ANALISI AMMORTIZZATA

- PER ANALIZZARE SEQUENZE DI n OPERAZIONI
- SI DETERMINA UN TEMPO COMPLESSIVO $T(n)$ CHE VIENE RIPARTITO IN QUALCHE MODO TRA LE n OPERAZIONI
- ES. $T(n)/n$ - COSTO AMMORTIZZATO PER OPERAZIONE
- LA STIMA OTTENUTA NON E' PROBABILISTICA, MA SI TRATTA DI UNA MEDIA NEL CASO PEGGIORE

TRE METODI:

- METODO DELL'AGGREGAZIONE
- METODO DEGLI ACCANTONAMENTI
- METODO DEL POTENZIALE

DUE ESEMPI:

- STACK CON MULTIPOP
- CONTATORE BINARIO CON INCREMENT

TABELLE DINAMICHE

STACK CON MULTIPOP

POP(S) $\longrightarrow$ COSTO $O(1)$

PUSH(S, x) $\longrightarrow$ COSTO $O(1)$

STACK_EMPTY(S) $\longrightarrow$ COSTO $O(1)$

```
MULTIPOP (S, k)
  while not STACK_EMPTY(S) and k ≠ 0 do
      POP(S)
      k := k - 1
```

MULTIPOP (S, k) $\longrightarrow$ COSTO $O(\min(|S|, k))$

ANALISI DI UNA SEQUENZA DI n OPERAZIONI SU UNO STACK INIZIALMENTE VUOTO

- $|S| = O(n)$
- COSTO DI UNA SINGOLA OPERAZIONE $= O(n)$
- COSTO DI n OPERAZIONI $= nO(n) = O(n^2)$

CONTATORE BINARIO CON INCREMENT

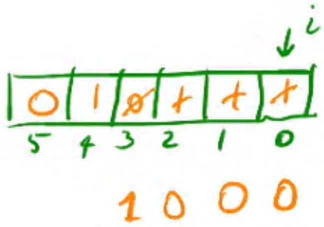
- SIA $A[0..k-1]$ UN ARRAY DI k BIT

$$VALUE[A] = \sum_{i=0}^{k-1} A[i] \cdot 2^i$$

$$VALUE[INCREMENT(A)] \equiv VALUE[A] + 1 \pmod{2^k}$$

$$1101011 \longmapsto 1101100$$

```
INCREMENT(A)
  i := 0
  while i<k and A[i]=1 do
      A[i] := 0
      i := i+1
  if i<k then
      A[i] := 1
```



COSTO DI UN INCREMENTO = $O(k)$

COSTO DI n INCREMENTI = $nO(k) = O(nk)$

METODO DELL'AGGREGAZIONE

- CONSISTE NELLO STIMARE IL COSTO $T(n)$ DI n OPERAZIONI E DI EQUIDISTRIBUIRE TALE COSTO TRA LE n OPERAZIONI $\left(T(n)/n \right)$

STACK CON MULTIPOP (INIZIALMENTE VUOTO)

$\left.\begin{array}{l} POP(S) \\ PUSH(S,x) \end{array}\right\}$ OPERAZIONI ELEMENTARI

MULTIPOP (S,k) - OPERAZIONE DERIVATA

$$\underbrace{op_1, op_2, op_3, \dots, op_{n-1}, op_n}_{\downarrow} \in \{POP, PUSH, MULTIPOP\}$$

$$op'_1, op'_2, op'_3, \dots, op'_{m-1}, op'_m \in \{POP, PUSH\}$$

$$COSTO(\langle op_1 .. op_n \rangle) = COSTO(\langle op'_1 .. op'_m \rangle) = m$$

$$\#POP(\langle op'_1 .. op'_m \rangle) \le \#PUSH(\langle op'_1 .. op'_m \rangle)$$

$$\#PUSH(\langle op'_1 .. op'_m \rangle) = \#PUSH(\langle op_1 .. op_m \rangle) \le n$$

PERTANTO:

$$\#POP(\langle op'_1 .. op'_m \rangle) \le n$$

DA CUI

$$\begin{aligned} COSTO(\langle op_1 .. op_m \rangle) &= COSTO(\langle op'_1 .. op'_m \rangle) \\ &= \#POP(\langle op'_1 .. op'_m \rangle) + \#PUSH(\langle op'_1 .. op'_m \rangle) \\ &\le n + n = 2n \end{aligned}$$

$$COSTO\_AMMORTIZZATO\_PER\_OPERAZIONE \le 2$$

CONTATORE BINARIO CON INCREMENT
(INIZIALMENTE NULLO)

OPERAZIONI ELEMENTARI: SET E RESET DI SINGOLI BIT

$\overbrace{\hphantom{00000}}^{k \text{ BIT}}$

..00000	SU n OPERAZIONI INCREMENT
..00001	
..00010	A[0] CAMBIA n VOLTE
..00011	
..00100	A[1] CAMBIA $\left\lfloor \frac{n}{2} \right\rfloor$ VOLTE
..00101	
..00110	A[2] CAMBIA $\left\lfloor \frac{n}{2^2} \right\rfloor$ VOLTE
..00111	
..01000	A[3] CAMBIA $\left\lfloor \frac{n}{2^3} \right\rfloor$ VOLTE
..01001	
..01010	...

$$T(n) = \sum_{i=0}^{k-1} \left\lfloor \frac{n}{2^i} \right\rfloor \leq \sum_{i=0}^{k-1} \frac{n}{2^i} < \sum_{i=0}^{\infty} \frac{n}{2^i} = 2n$$

COSTO_AMMORTIZZATO_PER_OPERAZIONE ≤ 2

ESERCIZI

1) SE L'INSIEME DELLE OPERAZIONI SULLO STACK INCLUDESSE UN'OPERAZIONE MULTIPUSH, CHE INSERISCE k ELEMENTI NELLO STACK, IL LIMITE $O(1)$ SUL COSTO AMMORTIZZATO DELLE OPERAZIONI SULLO STACK SAREBBE ANCORA VALIDO?

2) SI DIMOSTRI CHE, SE UN'OPERAZIONE DECREMENT FOSSE INCLUSA NELL'ESEMPIO DEL CONTATORE DI k BIT, n OPERAZIONI RICHIEDEREBBERO UN TEMPO $\Theta(nk)$.

3) UNA SEQUENZA DI n OPERAZIONI VIENE ESEGUITA SU UNA STRUTTURA DATI. LA i-ESIMA OPERAZIONE COSTA i SE i È UNA POTENZA ESATTA DI 2, ALTRIMENTI COSTA 1. SI APPLICHI IL METODO DELL'AGGREGAZIONE PER DETERMINARE IL COSTO AMMORTIZZATO PER OPERAZIONE.

METODO DEGLI ACCANTONAMENTI

$op_1, op_2, \ldots, op_n$

$c_i =_{df}$ COSTO-REALE (op_i)

$\hat{c}_i =_{df}$ COSTO_AMMORTIZZATO (op_i) (DEFINITO DA NOI)

OBIETTIVO

DEFINIRE I COSTI AMMORTIZZATI IN MODO TALE CHE VALGA

$$\sum_{i=1}^{n} c_i \leq \sum_{i=1}^{n} \hat{c}_i$$

ESERCIZIO

3) UNA SEQUENZA DI n OPERAZIONI VIENE ESEGUITA SU UNA STRUTTURA DATI. LA i-ESIMA OPERAZIONE COSTA i SE i E' UNA POTENZA ESATTA DI 2, ALTRIMENTI COSTA 1. SI APPLICHI IL METODO DELL'AGGREGAZIONE PER DETERMINARE IL COSTO AMMORTIZZATO PER OPERAZIONE.

SOLUZIONE CON IL METODO DELL'AGGREGAZIONE

$$
\begin{aligned}
T(n) = \sum_{j=0}^{\lfloor \lg n \rfloor} 2^j + n - (\lfloor \lg n \rfloor + 1) &= \frac{2^{\lfloor \lg n \rfloor + 1} - 1}{2 - 1} + n - \lfloor \lg n \rfloor - 1 \\
&= 2 \cdot 2^{\lfloor \lg n \rfloor} + n - \lfloor \lg n \rfloor - 2 \\
&\leq 2n + n - \lfloor \lg n \rfloor - 2 \\
&= 3n - \lfloor \lg n \rfloor - 2 \\
&\leq 3n
\end{aligned}
$$

$$\hat{c} = \frac{T(n)}{n} \leq 3$$

SOLUZIONE CON IL METODO DEL POTENZIALE

$$\phi(i) = 2\big(i - \underbrace{\text{massima potenza di 2 che risulta} \leq i}_{2^{\lfloor \lg i \rfloor}}\big)$$

$$= 2\left(i - 2^{\lfloor \lg i \rfloor}\right)$$

$$\phi(i) = \begin{cases} 0 & i = 0 \\ 2\left(i - 2^{\lfloor \lg i \rfloor}\right) & i \geq 1 \end{cases}$$

Vale: $\phi(i) \geq \phi(0)$

CASO: $i \notin 2^{\mathbb{N}}$

$$\lfloor \lg(i-1) \rfloor = \lfloor \lg i \rfloor$$

$$\hat{c}_i = c_i + \phi(i) - \phi(i-1)$$

$$= 1 + 2\left(i - 2^{\lfloor \lg i \rfloor}\right) - 2\left(i - 1 - 2^{\lfloor \lg(i-1) \rfloor}\right)$$

$$= 1 + \cancel{2i} - \cancel{2 \cdot 2^{\lfloor \lg i \rfloor}} - \cancel{2i} + 2 + \cancel{2 \cdot 2^{\lfloor \lg(i-1) \rfloor}}$$

$$= 3$$

CASO: $i \in 2^{\mathbb{N}}$, $i \neq 1$

$$\lfloor \lg i \rfloor = \lg i \qquad \lfloor \lg(i-1) \rfloor = \lg i - 1$$

$$\hat{c}_i = c_i + \phi(i) - \phi(i-1)$$

$$= i + 2\left(i - 2^{\lfloor \lg i \rfloor}\right) - 2\left(i - 1 - 2^{\lfloor \lg(i-1) \rfloor}\right)$$

$$= i + \cancel{2i} - 2 \cdot 2^{\lfloor \lg i \rfloor} - \cancel{2i} + 2 + 2 \cdot 2^{\lfloor \lg(i-1) \rfloor}$$

$$= i - 2 \cdot 2^{\lg i} + 2 \cdot \underbrace{2^{\lg i - 1}} + 2$$

$$2^{\lg i - 1} = \frac{2^{\lg i}}{2} = \frac{i}{2}$$

$$= i - 2i + i + 2$$

$$= 2$$

CASO: $i = 1$

$$\hat{c}_i = c_i + \phi(i) - \phi(i-1)$$
$$= 1 + \cancel{2(i - 2^{\lceil \log i \rceil})} - 0$$
$$= 1$$

In definitiva abbiamo:

$$\hat{c}_i = \begin{cases} 1 & \text{se } i = 1 \\ 2 & \text{se } i \neq 1 \ \& \ i \in 2^{\mathbb{N}} \\ 3 & \text{se } i \notin 2^{\mathbb{N}} \end{cases}$$

METODO DEGLI ACCANTONAMENTI (CNT)

SE $\hat{c}_i > c_i$, c_i UNITÀ DI COSTO SONO UTILIZZATE PER PAGARE IL COSTO DI op_i

$\hat{c}_i - c_i$ UNITÀ DI COSTO SONO IMMAGAZZINATE SU ELEMENTI SPECIFICI DELLA STRUTTURA DATI

SE $c_i > \hat{c}_i$, LA DIFFERENZA $c_i - \hat{c}_i$ VIENE RECUPERATA DA CREDITI IMMAGAZZINATI NELLA STRUTTURA DATI

$\therefore$ VIENE RAGGIUNTO L'OBIETTIVO

STACK CON MULTIPOP (INIZIALMENTE VUOTO)

$\hat{c}_{PUSH} = 2$ (1 UNITÀ PER IL COSTO REALE + 1 UNITÀ ASSEGNATA ALL'ELEMENTO)

$\hat{c}_{POP} = \hat{c}_{MULTIPOP} = 0$

IN OGNI ISTANTE: $\sum_{i=1}^{n} \hat{c}_i - \sum_{i=1}^{n} c_i = |S| \geq 0$

E QUINDI $\sum_{i=1}^{n} \hat{c}_i \geq \sum_{i=1}^{n} c_i$

PERTANTO $\sum_{i=1}^{n} c_i \leq 2n$

CONTATORE BINARIO CON INCREMENT
(INIZIALMENTE NULLO)

$\hat{c}_{SET} = 2$ (1 UNITÀ PER PAGARE L'OPERAZIONE + 1 UNITÀ IMMAGAZZINATA SUL BIT STESSO)

$\hat{c}_{RESET} = 0$

$\hat{c}_{INCREMENT} \leq 2$

$$\sum_{i=1}^{p} \hat{c}_i - \sum_{i=1}^{p} c_i \geq \#\ \text{BIT SUL CONTATORE UGUALI AD 1}$$

$$\geq 0$$

$\therefore$ VALE $\displaystyle\sum_{i=1}^{p} \hat{c}_i \geq \sum_{i=1}^{p} c_i$

ESERCIZI

1) UNA SEQUENZA DI OPERAZIONI VIENE ESEGUITA SU UNO STACK LA CUI DIMENSIONE NON SUPERA MAI k. DOPO OGNI k OPERAZIONI, VIENE FATTA UNA COPIA DI BACKUP DELL'INTERO STACK. DIMOSTRARE CHE IL COSTO DI n OPERAZIONI SU STACK, INCLUSA LA COPIA DELLO STACK, E' $O(n)$ ASSEGNANDO DEI COSTI AMMORTIZZATI APPROPRIATI ALLE VARIE OPERAZIONI.

2) UNA SEQUENZA DI n OPERAZIONI VIENE ESEGUITA SU UNA STRUTTURA DATI. LA i-ESIMA OPERAZIONE COSTA i SE i E' UNA POTENZA ESATTA DI 2, ALTRIMENTI COSTA 1. SI APPLICHI IL METODO DEGLI ACCANTONAMENTI PER DETERMINARE IL COSTO AMMORTIZZATO PER OPERAZIONE.

3) SI SUPPONGA NON SOLTANTO DI INCREMENTARE UN CONTATORE, MA ANCHE DI RIPORTARLO A 0. SI SPIEGHI COME IMPLEMENTARE UN CONTATORE COME UN ARRAY DI BIT IN MODO CHE UNA QUALSIASI SEQUENZA DI n OPERAZIONI INCREMENT E RESET IMPIEGHI UN TEMPO $O(n)$ CON UN CONTATORE INIZIALMENTE A ZERO.

METODO DEL POTENZIALE

- ALLA STRUTTURA DATI VIENE ASSEGNATA UNA FUNZIONE POTENZIALE

$$D_0 \xrightarrow[op_1]{} D_1 \xrightarrow[op_2]{} D_2 \xrightarrow[op_3]{} D_3 \xrightarrow[op_4]{} \quad \cdots \quad \xrightarrow[op_n]{} D_n$$

DEFINIZIONE

$\Phi : D \longmapsto \Phi(D) \in \mathbb{R}$ (POTENZIALE)

$op_i \longrightarrow c_i$ (COSTO REALE)

$$\hat{c}_i = c_i + \Phi(D_i) - \Phi(D_{i-1})$$

(COSTO AMMORTIZZATO)

$$\sum_{i=1}^{m} \hat{c}_i = \sum_{i=1}^{m} \left[c_i + \left(\Phi(D_i) - \Phi(D_{i-1}) \right) \right]$$

$$= \sum_{i=1}^{m} c_i + \Phi(D_m) - \Phi(D_0)$$

<u>LEMMA</u> $\sum_{i=1}^{m} \hat{c}_i \geq \sum_{i=1}^{m} c_i$ SSE $\Phi(D_n) \geq \Phi(D_0)$

$\Phi(D_k) \geq \Phi(D_0)$ per $k = 1, 2, \dots, n$

STACK CON MULTIPOP (INIZIALMENTE VUOTO)

PONIAMO: $\Phi(S) =_{df} |S|$

SE S_0 È LO STACK VUOTO, $\Phi(S_0) = 0$.

QUINDI $\Phi(S) \geq \Phi(S_0)$.

$$\hat{c}_{POP} = c_{POP} + \Phi(S_i) - \Phi(S_{i-1}) = 1 + |S_i| - |S_{i-1}|$$
$$= 0 \quad \text{IN QUANTO} \quad |S_i| = |S_{i-1}| - 1$$

$$\hat{c}_{MULTIPOP} = c_{MULTIPOP} + \Phi(S_i) - \Phi(S_{i-1})$$
$$= k + |S_i| - |S_{i-1}| = 0 \quad (\text{IN QUANTO} \ |S_i| = |S_{i-1}| - k)$$

$$\hat{c}_{PUSH} = c_{PUSH} + |S_i| - |S_{i-1}| = 1 + 1 = 2$$
$$(\text{IN QUANTO} \quad |S_i| = |S_{i-1}| + 1)$$

PERTANTO: $\sum_{i=1}^{m} c_i \leq \sum_{i=1}^{m} \hat{c}_i \leq 2n$

CONTATORE BINARIO CON INCREMENTO (INIZIALMENTE NULLO)

$\Phi(D_i) = \#$ BIT UGUALI AD 1

SIA D_0 LA CONFIGURAZIONE NULLA DEL CONTATORE

$\Phi(D_0) = 0.$

INOLTRE $\Phi(D_i) \geq \Phi(D_0)$

$$\hat{c}_i = c_i + \underbrace{\phi(D_i) - \phi(D_{i-1})}_{\Delta\phi_i}$$

SI HA:

$$c_i = \#\,SET_i + \#\,RESET_i$$

NOTA: $0 \leq \#SET_i \leq 1$

$$\Delta\phi_i = \#\,SET_i - \#\,RESET_i$$

PERTANTO:

$$\hat{c}_i = c_i + \Delta\phi_i = (\#\,SET_i + \#\,RESET_i) + (\#\,SET_i - \#\,RESET_i) = 2 \cdot \#\,SET_i \leq 2$$

DA CUI: $\displaystyle\sum_{i=1}^{n} c_i \leq \sum_{i=1}^{n} \hat{c}_i = 2 \cdot \sum_{i=1}^{n} \#SET_i \leq 2n,$

CONTATORE BINARIO CON INCREMENTO INIZIALMENTE NON NULLO

PONIAMO COME PRIMA

$\phi(D_i)$ # BIT UGUALI AD 1

SI HA

$$\sum_{i=1}^{n} c_i = \sum_{i=1}^{n} \hat{c}_i - \Phi(D_n) + \Phi(D_0)$$

$$\leq 2n + \Phi(D_0) \leq 2n + k$$

(DOVE k E' IL NUMERO DI BIT DEL CONTATORE)

SE $k = O(n)$ ALLORA $2n + k = O(n)$ E QUINDI

$$\sum_{i=1}^{n} c_i = O(n)$$

TABELLE DINAMICHE

- SI TRATTA DI TABELLE SOGGETTE A RIALLOCAZIONE PER RISOLVERE GLI OVERFLOW

- SIA T UNA TABELLA. PONIAMO:

$\mathrm{size}[T] =_{\mathrm{def}}$ DIMENSIONE DELLA TABELLA

$\mathrm{num}[T] =_{\mathrm{def}}$ NUMERO DEGLI ELEMENTI IN T

$$\alpha(T) =_{\mathrm{def}} \begin{cases} \dfrac{\mathrm{num}[T]}{\mathrm{size}[T]} & \text{SE } \mathrm{SIZE}[T] > 0 \\ 1 & \text{ALTRIMENTI} \end{cases}$$

(FATTORE DI CARICO)

```
Table-Insert (T, x)
if size[T] = 0 then
    - si allochi Table[T] di dimensione 1
    size[T] := 1
if num[T] = size[T] then
    - si allochi in new_table una tabella di dim. 2·size[T]
    - si copi Table[T] in new_table
    - si deallochi Table[T]
    Table[T] := new_table
    size[T] := 2 * size[T]

- si inserisca x in Table[T]
- num[T] := num[T] + 1
```

ANALISI DI n INSERIMENTI SU UNA TABELLA INIZIALMENTE NULLA

(I COSTI SONO VALUTATI IN TERMINI DI INSERIMENTI ELEMENTARI)

ANALISI GROSSOLANA

$n = 2^k$ INSERIMENTI

COSTO DELL'INSERIMENTO $(2^{k-1}+1)$-ESIMO $= 2^{k-1}+1$

$$= \frac{n}{2} + 1$$

COSTO DI UN INSERIMENTO $= O(n)$

COSTO DI n INSERIMENTI $= O(n^2)$

METODO DELL'AGGREGAZIONE

	C_i	
INSERIMENTO	COSTO COPIA	COSTO INSERIMENTO
1	/	1
2	1	1
3	2	1
4	/	1
5	4	1
6	/	1
7	/	1
8	/	1
9	8	1
10	/	1
...	...	...

$$c_i = \begin{cases} i & \text{SE } i-1 \text{ E' UNA POTENZA DI } 2 \\ 1 & \text{ALTRIMENTI} \end{cases}$$

$$c_i = \begin{cases} i & \text{SE } i-1 \text{ È UNA POTENZA DI } 2 \\ 1 & \text{ALTRIMENTI} \end{cases}$$

$$\sum_{i=1}^{n} c_i = \left(1+2+\dots+2^{\lfloor \lg(n-1) \rfloor}\right) + n$$

$$= 2^{\lfloor \lg(n-1) \rfloor + 1} - 1 + n$$

$$\leq 2^{\lg(n-1)+1} - 1 + n$$

$$= 2(n-1) - 1 + n$$

$$= 3n - 3 = O(n)$$

$$\hat{c} = \frac{\sum_{i=1}^{n} c_i}{n} \leq \frac{3n-3}{n} = 3 - \frac{3}{n} = O(1)$$

ANALISI CON IL METODO DEGLI ACCANTONAMENTI

$$\hat{c}_{ins} = 3 \begin{cases} 1 \text{ UNITÀ PER IL COSTO REALE} \\ 1 \text{ UNITÀ PER PAGARE IL COSTO DELLA COPIA} \\ 1 \text{ UNITÀ PER PAGARE IL COSTO DELLA COPIA} \\ \quad \text{DI UN ELEMENTO GIÀ RICOPIATO} \end{cases}$$

SI OSSERVI CHE:

$$\sum_{i=1}^{n} \hat{c}_i \geq \sum_{i=1}^{n} c_i$$

PERTANTO

$$T(n) \leq 3n$$

ANALISI CON IL METODO DEL POTENZIALE

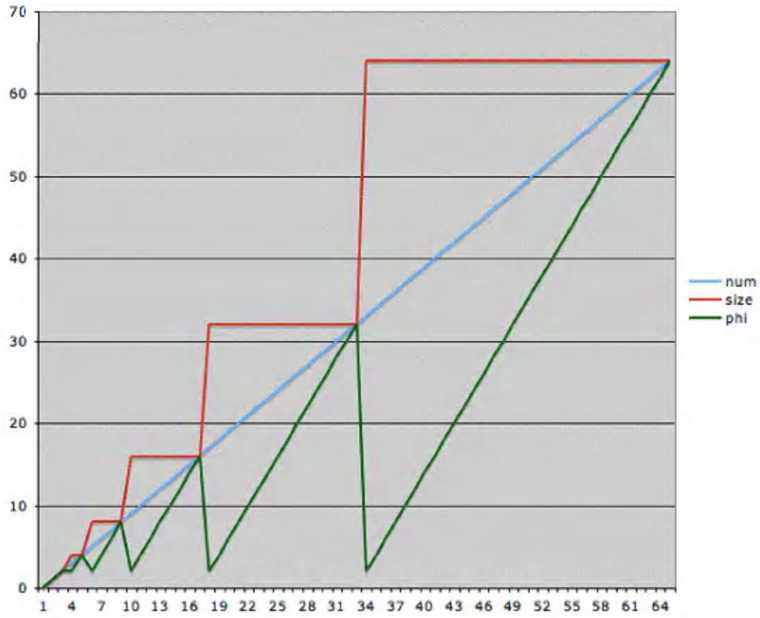
$\Phi(T) = 2 \cdot num[T] - size[T]$

SI HA: $\Phi(T_0) = 0$ (T_0 TABELLA VUOTA)

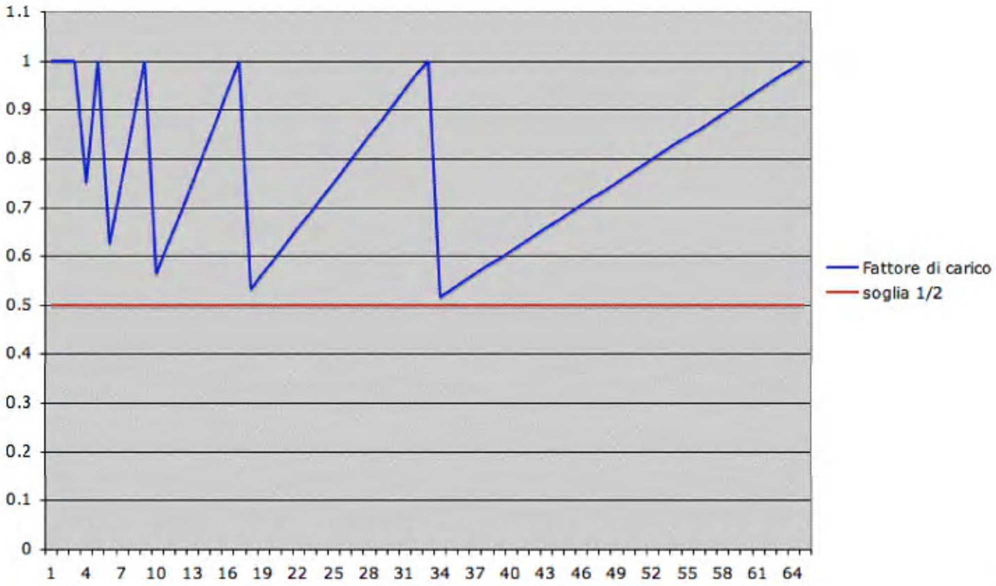
INOLTRE: $\frac{1}{2} size[T] \leq num[T]$

E PERTANTO $\Phi(T) \geq 0 = \Phi(T_0)$

CIOÈ IL METODO DEL POTENZIALE PUÒ ESSERE UTILIZZATO PER VALUTARE I COSTI AMMORTIZZATI



70
60
50
40
30
20
10
0
1 4 7 10 13 16 19 22 25 28 31 34 37 40 43 46 49 52 55 58 61 64
num
size
phi


1.1
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
1 4 7 10 13 16 19 22 25 28 31 34 37 40 43 46 49 52 55 58 61 64
Fattore di carico
soglia 1/2

INSERIMENTO SENZA ESPANSIONE

$$\hat{c}_{ins} = c_{ins} + \Phi(T_i) - \Phi(T_{i-1})$$

$s_i = s_{i-1}$

$n_i = n_{i-1} + 1$

$$= 1 + (2 \cdot n_i - s_i) - (2n_{i-1} - s_{i-1})$$

$$= 1 + 2(\cancel{n_{i-1}} + 1) - \cancel{s_{i-1}} - 2\cancel{n_{i-1}} + \cancel{s_{i-1}}$$

$$= 3$$

INSERIMENTO CON ESPANSIONE

$$\hat{c}_{ins} = c_{ins} + \Phi(T_i) - \Phi(T_{i-1})$$

$s_i = 2s_{i-1} = 2n_{i-1}$

$n_i = n_{i-1} + 1$

$$= n_i + (2 \cdot n_i - s_i) - (2n_{i-1} - s_{i-1})$$

$$= \cancel{n_{i-1}} + 1 + 2(\cancel{n_{i-1}} + 1) - 2\cancel{n_{i-1}} - \cancel{2n_{i-1}} + \cancel{n_{i-1}}$$

$$= 3$$

PERTANTO $T(n) \leq 3n$

TABELLE DINAMICHE CON INSERIMENTI E CANCELLAZIONI

SI CONSIDERI LA SEGUENTE SEQUENZA DI OPERAZIONI SU UNA TABELLA DINAMICA CHE SI DIMEZZA QUANDO IL FATTORE DI CARICO SCENDE AL DI SOTTO DI $1/2$.

SIA $n = 2^k$.

$$\underbrace{I\ I \ldots I}_{2^{k-1}}\ \overbrace{I\ C\ C\ I\ I\ C\ C\ I\ I \ldots}^{2^{k-1}}$$

$\downarrow$ $n = s = 2^{k-1}$

$\downarrow$ $s = 2^k$

$\downarrow$ $s = 2^{k-1}$

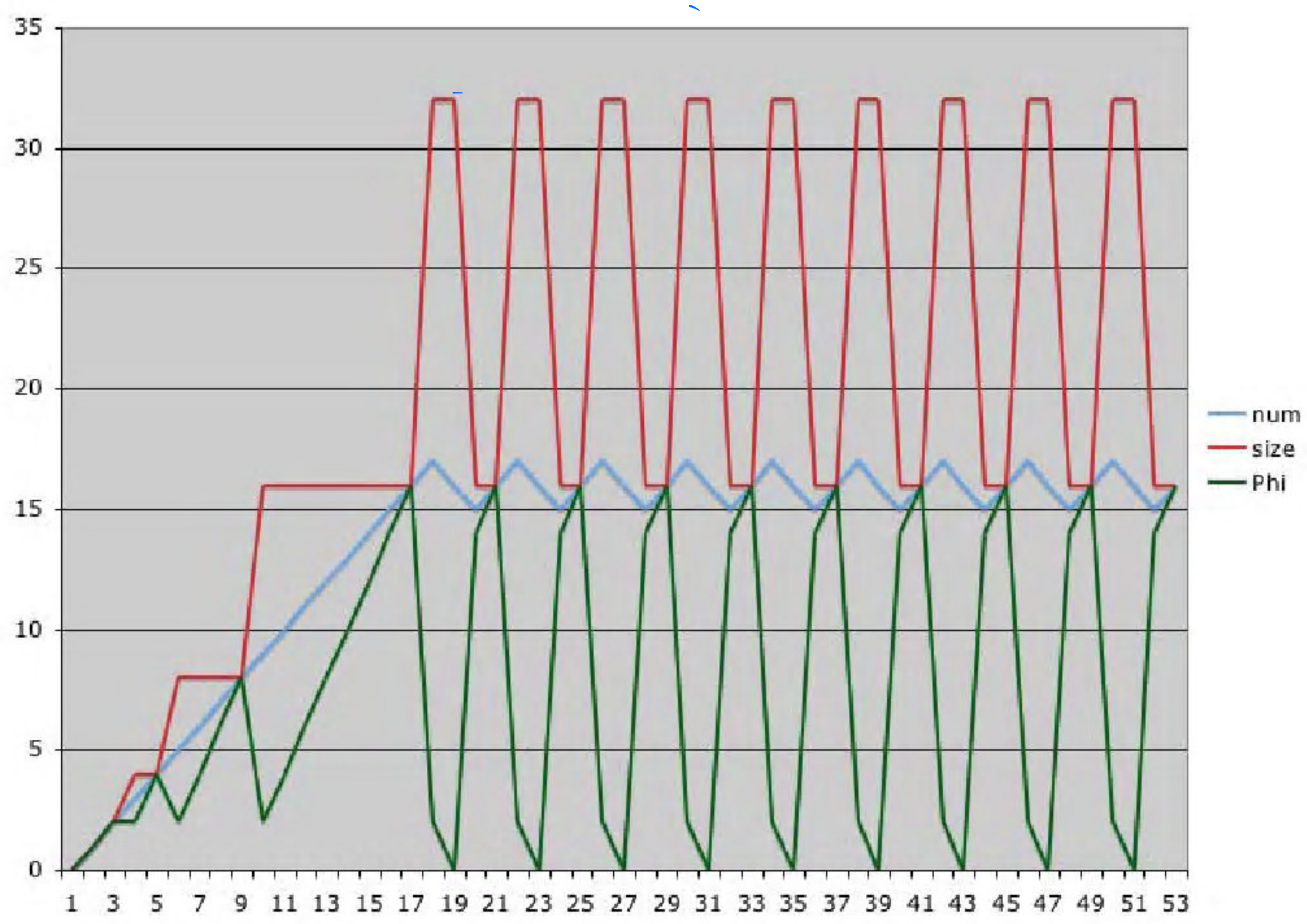
$T(2^{k-1}) \leq 3 \cdot 2^{k-1}$

$$2^{k-1} \left\{ \begin{array}{lll} c_{m+1} & = m+1 & \text{(ESPANSIONE)} \\ c_{m+2} & = 1 & \\ c_{m+3} & = m & \text{(CONTRAZIONE)} \\ c_{m+4} & = 1 & \\ c_{m+5} & = m+1 & \text{(ESPANSIONE)} \end{array} \right.$$

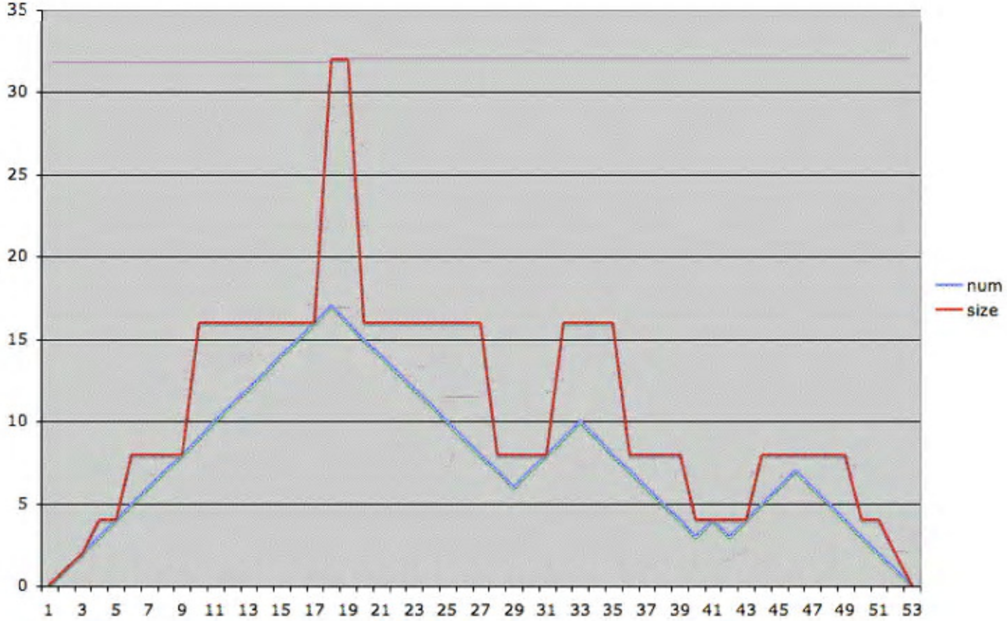
$2^{k-1} = m$

$\frac{n^2}{8} = \frac{n}{2} \cdot \frac{n}{4} = 2^{k-1} \cdot 2^{k-2} = m \cdot 2^{k-2}$ $\leftarrow$

COSTO REALE $\Theta(n^2)$



35
30
25
20
15
10
5
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31 33 35 37 39 41 43 45 47 49 51 53
num
size
Phi



35
30
25
20
15
10
5
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31 33 35 37 39 41 43 45 47 49 51 53
num
size

CI ACCONTENTIAMO DI AVERE $\alpha(T) \geq \frac{1}{4}$

PONIAMO:

$$\Phi(T) = \begin{cases} 2\,num[T] - size[T] & \text{SE} \quad \alpha(T) > \frac{1}{2} \\ \frac{1}{2}\,size[T] - num[T] & \text{SE} \quad \alpha(T) \leq \frac{1}{2} \end{cases}$$

OSSERVIAMO CHE SE $\alpha(T) = \frac{1}{2}$:

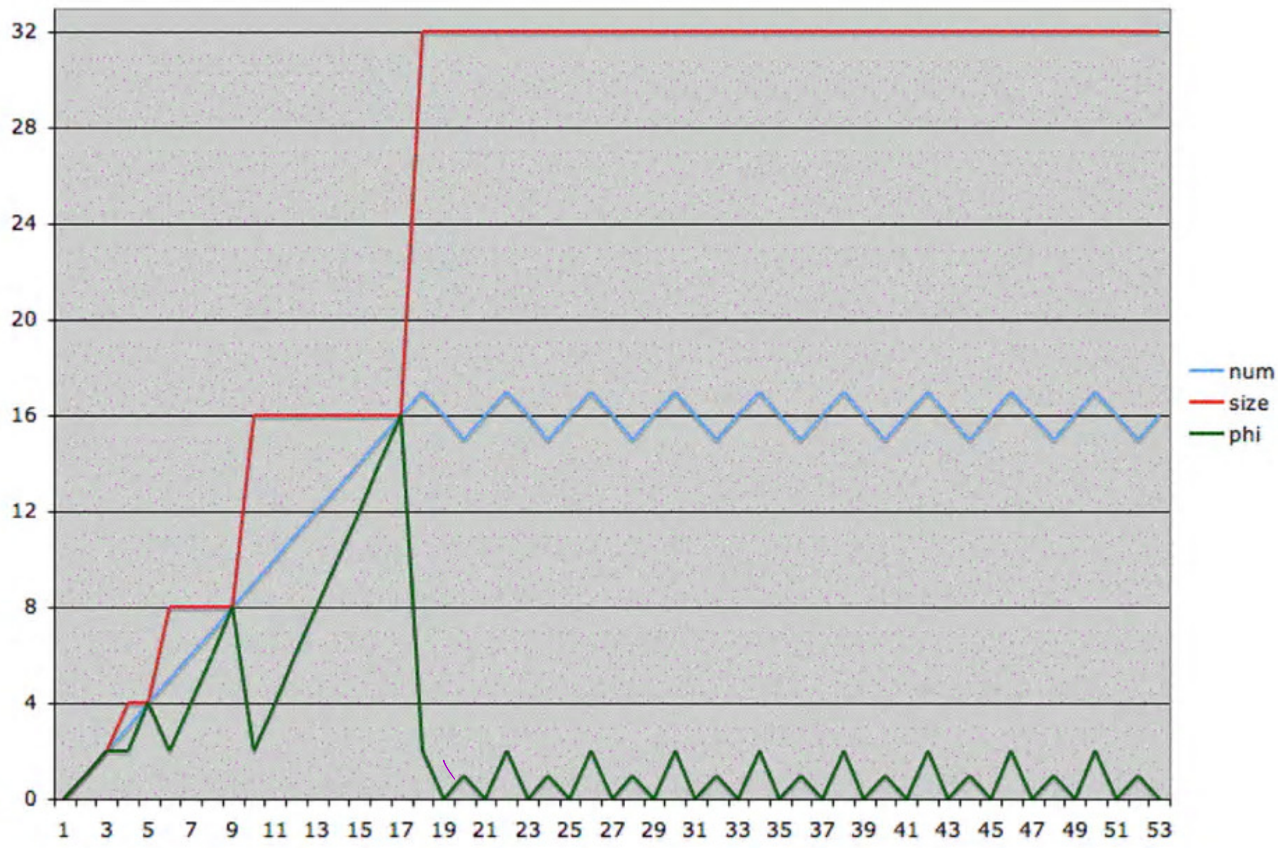
$\frac{num[T]}{size[T]} = \frac{1}{2}$, CIOE' $\frac{1}{2}\,size[T] - num[T] = 0$

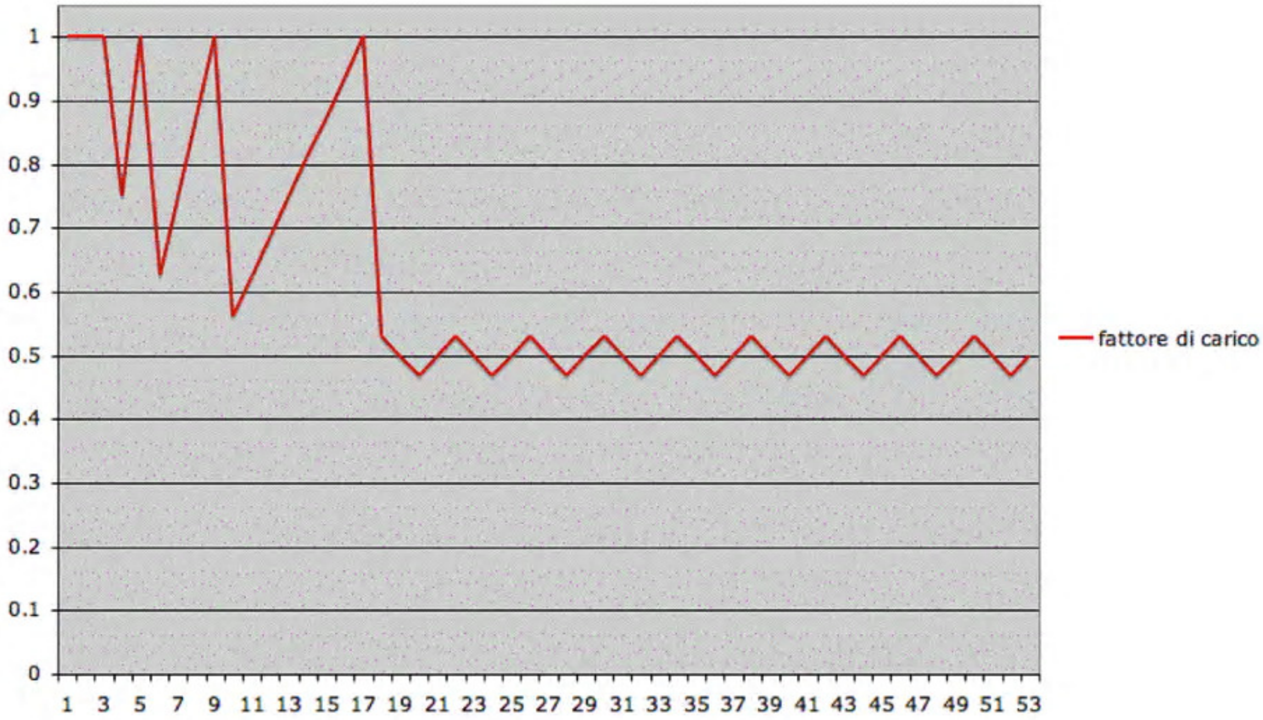
E $2\,num[T] - size[T] = 0$. PERTANTO

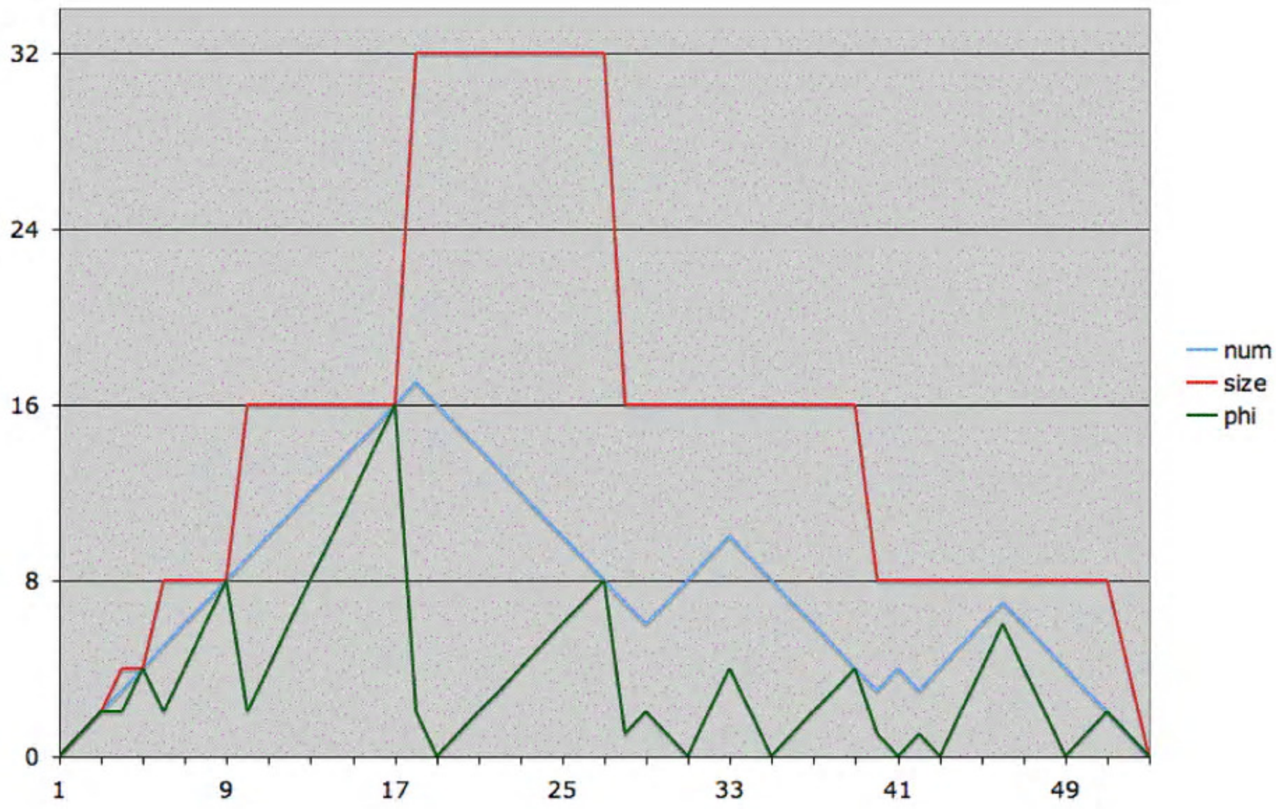
$$\Phi(T) = \begin{cases} 2\,num[T] - size[T] & \text{SE} \quad \alpha(T) \geq \frac{1}{2} \\ \frac{1}{2}\,size[T] - num[T] & \text{SE} \quad \alpha(T) \leq \frac{1}{2} \end{cases}$$

INOLTRE VALE SEMPRE

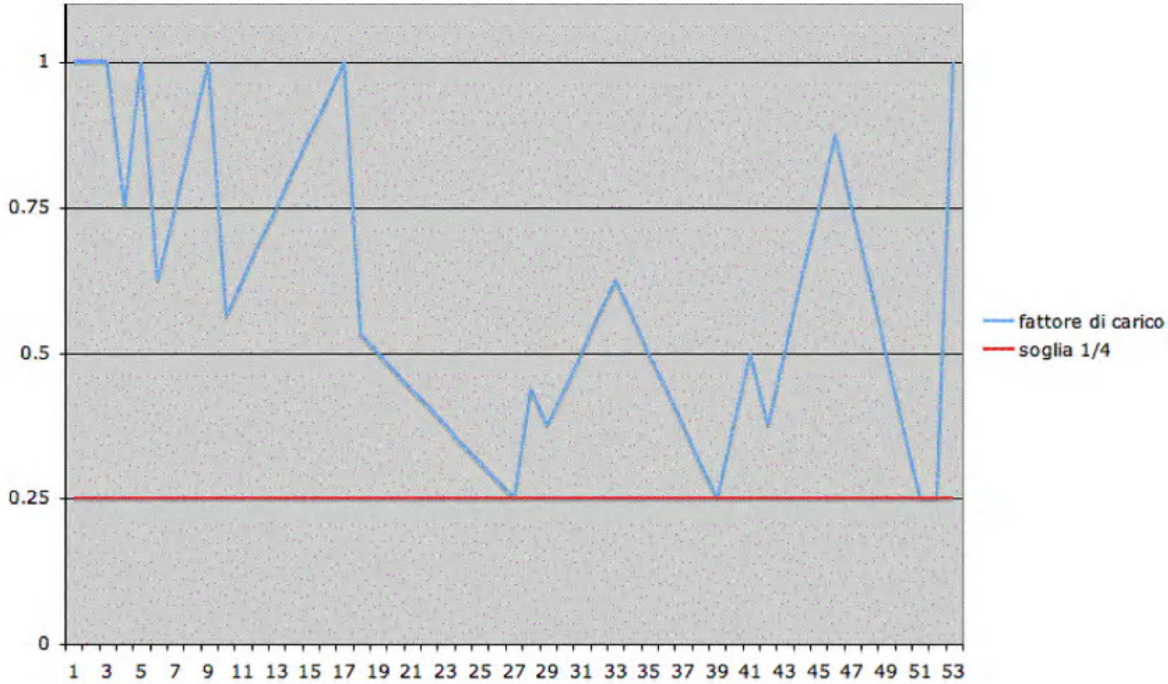
$\Phi(T) \geq \Phi(T_0) = 0$ (T_0 TABELLA VUOTA)


32
28
24
20
16
12
8
4
0
1
3
5
7
9
11
13
15
17
19
21
23
25
27
29
31
33
35
37
39
41
43
45
47
49
51
53
num
size
phi


1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31 33 35 37 39 41 43 45 47 49 51 53
fattore di carico

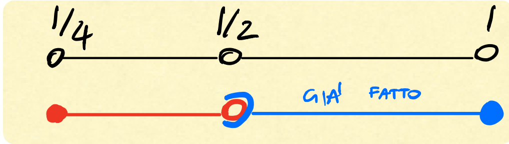


32
24
16
8
0
1
9
17
25
33
41
49
num
size
phi



1
0.75
0.5
0.25
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31 33 35 37 39 41 43 45 47 49 51 53
fattore di carico
soglia 1/4

ANALISI INSERIMENTI



CASO $\frac{1}{4} \le \alpha(T) < \frac{1}{2}$

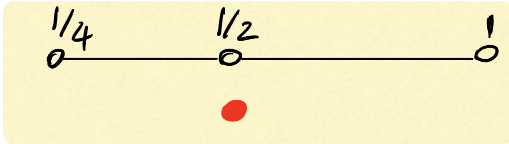
$$\hat{c}_{ins} = c_{ins} + \phi(T_i) - \phi(T_{i-1})$$

$$= 1 + \left(\frac{1}{2} s_i - n_i\right) - \left(\frac{1}{2} s_{i-1} - n_{i-1}\right)$$

$$= 1 + \frac{1}{2} s_{i-1} - n_{i-1} - 1 - \frac{1}{2} s_{i-1} + n_{i-1}$$

$$= 0$$

$$\begin{cases} n_i = n_{i-1} + 1 \\ s_i = s_{i-1} \end{cases}$$



CASO $\alpha(T) = \frac{1}{2}$

$$\hat{c}_{ins} = c_{ins} + \phi(T_i) - \phi(T_{i-1})$$

$$= 1 + (2 n_i - s_i) - (2 n_{i-1} - s_{i-1})$$

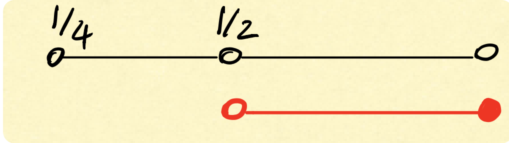
$$= 1 + 2(n_{i-1} + 1) - s_{i-1} - 2 n_{i-1} + s_{i-1}$$

$$= 3$$

$$\begin{cases} n_i = n_{i-1} + 1 \\ s_i = s_{i-1} \end{cases}$$

ANALISI CANCELLAZIONI

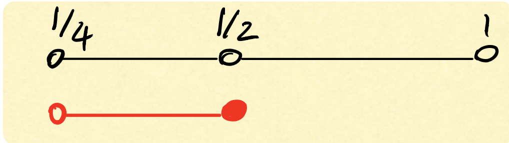
CASO $\frac{1}{2} < \alpha(T) \leq 1$



$$\hat{c}_{canc} = c_{canc} + \Phi(T_i) - \Phi(T_{i-1})$$
$$= 1 + (2n_i - s_i) - (2n_{i-1} - s_{i-1})$$
$$= 1 + 2n_{i-1} - 2 - s_{i-1} - 2n_{i-1} + s_{i-1}$$
$$= -1$$

$$\begin{cases} n_i = n_{i-1} - 1 \\ s_i = s_{i-1} \end{cases}$$

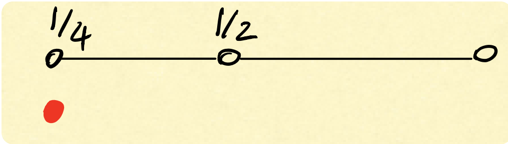
CASO $\frac{1}{4} < \alpha(T) \leq \frac{1}{2}$



$$\hat{c}_{canc} = c_{canc} + \Phi(T_i) - \Phi(T_{i-1})$$
$$= 1 + \left(\frac{1}{2} s_i - n_i\right) - \left(\frac{1}{2} s_{i-1} - n_{i-1}\right)$$
$$= 1 + \frac{1}{2} s_{i-1} - n_{i-1} + 1 - \frac{1}{2} s_{i-1} + n_{i-1}$$
$$= 2$$

$$\begin{cases} n_i = n_{i-1} - 1 \\ s_i = s_{i-1} \end{cases}$$

ANALISI CANCELLAZIONI (CNTD.)



CASO $\alpha(T) = \frac{1}{4}$

$$
\begin{aligned}
\hat{c}_{canc} &= c_{canc} + \Phi(T_i) - \Phi(T_{i-1}) \\
&= n_{i-1} + \left(\tfrac{1}{2} s_i - n_i\right) - \left(\tfrac{1}{2} s_{i-1} - n_{i-1}\right) \\
&= \frac{s_{i-1}}{4} + \frac{s_{i-1}}{4} - \frac{s_{i-1}}{4} + 1 - \frac{s_{i-1}}{2} + \frac{s_{i-1}}{4} \\
&= 1
\end{aligned}
$$

$$
\begin{cases}
n_{i-1} = s_{i-1}/4 \\
n_i = n_{i-1} - 1 = \dfrac{s_{i-1}}{4} - 1 \\
s_i = s_{i-1}/2
\end{cases}
$$

RIASSUMENDO:

$$\hat{c}_{ins} = \begin{cases} 0 & \text{SE } \frac{1}{4} \leq \alpha(T) < \frac{1}{2} \\ 3 & \text{SE } \frac{1}{2} \leq \alpha(T) \leq 1 \end{cases}$$

$$\hat{c}_{canc} = \begin{cases} -1 & \text{SE } \frac{1}{2} < \alpha(T) \leq 1 \\ 2 & \text{SE } \frac{1}{4} < \alpha(T) \leq \frac{1}{2} \\ 1 & \text{SE } \alpha(T) = \frac{1}{4} \end{cases}$$

PERTANTO

$$\hat{c}_{ins} \leq 3$$

$$\hat{c}_{canc} \leq 2$$

DA CUI $\quad T(n) \leq \sum_{i=1}^{n} \hat{c}_i \leq 3n$

ESERCIZI

1) SIA ϕ UNA FUNZIONE POTENZIALE TALE CHE $\phi(D_i) \geq \phi(D_0)$ PER OGNI i, MA $\phi(D_0) \neq 0$. SI DETERMINI UNA FUNZIONE POTENZIALE ϕ' TALE CHE $\phi'(D_0) = 0$, $\phi'(D_i) \geq 0$ PER OGNI $i \geq 1$, E CHE I COSTI AMMORTIZZATI CON ϕ' SIANO UGUALI AI COSTI AMMORTIZZATI CON ϕ.

2) UNA SEQUENZA DI n OPERAZIONI VIENE ESEGUITA SU UNA STRUTTURA DATI. LA i-ESIMA OPERAZIONE COSTA i SE i E' UNA POTENZA ESATTA DI 2, ALTRIMENTI COSTA 1. SI APPLICHI IL METODO DEL POTENZIALE PER DETERMINARE IL COSTO AMMORTIZZATO PER OPERAZIONE.

3) QUAL E' IL COSTO TOTALE PER ESEGUIRE n DELLE OPERAZIONI SU STACK PUSH, POP E MULTIPOP, SUPPONENDO CHE LO STACK INIZI CON s_0 OGGETTI E FINISCA CON s_n OGGETTI?

4) SI SUPPONGA DI AVERE UN NORMALE MIN-HEAP BINARIO CON n ELEMENTI CHE CONSENTE DI ESEGUIRE LE ISTRUZIONI INSERT E EXTRACT-MIN NEL TEMPO $O(\lg n)$ NEL CASO PEGGIORE.
SI DEFINISCA UN POTENZIALE Φ TALE CHE IL COSTO AMMORTIZZATO DI INSERT SIA $O(\lg n)$ E IL COSTO AMMORTIZZATO DI EXTRACT-MIN SIA $O(1)$.

5) SI SUPPONGA CHE UN CONTATORE INIZI DA UN NUMERO CON b BIT UGUALI A 1. SI DIMOSTRI CHE IL COSTO PER ESEGUIRE n OPERAZIONI INCREMENT E' $O(n)$ SE $n = \Omega(b)$.

6) SI SPIEGHI COME IMPLEMENTARE UNA CODA CON DUE STACK ORDINARI IN MODO CHE IL COSTO AMMORTIZZATO DI CIASCUNA OPERAZIONE ENQUEUE E DEQUEUE SIA $O(1)$.

7) SI PROGETTI UNA STRUTTURA DATI PER SUPPORTARE LE SEGUENTI OPERAZIONI PER UN INSIEME DINAMICO S DI INTERI:

INSERT(S, x) - INSERISCE x IN S

DELETE-LARGER-HALF(S) - CANCELLA GLI $\left\lceil \frac{|S|}{2} \right\rceil$ ELEMENTI PIÙ GRANDI DA S

SI SPIEGHI COME IMPLEMENTARE QUESTA STRUTTURA DATI IN MODO CHE QUALSIASI SEQUENZA DI m OPERAZIONI VENGA ESEGUITA NEL TEMPO $O(m)$.

SCALARE I COSTI

- SUPPONIAMO CHE $c_{PUSH} = 1$

$c_{POP} = k$ (k COSTANTE)

$c_{MULTIPOP} = k \cdot s$ (s NUMERO DI POP)

- ANALIZZIAMO CON IL METODO DEL POTENZIALE UNA SEQUENZA DI n OPERAZIONI

$$\boxed{\Phi(S) = |S|}$$

$$\hat{c}_{PUSH} = c_{PUSH} + \Delta\Phi = 2$$

$$\hat{c}_{POP} = c_{POP} + \Delta\Phi = k - 1$$

$$\hat{c}_{MULTIPOP} = c_{MULTIPOP} + \Delta\Phi = k \cdot s - s = (k-1)s = \Theta(s)$$

PROBLEMA!

$$c_{PUSH} = 1$$

$$c_{POP} = k \quad (k \text{ COSTANTE})$$

$$c_{MULTIPOP} = k \cdot s \quad (s \text{ NUMERO DI POP})$$

- ANALIZZIAMO CON IL METODO DEL POTENZIALE UNA SEQUENZA DI n OPERAZIONI

$$\boxed{\phi(S) = k\,|S| \;/\; \phi(S) \geq \phi(S_0) = 0}$$

$$\hat{c}_{PUSH} = c_{PUSH} + \Delta\phi = k+1$$

$$\hat{c}_{POP} = c_{POP} + \Delta\phi = 0$$

$$\hat{c}_{MULTIPOP} = c_{MULTIPOP} + \Delta\phi = k \cdot s - ks = 0 \longrightarrow \sum_{i=1}^{n} c_i = O(n)$$