ADT (Abstract Data Types)

$\boxed{\text{ADT}}$: Stack<T>

primitive

- boolean isEmpty()
- T top()
- void pop()
- void push(T x)

- $\forall S$

  S.push(x).top() == x

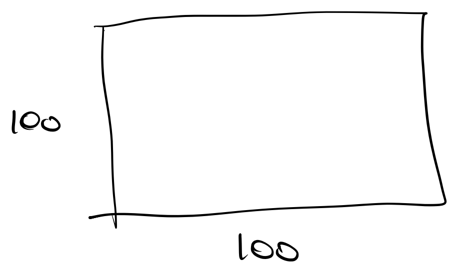
- $\forall S$

  S.isEmpty() == S.push(x).pop().isEmpty()

ADT $\longrightarrow$ IMPLEMENTAZIONI

INFORMATION-THEORETICAL LOWER BOUND

- Per codificare v valori servono in media $\log_2 v$ bit.



$2^{10\,000}$ $\left\{ \log_2 2^{10000} = 10000 \right.$

- $x_1, x_2, \dots, x_v$

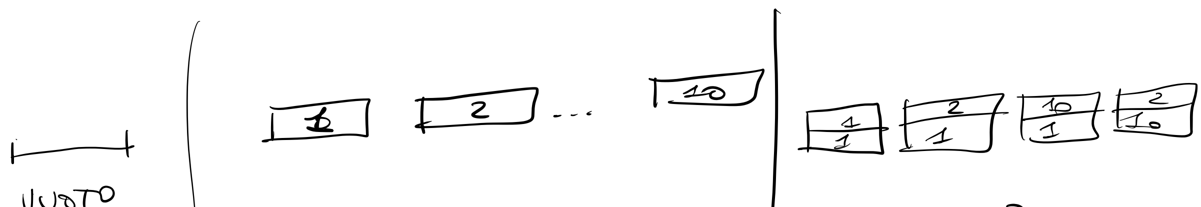
x_i = # bit necessari per rappresentare l'i-esimo valore

$$\geq \frac{x_1 + x_2 + \dots + x_v}{v} \geq \boxed{\log_2 v}$$

$[\log_2 v, 1 + \log_2 v)$

- Immaginate un ADT con v_n VALORI DI TAGLIA n

- STACK <{1, ..., 10}>



$v_0 = 1$ | $v_1 = 10$ | $v_2 = 10^2$

$$v_n = 10^n \to \log 10^n = n \log 10 \approx \boxed{4n}$$

$$\boxed{Z_n = (\log_2 10)^n \text{ bit}}$$

INF. THEORETICAL LOWER BOUND

$D_n \geq Z_n$ IMPLEMENTAZIONE CHE USA D_n BIT

Un'implementazione dell'ADT è chiamata

proprietà eff. care su str. "naturale"

1) implicita se $D_n = Z_n + O(1)$
2) succinte se $D_n = Z_n + o(Z_n)$
3) compatte se $D_n = O(Z_n)$

1) $Z_n + 3$ 2) $Z_n + \underbrace{\log Z_n}$ 3) $Z_n + \sqrt{Z_n}$

RANGO E SELEZIONE

ADT definiti da $\underline{b} \in 2^n$

$$\left.\begin{array}{l} rank_{\underline{b}} \\ select_{\underline{b}} \end{array}\right\} : \mathbb{N} \longrightarrow \mathbb{N}$$

1) $\forall p \leq n$

$$rank_{\underline{b}}(p) = |\{i \mid i < p \text{ e } b_i = 1\}|$$

2) $\forall k \leq$

$$select_{\underline{b}}(k) = \max\{p \mid rank_{\underline{b}}(p) \leq k\}$$

	0	1	2	3	4	5	6
$\underline{b} =$	0	1	1	0	1	0	1

p	$rank_{\underline{b}}(p)$
0	0
1	0
2	1
3	2
4	2
5	3
6	3
7	4

k	$sel_{\underline{b}}(k)$
0	1
1	2
2	4
3	6
4	7

a) $\forall k \quad rank(select(k)) = k$

b) $\forall p \quad select(rank(p)) \geq p$

è $=$ sse $b_p = 1$

PERMETTE DI DEDURRE $\underline{b}$

$n \log n$ ← TAB. RANK

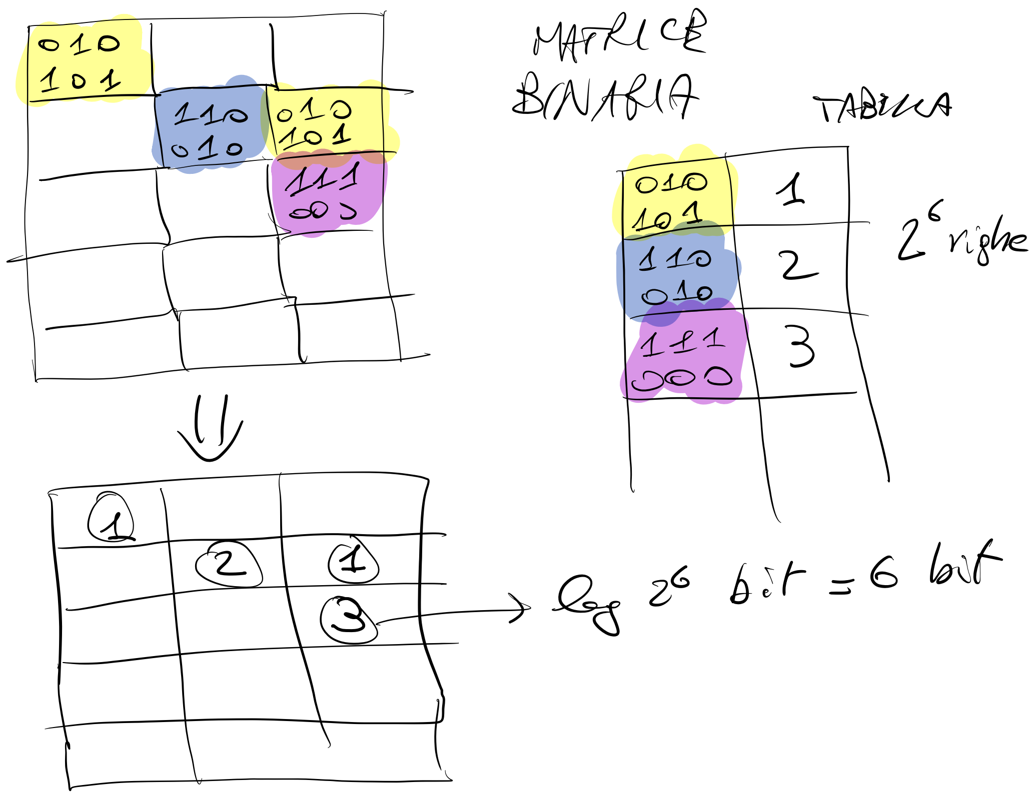
$n \log n$ ← TAB. SELECT

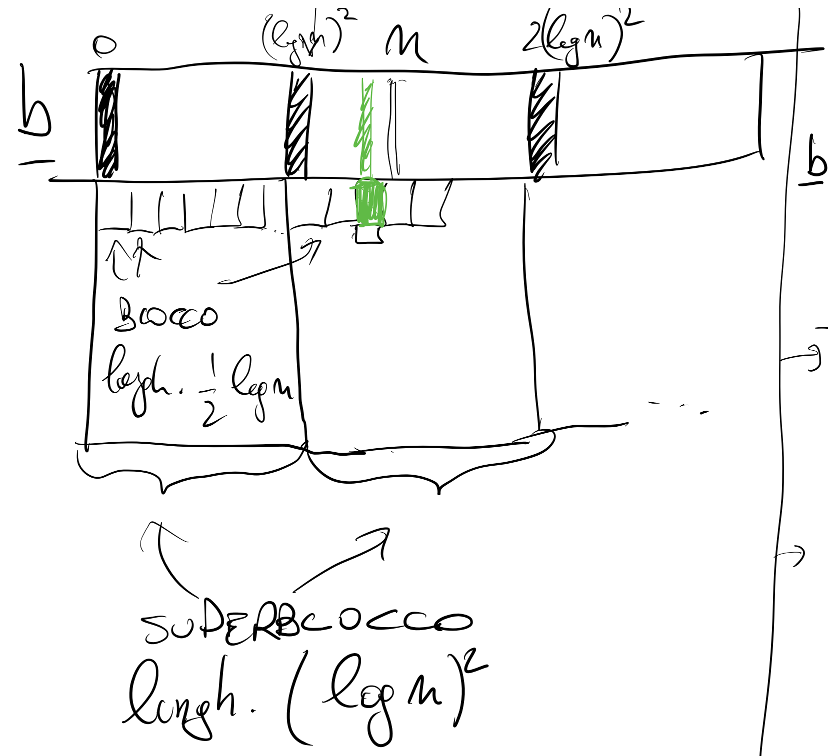
$$D_n = 2n \log n$$

$$Z_n = n$$

STRUTTURA DI JACOBSON PER IL RANGO

$\boxed{\text{IDEA}}$ Usa il "Four-Russians trick"!





0	1	2	3	4	5	6

$\underline{b} =$

0	1	1	0	1	0	1

p	$\mathrm{rank}_{\underline{b}}(p)$
0	0
1	0
2	1
3	2
4	2
5	3
6	3
7	4

...

① Per ogni superblocco i

S_i = rango dell'elemento iniziale del superblocco

② Per ogni blocco ℓ memorizzo

B_ℓ = differenza fra il rango del suo el. iniziale e il rango dell'elemento iniziale del superbl.

$\boxed{1}$ TABELLA S_i

$$\underbrace{\frac{n}{(\log n)^2}}_{\text{n° di superblocchi}} \log n = \frac{n}{\log n} = \underline{o(n)}$$

$\boxed{2}$ TABELLA B_e

$$\underbrace{\frac{n}{\frac{1}{2}\log n}}_{\text{n° di blocchi}} \log\left[(\log n)^2\right] =$$

$$= \frac{2n}{\log n} 2 \log\log n =$$

$$= \underline{o(n)}$$

TIPI DI BLOCCHI (four-Russians trick)

$$2^{\frac{1}{2}\log n} = 2^{\log \sqrt{n}} = \sqrt{n}$$

TIPO DI BLOCCO

$$\sqrt{n}\left\{\begin{array}{l} 0000 \\ 0001 \\ 0010 \\ 0011 \\ \vdots \\ 1111 \end{array}\right.$$

rank

$$\underbrace{\frac{1}{2}\log n}_{\text{righe}} \log\left(\frac{1}{2}\log n\right)$$

$$\sqrt{n}\ \frac{1}{2}\log n \log\left(\frac{1}{2}\log n\right) \le$$

$$\le \sqrt{n}\ \log n\ \log\log n = o(n)$$

STRUTTURA OCCUPA $Dn = o(n)$

$$\mathrm{rank}_{\underline{b}}(p) = S_{\frac{p}{(\log n)^2}} +$$

$$+ \quad B_{\frac{p}{\frac{1}{2}\log n}} +$$

$$+ \quad TAB_t\left[p \ \underline{\bmod} \ \tfrac{1}{2}\log n\right]$$

t è il blocco cui p appartiene

$$x = \left\lfloor \frac{p}{\frac{1}{2}\log n} \right\rfloor \left(\tfrac{1}{2}\log n\right)$$

$$t = \underbrace{\underline{b}\left[x, x+1, \ldots, x+\tfrac{1}{2}\log n - 1\right]}$$

Occupa spazio

- $\underbrace{n}_{\text{b}} + \underbrace{O(n)}_{\text{S, B, FAB}}$ SUCCINTA

- $O(1)$

STRUTTURA PER SELECT DI CLARKE

$\boxed{\text{I LIVELLO}}$

P_i = select per la posizione $i \underbrace{\log n \log\log n}$

$b = $

0	1	2	3	4	5	6
0	1	1	0	1	0	1

k	$sel_b(k)$
0	1
1	2
2	4
3	6
4	7
⋮	⋮

$7 = \log n \log\log n$

P_i

P_{i+1}

t

$t = \#$ di uni

MEMORIA:

$$\underbrace{\frac{t}{\cancel{\log n}\log\log n}}_{\text{righe della tabella } P} \cancel{\log n} = \frac{t}{\log\log n} \leq$$

$$\leq \frac{n}{\log\log n} = o(n)$$

II LIVELLO

$$r_i \triangleq \underbrace{P_{i+1} - P_i}$$

$$r_i \geq \log n \log\log n$$

$$\textcircled{*}\ r_i \geq (\log n \log\log n)^2$$

II A: CASO SPARSO

Memorizzo tutti gli r_i esplicitamente come differenze da P_i

MEMORIA

$$\underbrace{\log n \log\log n}_{\text{righe}} \underbrace{\log r_i}_{\text{n° di bit per scrivere una diff. da } P_i} =$$

$$= \frac{(\log n \log\log n)^2}{\log n \log\log n} \log r_i \leq \textcircled{*}$$

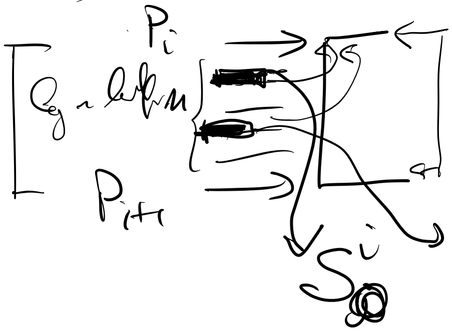
$$\leq \frac{r_i}{\log n \log\log n} \log r_i \leq$$

$$\frac{r_i}{\ } \ \cancel{\log n} =$$

$$\leq \frac{\quad}{\cancel{\log n}\,\log\log n}$$

$$= \frac{r_i}{\log\log n}$$

$\boxed{\text{IIB: CASO DENSO}}$ $\quad r_i < (\log n \log\log n)^2$



Memorizzo esplicitamente solo le entry multiple di $\log r_i \log\log n$

$S_1^i \ldots \underbrace{\log r_i \log\log n}$

MEMORIA: $(r_i \geq \log n \log\log n)$

$$\frac{\log n \log\log n}{\cancel{\log r_i}\,\log\log n}\,\cancel{\log r_i} \leq$$

$\leq r_i$

$$\leq \frac{r_i}{\log\log n}$$

$$\bar{r}_j^i = S_{j+1}^i - S_j^i$$

PROPR.

$$\bar{r}_j^i \geq \log r_i \log\log n \quad (1)$$

$$r_i < (\log n \log\log n)^2 \quad (2)$$

$\underline{\text{III}}$ A : CASO SPARSO

$$\rightarrow \bar{r}_j^i \geq \log \bar{r}_j^i \log r_i (\log\log n)^2$$

Memorizzo esplicitamente le entry fra S_j^i e S_{j+1}^i

MEMORIA

$$\underbrace{(\log r_i \log\log n)}_{n^\circ \text{ righe}} \underbrace{\log \bar{r}_j^i}_{n^\circ \text{ bit per riga}} =$$

$$= \frac{\boxed{\log r_i (\log\log n)^2 \log \bar{r}_j^i}^{\leq \bar{r}_j^i}}{\log\log n} \leq$$

$$\leq \frac{\bar{r}_j^i}{\log\log n} = O(n)$$

III B: CASO DENSO

(**) $\bar{r}^i_j < \log \bar{r}^i_j \log r_i (\log \log n)^2$

$\underbrace{\bar{r}^i_j}$ USIAMO il Four-Russians Trick

Oss.: $\log \bar{r}^i_j \leq \log r_i \leq$ (2)

$\leq \log (\log n \log \log n)^2 =$

$= 2 \log \log n + 2 \log \log \log n \leq$

$\leq 4 \log \log n$

$\Rightarrow \bar{r}^i_j \overset{(**)}{<} \underbrace{\log \bar{r}^i_j}_{\text{Oss. } \leq 4 \log \log n} \underbrace{\log r_i}_{\text{Oss. } \leq 4 \log \log n} (\log \log n)^2 \leq$

$\leq 16 (\log \log n)^4$

$\underbrace{2^{\bar{r}^i_j}} \quad \underbrace{\bar{r}^i_j} \quad \underbrace{\log \bar{r}^i_j} \leq$

N° TABELLE — RIGHE — BIT PER RIGA

$$\leq 2^{16(\log\log n)^4} \cdot 16 \cdot (\log \log n)^4$$

$$\log\left(16(\log\log n)^4\right) =$$

$$= \underbrace{\left[16(\log\log n)^4\right]^2}\ \underbrace{\log\left(16(\log\log n)^4\right)} =$$

$$= o(n)$$

$$\left.\underset{o(n)}{\mathrm{I}} + \overset{o(n)}{\mathrm{II}} + \underset{o(n)}{\mathrm{III}}\right] o(n)$$

$$\mathrm{II} \leq \underbrace{\frac{r_i}{\log\log n}}$$

$$\mathrm{II} \leq \frac{r_0}{\log\log n} + \frac{r_1}{\log\log n} + \ldots + \frac{r_{\frac{n}{\log n \log\log n}}}{\log\log n} =$$

$$= \sum^{\frac{n}{\log n \log\log n}} \frac{P_{i+1} - P_i}{\log\log n} =$$

$i=0$

$$= \frac{\boxed{P_{\frac{n}{\log n \log\log n}} - P_0}}{\log \log n} \le \frac{n}{\log \log n} =$$

$$= o(n)$$

SUCCINTA

SPAZIO: $n + o(n)$

ACCESSO: $O(1)$