PARENTESI BILANCIATE

Parole di Dyck

1) $\varepsilon \in D_0$
2) se $x \in D_n$, $y \in D_m$ allora $xy \in D_{n+m}$
3) se $x \in D_n$ $(x) \in D_{n+1}$

1) $\varepsilon \in D_0$
2) se $x \in D_n$, $y \in D_m$ $(x)y \in D_{n+m+1}$

1 1 0 1 1 0 0 1 0 0 1 0

(()(())()()

$w \in \{(,)\}^*$

$E_w(i) = \#\{j \mid j < i, w_j = "("\} - \#\{j \mid j \le i, w_j = ")"\}$

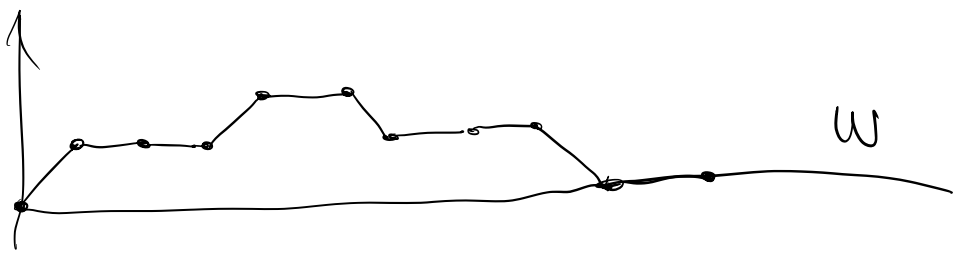
w_0 w_1 w_2

(( ) (( )) ( )) ( )

0 1 1 1 2 2 1 1 1 0 0 0

$\forall i$

1) $E_w(i) \ge 0$
2) $E_w(|w|) = 0$

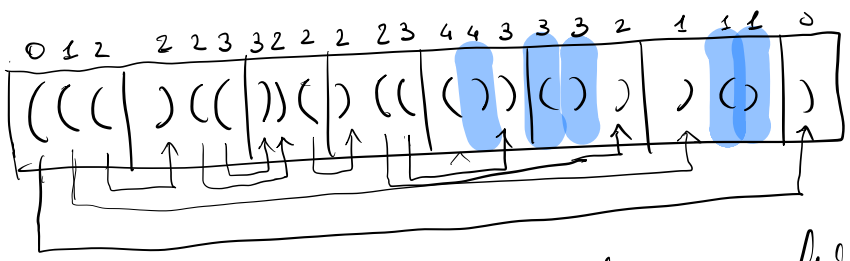


findClosed$_w(p)$ → data la posizione $w_p = "("$ restituisce la posiz. $q > p$ della p. chiusa corrispondente

findOpen$_w(p)$ → mutatis mutandis

enclose$_w(p)$ → data la posizione $w_p = "("$ restituisce la posiz. $q < p$ della par. aperta che la avvolge

e

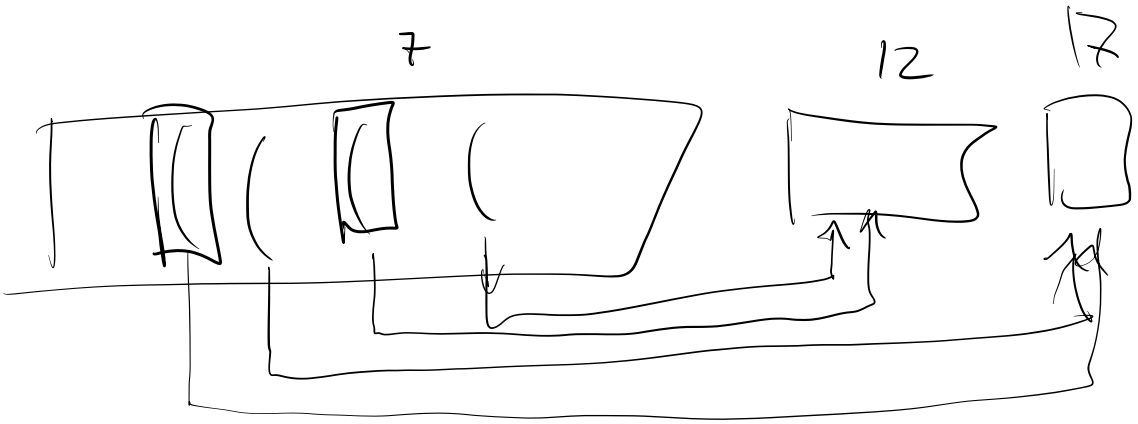


- vicine: aperte e chiuse nello stesso blocco
- lontane: altre

  - pioniere
  - non - pioniere

PIONIERA Una pioniera aperta è una aperta (lontana) che è la prima (di sx a dx) del suo blocco la cui chiusa in un certo altro blocco

blocco

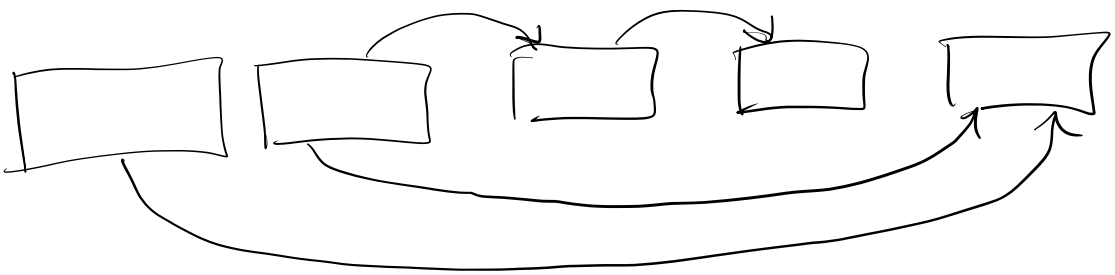


Teorema: Se ci sono $k \geq 1$ blocchi ci sono al massimo $2k-3$ pioniere.

Dim: $G = (V, E)$ orientato

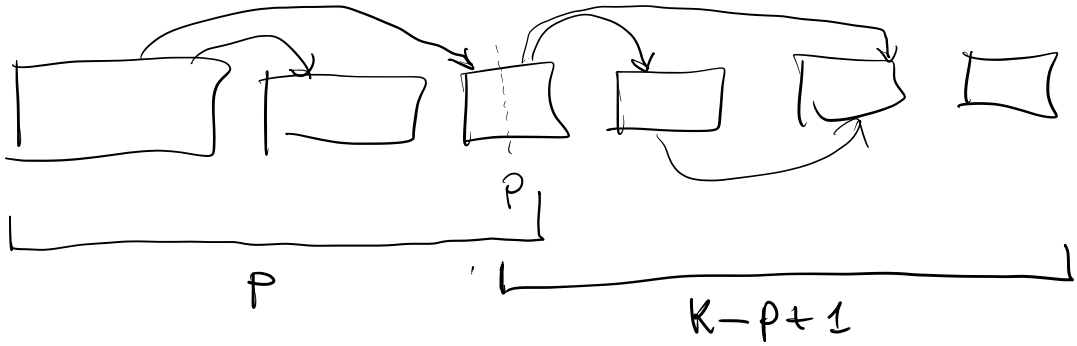
V = blocchi

E = (x,y) sse x contiene una apertura pioniera che si chiude in y



1° caso

Esiste una posizione p sopra cui non passa nessun arco



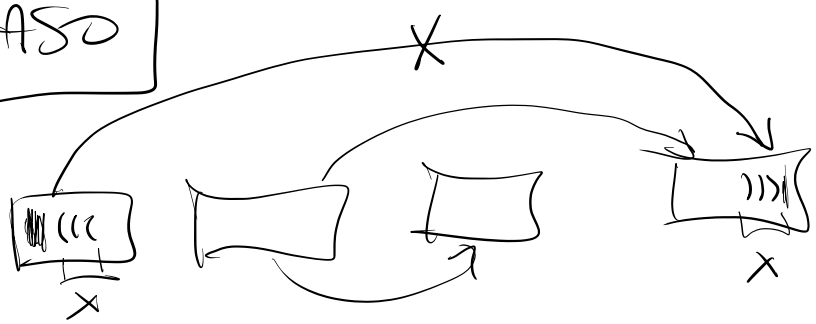
$pian_1 \leq 2p-3$

$pian_2 \leq 2(K-p+1)-3$

$\Downarrow$

$pian \leq 2p-3+2K-2p+2-3 = 2K-4$

2° CASO



$pian \leq 2K-4$ □

$2K-3$

STRUTTURE DATI

$w \in D$

$|w| = n$

$\ell = \log n$

$k = \left\lceil \frac{n}{\ell} \right\rceil$ n° di blocchi

1) w

2) $p \in \{0,1\}^*$ $\quad |p| = n$

1 nelle posizioni delle pioniere

+ rank/select

3) $M[\,]$ per ogni pioniera, la posizione della pioniera corrispondente

$M[7] = 153 \Rightarrow$ la 7-ma pioniera si trova in posiz. 153

4) $E[\,]$ per ogni blocco, l'eccesso all'inizio del blocco

5) $O[\,]$ per ogni blocco B, la posizione della prima parentesi a sinistra di B il cui eccesso è $m-1$ dove m è

il minimo eccesso in
B

MEMORIA

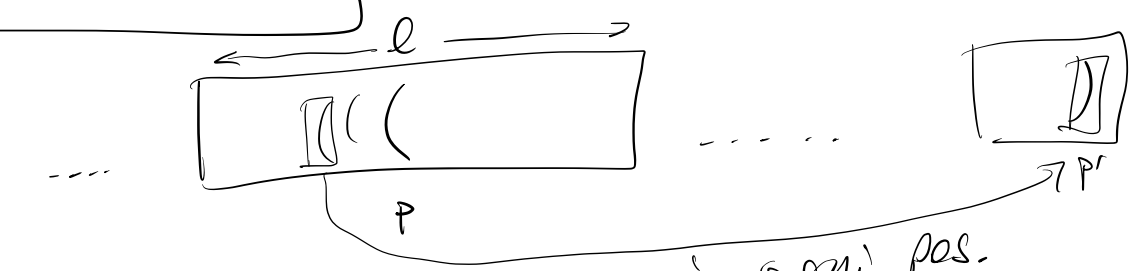
	≤ BIT
(1)	n
(2)	$n + o(n)$
(3)	#pcon · $\log n \le (2k-3)\log n$
(4)	$k \cdot \log n$
(5)	$k \cdot \log n$

$\Downarrow$

$\le 4k \log n$

$\le n + n + o(n) + 4k \log n \le 6n = O(n)$

findClosed$_w$(p)



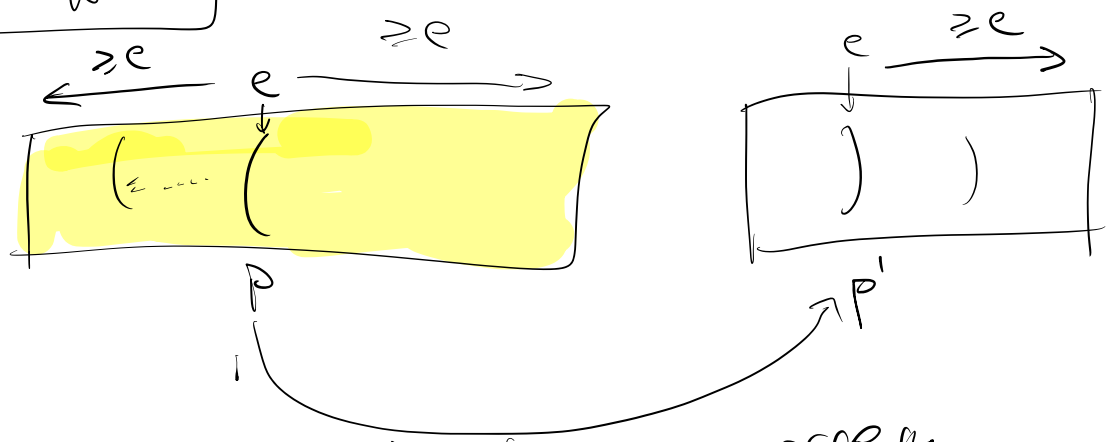
$O(\ell)$ 1) Calcolo l'eccesso in ogni pos. del blocco (a partire da E[])

$O(\ell)$ 2) si scopre se p è vicina

3) Se è lontana:

$p' = M[rank_t(p)]$

enclose$_w$(p)



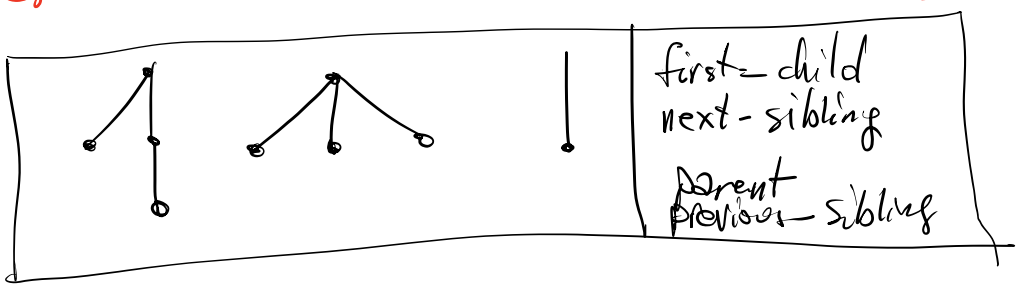
$\Rightarrow$ e_{del} è il minimo eccesso del blocco

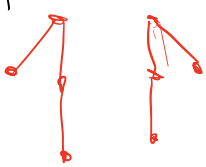
Uso O[]

Tempo $\Rightarrow$ $O(\log n)$

Spazio $\Rightarrow$ $6n + o(n) = O(n)$

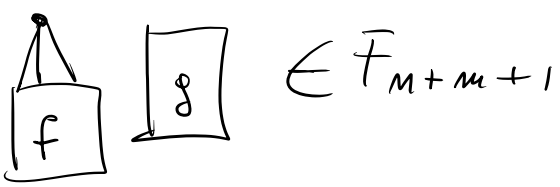
ISOMORFISMO FRA D_n (parole di Dyck di semilunghezza n) E F_n (foreste ordinate con n nodi)



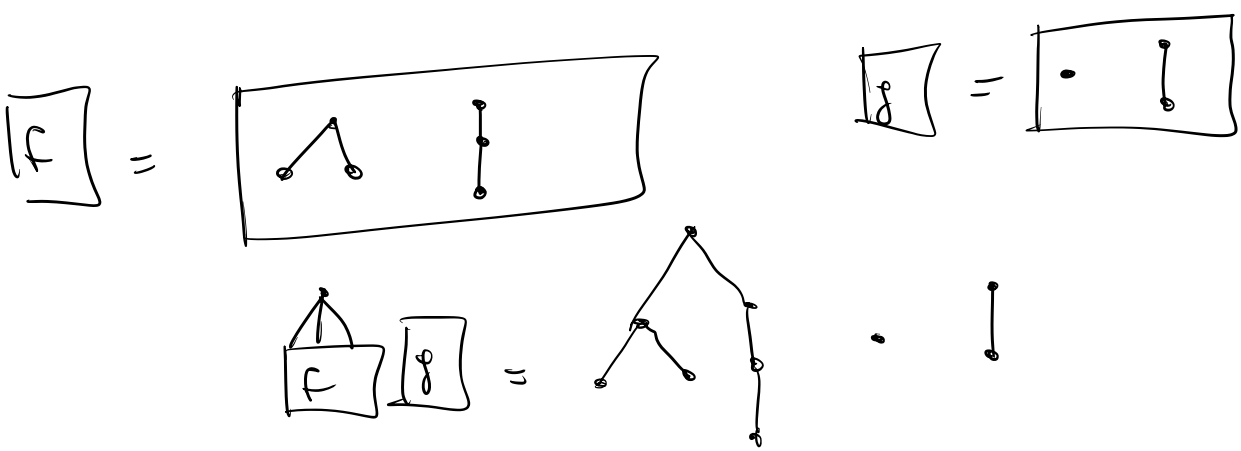


1) $\square \in F_0$

2) Se $f \in F_n$, $g \in F_m$

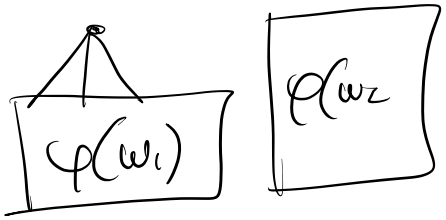


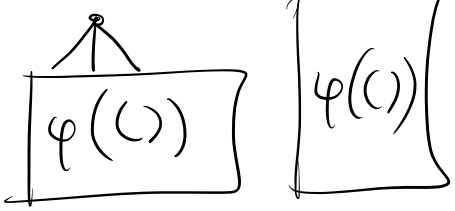
$\in F_{n+m+1}$



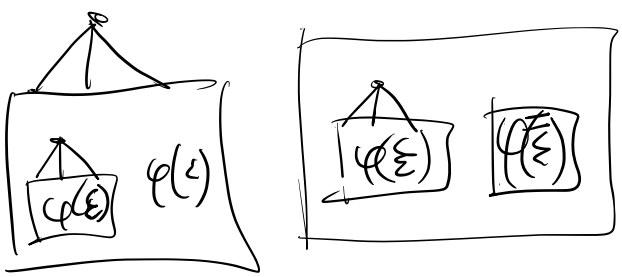
$\varphi : D_M \longrightarrow T_M$

$\varphi(\varepsilon) =$ 

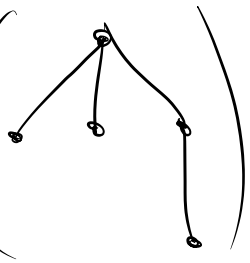
$\varphi((w_1)w_2) =$ 

$\varphi(((\,))()) =$ 

$=$

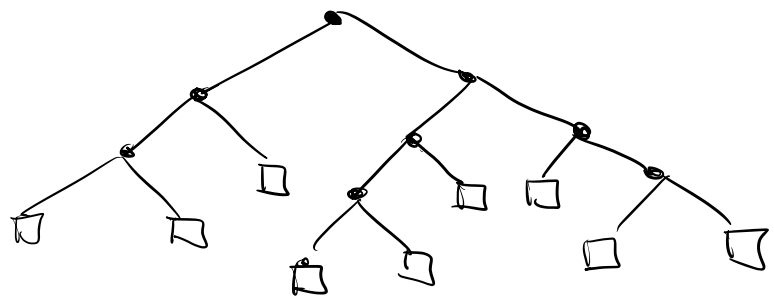
$=$ 

$=$ 
.

φ^{-1}  = (( ) ( ) (( )))

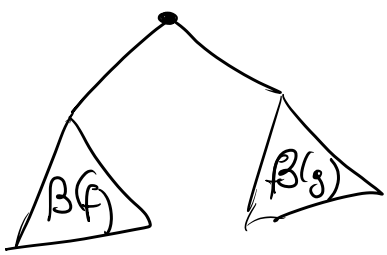
ISOMORFISMO TRA F_n E B_n
(alberi binari con n nodi interni)

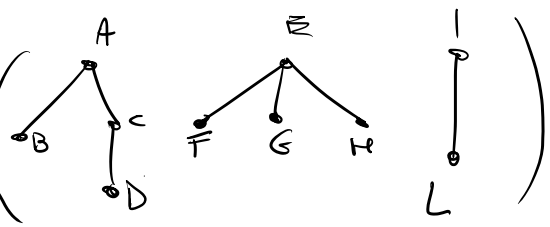
$$\mathcal{B}: F_n \to B_n$$

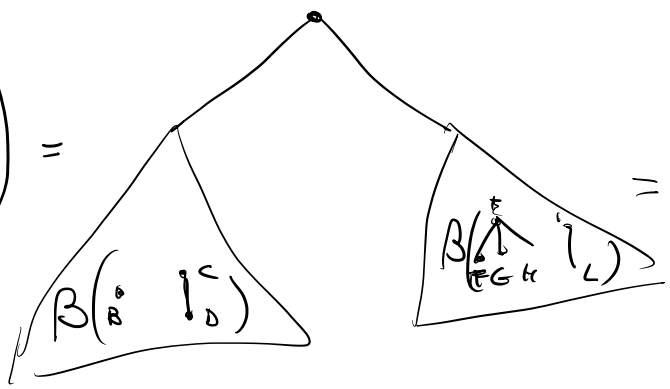


$\in B_8$

$\mathcal{B}(\square) =$ 

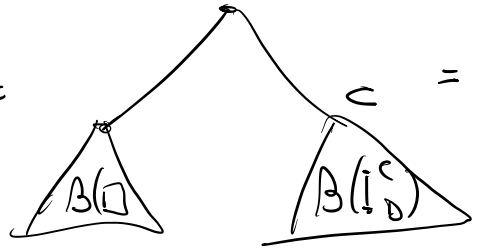
$\mathcal{B}(f\ g) =$ 

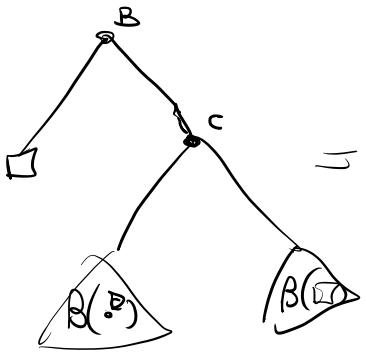
$\mathcal{B}$ 

$=$ 

$=$

$=$

$B(\bullet_B \; {}^{C}_{D}) =$ 

$=$ 

$=$

$=$

$|D_n| = |F_n| = |B_n| = C_n$