

Alberi red-black

1 Albero red-black

Un **albero red-black** (RB) è un albero binario di ricerca in cui:

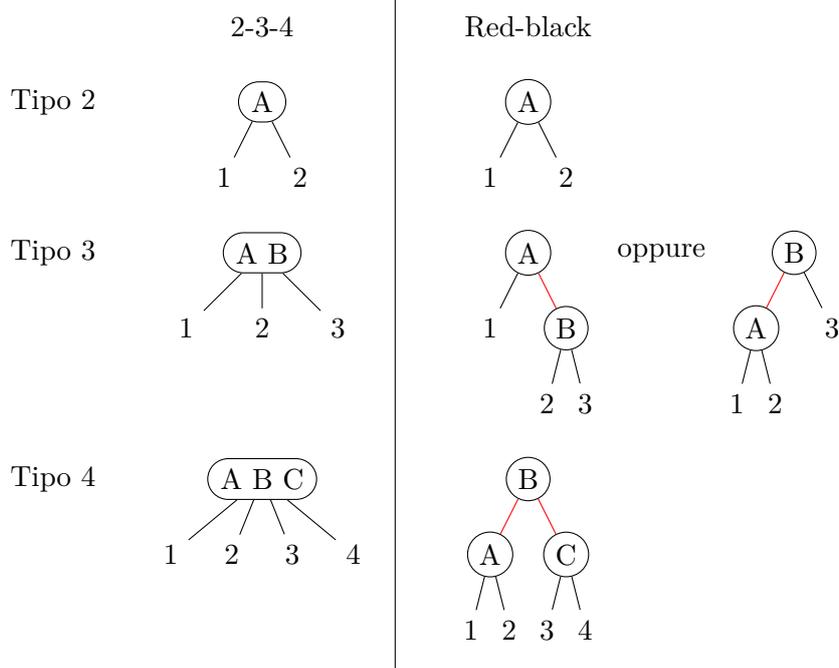
- ogni nodo contiene (oltre a un valore) un campo colore (rosso/nero), il quale indica il colore del lato che collega il nodo al padre;
- non ci sono due lati rossi consecutivi (cioè il cammino da un qualunque nodo a un suo nipote ha almeno un lato nero);
- ogni cammino dalla radice a una foglia ha lo stesso numero di lati neri (ma le foglie non sono necessariamente tutte allo stesso livello).

2 Corrispondenza con gli alberi 2-3-4

Per implementare gli alberi 2-3-4 si potrebbe rappresentare ciascun nodo mediante un record dotato di 3 campi per i valori e 4 riferimenti per i figli. Così facendo, però, si avrebbe uno spreco di spazio per tutti non saturi.

Gli alberi red-black sono invece un'implementazione efficiente per gli alberi 2-3-4: siccome ogni nodo contiene esattamente un valore, si evitano gli sprechi.

In un albero red-black, più nodi collegati da lati rossi corrispondono a un unico nodo di un albero 2-3-4:



2.1 Altezza

Lemma: Sia T un albero 2-3-4 di altezza h . Allora, ogni albero RB equivalente a T ha al massimo altezza $2h$.

Dimostrazione: Per attraversare il corrispondente di un nodo 2-3-4 in un albero RB servono al massimo due lati, quindi l'altezza può al massimo raddoppiare passando da un albero 2-3-4 a un albero RB equivalente.

Osservazione: Segue dal lemma che un albero RB con n valori ha altezza $\Theta(\log n)$, quindi gli alberi red-black sono **bilanciati**.

2.2 Alberi red-black equivalenti

A un particolare albero red-black corrisponde un unico albero 2-3-4.

Viceversa, a un dato albero 2-3-4 corrispondono più alberi RB, perché esistono due rappresentazioni diverse per i nodi di tipo 3 (e quale delle due viene usata in ogni caso dipende dalla sequenza di operazioni eseguite).

Di conseguenza, se un albero 2-3-4 ha n nodi di tipo 3, esistono 2^n alberi RB corrispondenti.

3 Inserimento

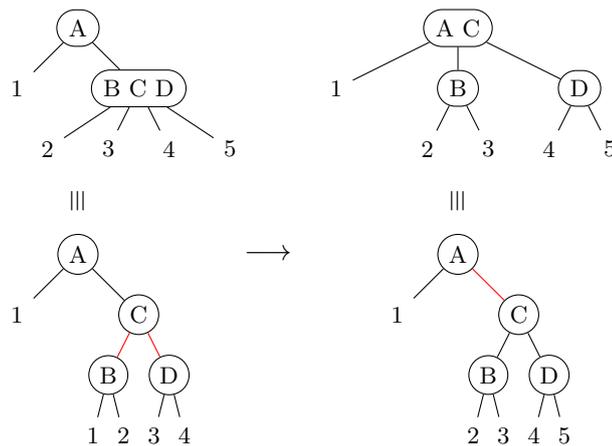
L'inserimento di un valore in un albero red-black si effettua come in un BST, ma in più:

1. si scompongono eventuali nodi saturi (cioè aventi entrambi i figli collegati da lati rossi) durante la discesa, come per gli alberi 2-3-4;
2. se, dopo l'inserimento, si hanno due lati rossi consecutivi, si eseguono delle rotazioni per ridisporre i nodi in modo da eliminare la situazione illegale.

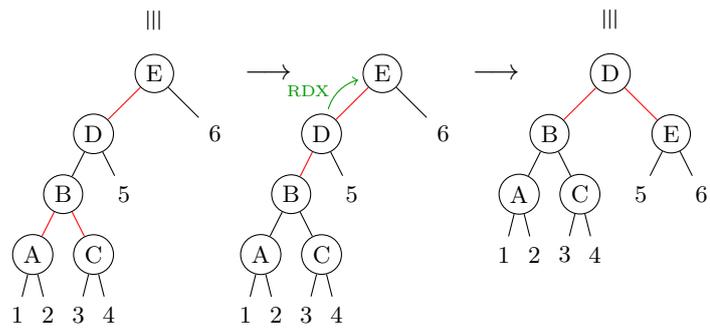
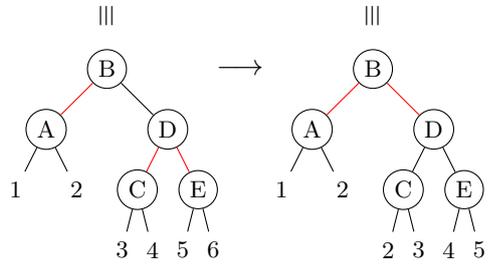
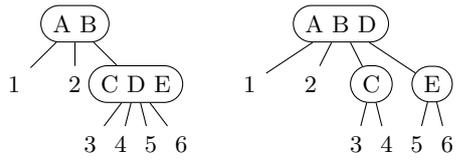
3.1 Scomposizione

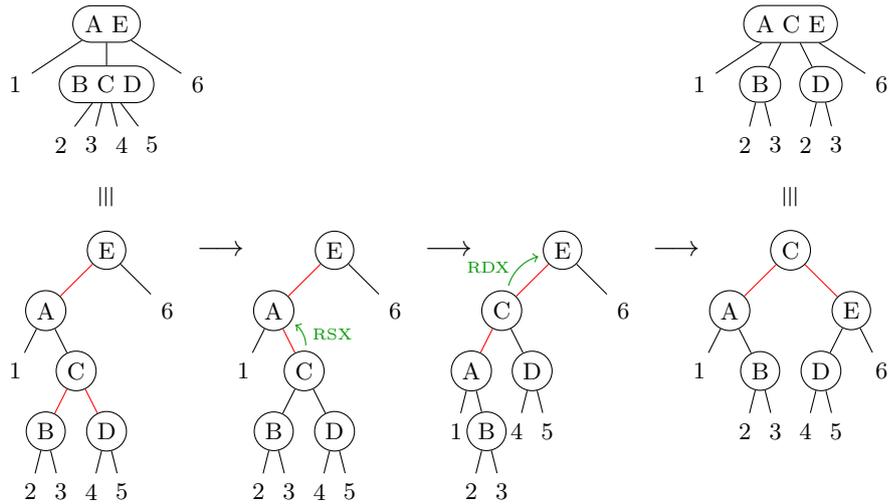
La scomposizione di un nodo saturo si effettua invertendo i colori di 3 lati (quelli che collegano il nodo al padre e ai due figli) e, se necessario, applicando delle rotazioni per evitare che ci siano due lati rossi consecutivi.

- Se il nodo saturo è figlio di un nodo di tipo 2 (o è la radice), la scomposizione è immediata:



- Se, invece, il nodo saturo è figlio di un nodo di tipo 3, la procedura di scomposizione dipende da come è orientato il padre, cioè da qual è il lato rosso:





- Infine, il nodo saturo non può essere figlio di un nodo di tipo 4, cioè di un altro nodo saturo, dato che quest'ultimo sarebbe già stato scomposto.

3.2 Complessità

Il costo della scomposizione di un nodo saturo è $O(1)$, perché si effettuano al massimo 3 cambiamenti di colore e 2 rotazioni, tutte operazioni a costo costante.

Di conseguenza, la complessità in tempo dell'inserimento è $\Theta(\log n)$ in ogni caso.

4 Cancellazione

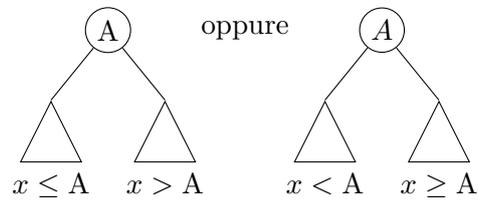
Per la cancellazione, si applica agli alberi red-black la stessa strategia usata per gli alberi 2-3-4:

1. si cerca il nodo da cancellare;
2. se esso non è una foglia, si cerca un sostituto;
3. si rimuove il valore/sostituto e si ribilancia l'albero.

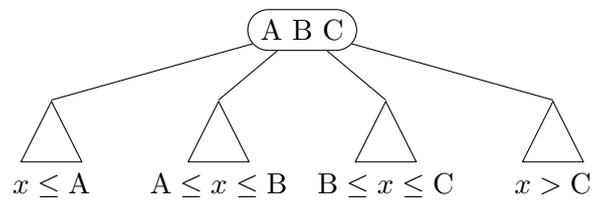
La complessità in tempo è sempre $\Theta(\log n)$.

5 Gestione di valori uguali

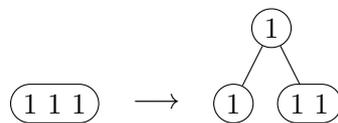
Nei BST, per poter gestire valori uguali è sufficiente determinare se inserirli sempre nel sottoalbero sinistro o in quello destro.



Negli alberi 2-3-4 e RB, invece, i valori uguali devono essere consentiti in tutti i sottoalberi, per via dell'operazione di scomposizione.



Ad esempio, se si costruisce un albero 2-3-4 dalla sequenza 1,1,1,1, con il quarto inserimento si creano due sottoalberi contenenti valori uguali a quello del padre:



Un altro problema è che la cancellazione diventa ambigua, perché non è ovvio quale dei valori uguali eliminare. Bisogna quindi scegliere in modo coerente.

In pratica, per questi motivi si tende a utilizzare una porzione univoca dei dati come chiave, in modo da garantire che tutti i valori inseriti siano diversi.