

Inserimento in B-tree (ordine m)

- Si verifica se la chiave è presente, altrimenti si scende alla foglia appropriata
- Se il nodo non è completo (cioè contiene meno di m-1 chiavi), allora vi si può inserire il record
- Se il nodo è completo, si crea un altro nodo (splitting) e si considera il valore mediano tra le chiavi del nodo completo più la chiave da inserire:
 - il valore mediano si inserisce nel nodo padre
 - se il padre è completo si esegue la suddivisione (splitting) del padre (processo ricorsivo)

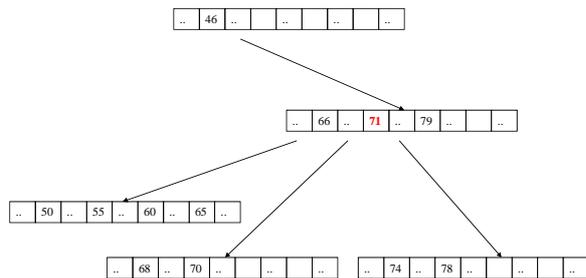
Inserimento in nodo saturo

- Se il nodo p' è saturo, cioè contiene m-1 chiavi, si esegue lo **splitting** del nodo p'
 - si crea un altro nodo p'' in cui si memorizzano le chiavi maggiori della chiave di **valore mediano** (oppure di posto mediano) k_c nella sequenza (considerando chiavi del nodo e chiave da inserire) togliendole dal nodo originario
 - la chiave di valore mediano k_c si inserisce nel nodo padre, o meglio si inserisce

$$\dots p'(k_c, r_{k_c}) p'' \dots$$
 e se il padre è completo si esegue la suddivisione (splitting) del padre e così via fino alla radice.

Inserimento in B-tree: splitting (inserzione della chiave 70)

$68 + 70 + 71 + 74 + 78 = 361$ valore mediano = 71

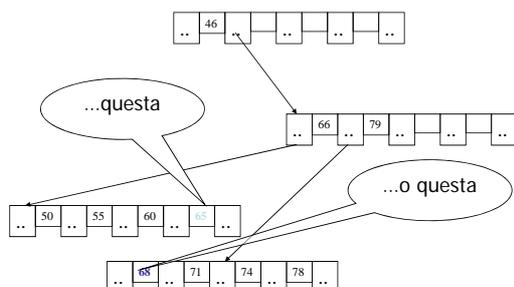


Cancellazione da B-tree di chiave k

- P0. la chiave k da cancellare sia nel nodo n;
 P1. se n è una foglia, si cancella k da n e si va a P3
 P2. se n NON è una foglia si sostituisce k prendendo il più piccolo (grande) fra gli elementi maggiori (minori) di k; tale chiave sostitutiva è in una foglia f, per costruzione dell'albero, e quindi le va applicato il punto P1.
 P3. controllo numero chiavi nella foglia
 se $\lceil m/2 \rceil - 1 \leq \text{numero chiavi} \leq m-1$
 allora fine cancellazione
 altrimenti bisogna P4) **concatenare** o P5) **bilanciare**

Cancellazione in B-tree

(cancellazione della chiave 66: al suo posto...)



Concatenazione o Merge (P4)

- Unisce due nodi in un solo nodo
 Numero di chiavi in n (al più) = $(m+1)/2 - 2$
- Cerco un nodo q adiacente ad n nell'albero tale che
 Numero chiavi in q (al più) = $(m+1)/2 - 1$
- Considero che una chiave verrà dal padre: +1
- Sommando il numero di chiavi ottengo

$$(m+1)/2 - 2 + (m+1)/2 - 1 + 1 = m + 1 - 3 + 1 = m - 1$$
 (m-1 è il massimo numero di chiavi possibile)

Concatenazione (cont.)

nodo padre: [... (k_{j-1}, r_{j-1}) p_{j-1} (k_j, r_j) p_j (k_{j+1}, r_{j+1}) p_{j+1} ...]

nodo-1 da conc: [p₀(k₁, r₁) p₁ ... p_{e-1}(k_e, r_e) p_e]

nodo-2 da conc: [p'₀(k_{e+1}, r_{e+1}) ...]

p_{j-1} e p_j sono puntatori ai nodi 1 e 2

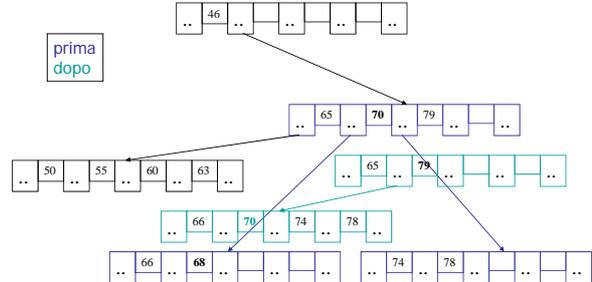
i nodi 1 e 2 sono concatenati in

[p₀(k₁, r₁) p₁ ... p_{e-1}(k_e, r_e) p_e (k_j, r_j) p'₀(k_{e+1}, r_{e+1}) ...]

e nel nodo padre: [... (k_{j-1}, r_{j-1}) p_{j-1} (k_{j+1}, r_{j+1}) p_{j+1} ...]

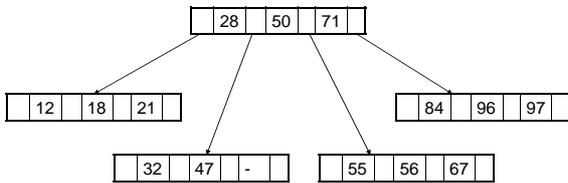
Cancellazione in B-tree: concatenazione

B-tree di ordine 5 per cui: 2 ≤ num. chiavi ≤ 4
cancellazione della chiave 68



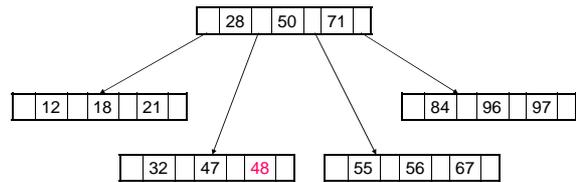
Esercizio su B-Tree

- Dato il seguente B-Tree (Fan-Out = 4) eseguire le operazioni indicate mantenendo l'albero bilanciato.



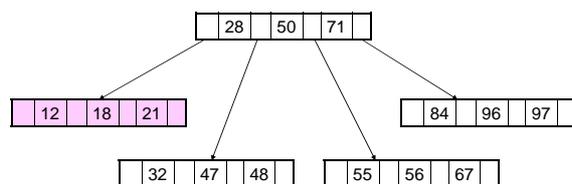
Inserimento di k=48

(Inserimento a nodo NON saturo)



Inserimento di k=8

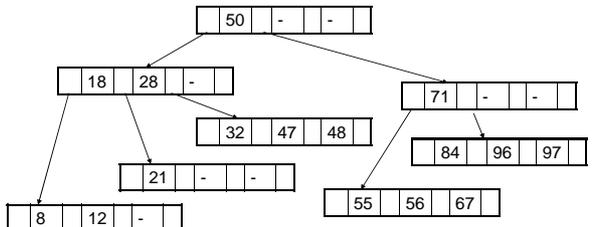
(Inserimento a nodo saturo)



8 – 12 – 18 – 21
Necessario SPLIT del nodo

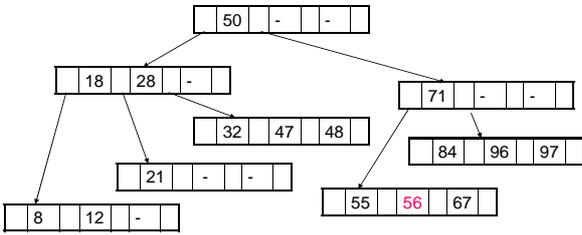
Inserimento di k=8

(Inserimento a nodo saturo)

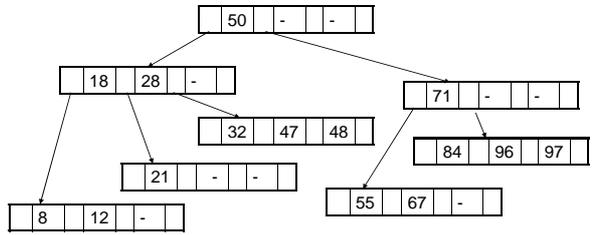


Suddivisione dell'informazione 8 – 12 – 18 – 21 in due 8 – 12 e 21
Aggiungiamo 18 al padre, facendo crescere di uno il numero dei puntatori
⇒ SPLIT del padre (18 – 28 – 50 – 71 in 18 – 28 e 71 più 50 nel padre)

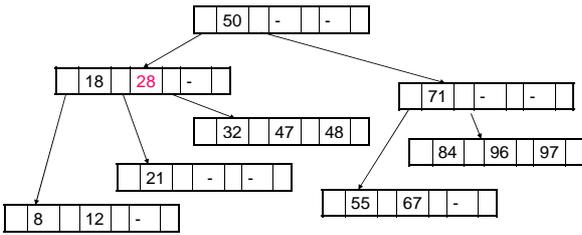
Rimozione di k=56



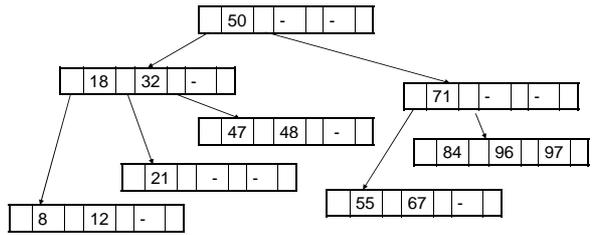
Rimozione di k=56



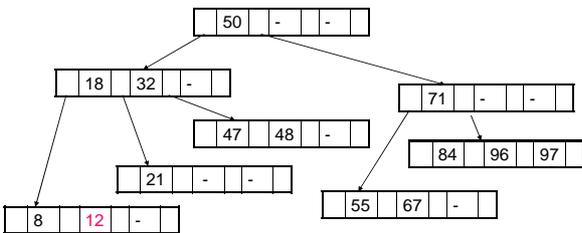
Rimozione di k=28



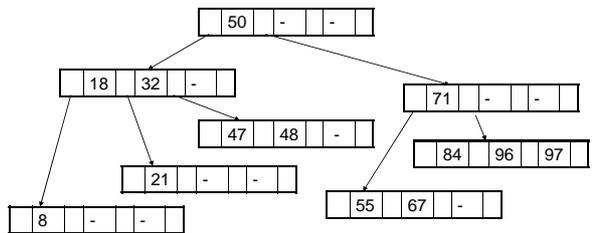
Rimozione di k=28

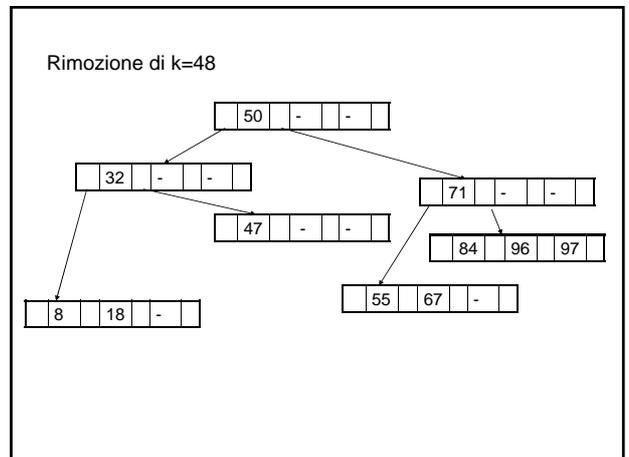
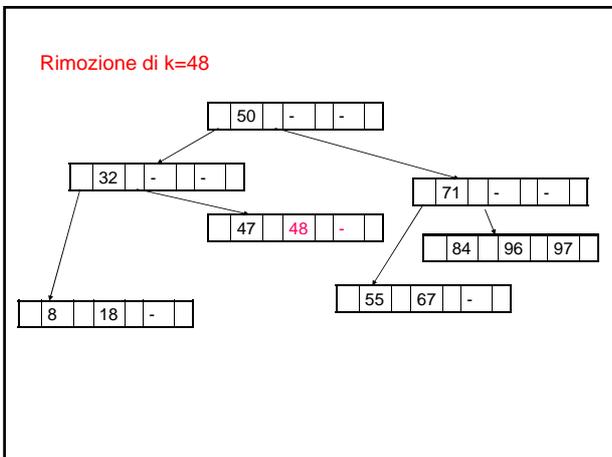
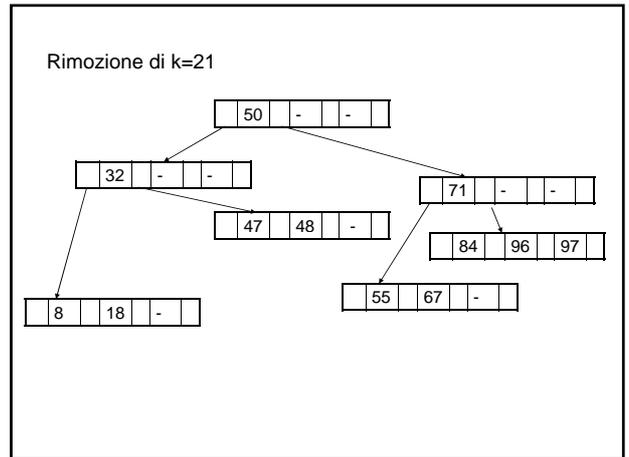
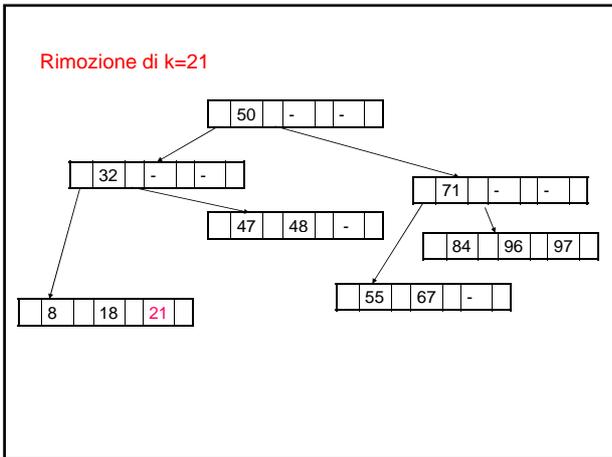
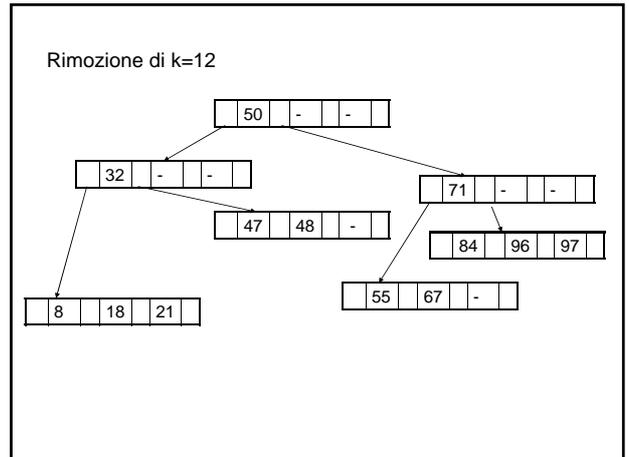
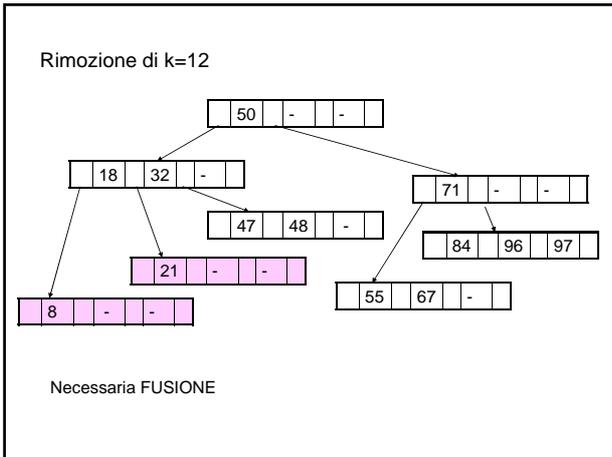


Rimozione di k=12

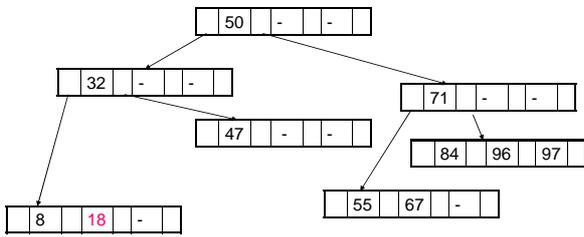


Rimozione di k=12

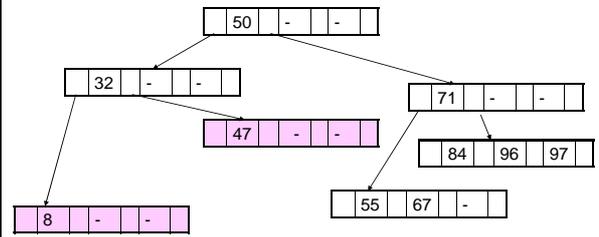




Rimozione di k=18

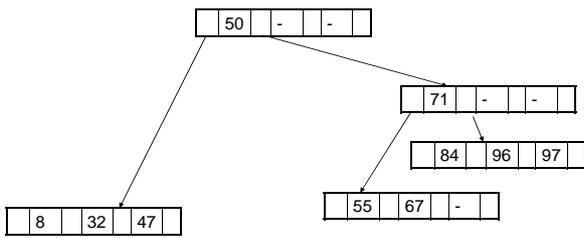


Rimozione di k=18

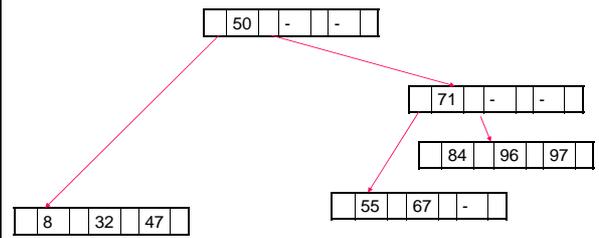


Necessaria FUSIONE

Rimozione di k=18



Rimozione di k=47



Necessaria BILANCIAMENTO

Rimozione di k=47

