

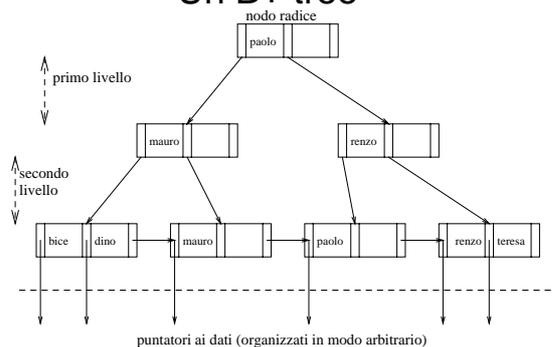
B+ Tree

B+ Tree

- B+ Tree:
 - Le foglie sono collegate in una lista
 - I puntatori ai dati sono **solo** nelle foglie
 - L'inserimento delle chiavi avviene nei nodi foglia
 - Ottimi per le ricerche su intervalli
 - Molto usati nei DBMS

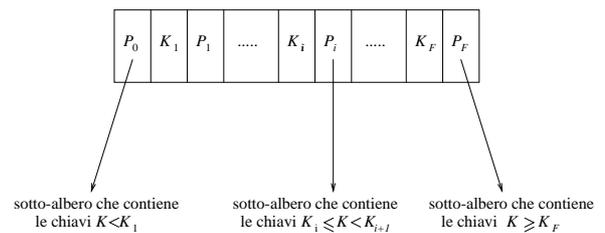
2

Un B+ tree



3

Organizzazione dei nodi



4

Inserimento di una chiave

- Non provoca problemi quando la nuova chiave si può inserire in una **foglia** non satura
- Quando la foglia non ha spazio disponibile, si esegue lo SPLIT della foglia
 - Si suddivide l'informazione presente nella foglia e la nuova informazione in due (in modo equilibrato) allocando due foglie al posto di una
 - Il valore mediano delle chiavi si **inserisce anche** (in copia) nel padre
- SPLIT causa il crescere del numero dei puntatori al livello superiore
 - Lo SPLIT può dover essere ripetuto sul nodo padre
 - Lo SPLIT del padre (nodo non foglia) viene gestito come per i BTree
 - Si suddivide l'informazione in due, mettendo le chiavi maggiori del valore mediano nel nuovo nodo
 - Il valore mediano delle chiavi viene **spostato** nel padre

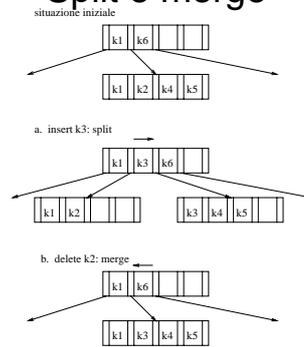
Rimozione di una chiave

- Se la rimozione della chiave lascia due nodi **foglia** con spazio inutilizzato da consentire che tutta l'informazione in esse presente possa essere concentrata in una sola foglia
 - Operazione di MERGE
 - Richiede la modifica della disposizione dei puntatori
 - MERGE causa il decrescere del numero dei puntatori al livello superiore
 - MERGE può dover essere ripetuto sul nodo padre

Bilanciamento

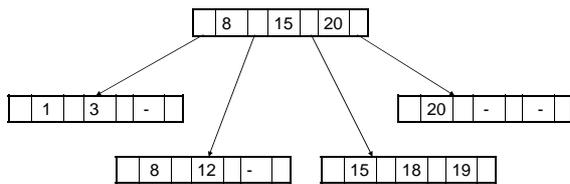
- Se le modifiche causano differenze tra le lunghezze dei cammini tra radice e foglie
 - Opportuno ri-bilanciare l'albero

Split e merge



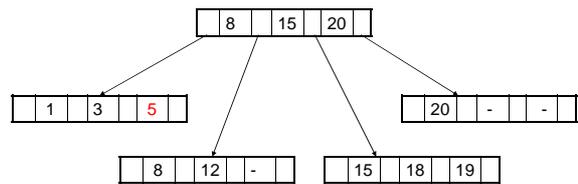
8

B+-Tree Fan-Out = 4 (ordine=4)
Alberi bilanciati (cammino costante tra radice e foglie)

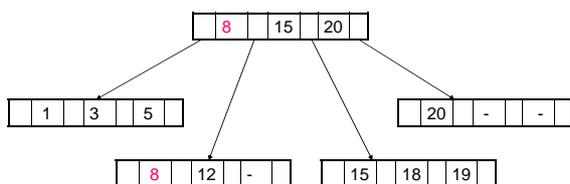


NB. non rappresentiamo i collegamenti tra le foglie

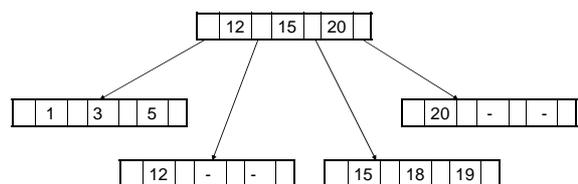
Inserimento di k=5
(Inserimento in foglia NON satura)



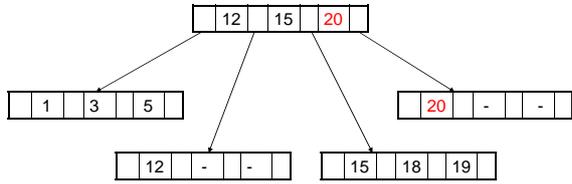
Rimozione di k=8



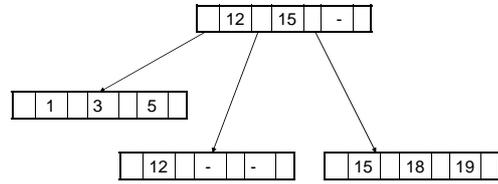
Rimozione di k=8



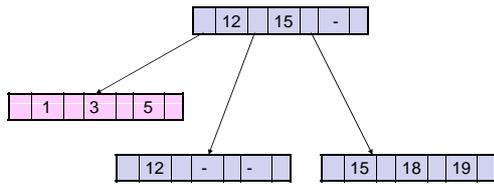
Rimozione di k=20



Rimozione di k=20

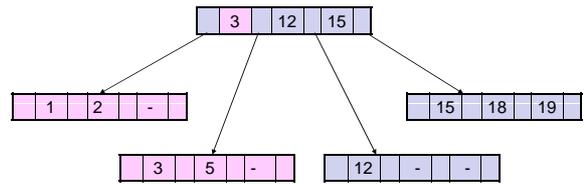


Inserimento di k=2
(Inserimento in foglia saturata)



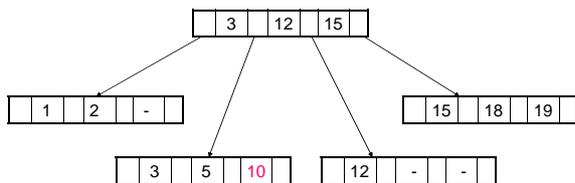
1 - 2 - 3 - 5
Necessario SPLIT della foglia

Inserimento di k=2

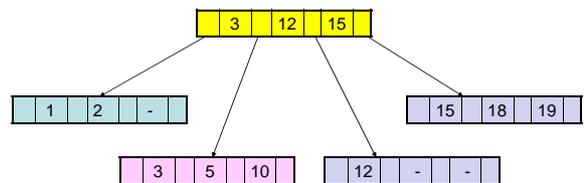


Suddivisione dell'informazione 1 - 2 - 3 - 5 in due 1 - 2 e 3 - 5
Inseriamo 3 anche nel padre, facendo crescere di uno il numero dei puntatori
=> chiavi nel padre 3 - 12 - 15

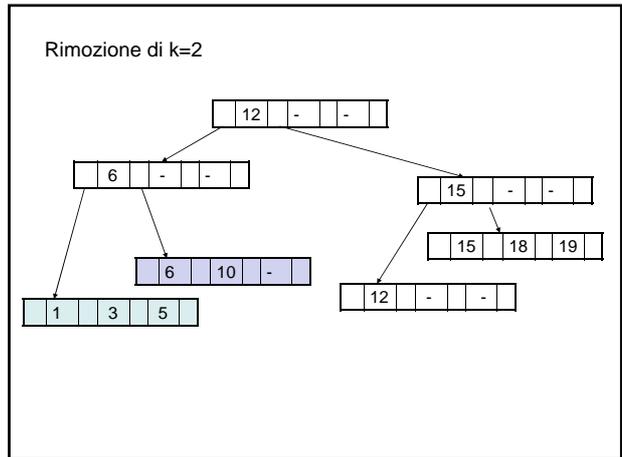
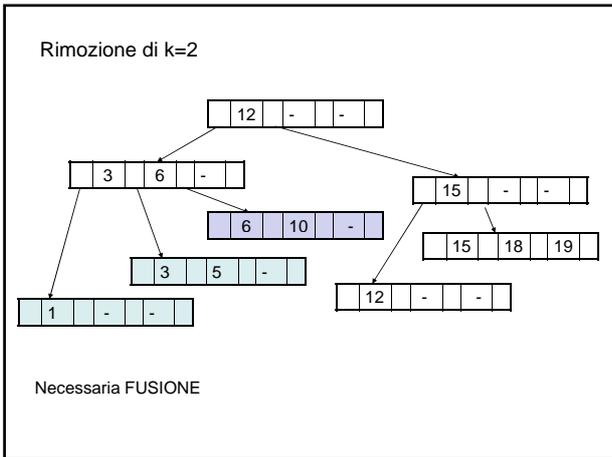
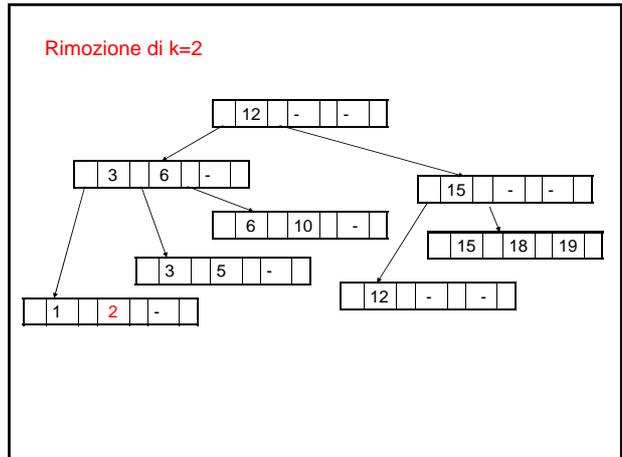
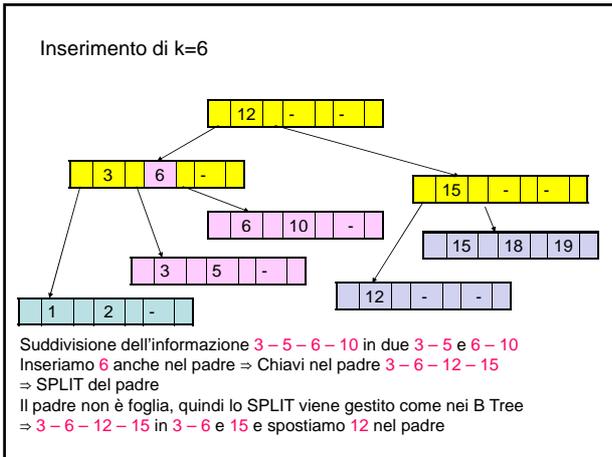
Inserimento di k=10



Inserimento di k=6
(Inserimento in foglia saturata con SPLIT ripetuto sul padre)



3 - 5 - 6 - 10
Necessario SPLIT della foglia



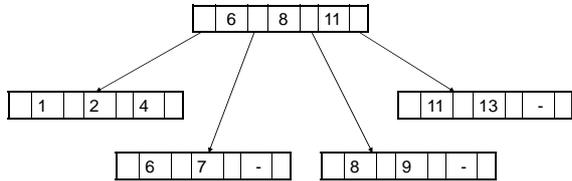
Esercizio

- Data la seguente lista di valori chiave $L=(1,6,2,9,4,7,11,8,13)$

a) costruire un possibile B⁺-tree (fan-out=4) che contenga i seguenti nodi foglia: (1,2,4), (6,7), (8,9), (11,13);

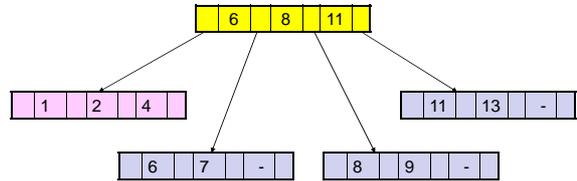
b) mostrare l'albero dopo l'inserimento del valore chiave 3.

B+-Tree Fan-Out = 4 (ordine=4)



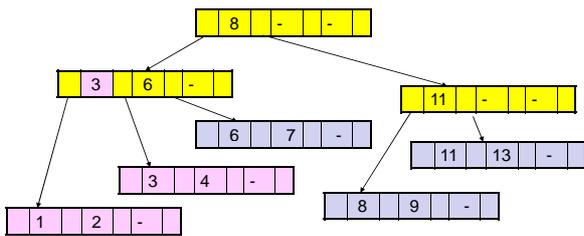
NB. non rappresentiamo i collegamenti tra le foglie

Inserimento di k=3



1-2-3-4
Necessario SPLIT della foglia

Inserimento di k=3



Suddivisione dell'informazione della foglia 1-2-3-4 in due 1-2 e 3-4
 Aggiungiamo 3 al padre, facendo crescere di uno il numero dei puntatori
 => SPLIT del padre (nodo non foglia)
 => 3-6-8-11 in due 3-6 e 11 e spostiamo 8 nel padre