

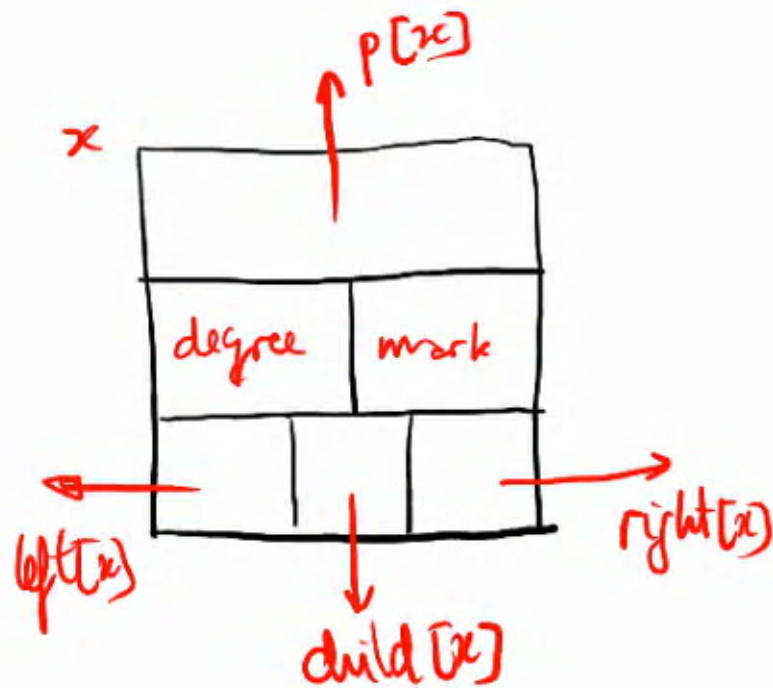
Heap di Fibonacci

- UNO HEAP DI FIBONACCI E' UNA COLLEZIONE DI ALBERI CON LA PROPRIETA' HEAP
- GLI ALBERI DI UNO HEAP DI FIBONACCI NON DEBBONO ESSERE NECESSARIAMENTE ALBERI BINOMIALI

	Binary heap	Binomial heap	Fibonacci heap
Procedure	(worst-case)	(worst-case)	(amortized)

MAKE-HEAP	$\Theta(1)$	$\Theta(1)$	$\Theta(1)$
INSERT	$\Theta(\lg n)$	$O(\lg n)$	$\Theta(1)$ (*)
MINIMUM	$\Theta(1)$	$O(\lg n)$	$\Theta(1)$
EXTRACT-MIN	$\Theta(\lg n)$	$\Theta(\lg n)$	$O(\lg n)$ (*)
UNION	$\Theta(n)$	$O(\lg n)$	$\Theta(1)$
DECREASE-KEY	$\Theta(\lg n)$	$\Theta(\lg n)$	$\Theta(1)$ (*)
DELETE	$\Theta(\lg n)$	$\Theta(\lg n)$	$O(\lg n)$ (*)

(*) Costi ammortizzati

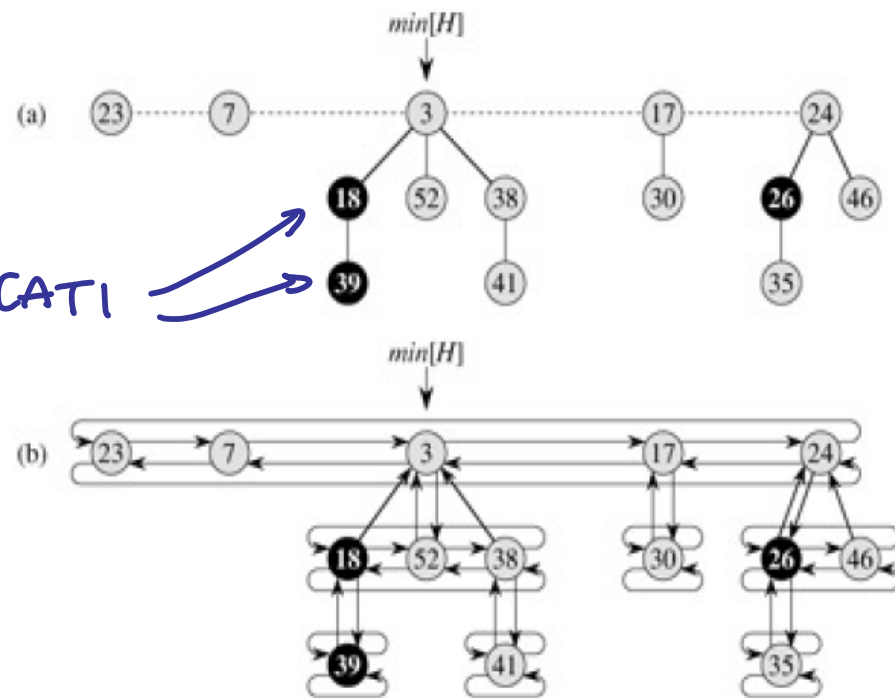


RAPPRESENTAZIONE
DI UN NODO

- $P(x)$ - padre
- $left(x)$ - fratello sinistro
- $right(x)$ - fratello destro
- $child(x)$ - (un) figlio
- $degree(x)$ - numero di figli
- $mark(x)$ - indica se il nodo x ha perduto un figlio dall'ultima volta in cui x è diventato figlio di un altro nodo

ESEMPIO

NODI MARCATI



- IL PUNTATORE $\text{min}[H]$ INDICA LA RADICE CONTENENTE LA CHIAVE MINIMA E DA' ACCESSO ALLA STRUTTURA
- VIENE ANCHE MANTENUTO IL CAMPO $n[H]$ CHE CONTIENE IL NUMERO DI NODI IN H

FUNZIONE POTENZIALE

$$\Phi(H) = t(H) + 2m(H)$$

DOVE

- $t(H) = \#$ ALBERI NELLA LISTA DELLE RADICI DI H

- $m(H) = \#$ NODI MARCATI IN H

- SIA $\{H_i\}_{i \in I}$ UNA COLLEZIONE FINITA DI HEAP. PONIAMO:

$$\Phi(\{H_i\}_{i \in I}) = \sum_{i \in I} \Phi(H_i)$$

MASSIMO GRADO DI UN NODO: $D(m)$

- L'ANALISI VERRA' EFFETTUATA IN FUNZIONE DI UN UPPER BOUND $D(m)$ SUL MASSIMO GRADO DI UN NODO QUALUNQUE IN UNO HEAP CON n NODI
- DIMOSTREMO CHE SI HA $D(m) = O(\log m)$

- SE VENGONO ESEGUITE SOLO OPERAZIONI DEL TIPO:

- MAKE-HEAP
- INSERT
- MINIMUM
- EXTRACT-MIN
- UNION

CLASCUNO HEAP DI FIBONACCI E' RAPPRESENTABILE
COME COLLEZIONE DI ALBERI BINOMIALI NON
ORDINATI

ALBERI BINOMIALI NON ORDINATI

DEFINIZIONE

PER OGNI $k \in \mathbb{N}$ ESISTE UN ALBERO BINOMIALE NON ORDINATO U_k DI GRADO k , DEFINITO IN BASE ALLA SEGUENTE RICORSIONE:

- U_0 E' L'ALBERO FORMATO DA UN SOLO NODO
- DATO U_{k-1} DEFINIAMO U_k COMBINANDO DUE COPIE DI U_{k-1} NELLA SEGUENTE MANIERA:



LEMMA (PROPRIETA' DEGLI ALBERI BINOMIALI NON ORDINATI)

PER OGNI $k=0,1,2,\dots$ VALGONO LE SEGUENTI

PROPRIETA':

1. U_k HA 2^k NODI

2. L'ALTEZZA DI U_k E' k

3. U_k HA $\binom{k}{i}$ NODI A PROFONDITA' i ($i=0,1,\dots,k$)

4. LA RADICE DI U_k HA GRADO k ED OGNI ALTRO NODO IN U_k HA GRADO $< k$,

INOLTRE I FIGLI DELLA RADICE DI U_k SONO RADICI DI $U_0, U_1, U_2, \dots, U_{k-1}$ (IN QUALCHE ORDINE). ■

- SI OSSERVI CHE DAL LEMMA PRECEDENTE SEGUE IMMEDIATAMENTE CHE $D(n) = O(\lg n)$ ALMENO QUANDO LO HEAP È FORMATO SOLTANTO DA ALBERI BINOMIALI NON ORDINATI
- LA STRATEGIA DI MANTENIMENTO DEGLI HEAP DI FIBONACCI PREVEDE DI RITARDARE IL LAVORO IL PIÙ POSSIBILE

MAKE-FIBONACCI-HEAP(c)

$H := \text{allocate_node}();$

$m[H] := 0;$

$\text{min}[H] := \text{NIL};$

return H

COMPLESSITA':

$$\left. \begin{array}{l} \Delta t = 0 \\ \Delta m = 0 \end{array} \right\} \Rightarrow \Delta \phi = 0$$

PERTANTO: $\hat{c} = c + \Delta \phi = c = O(1)$

FIB-HEAP-INSERT(H, x)

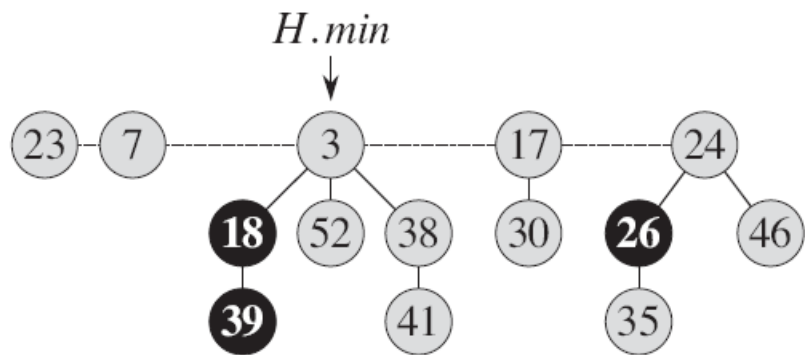
```
1  degree[x] ← 0
2  p[x] ← NIL
3  child[x] ← NIL
4  left[x] ← x
5  right[x] ← x
6  mark[x] ← FALSE
7  concatenate the root list containing x with root list H
8  if min[H] = NIL or key[x] < key[min[H]]
9     then min[H] ← x
10 n[H] ← n[H] + 1
```

COMPLESSITA'

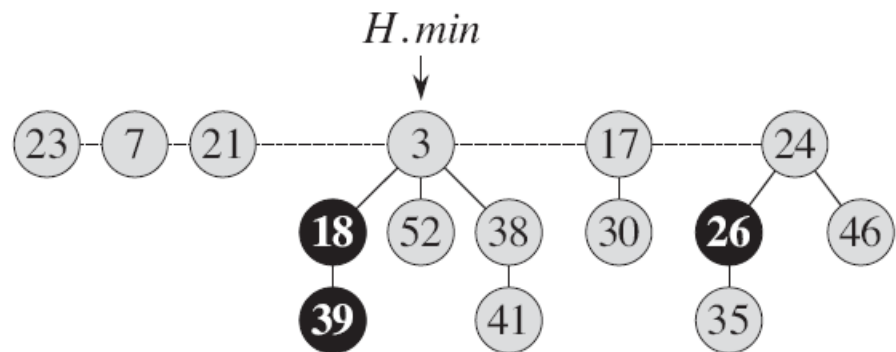
$$\Delta t = 1, \Delta m = 0 \Rightarrow \Delta \phi = 1$$

$$\hat{c} = c + \Delta \phi = c + 1 = O(1)$$

ESEMPIO



FIB-HEAP-INSERT(H, x)
(con $key[x] = 21$)



MINIMUM(H)

return (min[H])

COMPLEXITY

$$\Delta\phi = 0$$

$$\hat{c} = c + \Delta\phi = c = \Theta(1)$$

UNION

FIB-HEAP-UNION(H_1, H_2)

1 $H \leftarrow \text{MAKE-FIB-HEAP}()$

2 $\text{min}[H] \leftarrow \text{min}[H_1]$

3 concatenate the root list of H_2 with the root list of H

4 **if** ($\text{min}[H_1] = \text{NIL}$) or ($\text{min}[H_2] \neq \text{NIL}$ and $\text{min}[H_2] < \text{min}[H_1]$)

5 **then** $\text{min}[H] \leftarrow \text{min}[H_2]$

6 $n[H] \leftarrow n[H_1] + n[H_2]$

7 free the objects H_1 and H_2

8 **return** H

COMPLESSITA':

$$\left. \begin{array}{l} \Delta t = 0 \\ \Delta m = 0 \end{array} \right\} \Rightarrow \Delta \phi = 0$$

PERTANTO: $\hat{c} = c + \Delta \phi = c = \textcircled{n}(1)$

EXTRACT-MIN

FIB-HEAP-EXTRACT-MIN(H)

```
1   $z \leftarrow \text{min}[H]$ 
2  if  $z \neq \text{NIL}$ 
3      then for each child  $x$  of  $z$ 
4          do add  $x$  to the root list of  $H$ 
5               $p[x] \leftarrow \text{NIL}$ 
6          remove  $z$  from the root list of  $H$ 
7      if  $z = \text{right}[z]$ 
8          then  $\text{min}[H] \leftarrow \text{NIL}$ 
9          else  $\text{min}[H] \leftarrow \text{right}[z]$ 
10         CONSOLIDATE( $H$ )
11          $n[H] \leftarrow n[H] - 1$ 
12 return  $z$ 
```

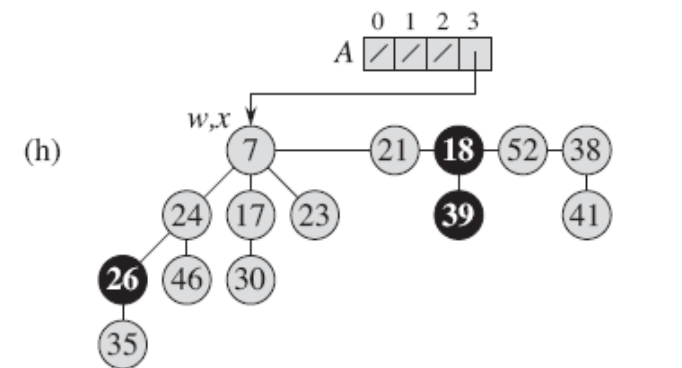
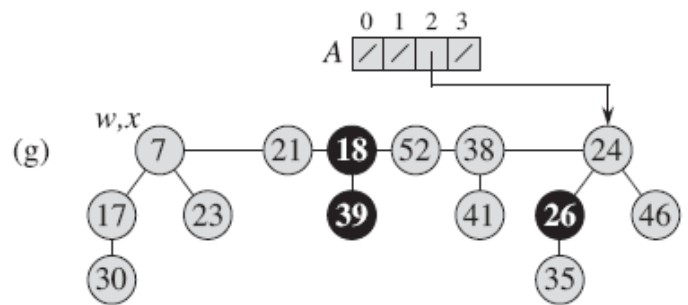
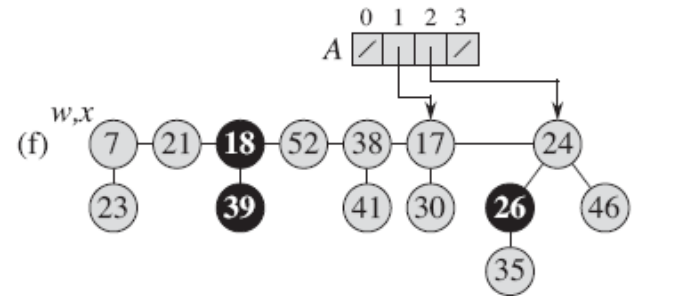
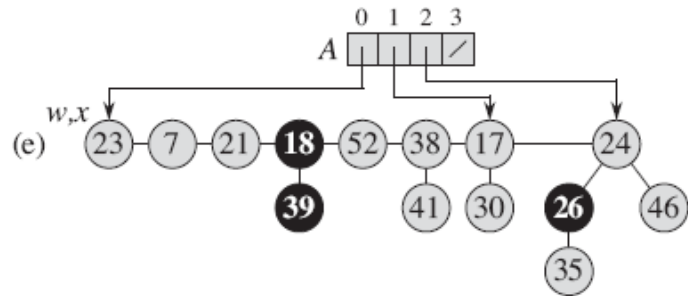
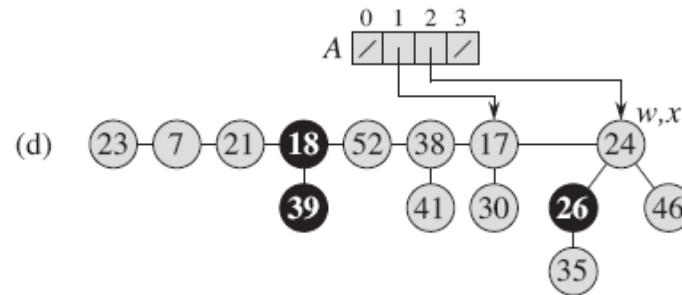
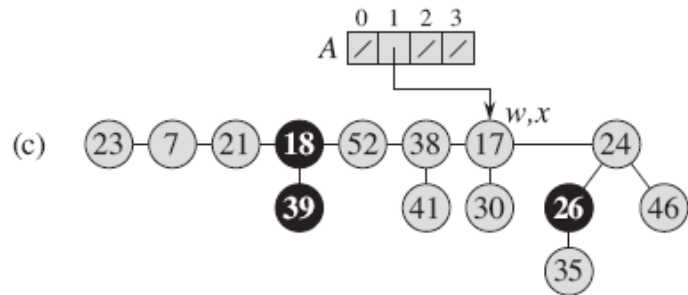
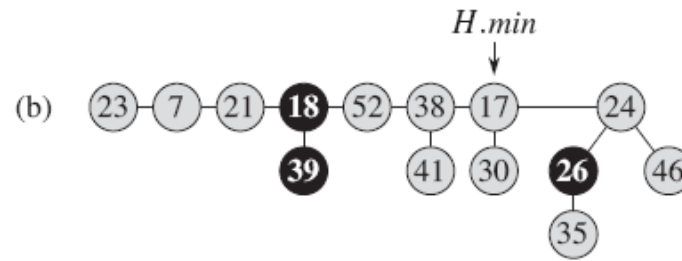
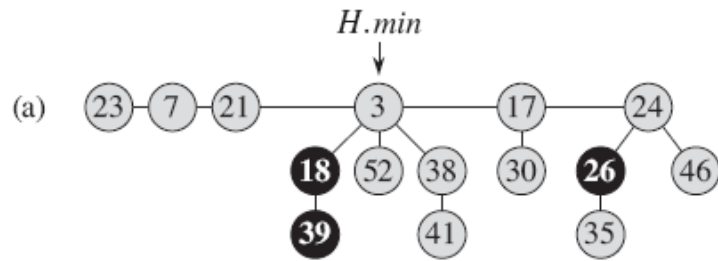
CONSOLIDATE(H)

```
1 for  $i \leftarrow 0$  to  $D(n[H])$ 
2     do  $A[i] \leftarrow \text{NIL}$ 
3 for each node  $w$  in the root list of  $H$ 
4     do  $x \leftarrow w$ 
5          $d \leftarrow \text{degree}[x]$ 
6         while  $A[d] \neq \text{NIL}$ 
7             do  $y \leftarrow A[d]$             $\triangleright$  Another node with the same degree as  $x$ .
8                 if  $\text{key}[x] > \text{key}[y]$ 
9                     then exchange  $x \leftrightarrow y$ 
10                    FIB-HEAP-LINK( $H, y, x$ )
11                     $A[d] \leftarrow \text{NIL}$ 
12                     $d \leftarrow d + 1$ 
13                 $A[d] \leftarrow x$ 
14  $\text{min}[H] \leftarrow \text{NIL}$ 
15 for  $i \leftarrow 0$  to  $D(n[H])$ 
16     do if  $A[i] \neq \text{NIL}$ 
17         then add  $A[i]$  to the root list of  $H$ 
18             if  $\text{min}[H] = \text{NIL}$  or  $\text{key}[A[i]] < \text{key}[\text{min}[H]]$ 
19                 then  $\text{min}[H] \leftarrow A[i]$ 
```

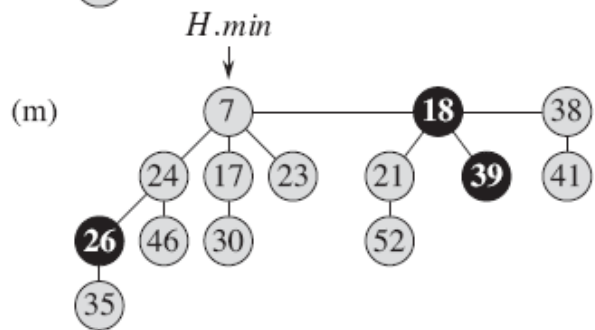
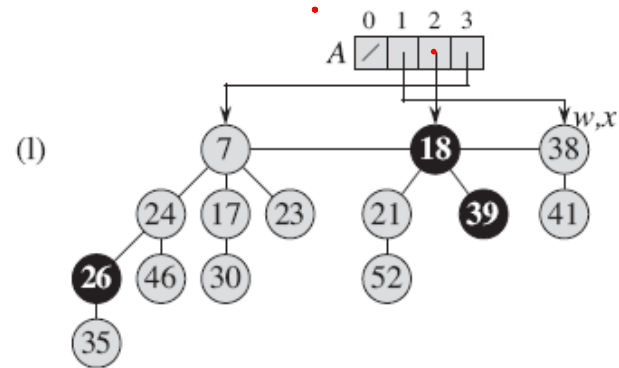
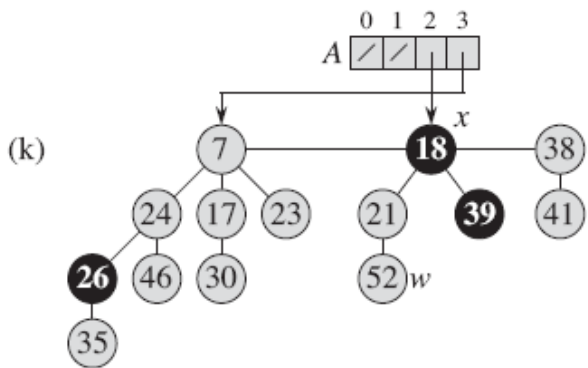
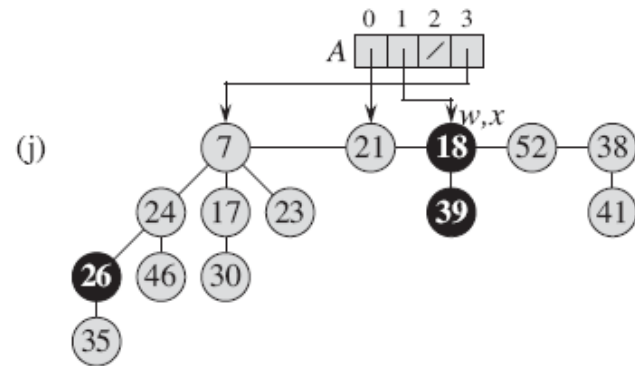
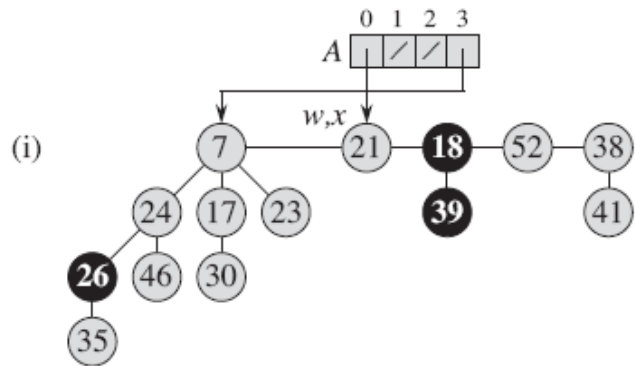
FIB-HEAP-LINK(H, y, x)

```
1 remove  $y$  from the root list of  $H$ 
2 make  $y$  a child of  $x$ , incrementing  $\text{degree}[x]$ 
3  $\text{mark}[y] \leftarrow \text{FALSE}$ 
```

ESEMPIO



ESEMPLO (CNTD)



COMPLESSITA' DI FIB-HEAP-EXTRACT-MIN

$$\begin{aligned} \text{COSTO REALE} &= \mathcal{O}(D(m)) + \mathcal{O}(D(m) + t(H)) \\ &\quad \uparrow \qquad \qquad \qquad \uparrow \\ &\quad \text{PROCESSAMENTO} \qquad \text{PROCESSAMENTO} \\ &\quad \text{FIGLI DI } \min(H) \qquad \text{LISTA DI RADICI} \\ &\qquad \qquad \qquad \qquad \text{IN CONSOLIDATE} \\ &= \mathcal{O}(D(m) + t(H)) = \mathcal{O}(m) \end{aligned}$$

$$\Delta t \leq (D(m) + 1) - t(H), \quad \Delta m \leq 0$$

$$\Delta \phi = \Delta t + 2\Delta m \leq D(m) + 1 - t(H)$$

$$\hat{c} = c + \Delta \phi = \mathcal{O}(D(m) + t(H)) - t(H) = \mathcal{O}(D(m))$$

(PUR DI SCALARE OPPORTUNAMENTE IL POTENZIALE)

DECREASE-KEY

```
FIB-HEAP-DECREASE-KEY( $H, x, k$ )
1  if  $k > \text{key}[x]$ 
2      then error "new key is greater than current key"
3   $\text{key}[x] \leftarrow k$ 
4   $y \leftarrow p[x]$ 
5  if  $y \neq \text{NIL}$  and  $\text{key}[x] < \text{key}[y]$ 
6      then CUT( $H, x, y$ )
7          CASCADING-CUT( $H, y$ )
8  if  $\text{key}[x] < \text{key}[\text{min}[H]]$ 
9      then  $\text{min}[H] \leftarrow x$ 
```

CUT(H, x, y)

1 remove x from the child list of y , decrementing $degree[y]$

2 add x to the root list of H

3 $p[x] \leftarrow \text{NIL}$

4 $mark[x] \leftarrow \text{FALSE}$

CASCADING-CUT(H, y)

1 $z \leftarrow p[y]$

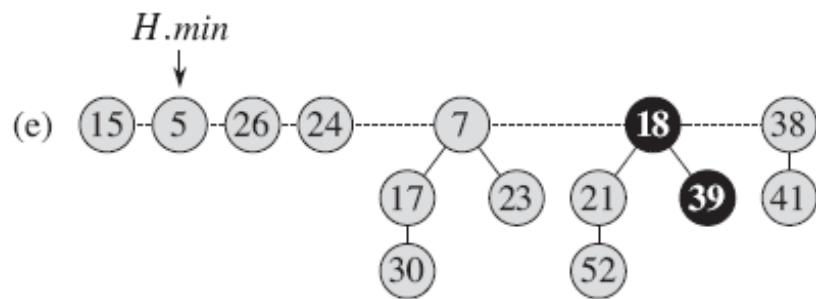
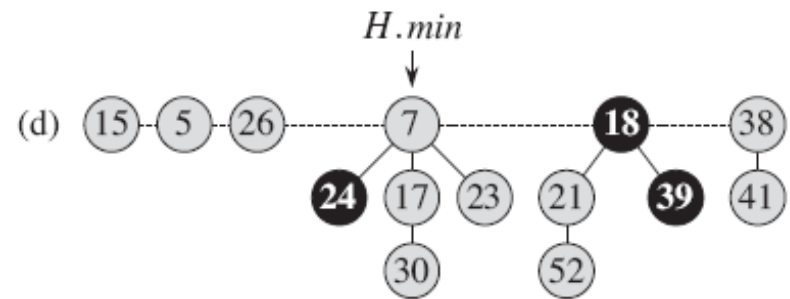
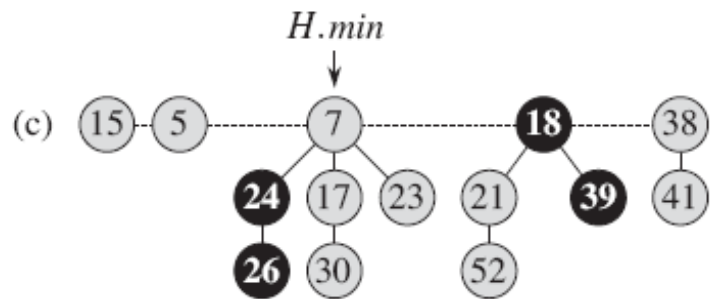
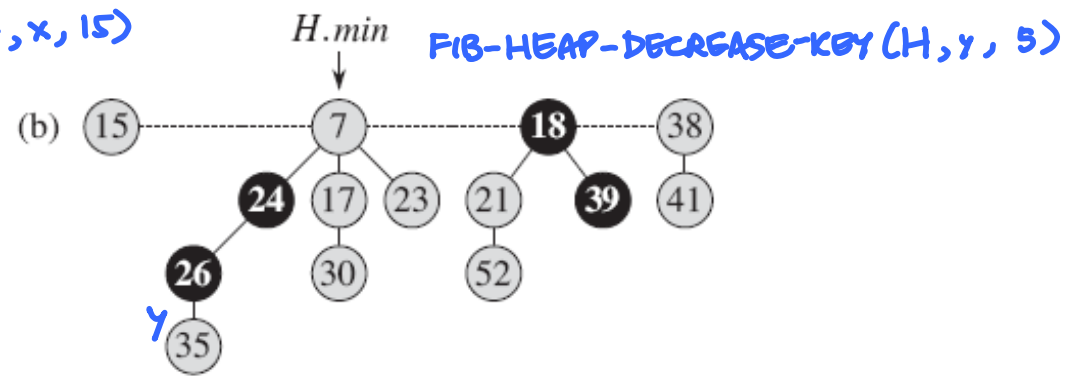
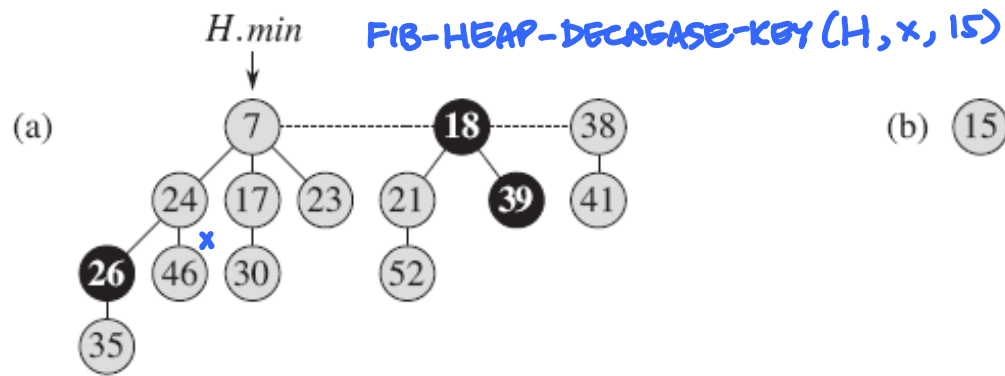
2 **if** $z \neq \text{NIL}$

3 **then if** $mark[y] = \text{FALSE}$

4 **then** $mark[y] \leftarrow \text{TRUE}$

5 **else** CUT(H, y, z)

6 CASCADING-CUT(H, z)



COMPLESSITA' DI FIB-HEAP-DECREASE-KEY
 SUPPONIAMO CHE LA PROCEDURA CASCADING-CUT VENGA
 CHIAMATA d VOLTE

$$\text{COSTO REALE} = \mathcal{O}(d)$$

$$\phi(H) = t(H) + 2m(H)$$

$$\begin{aligned}
 \phi(H') &\leq (t(H) + d) + 2(m(H) - (d-1) + 1) \\
 &= t(H) + d + 2m(H) - 2d + 2 + 2 \\
 &= t(H) + 2m(H) - d + 4
 \end{aligned}$$

$$\hat{c} = \mathcal{O}(d) + \phi(H') - \phi(H) = \mathcal{O}(d) - d + 4 = \mathcal{O}(1)$$

(PUR DI SCALARE OPPORTUNAMENTE IL POTENZIALE)

DELETE

FIB-HEAP-DELETE (H, x)

1 FIB-HEAP-DECREASE-KEY ($H, x, -\infty$)

2 FIB-HEAP-EXTRACT-MIN (H)

COMPLESSITA' AMMORTIZZATA = $O(D(m))$

ALCUNE PROPRIETÀ SUI NUMERI DI FIBONACCI

$$F_0 = 0, F_1 = 1$$

$$F_k = F_{k-1} + F_{k-2}, \text{ PER } k \geq 2$$

LEMMA 1 $F_{k+2} = 1 + \sum_{i=0}^k F_i, \text{ PER } k \geq 0$

DIM

CASO BASE $k=0$

$$F_{k+2} = F_2 = 0 + 1 = 1, \quad 1 + \sum_{i=0}^k F_i = 1 + \sum_{i=0}^0 F_i = 1 + 0 = 1$$

PASSO INDUTTIVO

$$F_{(k+1)+2} = F_{k+3} = F_{k+2} + F_{k+1} = 1 + \sum_{i=0}^k F_i + F_{k+1} = 1 + \sum_{i=0}^{k+1} F_i.$$

■

LEMMA 2 $F_{k+2} \geq \phi^k$, PER $k \geq 0$

(DOVE $\phi = \frac{1+\sqrt{5}}{2}$ È IL RAPPORTO AUREO)

DIM.

CASO BASE

$$k=0 \rightarrow F_2=1, \phi^0=1 \quad \checkmark$$

$$k=1 \rightarrow F_3=2, \phi^1=\phi < 2$$

PASSO INDUTTIVO ($k \geq 1$)

$$F_{k+3} = F_{k+1} + F_{k+2} \geq \phi^{k-1} + \phi^k = \phi^{k-1}(1+\phi) = \phi^{k-1} \cdot \phi^2 = \phi^{k+1}$$

IN QUANTO $1+\phi = \phi^2$ ■

STIMA DI $D(n)$

LEMMA 3 SIA x UN NODO IN UNO HEAP DI FIBONACCI
E SIA $\text{degree}[x] = k$,
SIANO y_1, y_2, \dots, y_k I FIGLI DI x NELL'ORDINE IN
CUI SONO STATI INNESTATI IN x .
ALLORA $\text{degree}[y_i] \geq i-2$, PER $i=3, 4, \dots, k$.

DM.

- SIA $i \geq 3$; NOTIAMO CHE QUANDO y_i E' INNESTATO IN x (AL TEMPO T), IL NODO x HA GIA' y_1, \dots, y_{i-1} TRA I SUOI FIGLI, PER CUI $\text{degree}_{T-1}[y_i] = \text{degree}[x] \geq i-1$.
DALL'ISTANTE T , y_i PUO' AVERE PERDUTO AL PIU' UN FIGLIO
E QUINDI $\text{degree}[y_i] \geq i-2$. ■

LEMMA SIA x UN NODO IN UNO HEAP DI FIBONACCI

E SIA $\text{degree}[x] = k$,

ALLORA $\text{size}(x) \geq F_{k+2}$, DOVE $\text{size}(x)$ È IL NUMERO
DI NODI NEL SOTTOALBERO RADICATO IN x ,

DIM

PONIAMO
$$s_j = \min_{\substack{\text{degree}[z]=j \\ z \in H \\ H \in \mathcal{H}}} \text{size}[z]$$

CON \mathcal{H} FAMIGLIA DI TUTTI GLI HEAP DI FIBONACCI

SI HA: $s_0 = 1$, $s_1 \geq 2$ E $s_{j+1} > s_j$ ($j \in \mathbb{N}$)

DIMOSTRIAMO CHE $s_j \geq F_{j+2}$, PER $j=0, 1, 2, \dots$

CASO $j=0$: $s_j = 1$, $F_{j+2} = 1$ ✓

CASO $j=1$: $s_j \geq 2$, $F_{j+2} = F_3 = 2$ ✓

CASO $j \geq 2$: SIA z UN NODO IN UNO HEAP DI FIBONACCI TALE $\text{degree}[z] = j$, $\text{size}[z] = s_j$ E SIANO y_1, y_2, \dots, y_j I FIGLI DI z NELL'ORDINE IN CUI SONO STATI INNESTATI IN z .

$$\begin{aligned} s_j = \text{size}[z] &= \text{size}[y_1] + \text{size}[y_2] + \text{size}[y_3] + \dots + \text{size}[y_j] + 1 \\ &\geq 1 + s_{2-2} + s_{3-2} + \dots + s_{j-2} + 1 \\ &= 2 + \sum_{i=2}^j s_{i-2} \geq 2 + \sum_{i=2}^j F_i = 1 + \sum_{i=0}^j F_i = F_{j+2}. \end{aligned}$$

POICHE' $\text{degree}(x) = k$, SI HA

$$\text{size}(x) \geq S_k \geq F_{k+2} \quad \blacksquare$$

COROLLARIO $\text{size}(x) \geq \phi^{\text{degree}(x)}$

COROLLARIO $D(n) \leq \lfloor \log_{\phi} n \rfloor$, DA CUI $D(n) = O(\log n)$.

DM SIA x UN NODO IN UNO HEAP DI FIBONACCI CON

n NODI,

SI HA: $n \geq \text{size}(x) \geq \phi^{\text{degree}(x)}$

PERTANTO $\text{degree}(x) \leq \lfloor \log_{\phi} n \rfloor$ E QUINDI

$$D(n) = \max_{\substack{x \in H \\ H \in \mathcal{H}(n)}} \text{degree}(x) \leq \lfloor \log_{\phi} n \rfloor \quad \blacksquare$$