

# Algoritmi e Strutture Dati (10/11/2021)

RED BLACK TREES : ABR che soddisfano

- ① ogni nodo ha uno ed un solo colore RED/BLACK
- ② la radice è BLACK
- ③ le foglie (T.m.l) sono BLACK
- ④ i figli di un nodo RED sono BLACK
- ⑤ per ogni nodo  $x$  tutti i cammini  $x \leadsto$  foglia hanno lo stesso numero di nodi BLACK

$bh(x)$

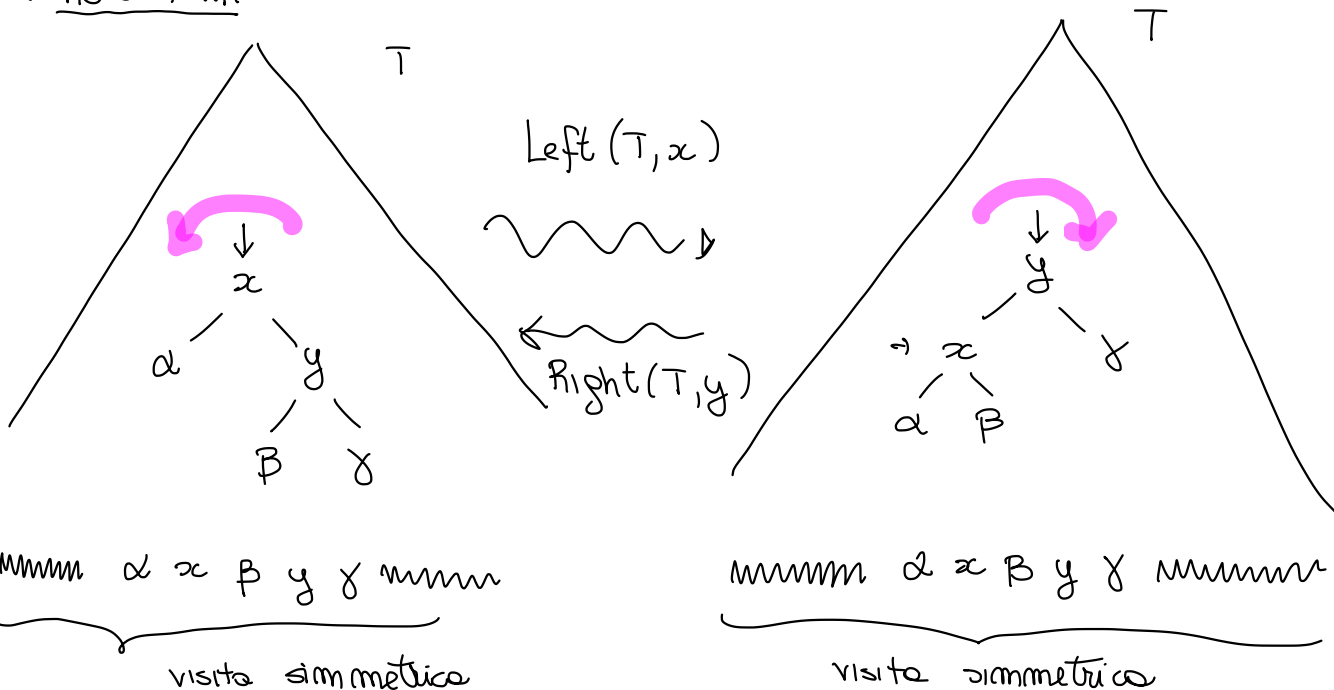
\* Altezza  $h = O(\log m)$

$$h \leq 2 \log_2 (m+1)$$

\* Operazioni search, Min, Max, Pred, Succ  $O(h) = O(\log m)$

Pb.: Come realizzare Insert/Delete mantenendo le proprietà di RB-Tree?

\* Rotazioni



OSSERVAZIONE: Le rotazioni mantengono le proprietà di ABR

dim osserva che il risultato di una visita InOrder è immutato

Left( $T, x$ ) // assunzione:  $x.right \neq T.nil$

$y = x.right$

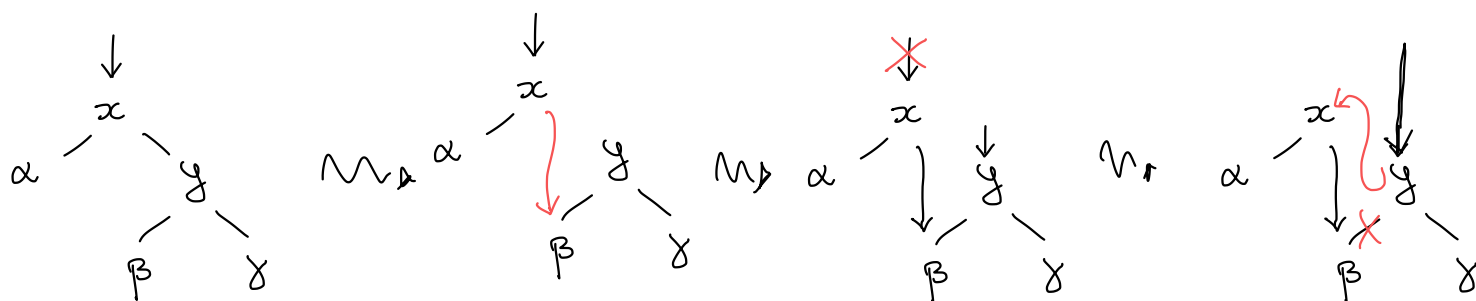
$x.right = y.left$

$x.right.p = x$

Transplant( $T, x, y$ )

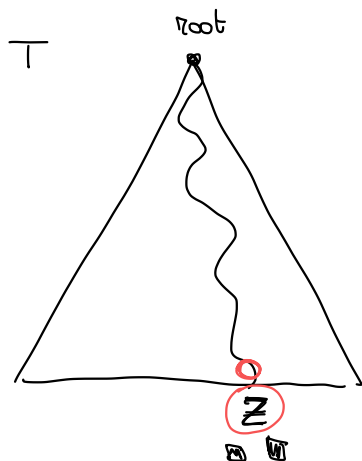
$y.left = x$

$x.p = y$



\* Insert

Insert( $T, z$ )



$z.color$  ?

- **BLACK**: altera proprietà (5)
- **RED** problemi

② se  $z$  è radice



④ parent di  $z$  è RED

idea: cambi di colore + rotazioni

risolvere il pb oppure lo si trasferisce ad un nodo "più alto"

RB-Insert ( $T, z$ )

Insert ( $T, z$ )

$z.color = RED$

RB-Insert FixUp ( $T, z$ )

Convenzione:

$z$

RED

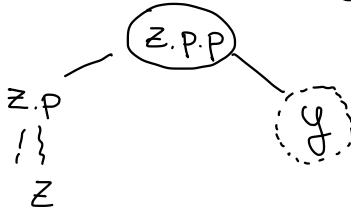
$\bar{z}$

BLACK

problema:  $z.color = RED$  e  $z.p.color = RED$

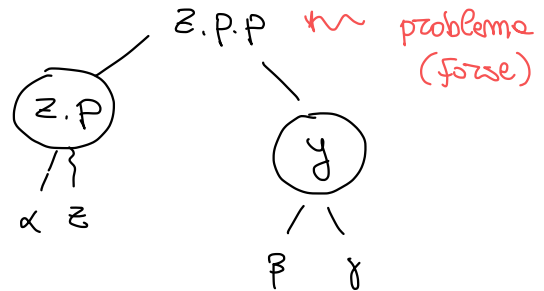
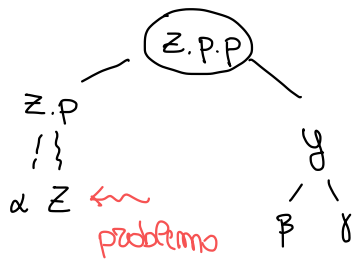
MACROCASO:  $z.p$  è figlio sx

(l'altro è simmetrico e non lo vediamo)

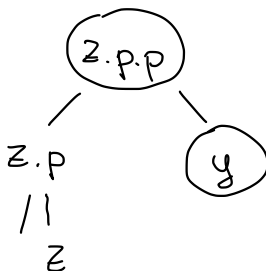


2 sotto casi a seconda del colore di  $y$

**CASO 1**  $y.color = RED$



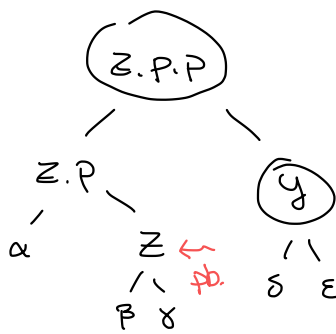
**CASO 2**  $y.color = BLACK$



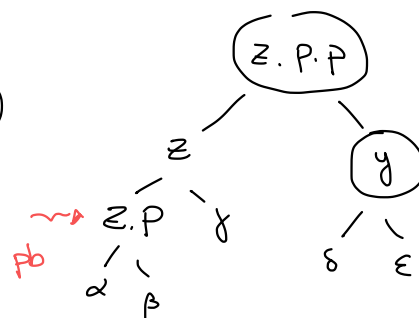
sotto casi a seconda che  $z$  sia figlio sx o dx

**CASO 2.1**  $z$  è figlio dx

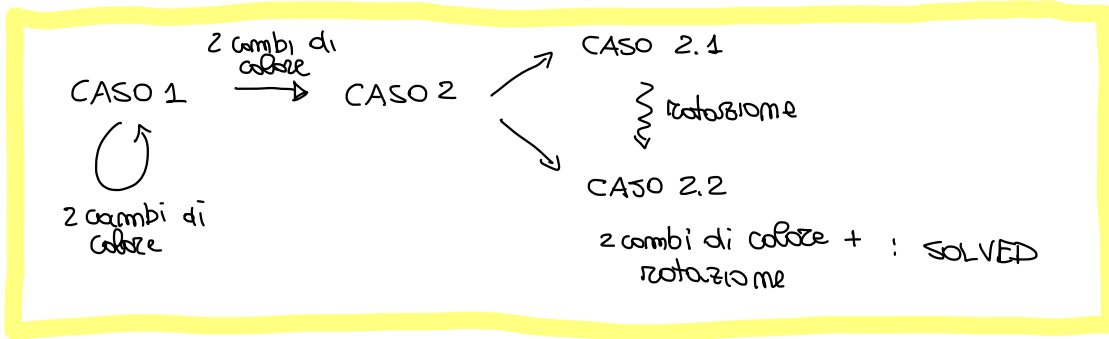
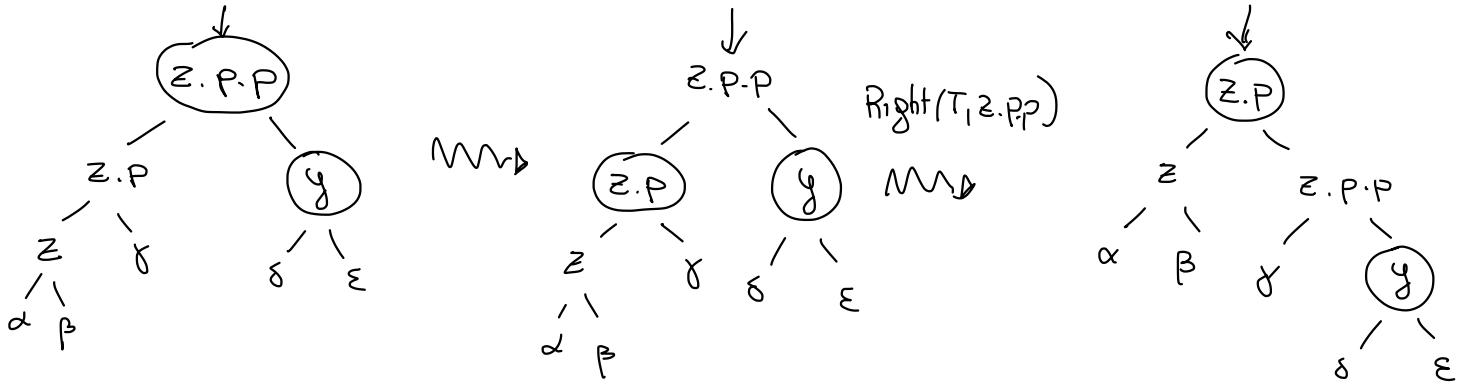
(mi riduco al caso 2.2)



Left( $T, z.p$ )



CASO 2.2  $z$  è figlio sx



Costo :  $O(1)$   
 $\downarrow$   
 max  $h/2$  iterazioni  
 e due rotazioni  $\nwarrow O(1)$

$O(h/2) = O(\log n)$

## PSEUDO CODICE

RB-Insert-FixUp ( $T, z$ )

while ( $z.p.color = RED$ )

if ( $z.p = z.p.p.pf$ )

$y = z.p.p.right$

if  $y.color = RED$

$z.p.color = BLACK$

$y.color = BLACK$

$z.p.p.color = RED$

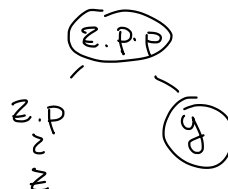
$z = z.p.p$

else  $\wedge y.color = BLACK \wedge$

MACROCASO :  $z.p$  è figlio sx



CASO 2



↓ y. ← uso sistema colore di z

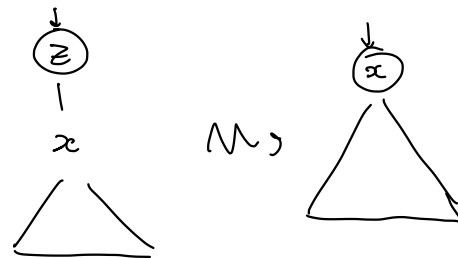
\* un solo problema



• se  $z$  è RED

- mom è radice (2)
- mom crea nodi RED consecutivi (4)
- mom cambia lunghezza BLACK (5)

• se  $z$  è BLACK



→ se  $x$  è RED OK!

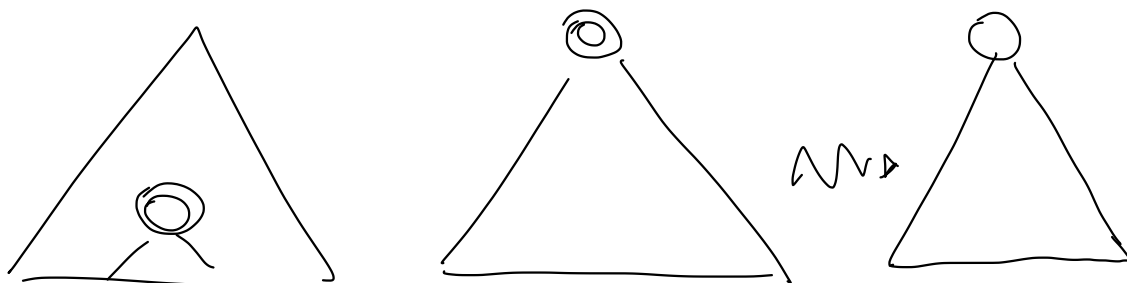
→ se  $x$  è BLACK



double black : video 1

con ROTAZIONI & cambi di colore

sposto il problema in alto



# iterazioni  $O(h)$   
# rotazioni 3

} →  $O(h) = O(\log n)$