H.heapsize \leftarrow H.heapsize - 1
 \Rightarrow Heapify (H, 1)
 return H[H.heapsize + 1]
 } else {
 return "Heap vuoto"
 }
 }

Precondizione

Heapify (H, i) {

% H[2i] e H[2i+1] radici di due max heap

$O(\log n)$

l \leftarrow Left(i)
 r \leftarrow Right(i)

if (l \leq H.heapsize \wedge H[l] $>$ H[i]) {
 m \leftarrow l

} else {
 m \leftarrow i

④(1)

}

if (r \leq H.heapsize \wedge H[r] $>$ H[m]) {
 m \leftarrow r

}

```

    ↪ if (m ≠ i) {
        swap (H, i, m)
        ↪ Heapify (H, m)
    }

```

∴ $H[i]$ è radice di Max Heap

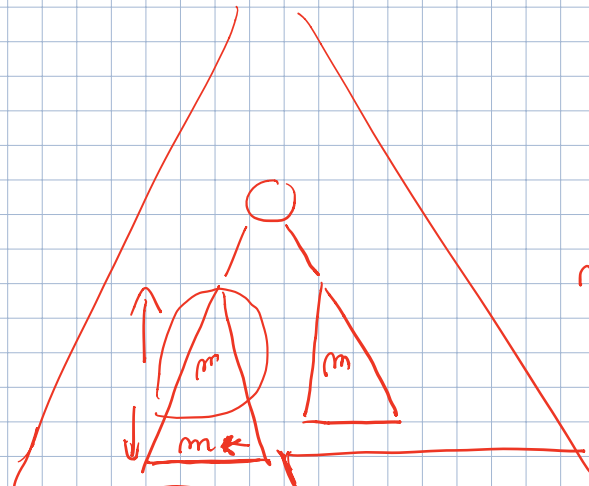
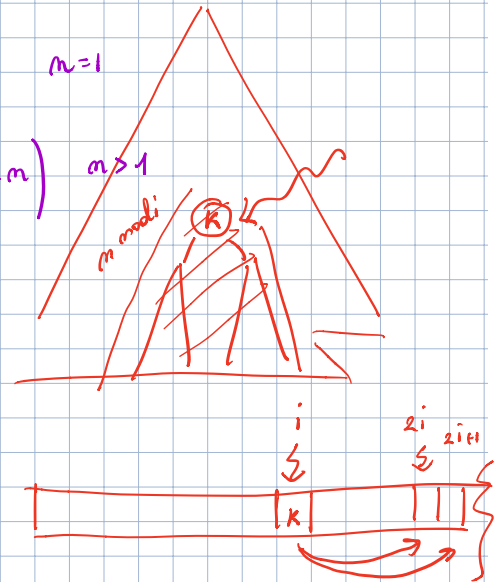
1) Complessità *

2) Correttezza ~~✗~~

* $T(n) \leq \begin{cases} \textcircled{u}(1) & n=1 \\ \textcircled{u}(1) + \cancel{T(\frac{n}{2})} + T(\frac{2}{3}n) & n > 1 \end{cases}$

↙
Heapify
↘
ie n.ro
di nodi
della Heap
 $H[i]$

↓
non
è così!



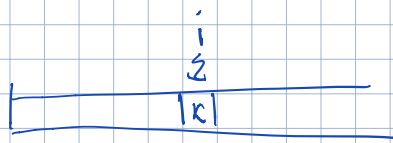
$$n \approx 3m$$

$2m$ el.
ch. ric.

$$\frac{2}{3} n$$

Esercizio Risolvere l'equazione di Heapify

$$T(n) = O(\log n)$$



h altezza dell'albero radicato
in $H[i]$

$$T(h) \leq \begin{cases} O(1) & h=0 \\ O(1) + T(h-1) & h>0 \end{cases}$$

$$T(h) = O(h) \rightsquigarrow T(n) = O(\log n)$$

$h = O(\log n)$

~~✗~~ Correttezza Heapify

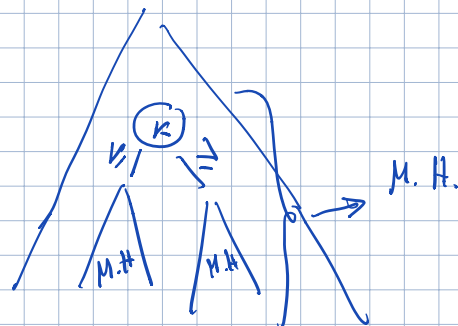
Se $H[2i]$ e $H[2i+1]$ sono radici di max-heap
allora $\text{Heapify}(H, i)$ termina con $H[i]$ radice di
max heap.

Dim

Per induzione sul n.ro di ric.

BASE 0 ch. ric.

Heapify (H, i) non active ch. ric.

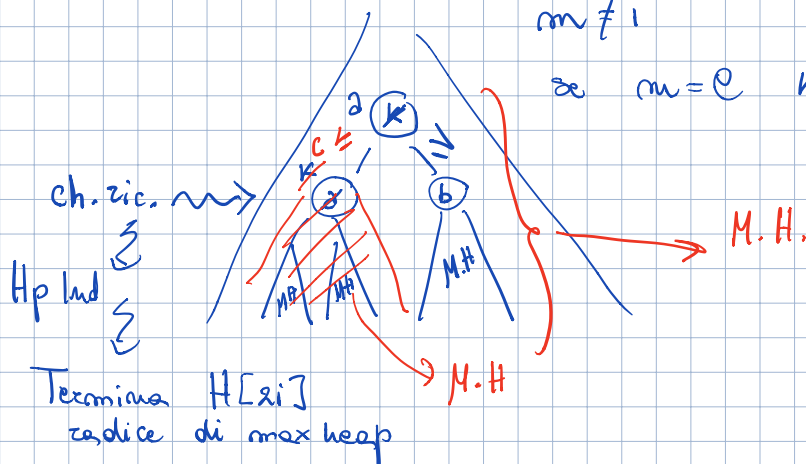
$$m = i$$


PASSO

Hphnd) Se vengono effettuate al più 2 ch. ric.
Hearpfy è corretta

3) Se reagiamo effettate $\tau+1$ ch. ric...

$m \neq i$

$$\text{Se } m=0 \quad k \leq 2 \text{ e } b \leq 2$$


Esercizio

Scirese Max-Heap-modify ($H, i, 1$)

- Modifica $H[i] \rightsquigarrow H[i] + 1$

- Ripristina la Max heap

Nota: si può essere sia > 0 che < 0

$$O(\log n)$$

~ . ~ . ~ .

HeapSort

Idea

