ES 11

STRUTTURA S

NEW (S)

INS (S, x)

HALF (S)

CREARE UNA STRUTTURA DATI che abbia complessità in logn.

NEW (S)
 CREA MIN HEAP S

USO MIN HEAP

INS (S, x)
 HEAP INSERT (S, x)

HALF (S)
 $n = S.HEAPSIZE$
 FOR $i = 1$ to $\lceil k/2 \rceil$
 EXTRACT-MIN (S)

Appunti parziali presi durante la correzione del 4° appello. Non garantisco la correttezza di quanto riportato, anzi vi invito di usarli solo come spunto per formulare una soluzione più chiara e comprensibile. Fava

Con Analisi ammortizzata dimostrare che sequenza di operazioni costa logn.

- PER TOGLIERE DEGLI ELEMENTI DEVO INSERIRLI PRIMA

$Ins(S, x) = 1 + \log|S| \rightarrow$ DIMENSIONE (NUMERO ELEMENTI)

↳ PAGO IN ANTICIPO il COSTO DELL' ESTRAZIONE.

↳ Perciò avere log o 2log non cambia la complessità.

$\Phi(S) = \sum_{j=1}^{|S|} \log j$

↳ FUNZIONE POTENZIALE

	COSTO (VERO)	$\Delta\Phi$ (POTENZIALE)	$\hat{C}$ (COSTO AMMORTIZZATO)
NEW	1	0	1
INS (S, x)	$1 + \log\lvert S\rvert$	$\log\lvert S\rvert$	$1 + 2\log\lvert S\rvert$
EXTRACT (S)	$1 + \log\lvert S\rvert$	$-\log\lvert S\rvert$	1
HALF (S)	$1 + \sum_{j=k/2}^{\lvert S\rvert} \log j$	$\sum_{j=k/2}^{\lvert S\rvert} \log j$	1

$\Phi(S) = 0$ ~~[illegible]~~ S VUOTA

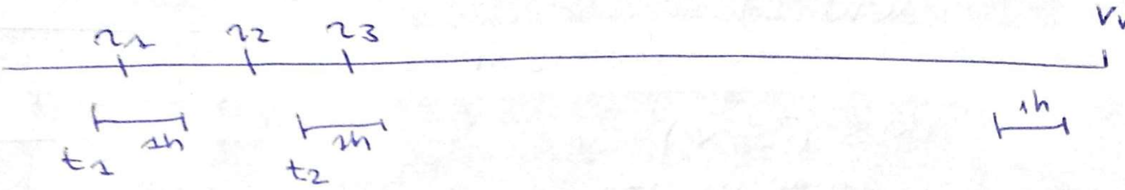
$\Phi(S) \geq 0 \quad \forall S$

M OPERAZIONI

$O(1 + 2\log|S|) = O(\log n)$

ES 2. → richieste temporizzate, con dei turni di un'ora

$r_1 \dots r_n$



- Prima cosa che viene in mente, primo turno a r_1 e poi la prossima faccio cominciare.

↳ Richieste non sono ordinate, se lo si vuole si ordinano con costo $n \log n$

FORMALIZZARE la soluzione per il problema

$\vec{r} = r_1 \dots r_n$

soluzione è $\vec{t} = t_1 \dots t_k$

$\forall i = 1, \dots, n \ \exists j$ t.c. $t_j \le r_i \le t_j + 1h$

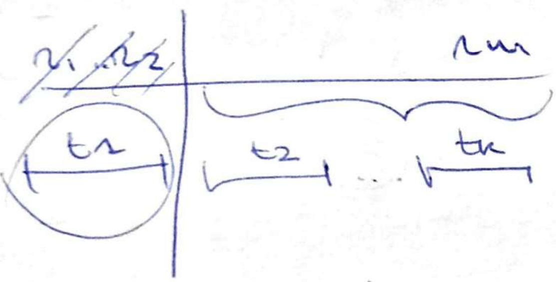
Costo$(t) = k$

SOTTOSTRUTTURA OTTIMA

sia $\vec{t}$ soluzione ottima per $\vec{r}$

$\vec{r'} = r_j \dots r_n$

t.c. $1 \le i \le j-1 \Rightarrow r_i \in [t_1, t_1 + 1h]$



allora $\vec{t'} = t_2 \dots t_k$ soluzione ottima per $\vec{r'}$

SCELTA GOLOSA: Primo turno comincia alla prima richiesta.

↳ Esiste sempre una soluzione ottima a una scelta golosa...

SCELTA GREEDY $t_1 = r_1$

Sia $\vec{t}$ ottima per $\vec{r} = r_1 \dots r_n$

$t_1' \dots t_k'$

$t_1' \le r_1$

$r_1 \quad r_2 \quad \dots \quad r_n$

$t_1' \quad t_i' \quad t_k'$

ARABO

→ soluzione $\vec{t} = t_1\, t_2' \dots t_k'$ è soluzione per $\vec{r}$ (e ottima)

⇒ Se r_J soddisfatto da t_1' → $r_J \in [t_1', t_1' + 1h]$

$t_1' \leq r_J \leq t_1' + 1$

↳ $r_1 \leq r_J \leq t_1' + 1 \leq r_1 + 1$

$t_1 \leq r_J \leq t_1 + 1$

```
TIME (r, n)
    t[1] = r[1]
    turni = 1
    FOR i=2 to n
        iF t[TURN]+1 ≤ r[i]
            TURNI++
            t[TURNI] = r[i]
    RETURN
```