PAROLE DI DYCK

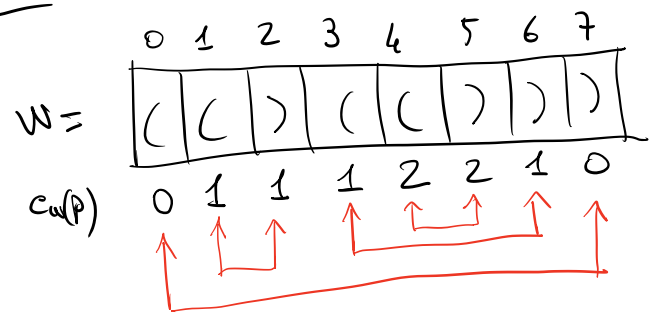
Def. $D \subseteq \{(,)\}^*$ linguaggio di Dyck (Parentesi bilanciate)

$w \in D$ sse

(*) 1) $\#_(w = \#_) w$

2) per ogni prefisso w' di w $\#_(w' \geq \#_) w'$

$n = |w|$



$$e_w(p) = |\{i \mid w_i = '(' \text{ e } i < p\}| - |\{i \mid w_i = ')' \text{ e } i \leq p\}|$$

(*) equivale a

1) $e_w(|w|-1) = 0$

2) $e_w(p) \geq 0 \quad \forall p$

Data $w \in D$, voglio rispondere a

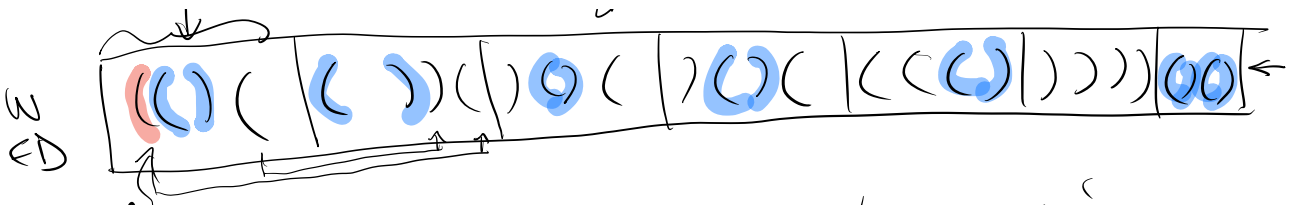
$findClose_w(p)$ = data la posizione p di una parentesi aperta restituisce la posizione $q > p$ della par. chiusa corrispondente

$findOpen_w(p)$ = data la posizione p di una parentesi chiusa restituisce la posizione $q < p$ della par. aperta corrispondente

$enclose_w(p)$ = restituisce la posizione della prima par. aperta $< p$ che racchiude p (e la corrispondente)

$\ell = \log n$

$\#Blocchi = k = \left\lceil \frac{n}{\ell} \right\rceil$

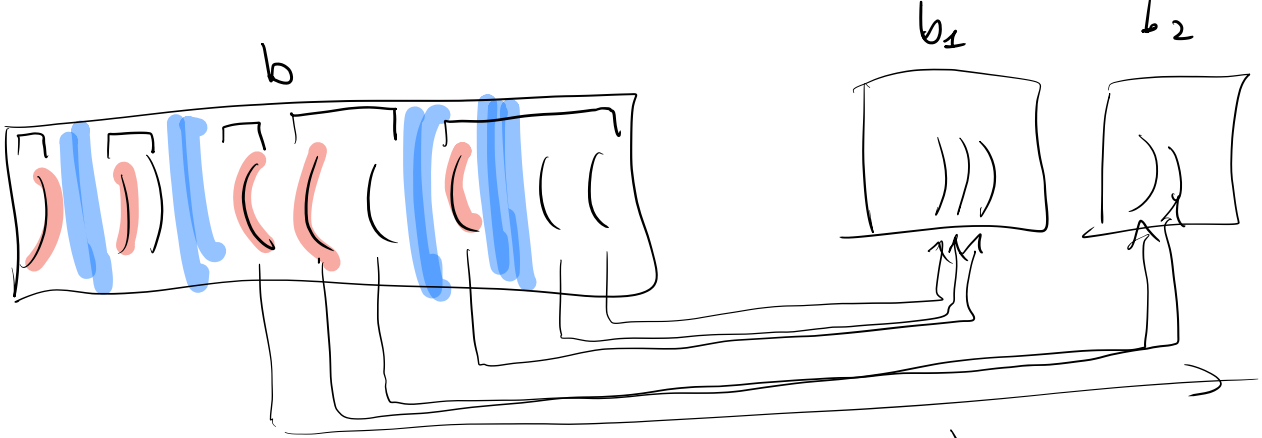


$w \in D$

$|w| = n$

$()$ → parentesi vicine: si chiudono nello stesso blocco in cui si aprono

piniera:
- è una parentesi lontana (la prima del blocco, o
- è la prima a chiudersi in un certo blocco



MEMORIZZIAMO (dove $2 \quad w$)

1) P vettore di n bit con 1 nelle posizioni delle piniere

2) $E[i]$ accesso all'inizio del blocco i

3) $M[i]$ posizione della parentesi corrispondente all'i-esima piniera

4) O[i] di ogni blocco i la posizione della prima copia a sx dell'inizio di i vale sempre $x-1$ dove x è il minimo degli elementi del blocco

---

SPAZIO

W

(1) P

(2) E[]

(4) O[]

(3) M[]

n

$n + \overbrace{O(n)}^{\text{RANK}}$

$k \log n$

$k \log n$

pion. $\log n$
$\leq (4k-6) \log n$
$\leq 4k \log n$

Teorema: Se ci sono k blocchi ci sono al massimo $2k-3$ coppie di pionieri

SPAZIO

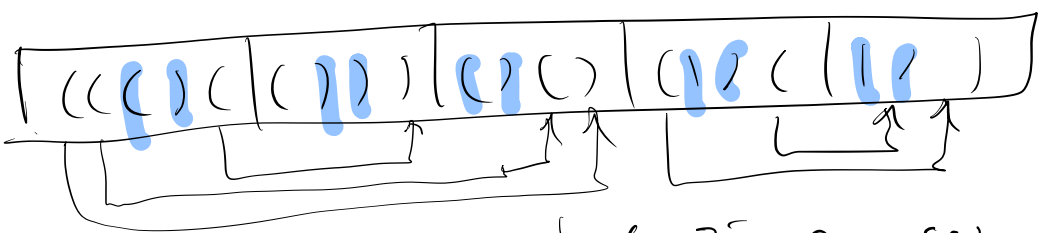
$$2n + o(n) + 6k \log n =$$

$$= 2n + 6 \frac{n}{\log n} \log n + o(n) =$$

$$= \boxed{8n + o(n)}$$

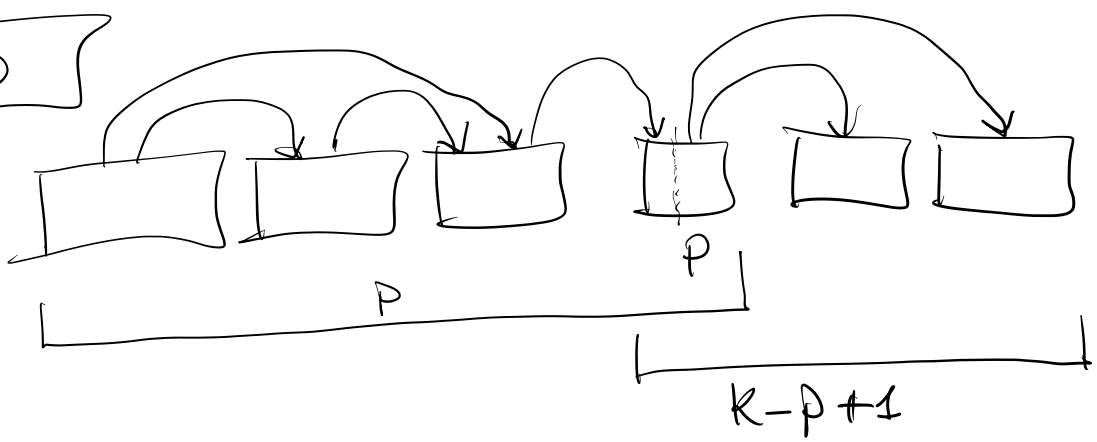
Dim (Thm): Costruiamo $G = (V, E)$ dove V sono i blocchi e

$xy \in E$ sse x contiene una parentesi che ha in y la sua corrisp.



Dimostrazione per induzione su k.

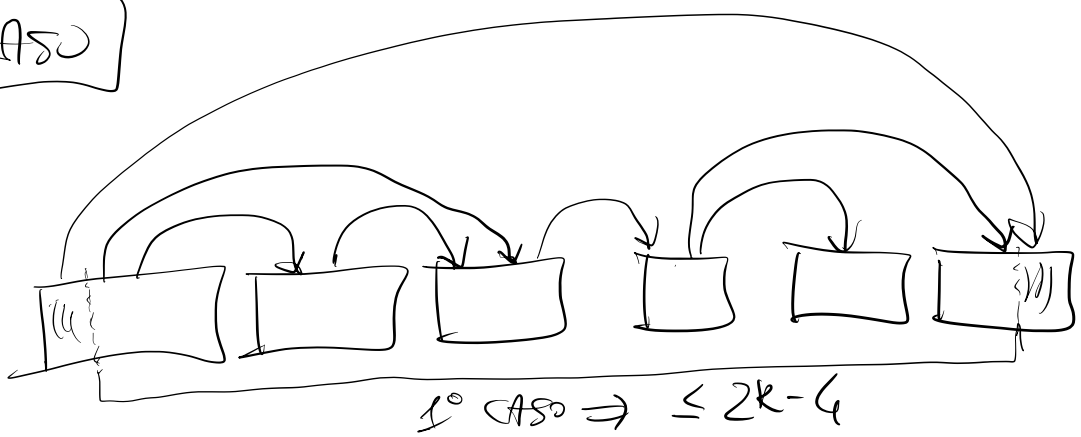
1° caso



$$\#\text{pian}_1 \leq 2p - 3$$
$$\#\text{pian}_2 \leq 2(k - p + 1) - 3$$

$$\#\text{pian} \leq 2p - 3 + 2k - 2p + 2 - 3 = 2k - 4$$ ✓

2° caso



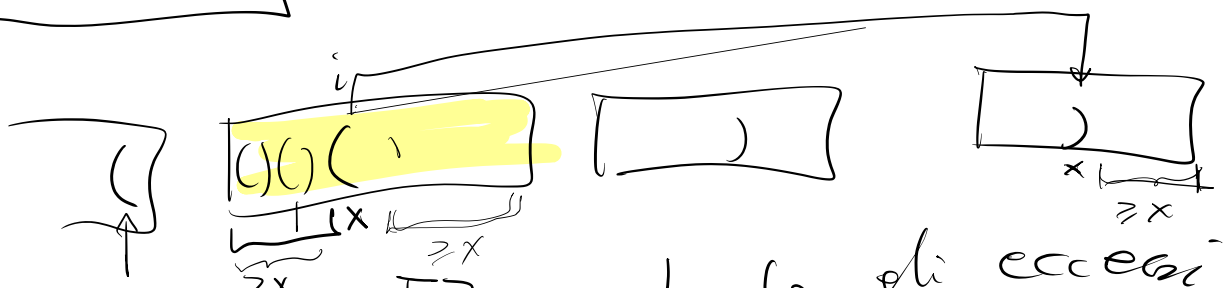
$$\leq 2k - 4 + 1 = 2k - 3$$ □

findClose$_w$(i) $O(\log n)$

1) Calcolare gli eccessi nel blocco di i (tempo $O(\log n)$)
usando E[]
2) Se i è vicino, fatto
3) j = rank$_p$(i) (indice della posizione che precede i)

M[j] = i'
Ripeti (1) per il blocco di i'

encloseW(i) $O(\log n)$



1) Usando E[] calcolo gli eccessi nel blocco, e cerco a sx di i per vedere se c'è un'apertura di eccesso x − 1

(dove x è l'eccesso a ')

2) Calcolo findClosed w (i)
e ritaccio a dx della chiusa

3) se non trovo ⇒ x è il minimo livello del blocco

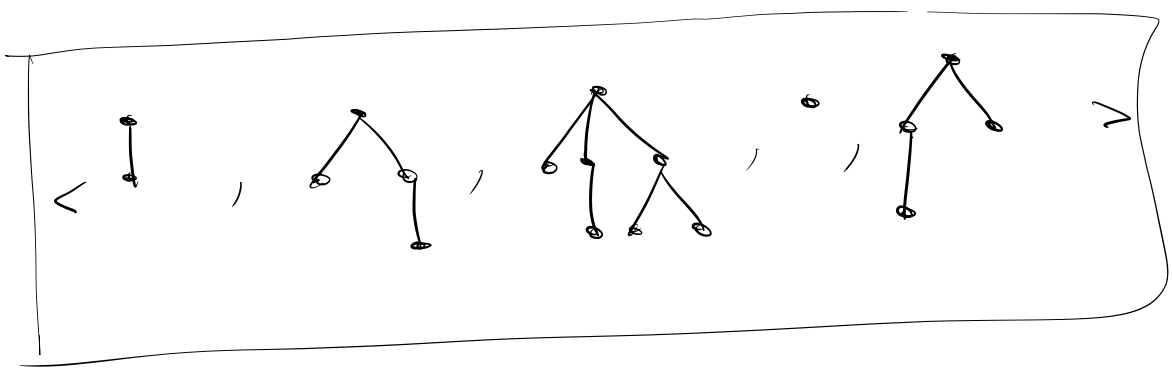
⇒ Uso ( ) [ ] per trovare le parentesi

INFORMATION-THEORETICAL LOWER BOUND

D_n = insieme delle parole di Dyck di lungh. n

B_n = insieme degli alberi binari con n nodi interni

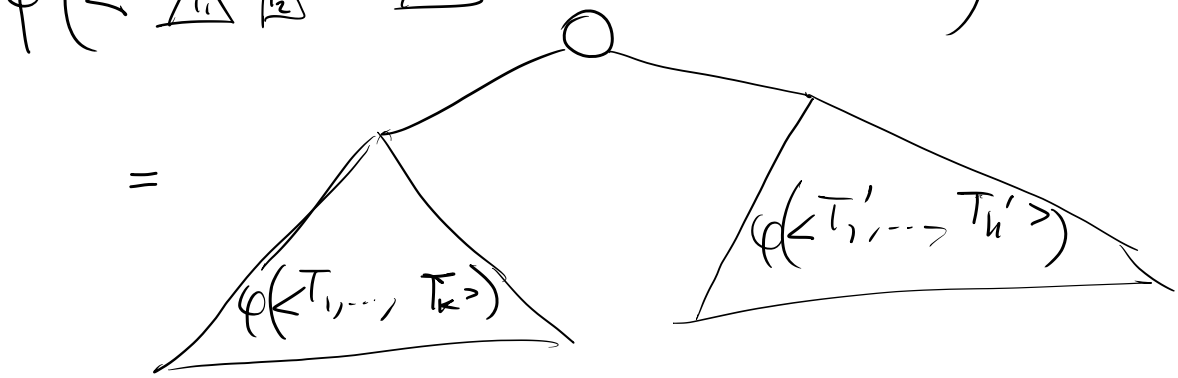
F_n = insieme delle foreste ordinate con n nodi



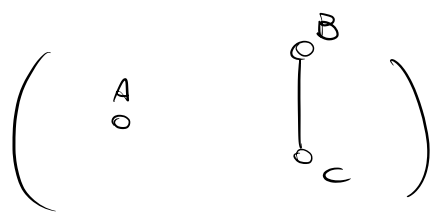
$\varphi : F_m \xrightarrow{1-1} B_m$

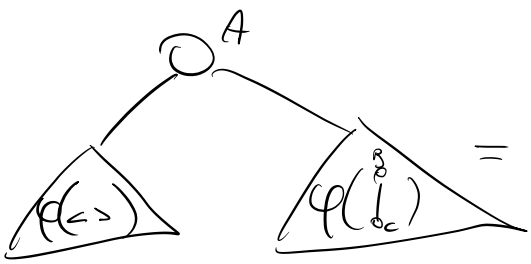
$\varphi(\langle\rangle) =$ •

$\varphi(\langle T_1, T_2, \dots, T_k \rangle, \langle T_1', \dots, T_h' \rangle) =$

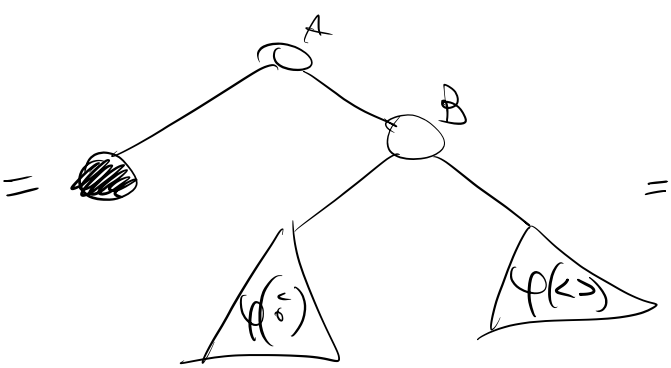


$=$

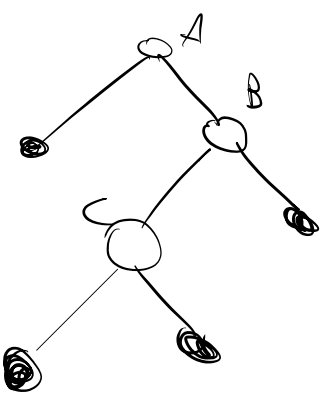
φ  $=$

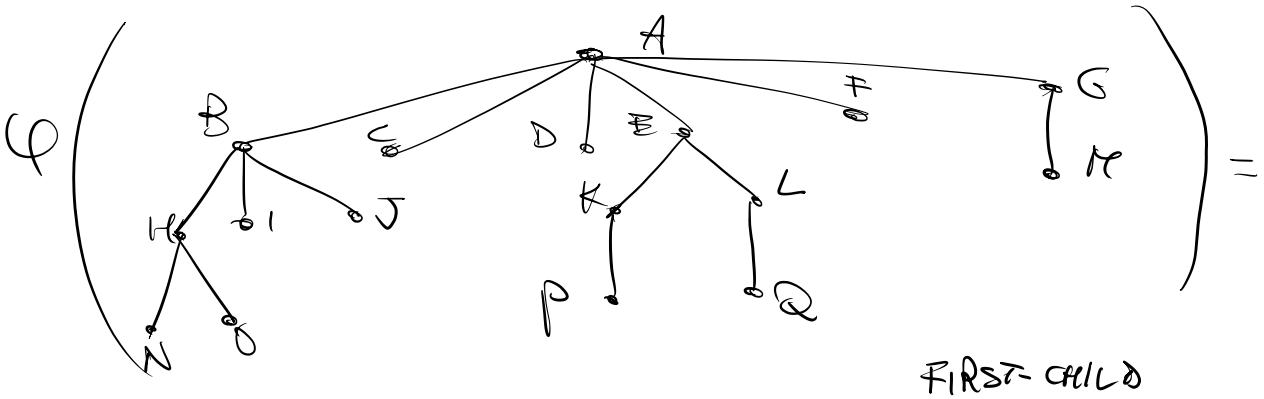


$=$

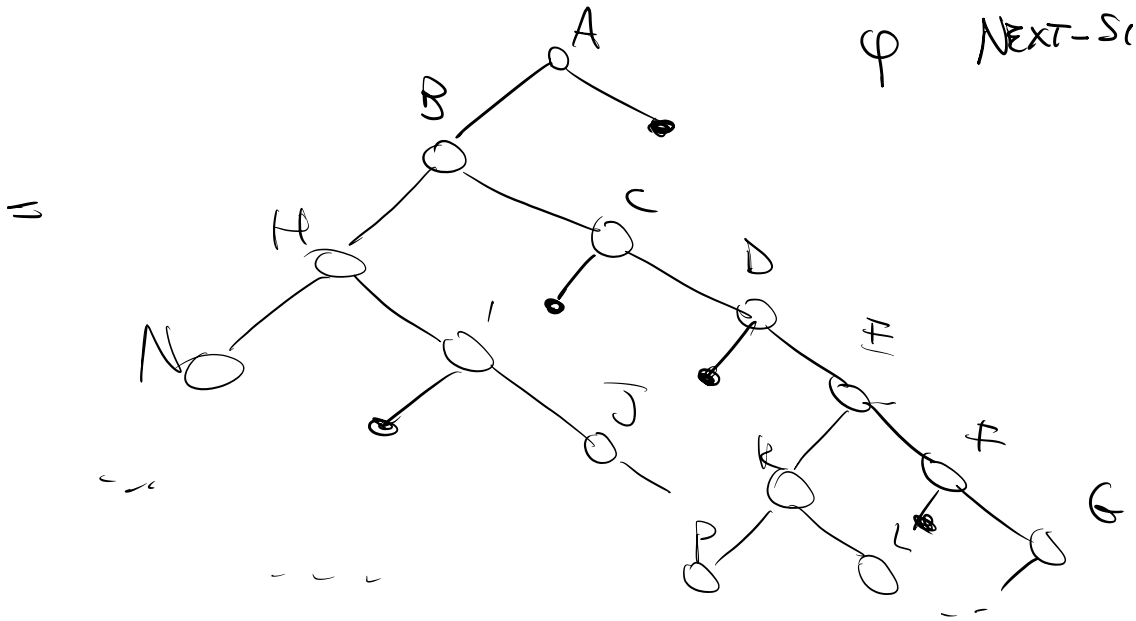


$=$





FIRST-CHILD
NEXT-SIBLING

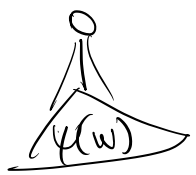


$$\psi : D_{2m} \xrightarrow{1-1} F_m$$

$$\psi(\varepsilon) = \langle\rangle$$

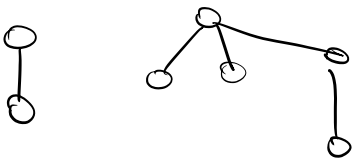
$$\psi\left(\underbrace{w_1}_{D}\ \underbrace{w_2}_{D} \cdots \underbrace{w_k}_{D}\right) = \langle \psi(w_1), \ldots, \psi(w_k) \rangle$$

$$\psi\left(\underbrace{(w}_{D}\right) = \langle$$



$\rangle$

w $\quad$ $()$ $(())$ $((()(())))$

$\varphi(w)$ $\quad$ 

$$D_{2n} \overset{\varphi}{\cong} F_n \overset{\varphi}{\cong} B_n$$

$$|B_n| = C_n$$

$$\rightarrow 2n + O(\log n)$$

D_n $\quad$ richiedono $\quad$ $n + O(\log n)$ bit

Noi usiamo $\quad$ $S(n) + O(n)$

$\Rightarrow$ STRUTTURA COMPATTA