

## 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 È PROBABILISTICA, MA SI TRATTA DI UNA MEDIA NEL CASO PEGGIORE

## TRE METODI:

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

## DUE ESEMPLI:

- STACK CON MULTIPOP
- CONTATORE BINARIO CON INCREMENT

## TABELLE DINAMICHE

## STACK CON MULTIPOP

POP(S) → COSTO  $O(1)$   
PUSH(S, x) → COSTO  $O(1)$   
STACK\_EMPTY(S) → COSTO  $O(1)$

MULTIPOP(S, k)

while not STACK\_EMPTY(S) and k  $\neq 0$  do

POP(S)

k := k - 1

MULTIPOP(S, k) → 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

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

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

1101011  $\mapsto$  1101100

INCREMENT (A)

$i := 0$

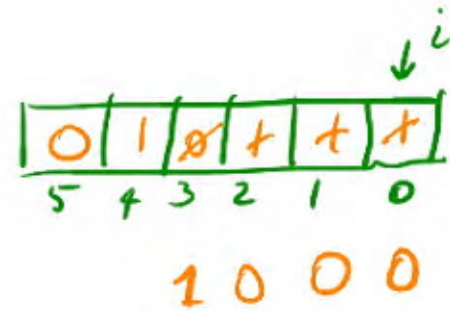
while  $i < k$  and  $A[i] = 1$  do

$A[i] := 0$

$i := i + 1$

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 ( $T(n)/n$ )

## STACK CON MULTIPOP (INIZIALMENTE VUOTO)

POP(S) } OPERAZIONI ELEMENTARI  
PUSH(S,x) }

MULTIPOP(S, k) - OPERAZIONE DERIVATA

$op_1, op_2, op_3, \dots, op_{m-1}, op_m \in \{POP, PUSH, MULTIPOP\}$



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

$$\text{COSTO}(\langle op_1 \dots op_m \rangle) = \text{COSTO}(\langle op'_1 \dots op'_m \rangle) = m$$



$$\# \text{POP} (\langle op'_1 \dots op'_m \rangle) \leq \# \text{PUSH} (\langle op'_1 \dots op'_m \rangle)$$

$$\# \text{PUSH} (\langle op'_1 \dots op'_m \rangle) = \# \text{PUSH} (\langle op_1 \dots op_m \rangle) \leq n$$

PERTANTO:

$$\# \text{POP} (\langle op'_1 \dots op'_m \rangle) \leq n$$

DA CUI

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

COSTO\_AMMORTIZZATO\_PER\_OPERAZIONE  $\leq 2$

# CONTATORE BINARIO CON INCREMENT (INIZIALMENTE NULLO)

OPERAZIONI ELEMENTARI: SET E RESET DI SINGOLI BIT

<u>K BIT</u>	SU	$n$	OPERAZIONI	INCREMENT
.. 00000				
.. 00001				
.. 00010				
.. 00011				
.. 00100	A[0]	CAMBIA	$n$	VOLTE
.. 00101				
.. 00110	A[1]	CAMBIA	$\lfloor \frac{n}{2} \rfloor$	VOLTE
.. 00111				
.. 01000	A[2]	CAMBIA	$\lfloor \frac{n}{2^2} \rfloor$	VOLTE
.. 01001				
.. 01010	A[3]	CAMBIA	$\lfloor \frac{n}{2^3} \rfloor$	VOLTE
			---	

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

COSTO\_AMMORTIZZATO\_PER\_OPERAZIONE  $\leq 2$

## 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  $i$  E' UNA POTENZA ESATTA DI  $2$ , ALTRIMENTI COSTA  $1$ . SI APPLICHI IL METODO DELL'AGGREGAZIONE PER DETERMINARE IL COSTO AMMORTIZZATO PER OPERAZIONE.

## METODO DEGLI ACCANTONAMENTI

$op_1, op_2, \dots, op_m$

$c_i =_{df}$  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

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

## METODO DEGLI ACCANTONAMENTI (CNT)

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

$\hat{c}_i - c_i$  UNITA' 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)

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

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

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

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

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

## CONTATORE BINARIO CON INCREMENT (INIZIALMENTE NULLO)

$$\hat{C}_{\text{SET}} = 2$$

$$\hat{C}_{\text{RESET}} = 0$$

---

$$\hat{C}_{\text{INCREMENT}} \leq 2$$

(1 UNITA' PER PAGARE L'OPERAZIONE +  
1 UNITA' IMMAGAZZINATA SUL BIT STESSO)

$$\sum_{i=1}^P \hat{C}_i - \sum_{i=1}^P C_i \geq \# \text{ BIT SUL CONTATORE UGUALI AD 1}$$
$$\geq 0$$

$\therefore$  VALE  $\sum_{i=1}^P \hat{C}_i \geq \sum_{i=1}^P C_i$



## 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'  $O(m)$  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  $O(m)$  CON UN CONTATORE INIZIALMENTE A ZERO.

## METODO DEL POTENZIALE

- ALLA STRUTTURA DATI VIENE ASSEGNATA UNA FUNZIONE POTENZIALE

$$D_0 \xrightarrow{op_1} D_1 \xrightarrow{op_2} D_2 \xrightarrow{op_3} D_3 \xrightarrow{\dots} \xrightarrow{op_n} D_n$$

## DEFINIZIONE

$$\phi : D \mapsto \phi(D) \in \mathbb{R} \quad (\text{POTENZIALE})$$

$$op_i \mapsto c_i \quad (\text{COSTO REALE})$$

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

(COSTO AMMORTIZZATO)

$$\begin{aligned}\sum_{i=1}^M \hat{c}_i &= \sum_{i=1}^M [c_i + (\Phi(D_i) - \Phi(D_{i-1}))] \\ &= \sum_{i=1}^M c_i + \Phi(D_m) - \Phi(D_0)\end{aligned}$$

LEMMA  $\sum_{i=1}^M \hat{c}_i \geq \sum_{i=1}^M c_i$  SSE  $\Phi(D_m) \geq \Phi(D_0)$

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

## STACK CON MULTIPOP (INIZIALMENTE VUOTO)

PONIAMO:  $\Phi(S) \stackrel{\text{def}}{=} |S|$

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

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

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

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

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

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

---

CONTATORE BINARIO CON INCREMENTO  
(INIZIALMENTE NULLO)

$$\phi(D_i) = \# \text{ BIT UGUALI AD 1}$$

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:

$$c_i = \# \text{SET}_i + \# \text{RESET}_i$$

NOTA:  $0 \leq \# \text{SET}_i \leq 1$

$$\Delta\phi_i = \# \text{SET}_i - \# \text{RESET}_i$$

PERTANTO:

$$\begin{aligned}\hat{c}_i &= c_i + \Delta\phi_i = (\# \text{SET}_i + \# \text{RESET}_i) \\ &\quad + (\# \text{SET}_i - \# \text{RESET}_i) \\ &= 2 \cdot \# \text{SET}_i \leq 2\end{aligned}$$

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

CONTATORE BINARIO CON INCREMENTO INIZIALMENTE  
NON NULLO

PONIAMO COME PRIMA

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

71 HA

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

$$\leq 2n + \Phi(D_0) \leq 2n + k \quad (\text{DOVE } k \text{ È IL NUMERO DI BIT DEL CONTATORE})$$

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:

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

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

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

(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.  $2 \cdot \text{size}[T]$
- si copi Table[T] in new-table
- si deallochi Table[T]

Table[T] := new-table

size[T] :=  $2 \times \text{size}[T]$

- 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

INSERIMENTO	$C_i$	
	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
...	...	...

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

SE  $i-1$  È  
UNA POTENZA  
DI 2

ALTRIMENTI

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

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

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

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

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

$$= 3n - 3 = \mathcal{O}(n)$$

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

## ANALISI CON IL METODO DEGLI ACCANTONAMENTI

$$\hat{c}_{ins} = 3 \quad \left\{ \begin{array}{l} 1 \text{ UNITA' PER IL COSTO REALE} \\ 1 \text{ UNITA' PER PAGARE IL COSTO DELLA COPIA} \\ 1 \text{ UNITA' PER PAGARE IL COSTO DELLA COPIA} \\ \text{DI UN ELEMENTO GIÀ RICOPiato} \end{array} \right.$$

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

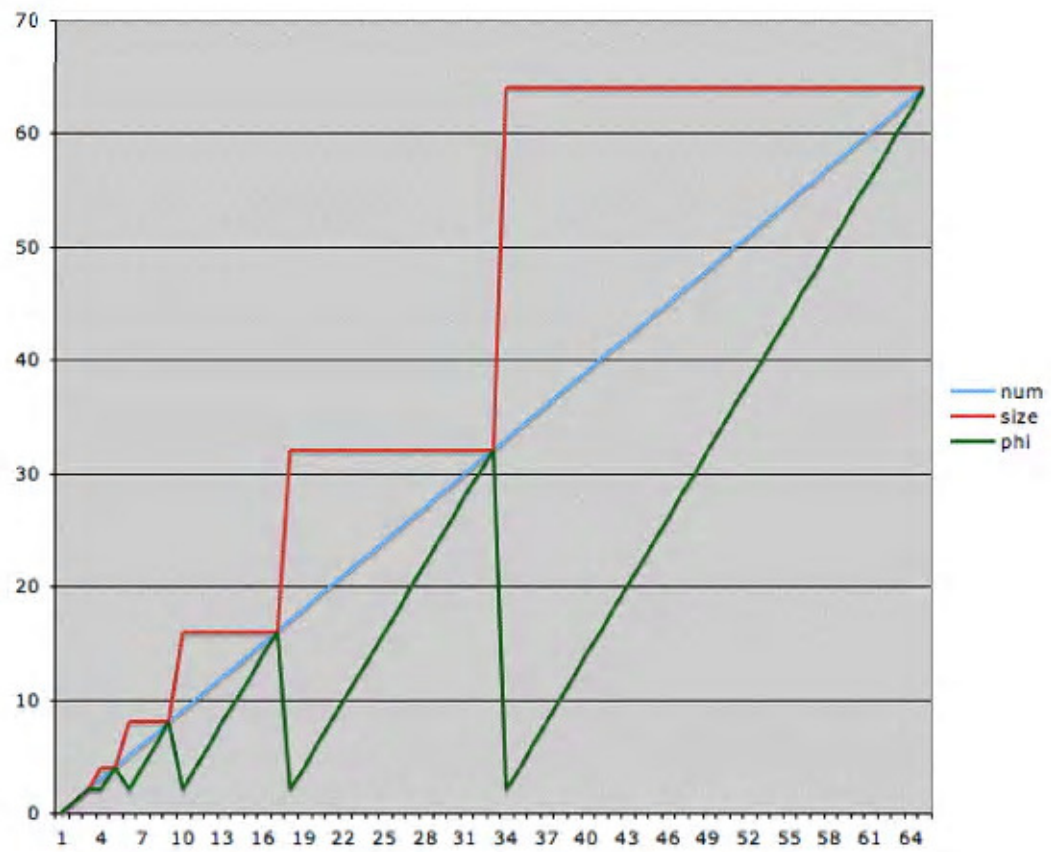
$$\Phi(T) = 2 \cdot \text{num}[T] - \text{size}[T]$$

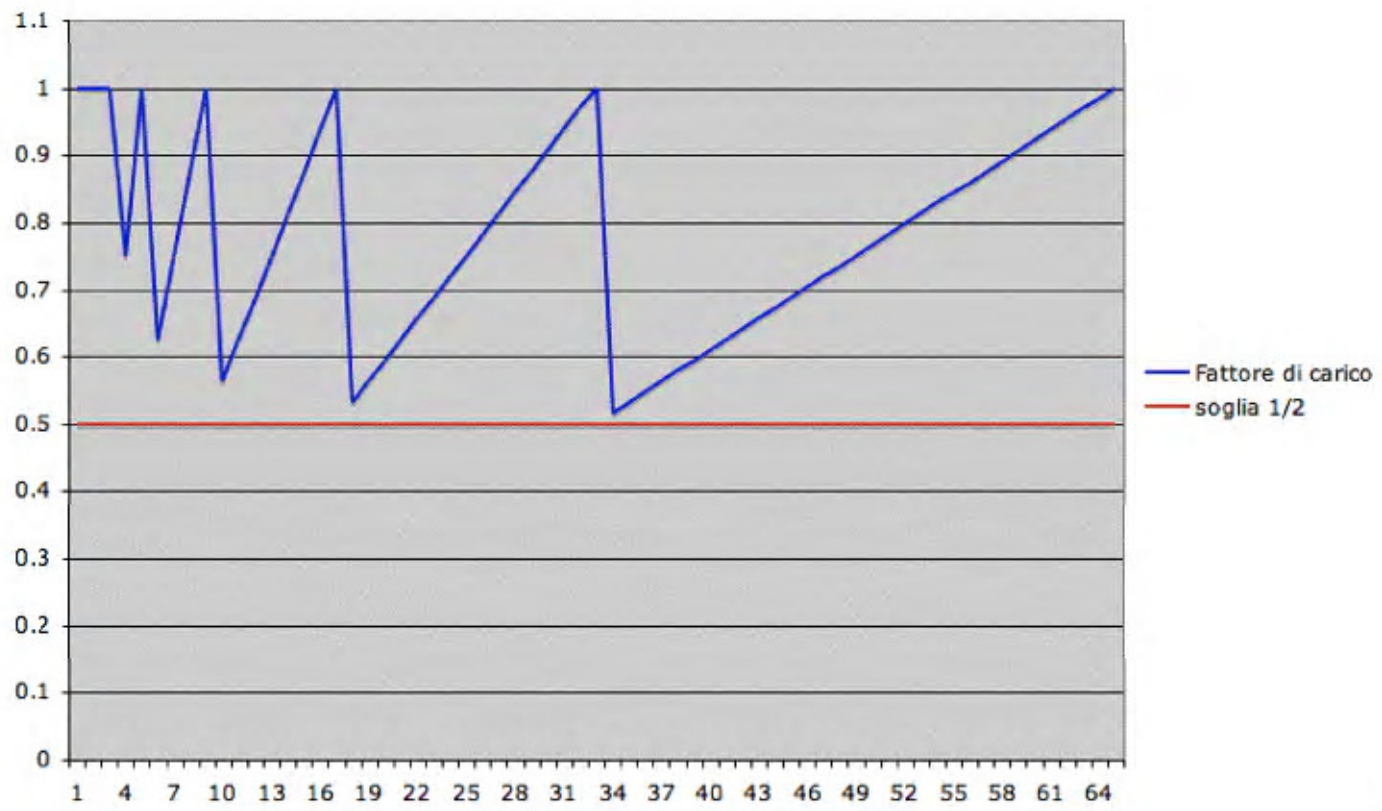
SI HA:  $\Phi(T_0) = 0$  ( $T_0$  TABELLA VUOTA)

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

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

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







INSERIMENTO SENZA ESPANSIONE

$$\begin{aligned}\hat{c}_{ins} &= c_{ins} + \Phi(T_i) - \Phi(T_{i-1}) \\ &= 1 + (2 \cdot n_i - s_i) - (2n_{i-1} - s_{i-1}) \\ &= 1 + 2(\cancel{n_{i-1}} + 1) - \cancel{s_{i-1}} - 2\cancel{n_{i-1}} + s_{i-1} \\ &= 3\end{aligned}$$

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

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

INSERIMENTO CON ESPANSIONE

$$\begin{aligned}\hat{c}_{ins} &= c_{ins} + \Phi(T_i) - \Phi(T_{i-1}) \\ &= n_i + (2 \cdot n_i - s_i) - (2n_{i-1} - s_{i-1}) \\ &= \cancel{n_{i-1}} + 1 + 2(\cancel{n_{i-1}} + 1) - 2\cancel{n_{i-1}} - 2\cancel{n_{i-1}} + \cancel{n_{i-1}} \\ &= 3\end{aligned}$$

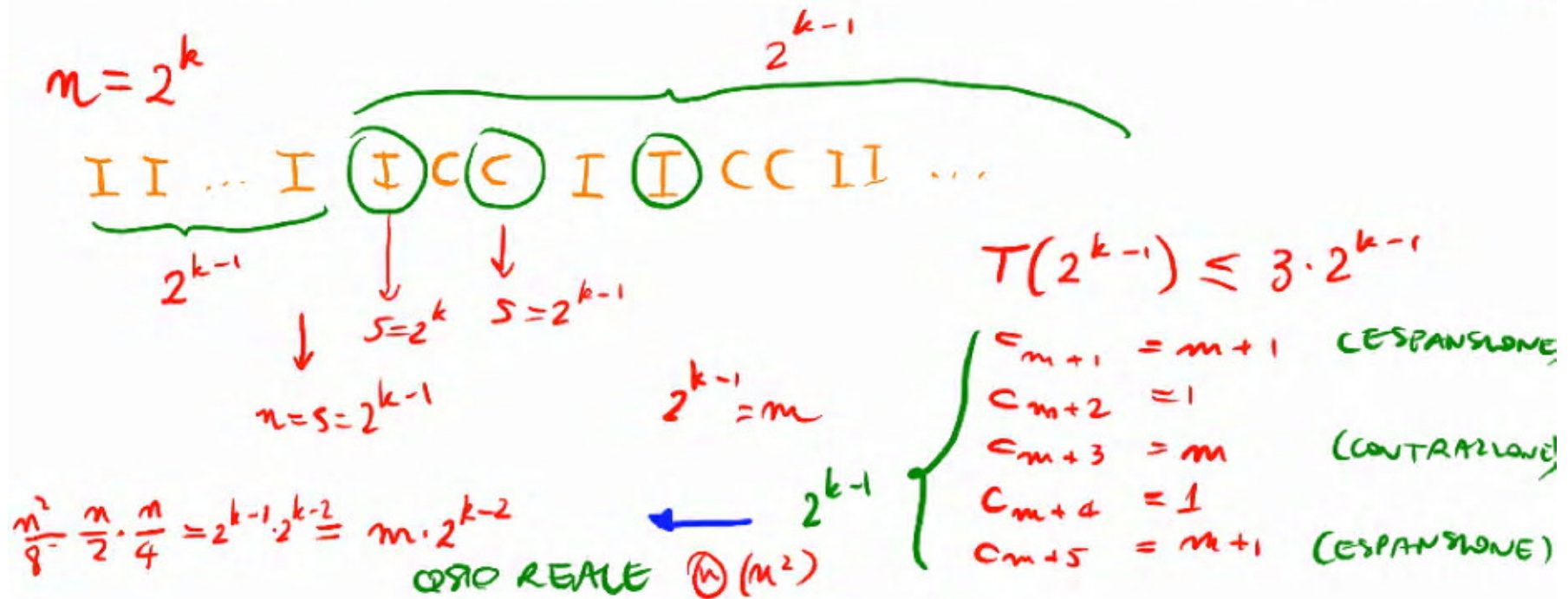
$$s_i = 2s_{i-1} = 2n_{i-1}$$

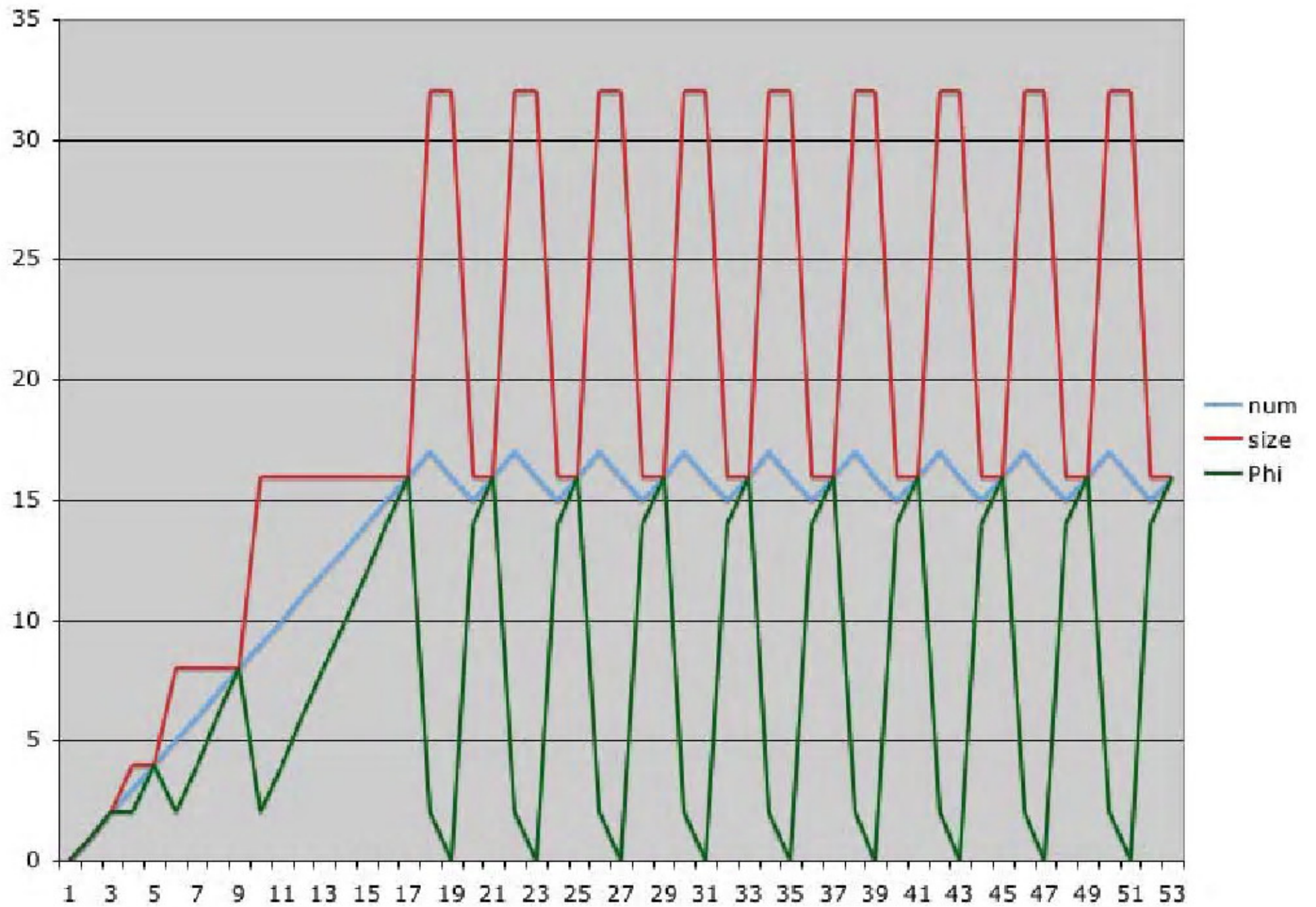
$$n_i = n_{i-1} + 1$$

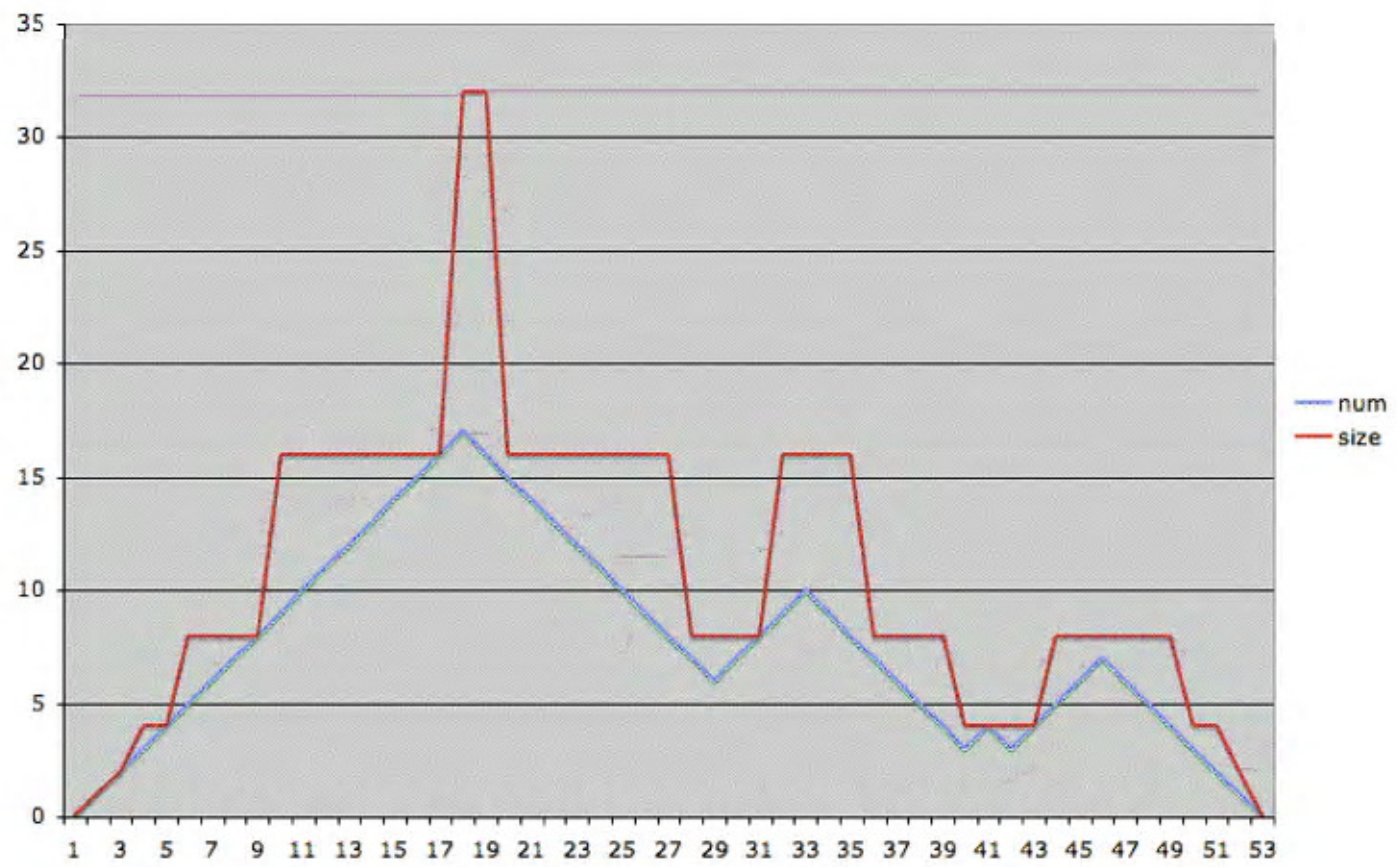
PERTANTO  $T(n) \leq 3n$

# TABELLE DINAMICHE CON INSERIMENTI E CANCELLAZIONI

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







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

PONIAMO:

$$\Phi(T) = \begin{cases} 2 \text{ num}(T) - \text{size}(T) & \text{SE } \alpha(T) \geq \frac{1}{2} \\ \frac{1}{2} \text{ size}(T) - \text{num}(T) & \text{SE } \alpha(T) < \frac{1}{2} \end{cases}$$

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

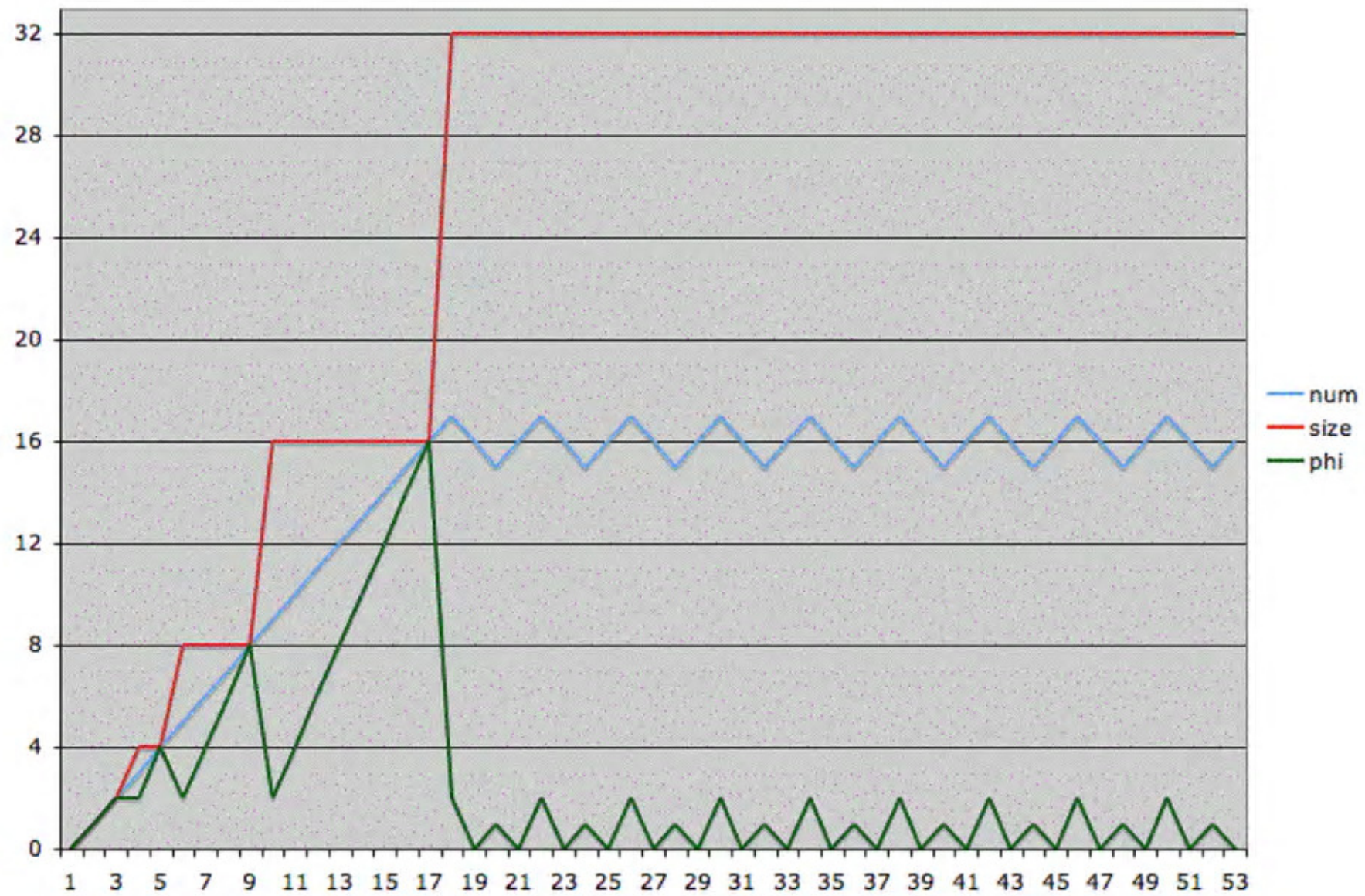
-  $\frac{\text{num}(T)}{\text{size}(T)} = \frac{1}{2}$ , CIOE'  $\frac{1}{2} \text{ size}(T) - \text{num}(T) = 0$

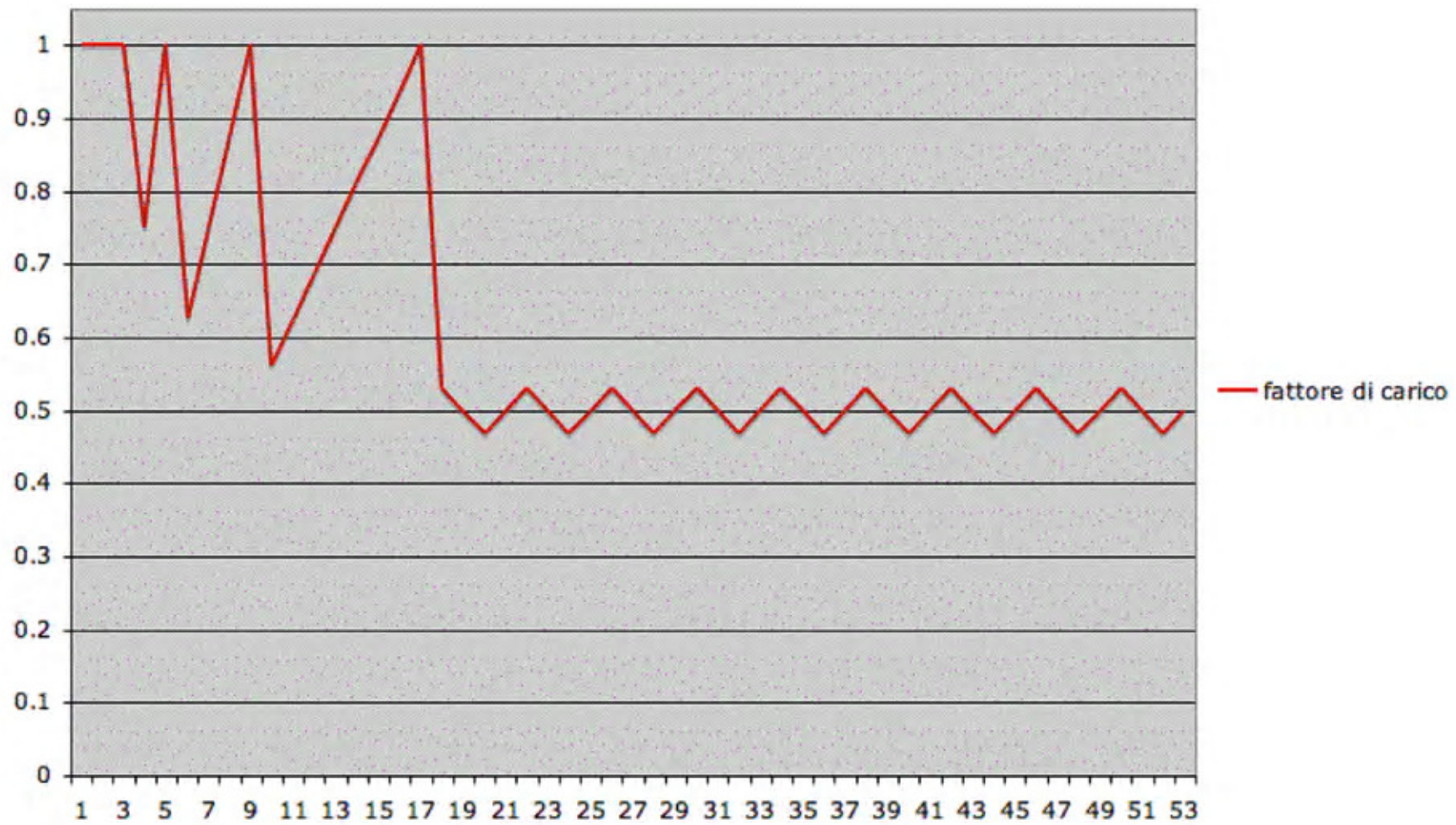
-  $\Phi(T) = 2 \text{ num}(T) - \text{size}(T) = 0$  . PERTANTO

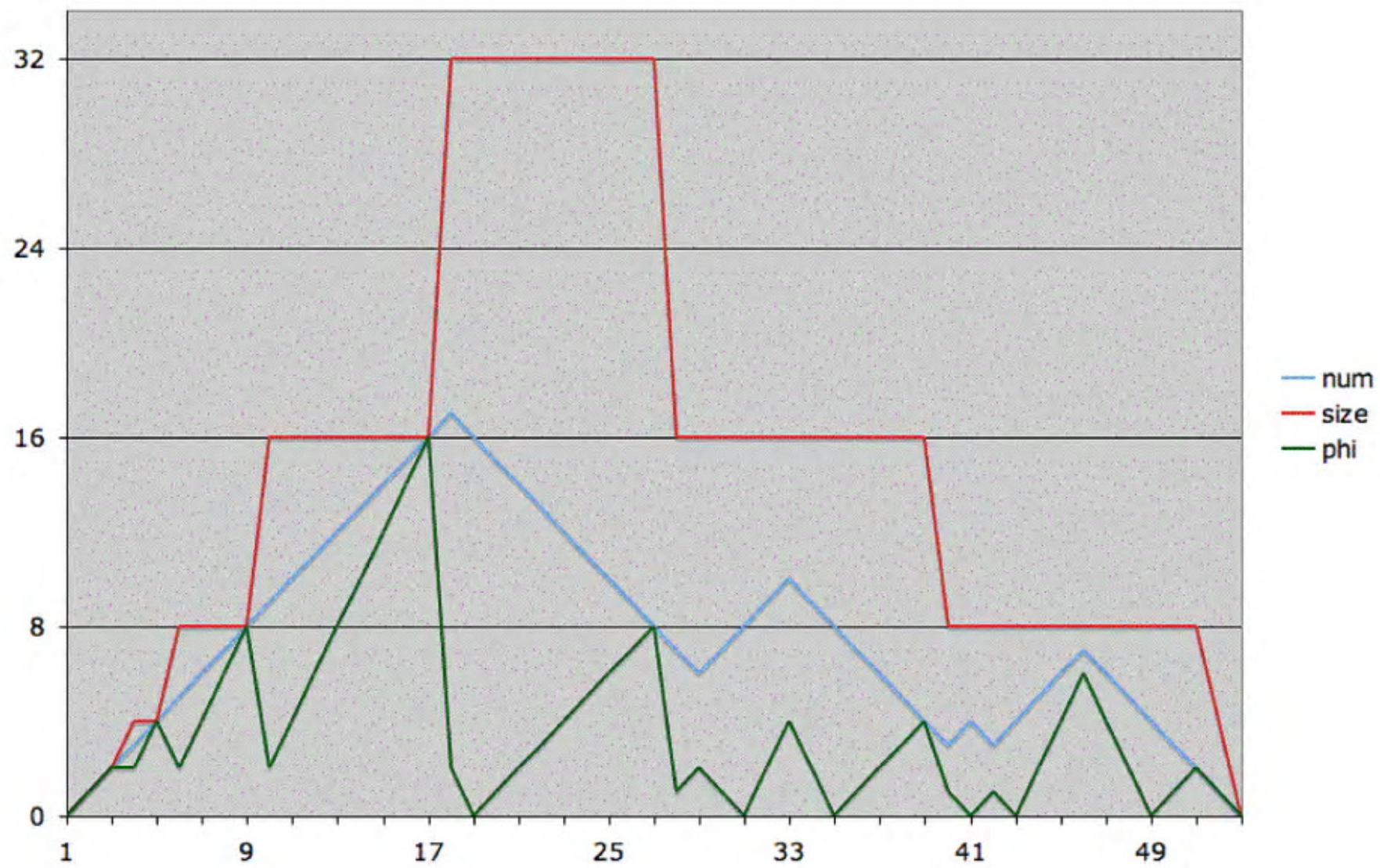
$$\Phi(T) = \begin{cases} 2 \text{ num}(T) - \text{size}(T) & \text{SE } \alpha(T) \geq \frac{1}{2} \\ \frac{1}{2} \text{ size}(T) - \text{num}(T) & \text{SE } \alpha(T) \leq \frac{1}{2} \end{cases}$$

INOLTRE VALE SEMPRE

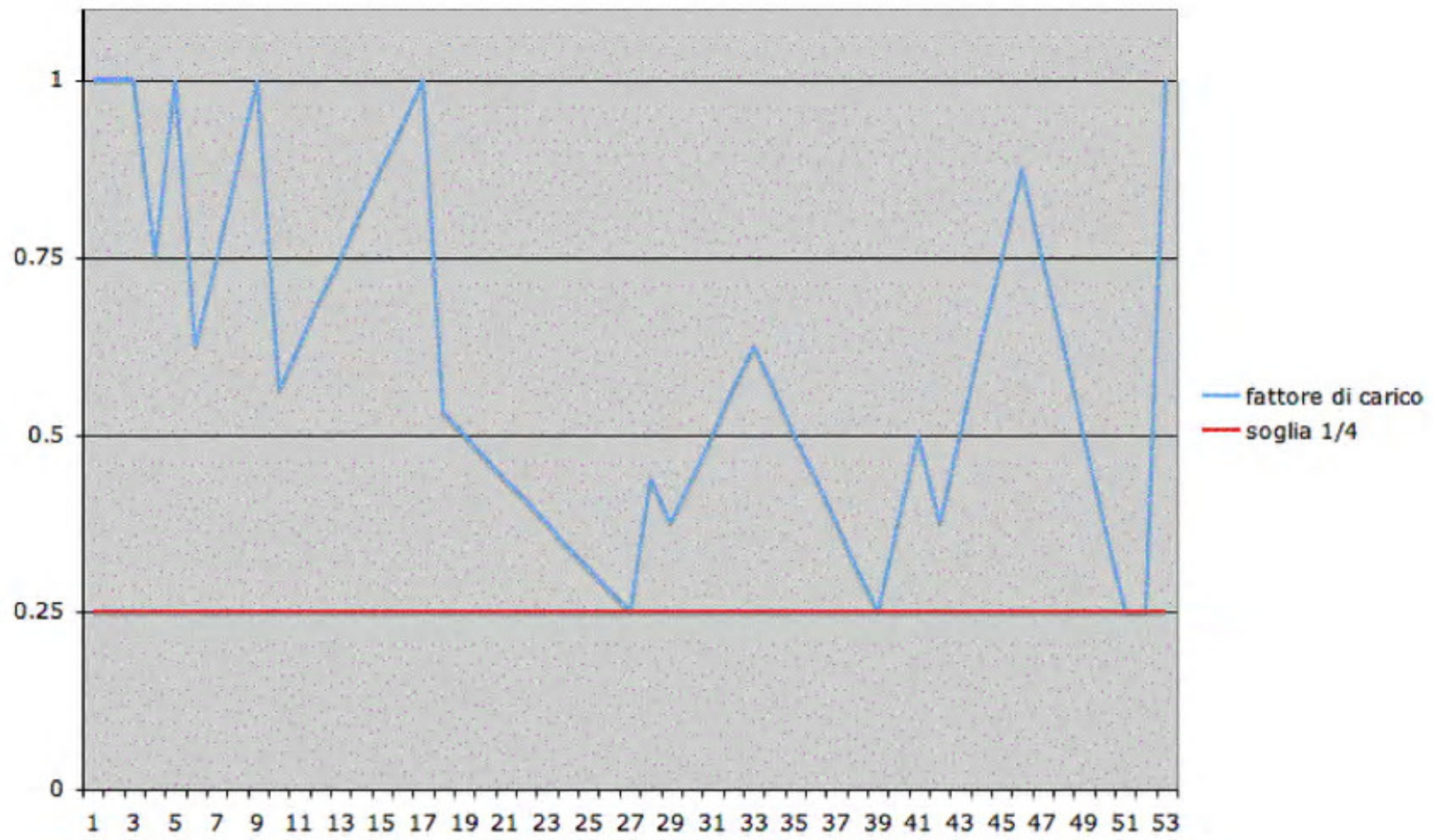
$$\Phi(T) \geq \Phi(T_0) = 0 \quad (T_0 \text{ TABELLA VUOTA})$$











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

$$\begin{aligned}\hat{c}_{ins} &= c_{ins} + \Phi(T_i) - \Phi(T_{i-1}) \\ &= 1 + \left(\frac{1}{2}s_i - n_i\right) - \left(\frac{1}{2}s_{i-1} - n_{i-1}\right) \\ &= \cancel{1} + \cancel{\frac{1}{2}s_{i-1}} - \cancel{n_{i-1}} - \cancel{1} - \cancel{\frac{1}{2}s_{i-1}} + \cancel{n_{i-1}} \\ &= 0\end{aligned}$$

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

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

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

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

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

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

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

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

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

$$\begin{aligned} s_i &= \frac{s_{i-1}}{2} \\ n_i &= n_{i-1} - 1 \\ n_{i-1} &= \frac{1}{4} s_{i-1} \\ n_i &= \frac{1}{4} s_{i-1} - 1 \end{aligned}$$

PERTANTO

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

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

DA CUI

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

## ESERCIZI

- 1) SIA  $\phi$  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  $\phi'$  TALE CHE  $\phi'(D_0) = 0$ ,  $\phi'(D_i) \geq 0$  PER OGNI  $i \geq 1$ , E CHE I COSTI AMMORTIZZATI CON  $\phi'$  SIANO UGUALI AI COSTI AMMORTIZZATI CON  $\phi$ .
- 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  $m$  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  $m$  ELEMENTI CHE CONSENTE DI ESEGUIRE LE ISTRUZIONI INSERT E EXTRACT-MIN NEL TEMPO  $O(\lg m)$  NEL CASO PEGGIORE.

SI DEFINISCA UN POTENZIALE  $\phi$  TALE CHE IL COSTO AMMORTIZZATO DI INSERT SIA  $O(\lg m)$  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  $m$  OPERAZIONI INCREMENT E'  $O(m)$  SE  $m = \Omega(b)$ .

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  $c_{\text{PUSH}} = 1$

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

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

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

$$\phi(S) = |S|$$

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

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

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

PROBLEMA!



$$c_{\text{PUSH}} = 1$$

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

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

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

$$\phi(S) = k |S| \quad / \quad \phi(S) \geq \phi(S_0) = 0$$

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

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

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