



**PEGASO**  
Università Telematica





## Indice

1. ALBERO ROSSO-NERO .....	3
2. COSTRUZIONE.....	5
3. ANALISI ALBERO ROSSO-NERO .....	8
BIBLIOGRAFIA .....	10

## 1. ALBERO ROSSO-NERO

Si dice che un albero binario di ricerca è **rosso-nero** se:

- Ogni nodo è colorato di rosso o di nero
- Le chiavi vengono mantenute solo nei nodi interni dell'albero
- Le foglie sono costituite da nodi speciali Nil: tali nodi sono detti "nodi sentinella" ed il loro contenuto è di fatto "irrilevante". Di fatto, quando in un albero rosso-nero un nodo ha come figli nodi NIL, quel nodo sarebbe una foglia dell'albero binario di ricerca corrispondente
- Vengono rispettati i seguenti vincoli:
  - ✓ La radice è nera
  - ✓ Tutte le foglie sono nere
  - ✓ Entrambi i figli di un nodo rosso sono neri
  - ✓ Ogni cammino semplice da un nodo ad una delle foglie contenute nel suo sottoalbero ha lo stesso numero di nodi neri

Dalle regole di cui sopra ricaviamo che non possono esserci 2 nodi rossi adiacenti, anche se più nodi neri possono esserlo.

Un albero **rosso-nero** è dunque un albero binario di ricerca con un bit aggiuntivo di memoria per ogni nodo che rappresenta il colore del nodo, che può essere ROSSO (red) o NERO (black).

Assegnando dei vincoli al modo in cui i nodi possono essere colorati lungo qualsiasi percorso che va dalla radice a una foglia, gli alberi red-black garantiscono che nessuno di tali percorsi sia più di due volte più lungo di qualsiasi altro, quindi l'albero è approssimativamente bilanciato.

Facciamo alcuni esempi di alberi rosso-nero:

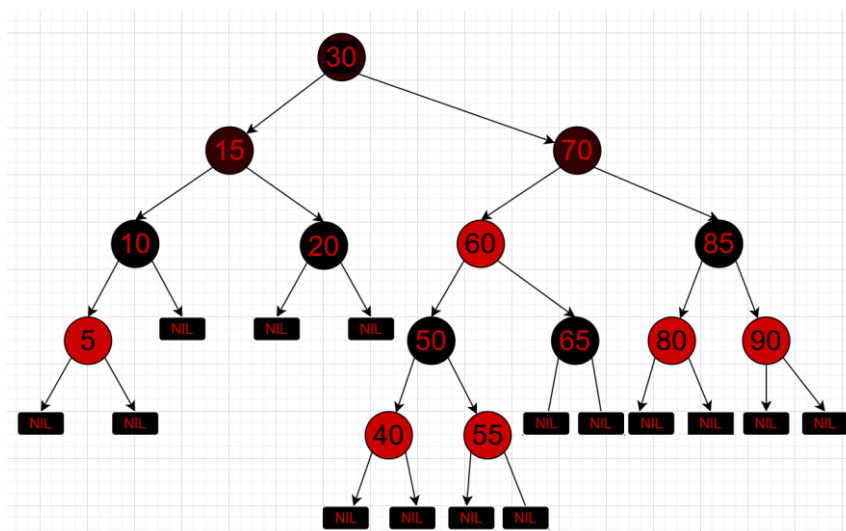


Figura 1: Albero Rosso-Nero

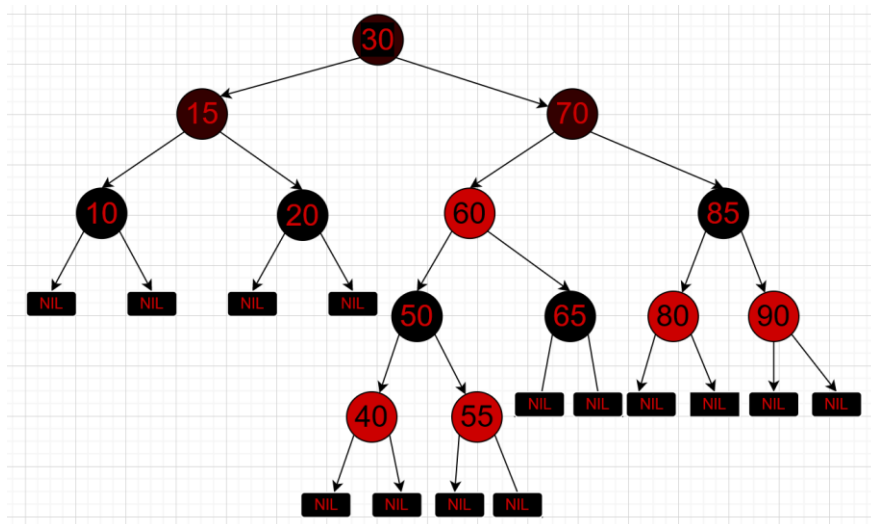


Figura 2: Albero rosso-nero

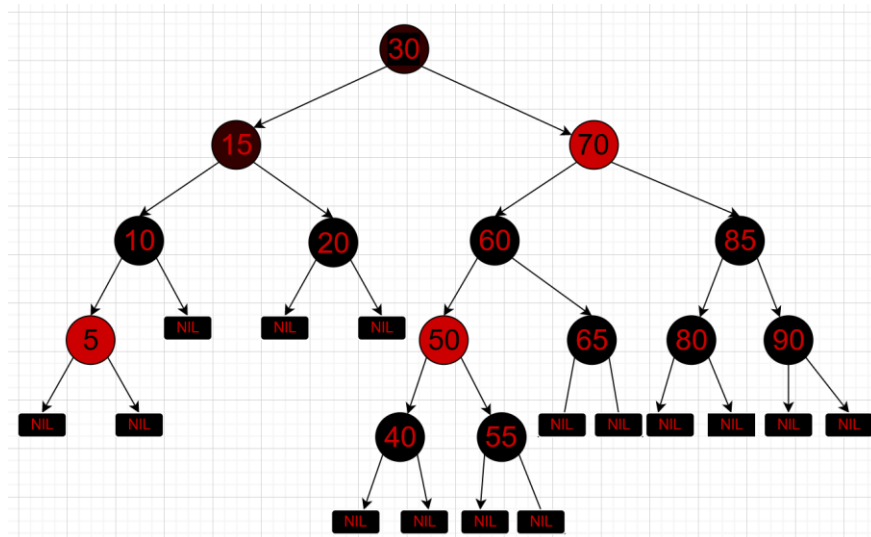


Figura 3: Albero rosso-nero

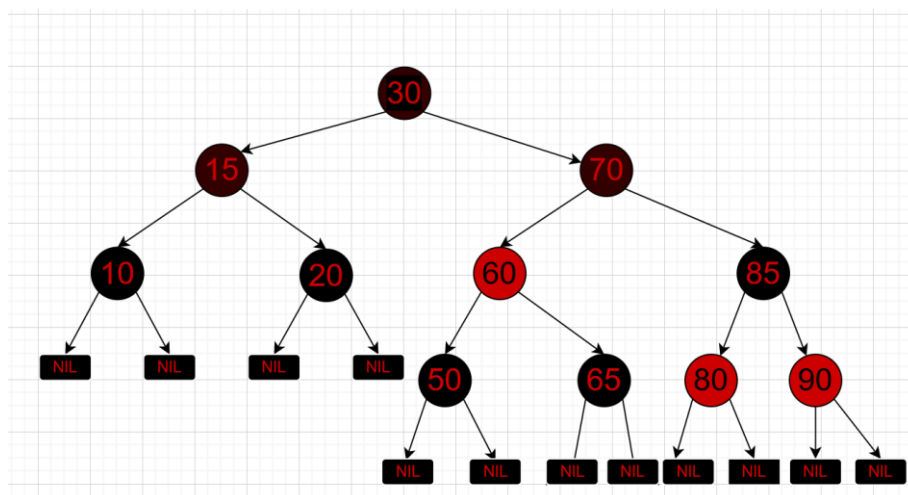


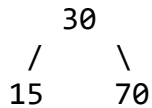
Figura 4: Albero rosso-nero

## 2. COSTRUZIONE

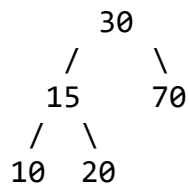
Proviamo a costruire un albero rosso-nero partendo da una sequenza di numeri:

**30, 15, 70, 10, 20, 60, 85, 50, 65, 80, 90**

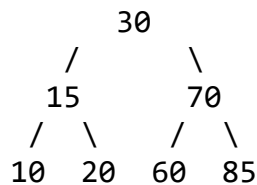
Costruiamo intanto l'albero di ricerca:



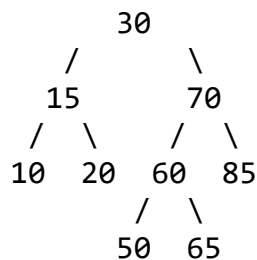
Inseriamo 30 come radice ed i successivi 15 e 70 come figlio sinistro e destro. Inseriamo ora 10 e 20 come figlio sinistro e destro di 15:



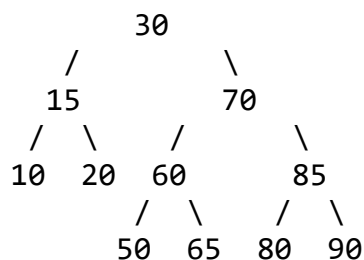
Inseriamo 60 ed 85:



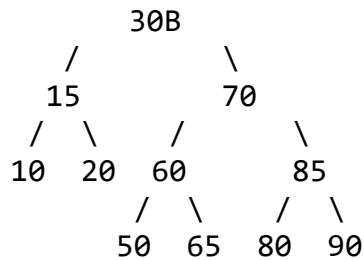
Inseriamo 50 e 65:



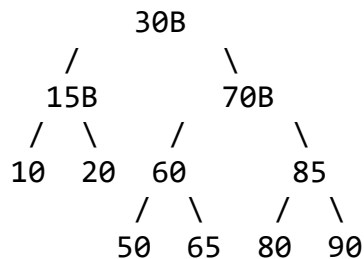
Inseriamo 80 e 90:



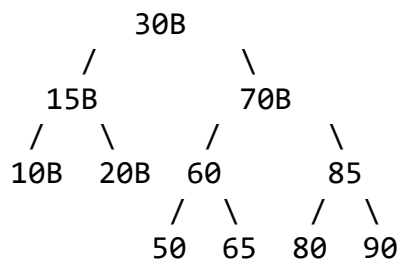
Inseriamo ora le etichette R e N ragionando nella seguente modalità: radice nera e foglie nere (sono NIL e sono state omesse):



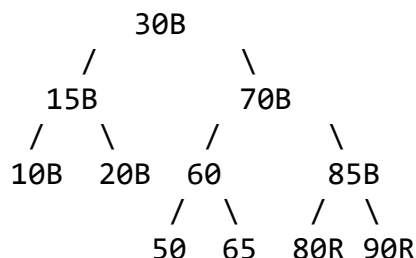
Se scegliamo per 15 e 70 il colore nero:



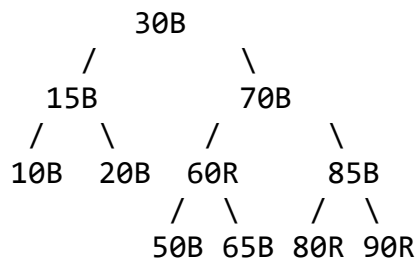
A questo punto ragiono sul 10: se inserisco il nero in 10 devo farlo anche sul 20, dovendo garantire che il numero di neri da radice a foglia deve essere lo stesso:



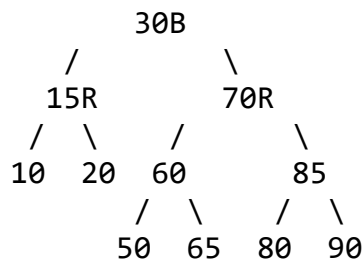
A questo punto posso inserire per 85 il nero e direttamente rosso per 80 e 90 in modo da garantire che il numero di neri da radice a foglia nel sotto albero destro sia esattamente lo stesso del sotto albero sinistro, ed in modo che i figli di 80 e 90 siano entrambi rossi (le foglie NIL):



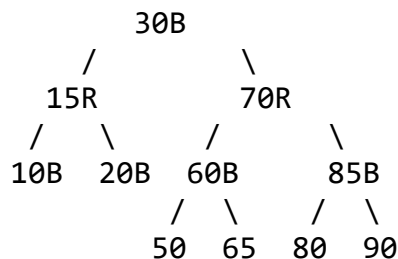
A questo punto se 60 fosse nero, avrei esaurito il numero di neri a disposizione per il sotto albero relativo ma non potrei continuare con soli rossi (dovendo mettere come vincolo dei figli neri per un nodo rosso), pertanto il nodo 60 è rosso, 50 e 65 sono di conseguenza neri:



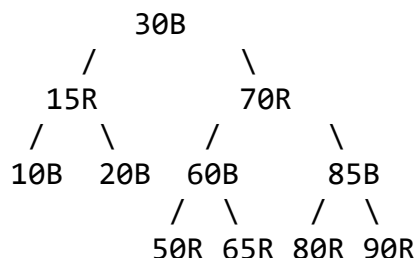
Tuttavia questa non è l'unica configurazione possibile; se infatti scegliessi per 15 e 70 la colorazione rossa potrei comunque costruire un albero rosso-nero:



I nodi 10, 20, 60 ed 85 dovrebbero essere neri:



E necessariamente i nodi 50, 65, 80 e 90 devono essere settati come rossi per rispettare il vincolo del numero dei neri dalla radice alle foglie:





### 3. ANALISI ALBERO ROSSO-NERO

Proviamo ora ad analizzare il seguente albero e cerchiamo di capire se può essere un albero rosso-nero:

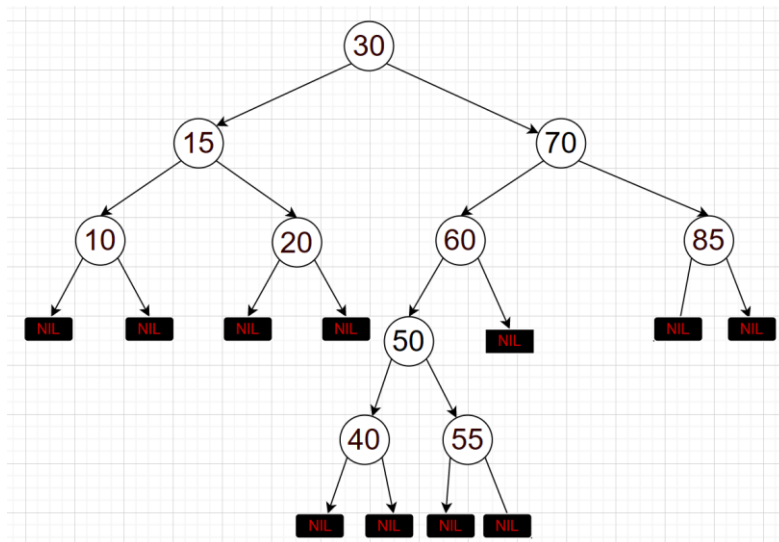
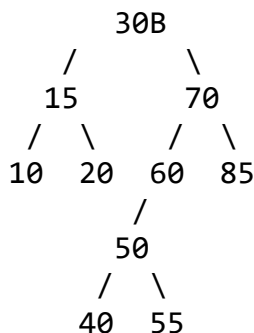
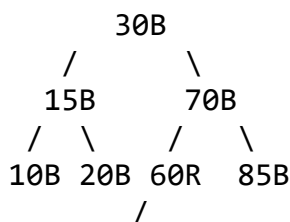


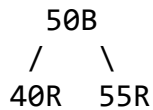
Figura 5: Analisi albero di ricerca

Iniziamo con il settaggio della radice:

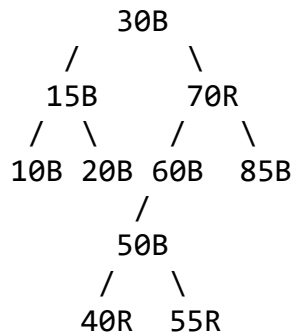


Analizziamo il sottoalbero contenente il nodo 50: qui posso al più inserire 2 neri infatti se ve ne fossero 3 la regola rosso-nero non sarebbe soddisfatta per il sottoalbero contenente il nodo 15 (che al più può avere 2 nodi neri) quindi forzatamente uno dei nodi tra 60 e 70 deve essere rosso, così come tra 40 e 50. Proviamo con 70 nero:





Tuttavia, abbiamo un problema con il nodo 60 infatti il cammino che dalla radice 30 termina al figlio destro del nodo 60 (che ricordiamo, è un NIL) avrebbe un solo nodo nero (il 70) mentre tutti gli altri (correttamente) ne hanno 2 (sempre al netto della radice); quindi tale configurazione non va bene; proviamo con il 70 rosso:



Ma anche in questo caso non va bene perché il cammino dalla radice 30 al nodo 85 conterrebbe un solo nero (l'85) mentre tutti gli altri (correttamente) ne hanno 2 (sempre al netto della radice).

Ne consegue dunque che questo albero non può essere rosso-nero.

## Bibliografia

- Alan Bertossi, Alberto Motresor: Algoritmi e strutture di dati, Città Studi Edizioni, terza edizione;
- C. Demetrescu, I. Finocchi, G. F. Italiano: Algoritmi e strutture dati, McGraw-Hill, seconda edizione;
- Crescenzi, Gambosi, Grossi: Strutture di Dati e Algoritmi, Pearson/Addison-Wesley;
- Sedgewick: Algoritmi in C, Pearson, 2015;
- Cormen Leiserson Rivest Stein-Introduzione Agli Algoritmi E Strutture Dati-Prima Edizione.