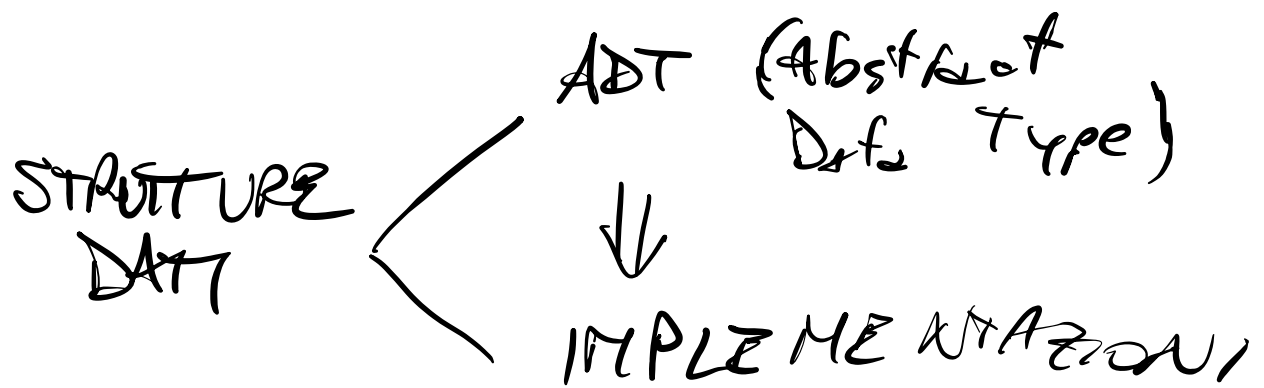




Esempio di ADT

Stack<T>

[	<u>void</u>	push (T x)	
	<u>T</u>	pop()	<u>throws</u> No Such Element Exception
	<u>T</u>	top()	<u>throws</u> No Such Element Exception
	<u>boolean</u>	isEmpty()	

$\forall \alpha \in T$

ASSIOMA

[		Stack<T> s;
		s.push(α)
	assert	s.pop() == α
	assert	s.top() == α

SPAZIO OCCUPATO DA UN'IMPLEMENTAZIONE (INFORMATION - THEORETICAL LOWER BOUND)

ADT con valori che dipendono da una "taglia" n

Per ogni taglia n numero $\sqrt{n}$ ADT
 avrà un certo numero $\sqrt{n}$
 di valori possibili.

Quanti bit servono per rappresentare uno di questi valori?

$V_1, V_2, \dots, V_{\sqrt{n}}$
 $\downarrow \quad \downarrow \quad \dots \quad \downarrow$
 $b_1, b_2, \dots, b_{\sqrt{n}}$

Teorema di Shannon

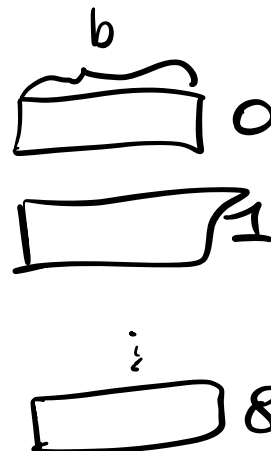
$$\frac{\sum_{i=1}^{\sqrt{n}} b_i}{\sqrt{n}} \geq \log_2 \sqrt{n}$$

information-theoretical lower bound

ESEMPIO

$$\sqrt{3} = 9$$

$$b = 4$$



Possibili valori di tag n

STRUTTURE COMPRESSE

Una implementazione di un ADT
occupa D_n bit e necessariamente

$$D_n \geq Z_n$$

Implementazione

- 1) IMPLICITA
- 2) SUCCINTA
- 3) COMPATTA

se $D_n = Z_n + O(1)$
se $D_n = Z_n + o(Z_n)$
se $D_n = O(Z_n)$

è detto che la velocità di accesso
sia proporzionale a una implementazione
veloce.

STRUTTURA DI RANGO E SELEZIONE

$$\underline{b} \in 2^m$$

$$\text{rank}_{\underline{b}}, \text{select}_{\underline{b}} : \mathbb{N} \rightarrow \mathbb{N}$$

$$1) \forall p \leq m \quad \text{rank}_{\underline{b}}(p) = |\{i \mid i < p \text{ e } b_i = 1\}|$$

$$2) \forall k \leq \text{rank}_{\underline{b}}(m) \quad \text{select}_{\underline{b}}(k) = \max \{p \mid \text{rank}_{\underline{b}}(p) \leq k\}$$

$\underline{b}$

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

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



$m \cdot \log m$
BIT

$m=7$

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

$m \cdot \log m$
BIT

$$2m \log m$$

• $\forall k$

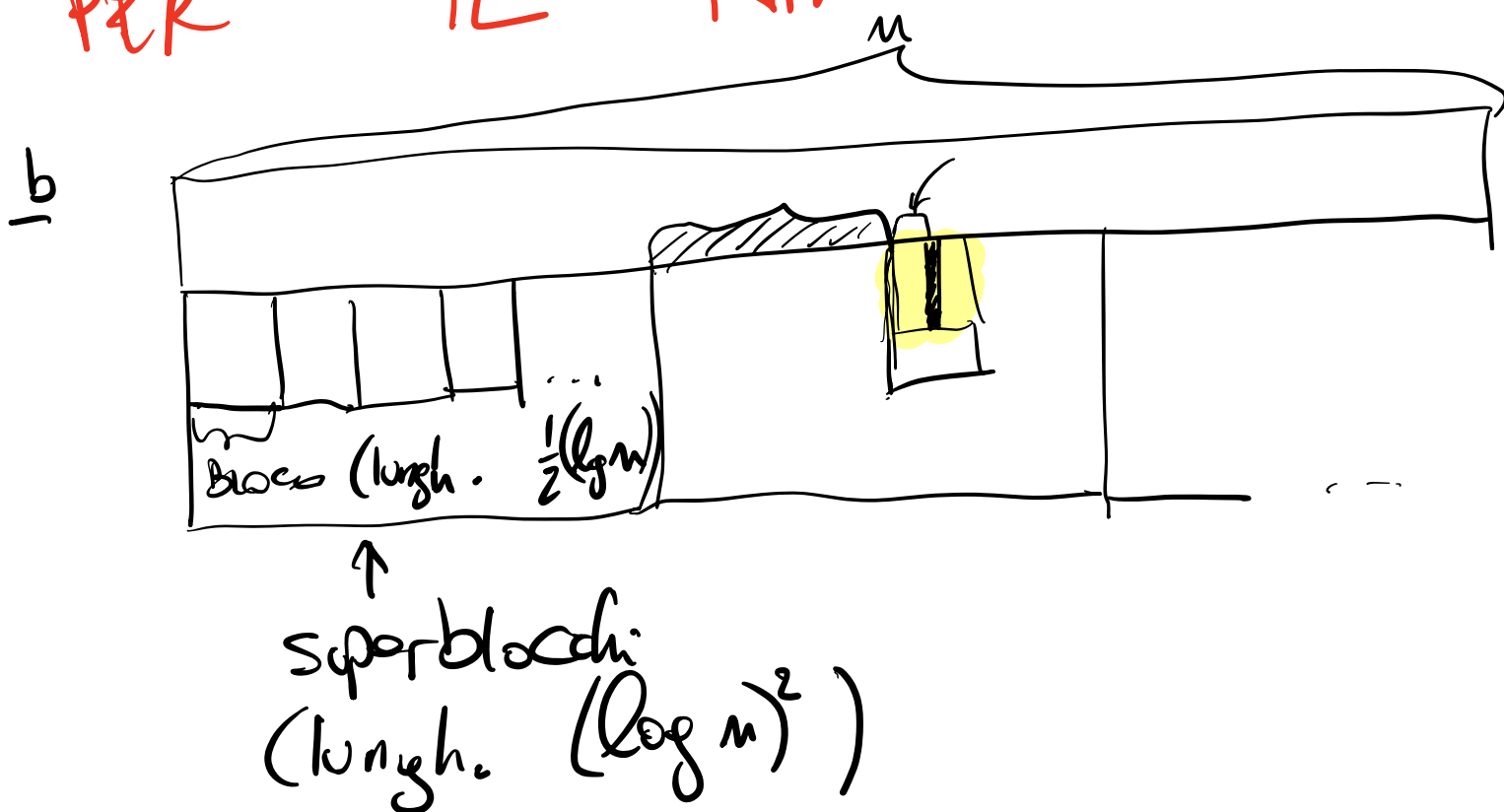
$$\text{rank}(\text{select}(k)) = k$$

• $\forall p$

$$\text{select}(\text{rank}(p)) \geq p$$

$$\uparrow \text{sse } b_p = 1$$

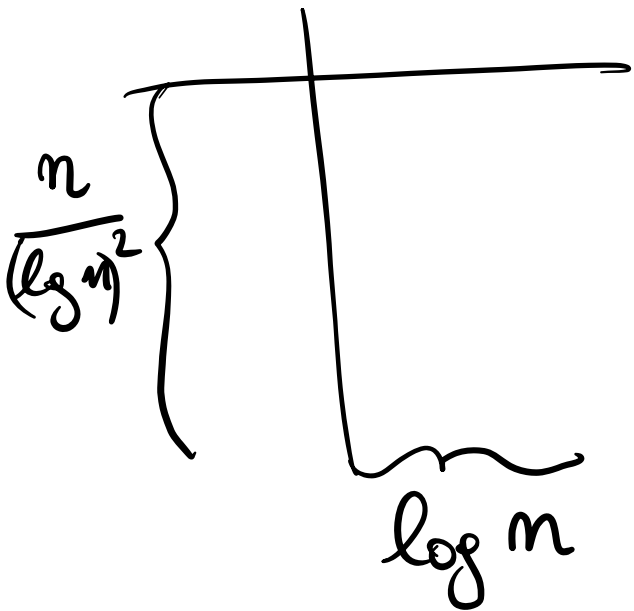
STRUTTURA DI JACOBSON PER IL RANGO



SUPERBLOCCO $\rightarrow$ memorizziamo il numero di 1 prima dell'inizio del super blocco

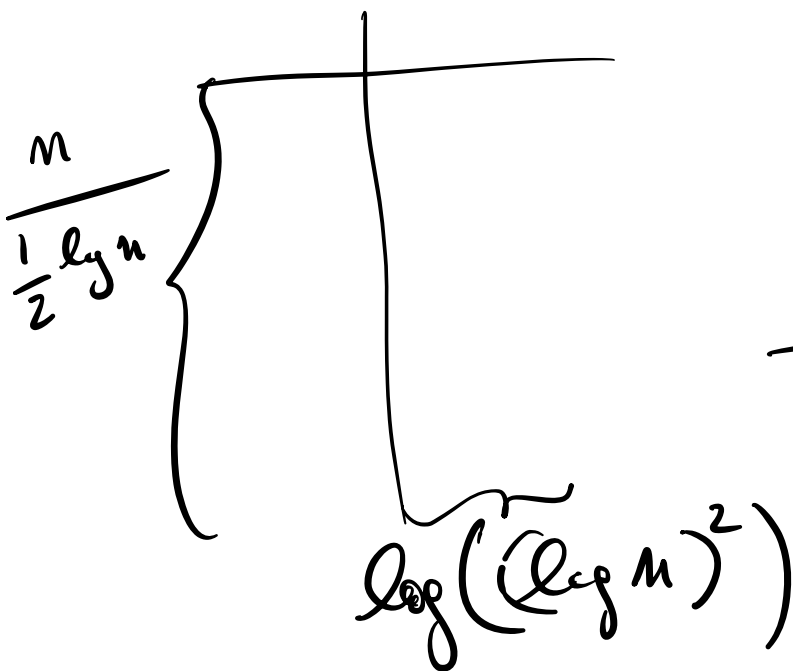
BLOCCO $\rightarrow$ memorizziamo il numero di 1

dall'inizio del
 superblocco fino
 all'inizio del
 blocco (escluso)



SUPERBLOCCO

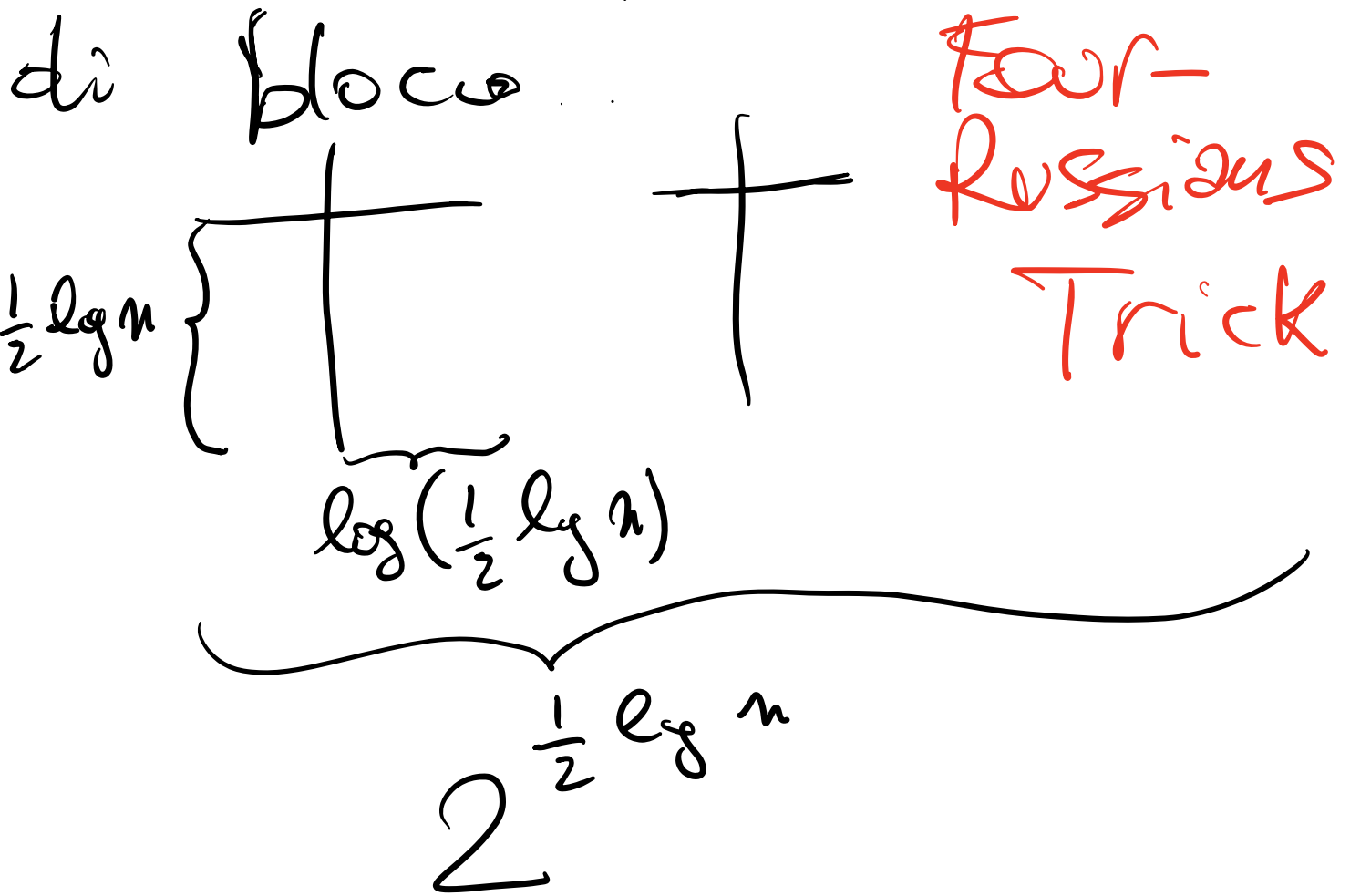
$$\frac{n}{(\log n)^2} \log n = \frac{n}{\log n} = o(n)$$



BLOCCO

$$\frac{n}{\frac{1}{2} \log n} 2 \log \log n = o(n)$$

Costruisco una minitabella
 di rank per ogni tipo



Bit $2^{\frac{1}{2} \lg n} \cdot \frac{1}{2} \lg n \cdot \log(\frac{1}{2} \lg n)$

$$= \sqrt{n} \cdot \frac{1}{2} \lg n \cdot \log \log \sqrt{n} =$$

$$= o(n)$$

SPAZZO COMPRESSIVO

$$D_n = M + o(n) = \underbrace{Z_n}_n + o(Z_n)$$

STRUTTURA STATICA ⁿ SUCCINTA
PER RANGO CON ACCESSO
 $O(1)$