

Alberi Rosso-Neri

E' un implementazione particolare di un albero binario di ricerca.

E' un albero di tipo posizionale e rispetto agli alberi binari di ricerca è autobilanciante, questo consente di non sfociare nel caso peggiore di ricerca in une liste $O(\lg n)$

Ovviamente le operazioni de fare per mantenere l'albero bilanciato devono essere leggere se no si avrebbe uno spreco de un' altra parte

PROPRIETA'

- 1) Radice nera
- 2) Un nodo o è rosso o è nero
- 3) Ogni foglia è nera
- 4) Un nodo rosso ha solo figli neri
- 5) Qualsiasi cammino percorso da un nodo n , avremo lo stesso numero di nodi neri (Altezza Nera)

Ricerca

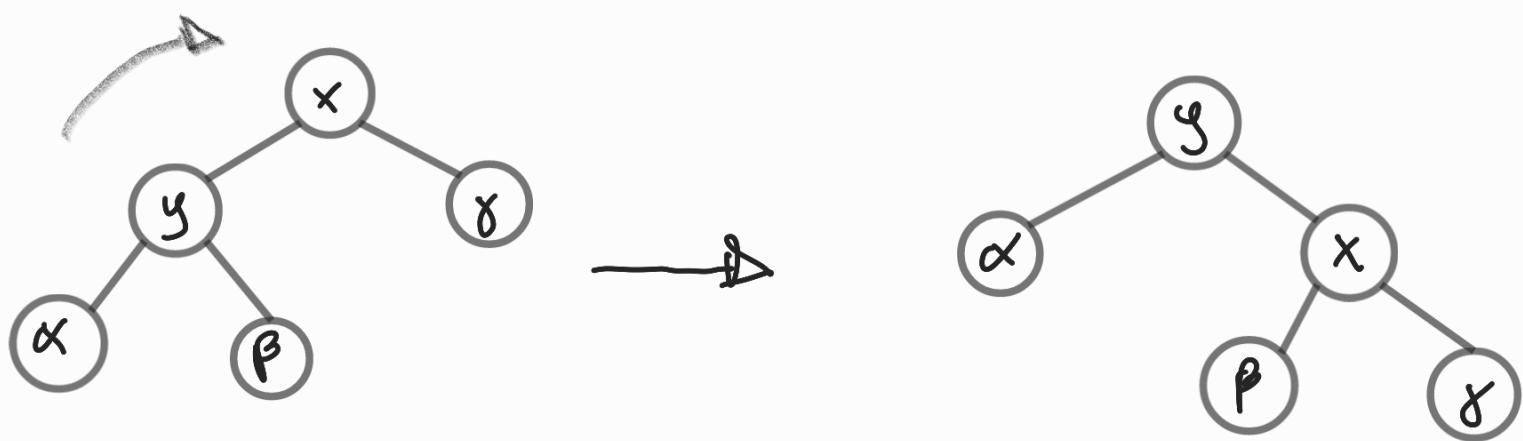
L'operazione di ricerca risulta uguale alle ricerche in un BST, con le differenze che sarebbe sempre $O(\log n)$

INSERIMENTO

Può violare qualche proprietà per questo bisogna gestirlo

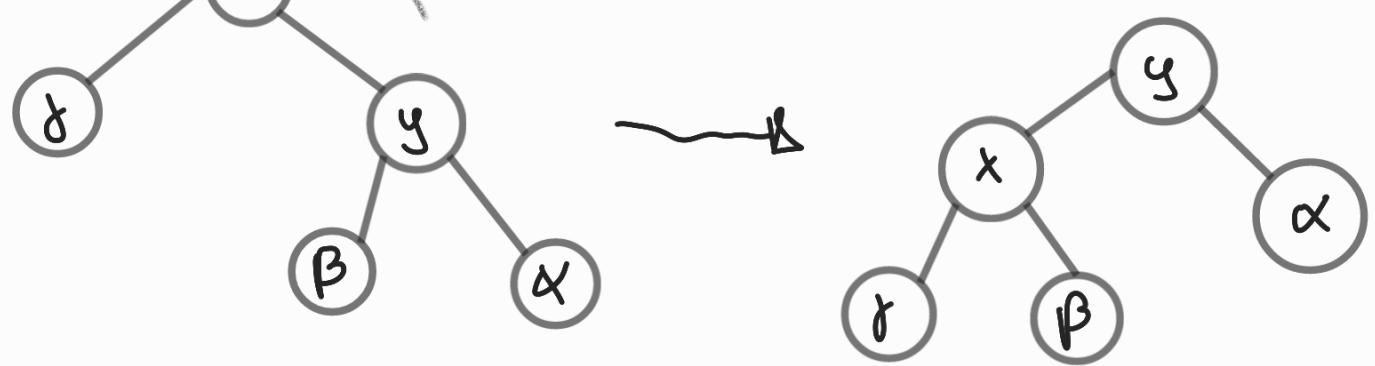
I nodi inseriti sono sempre rossi perché è più semplice sistemare la proprietà a che la s per sistemare gli inserimenti usiamo le rotazioni, operazioni che non violano le proprietà di albero posizionale

RIGHT ROTATE



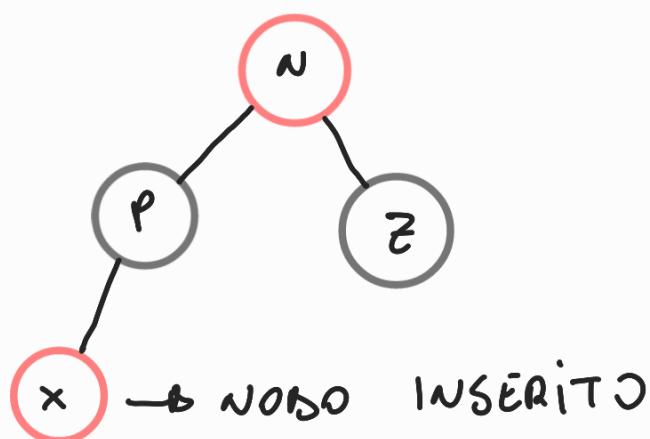
LEFT ROTATE





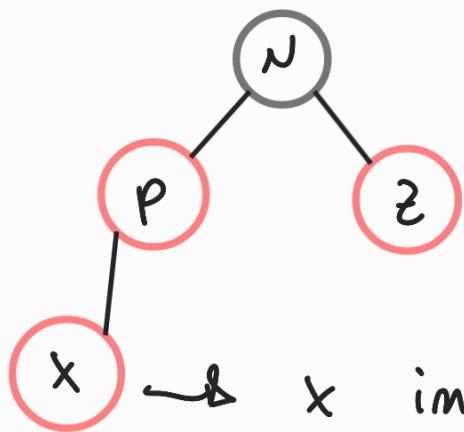
CASI INSERIMENTO

CASO 0



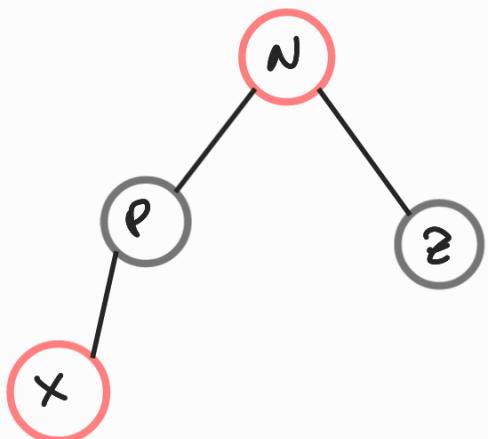
L'inserimento va a buon fine e non si fa niente, in caso le strutture ve aggiustate ci chiamate ricorsivamente RB-TREE FIX-UP(x)

CASO 1



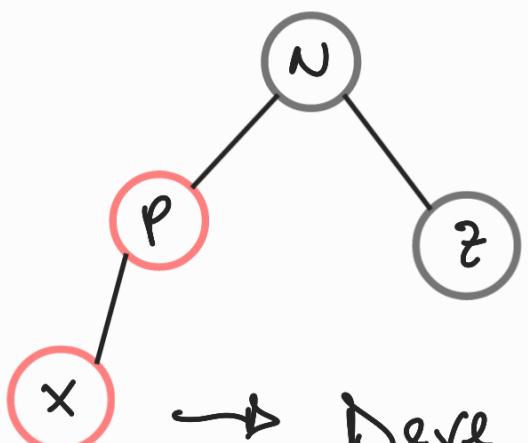
x inserito non va bene perché il padre è rosso, me essendo lo zio

tono, & nonno N regole il
colore ai figli me segue



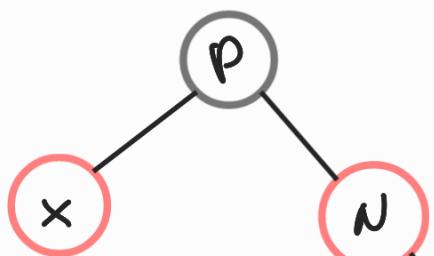
Le strutture s'è state
sistematice

CASO 2



→ Deve essere sistematico le
strutture quindi ruoto a
destra

Dunque le rotation i colori:
dei modi che si cambiano
vengono cambiati

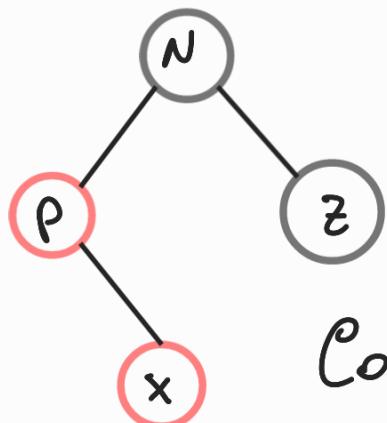


Le strutture ora va

bene !

z

CASO 3



Come nel caso 2 le strutture
non va bene

Riusto p ed x e mi ricordo
el CASO 2

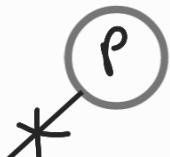
Con queste procedure riusciamo ad avere
una struttura con inserimenti sicuri

CANCELLAZIONE

Come per l'inserimento puo' causare le
perdite delle proprietà delle strutture

Abbiamo vari casi :

CASO 1

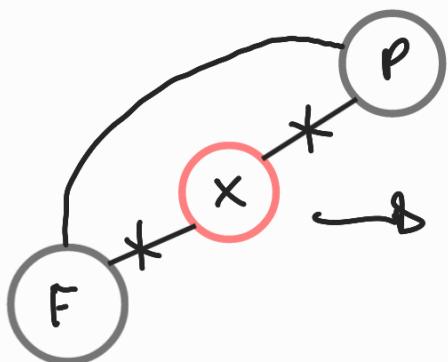


Dobbiamo staccare una foglia

x

In questo caso non abbiamo nessun problema

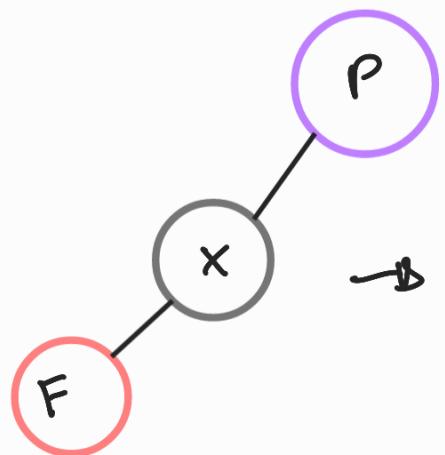
CASO 2



Dobbiamo staccare il nodo rosso. Non crea problemi lo stacchiamo e collegiamo i due nodi neri

I NODI ROSSI NON RECANO PROBLEMI PER LA CANCELLAZIONE

CASO 3



Il viola indica un colore generico quindi sié con rosso che con nero vale il caso

→ X deve essere eliminato, essendo il figlio rosso x cede il colore e non reca problemi

Nel caso in cui F è nero si cede ugualmente il colore e si ha un doppio nero

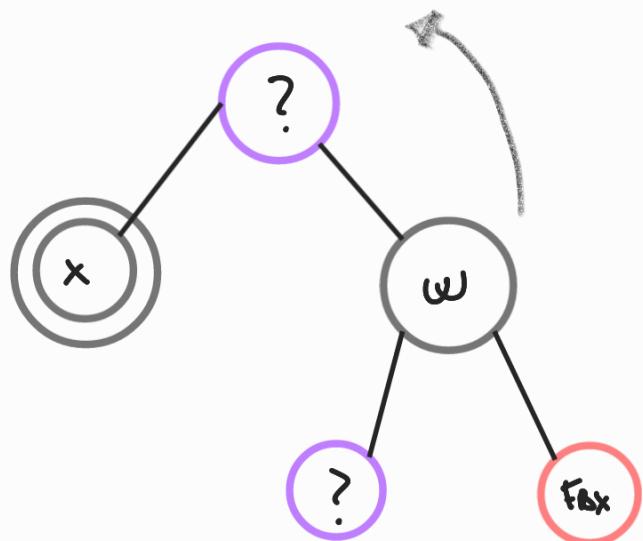


ne



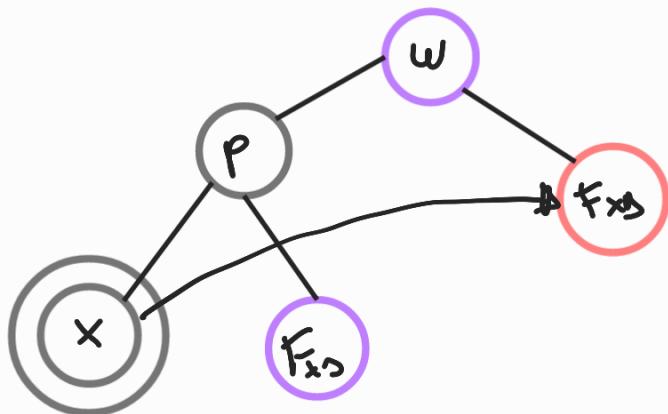


CASO 1



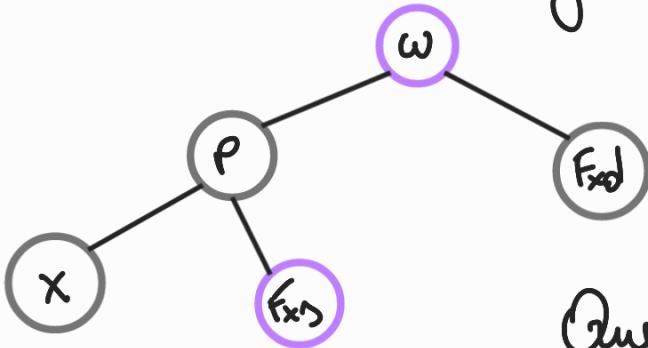
Se x ha un doppio nero e il fratello ω ha il figlio dx rosso si risolve così:

PER PRIMA COSA APPlichiamo UNA ROTATION



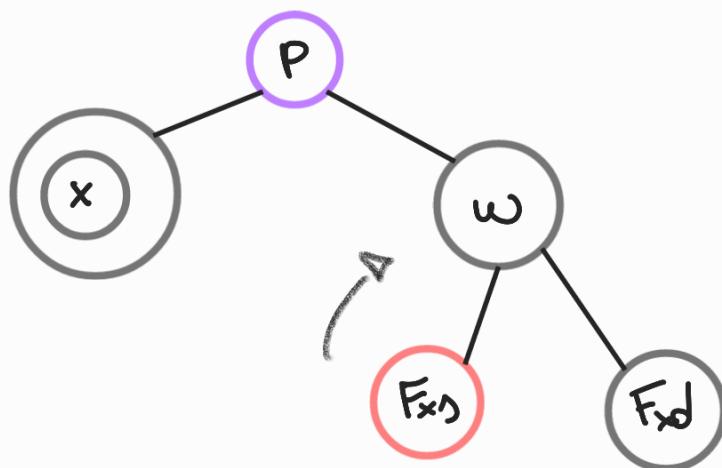
Adesso cediamo il doppio nero al $F_{x\omega}$

Ne segue:



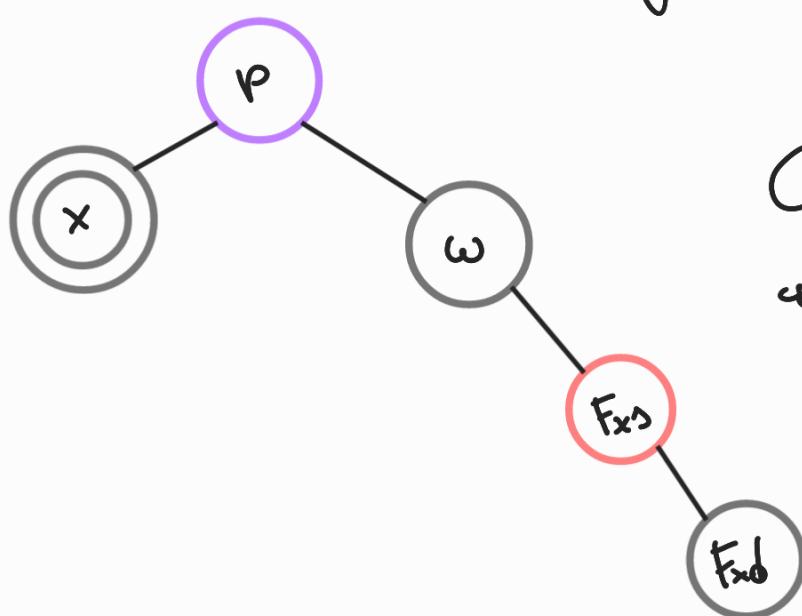
Questa configuratione è accettabile

CASO 2

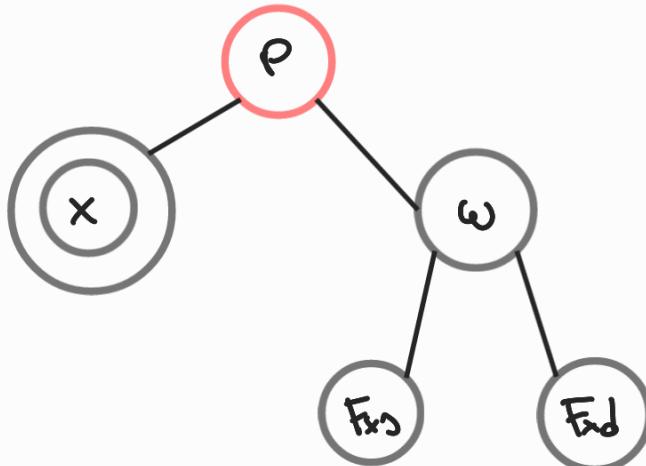


In questo caso
Faccio una
right-rotate e
mi ricordo del
caso 1

Ne segue



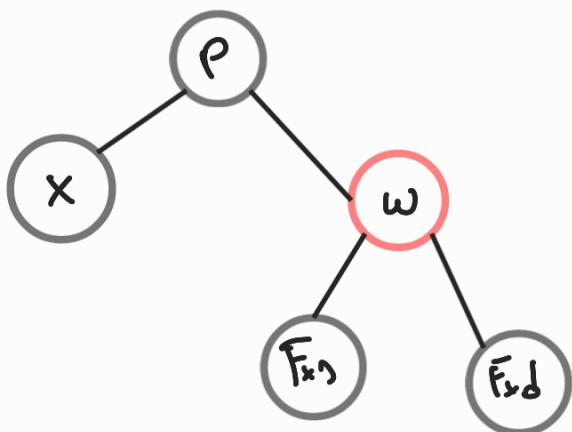
CASO 3



Essendo che non
possiamo applicare i
precedenti casi que
per risolvere il problema
x e ω devono d...

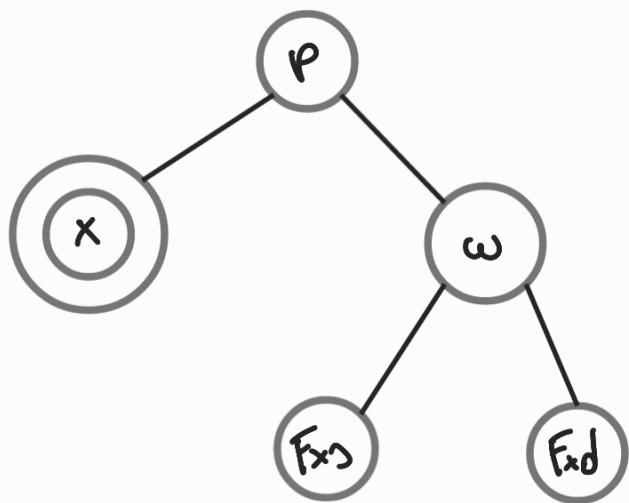
il pedre

Ne segue



Ora le strutture va
bene

CASO 4

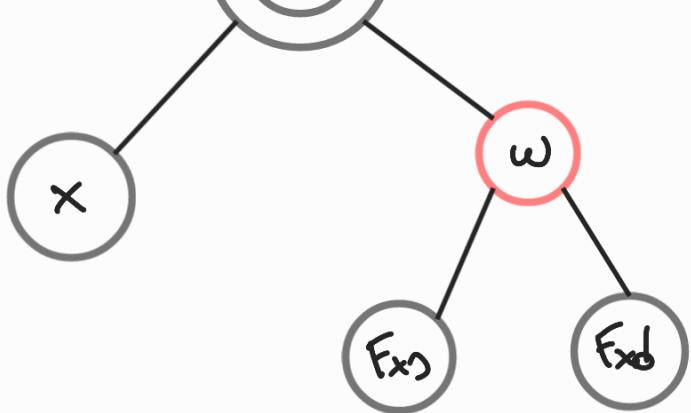


In questo caso
l'unica cosa che
possiamo fare è
donare il doppio-nero
al pedre e lasciare
a lui il problema

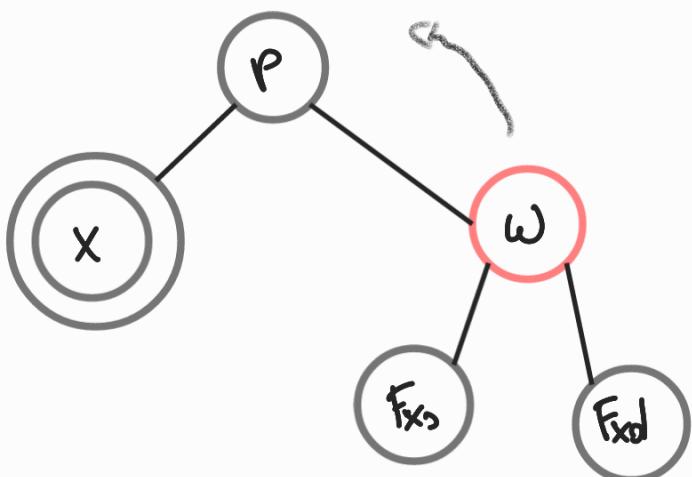
Nel caso in cui il doppio-nero arriva
alla radice si elimina ed il problema
è tolto

Da queste procedure segue





CASO 5



Riottiamo P e ω
e ci ricordiamo
el caso 3

Ne segue che

