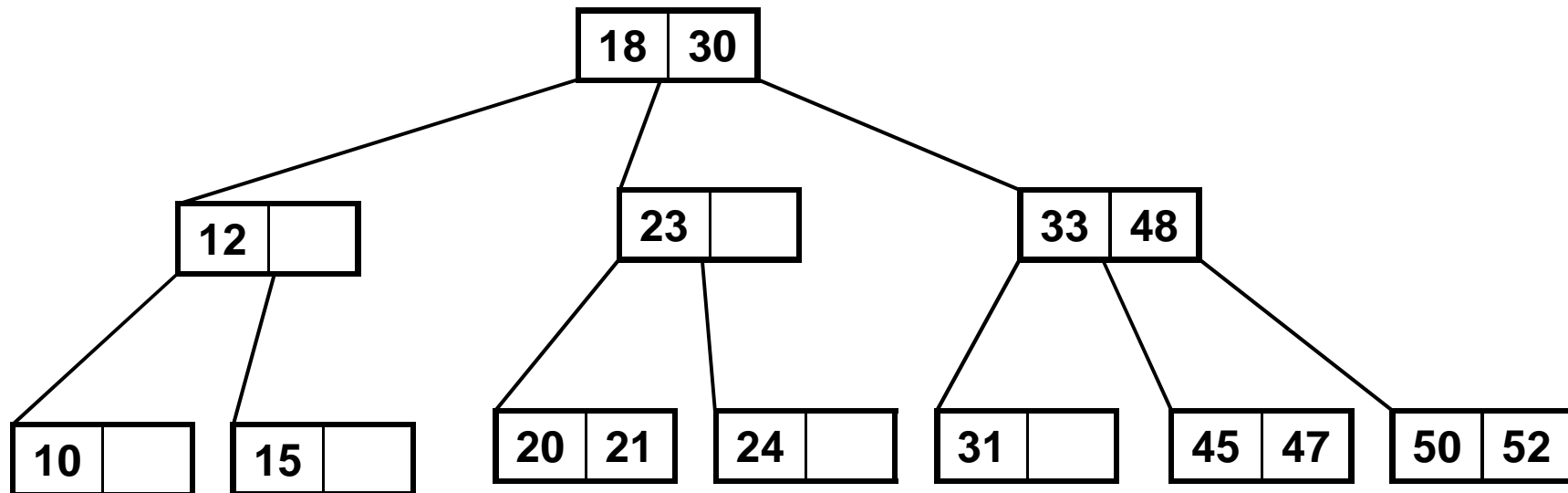


Indexing Trees

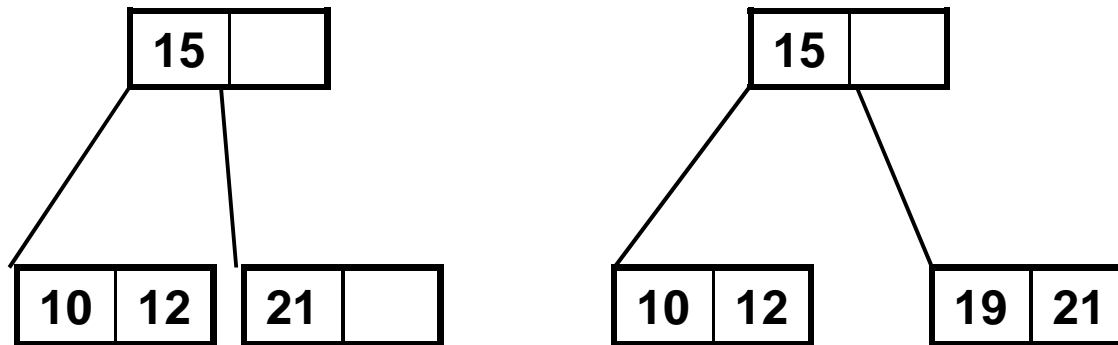
2-3 tree e B tree

- 2-3 tree
 - Numero di figli f : $2 \leq f \leq 3$
 - Se nodo pieno split dei nodi (ricorsivo)
- B tree
 - Numero di figli massimo di un nodo: m
 - Numero di figli f : $m/2 \leq f \leq m$
(Convenzione: $m/2$ in genere viene arrotondato per eccesso)
 - Ma la radice o è una foglia o puo' averne anche solo 2
 - Numero di chiavi $k = f-1$
 - quindi $m/2-1 \leq k \leq m-1$
 - Se nodo pieno split dei nodi (ricorsivo)
- In entrambi i casi non vediamo la cancellazione (troppo complicata)

Esempio 2-3 tree

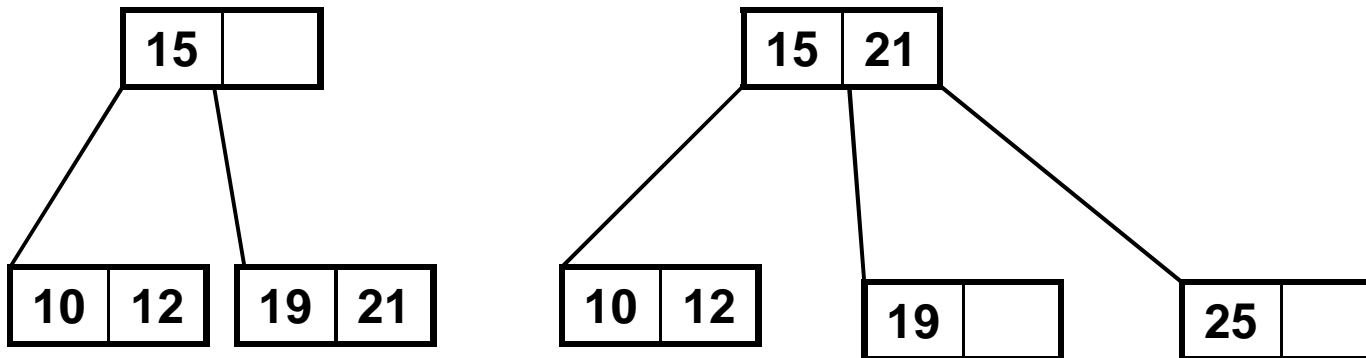


2-3 tree: inserimento senza splitting



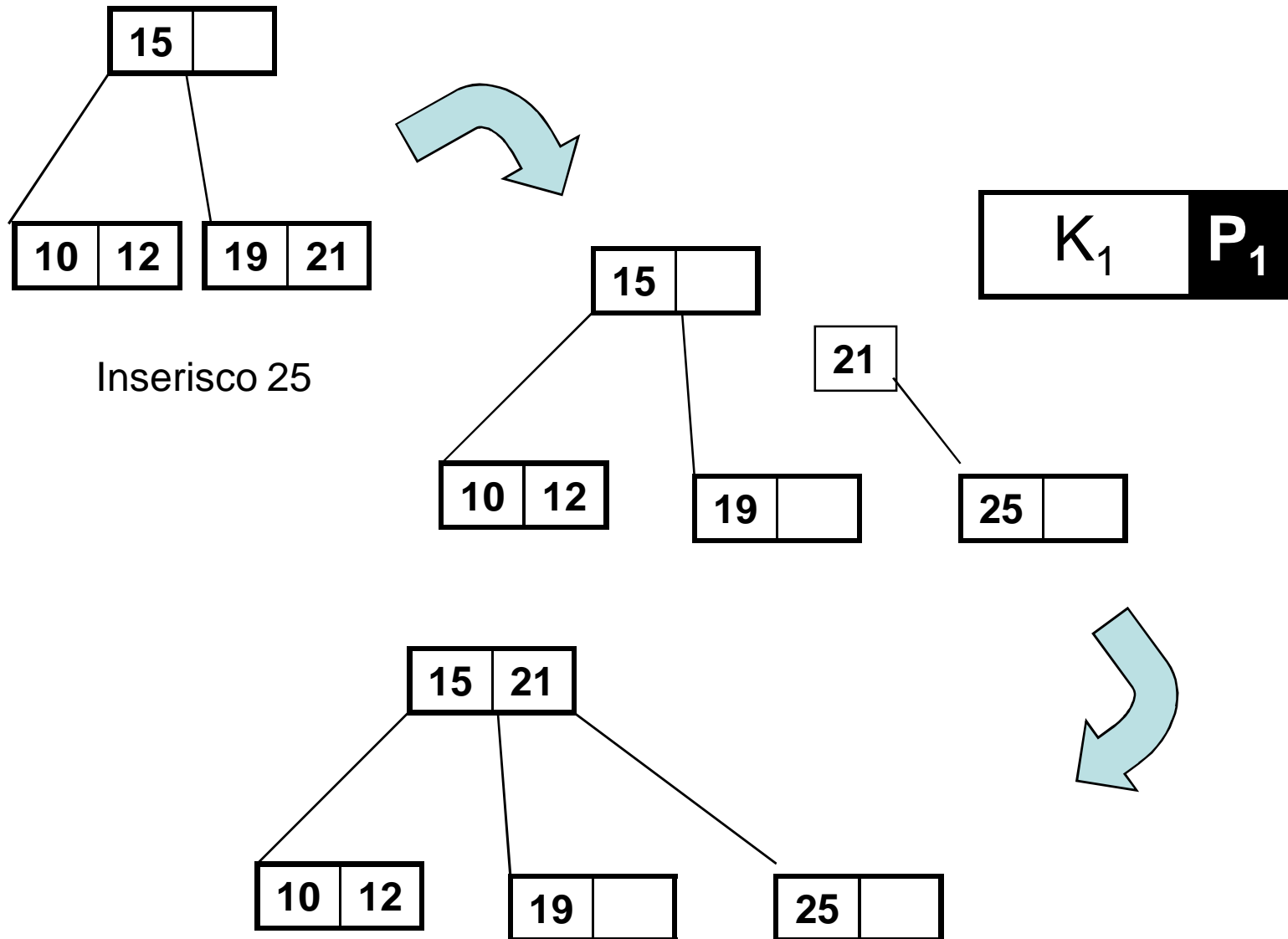
Inserisco 19

2-3 tree: splitting

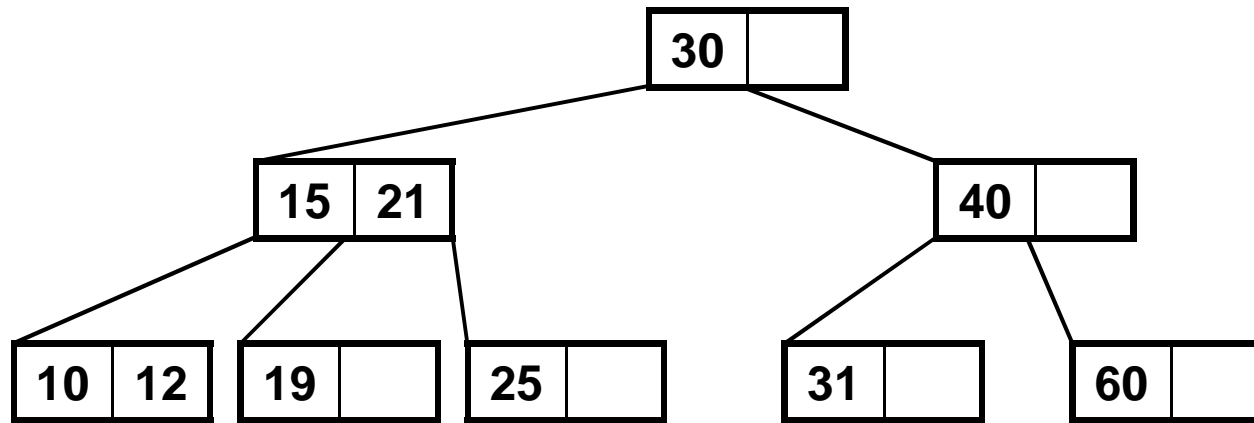


Inserisco 25

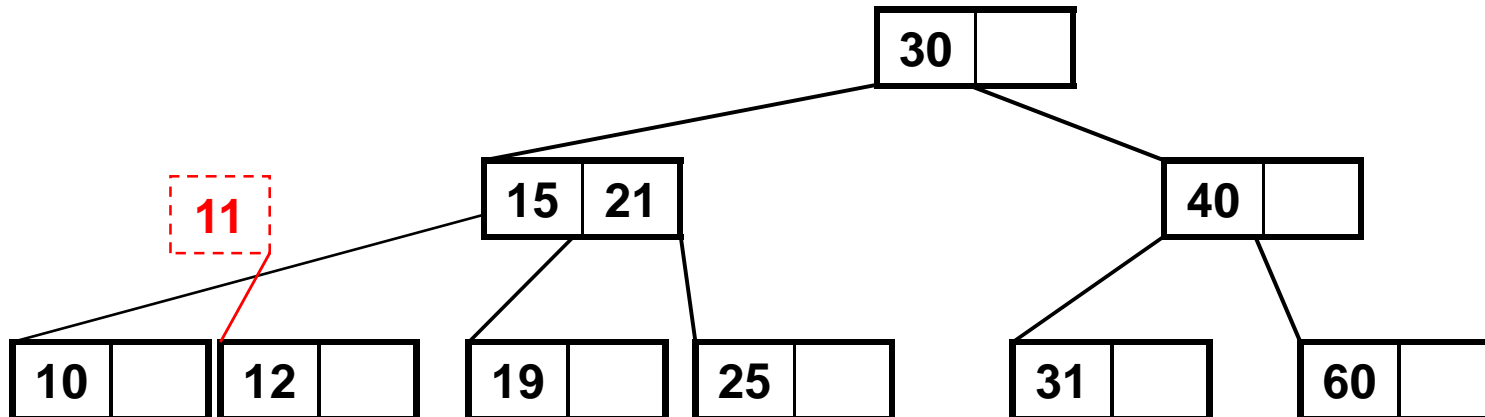
2-3 tree: splitting



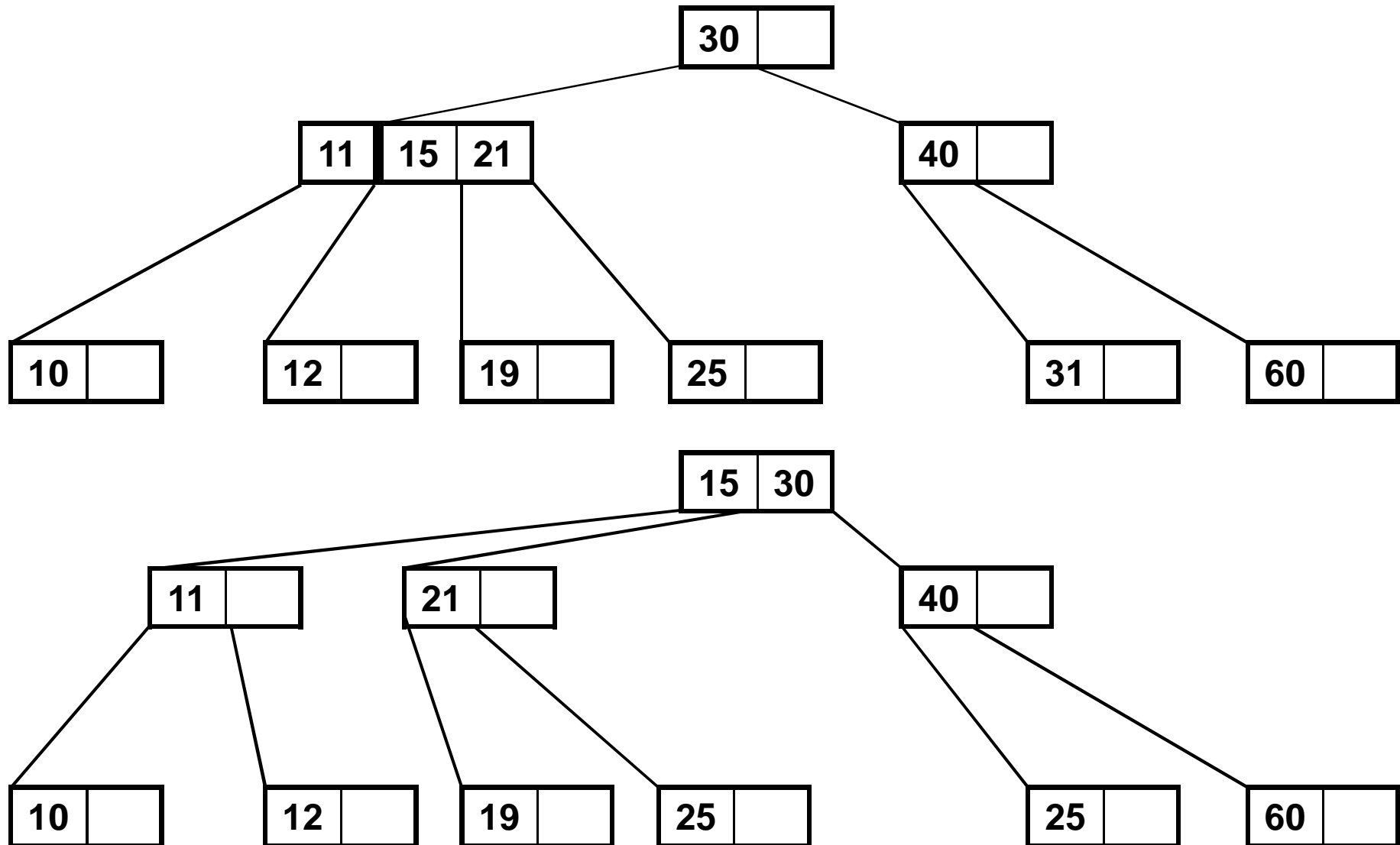
2-3 tree: splitting a catena (1)



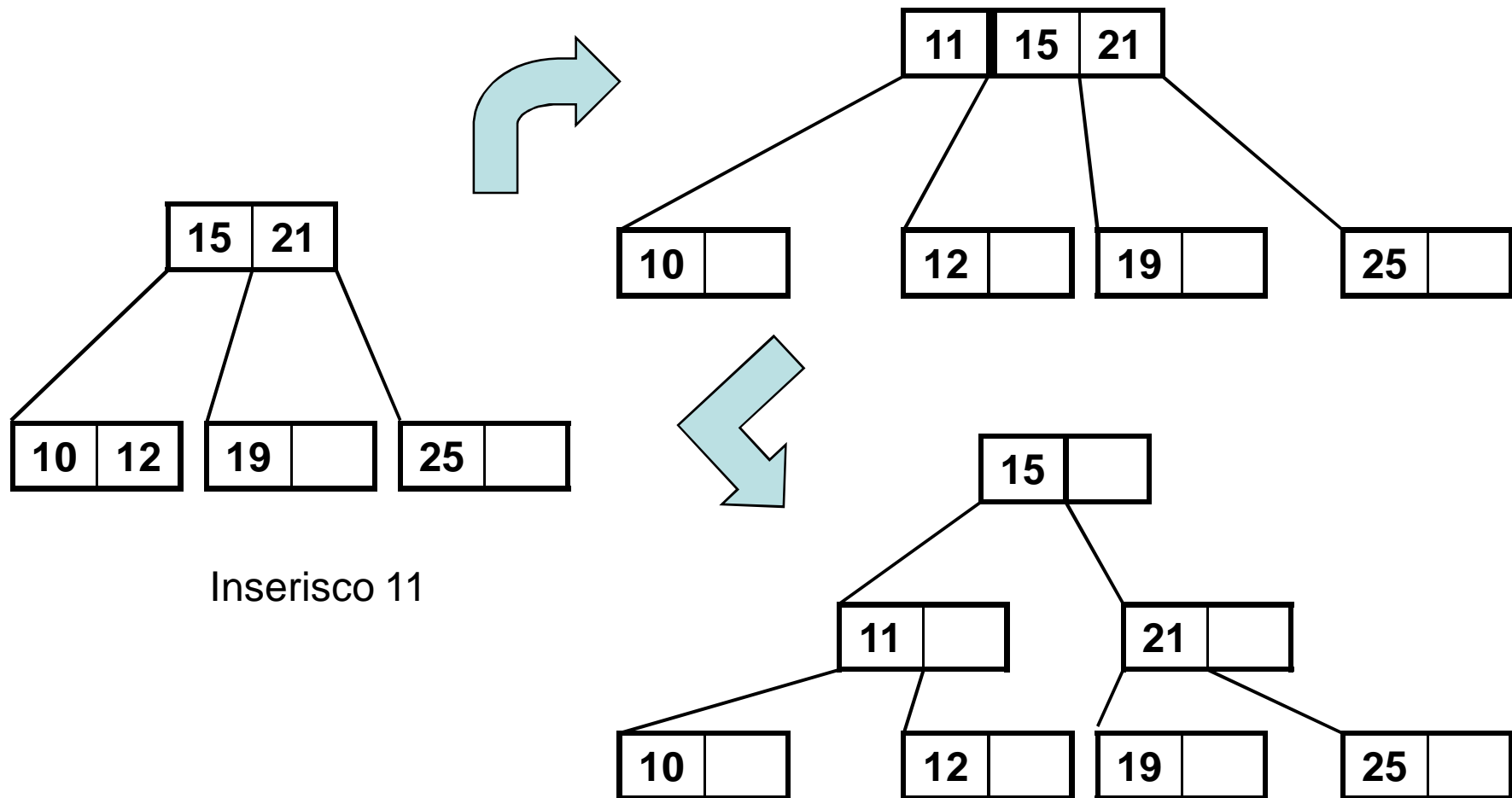
Inserisco 11



2-3 tree: splitting a catena (2)



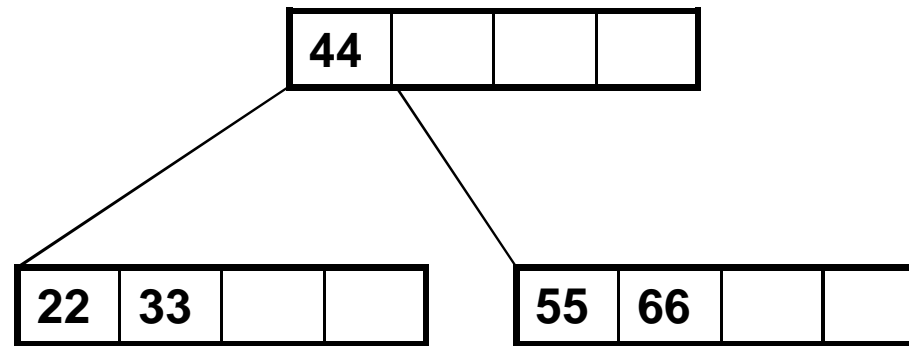
2-3 tree: splitting con aumento di livello



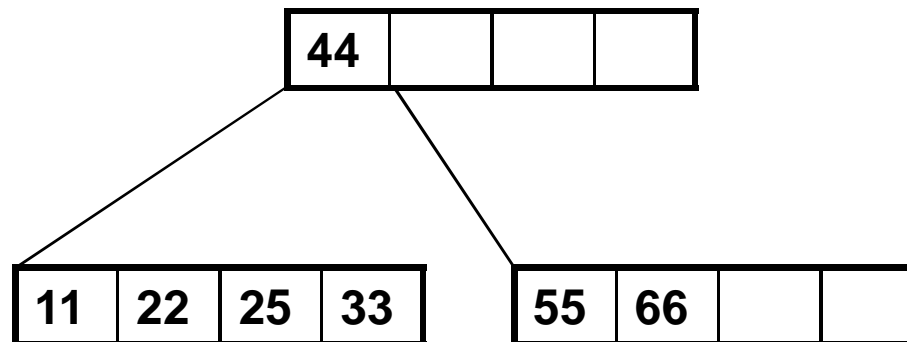
B tree: inserimento senza splitting

$m=5$

Il procedimento è identico ai 2-3 tree

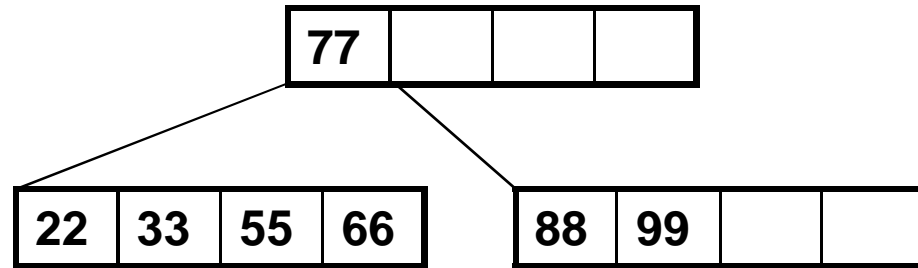


Inserisco 11, 25

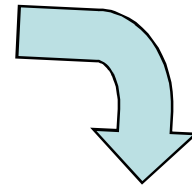


B tree: splitting

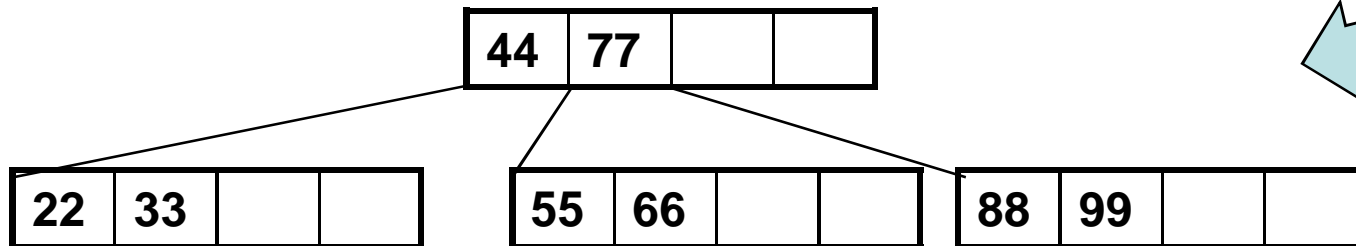
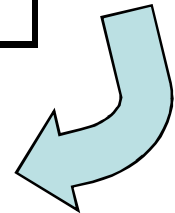
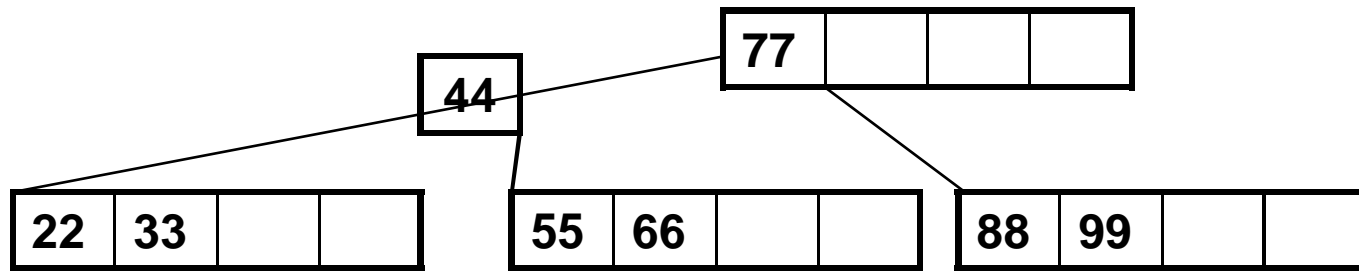
m=5



Il procedimento è identico ai 2-3 tree



Inserisco 44



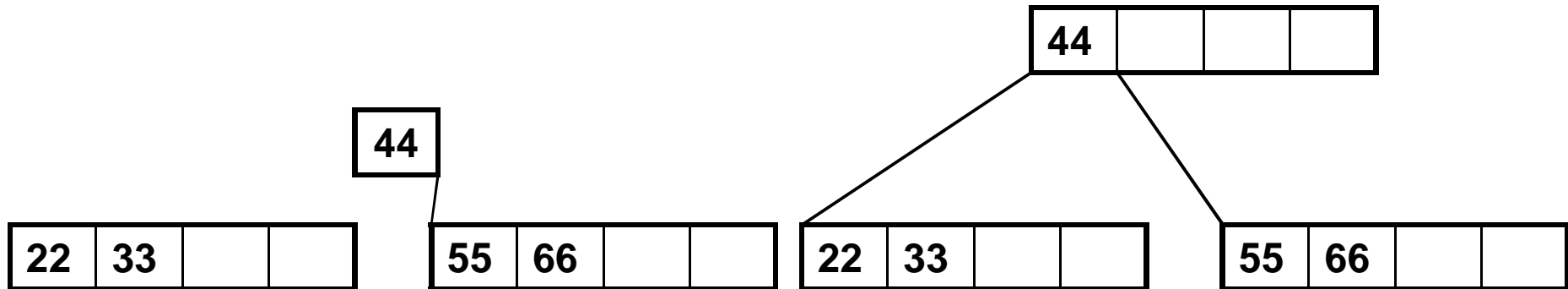
B tree: splitting con aumento di livello

$m=5$

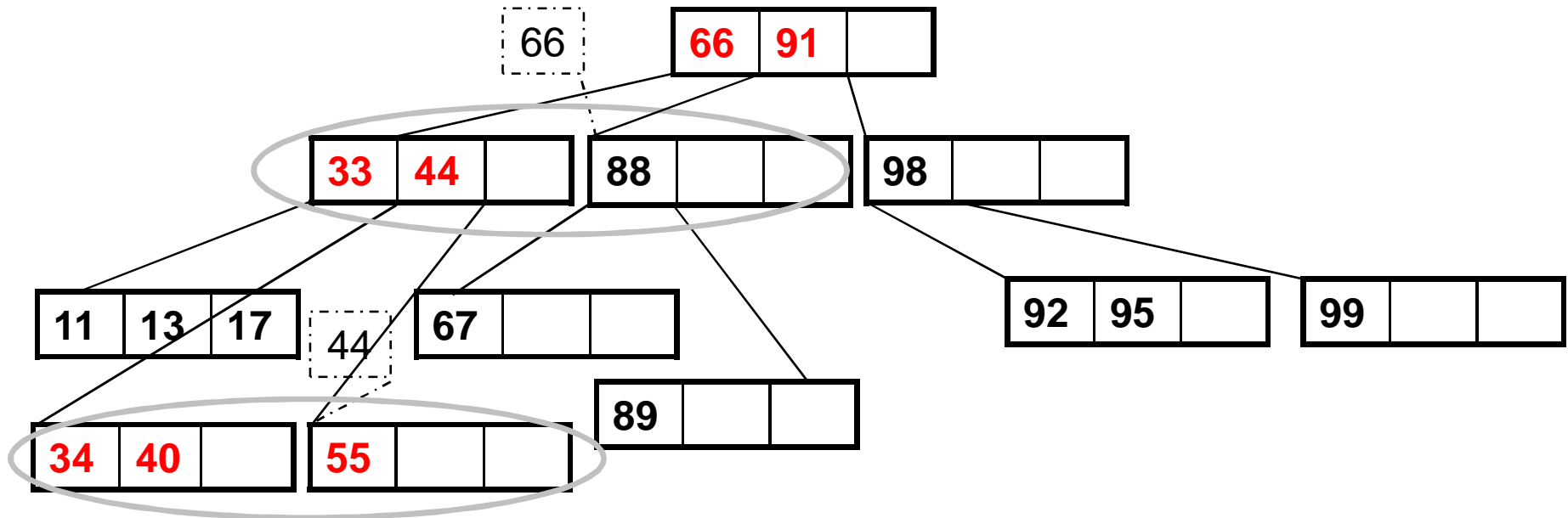
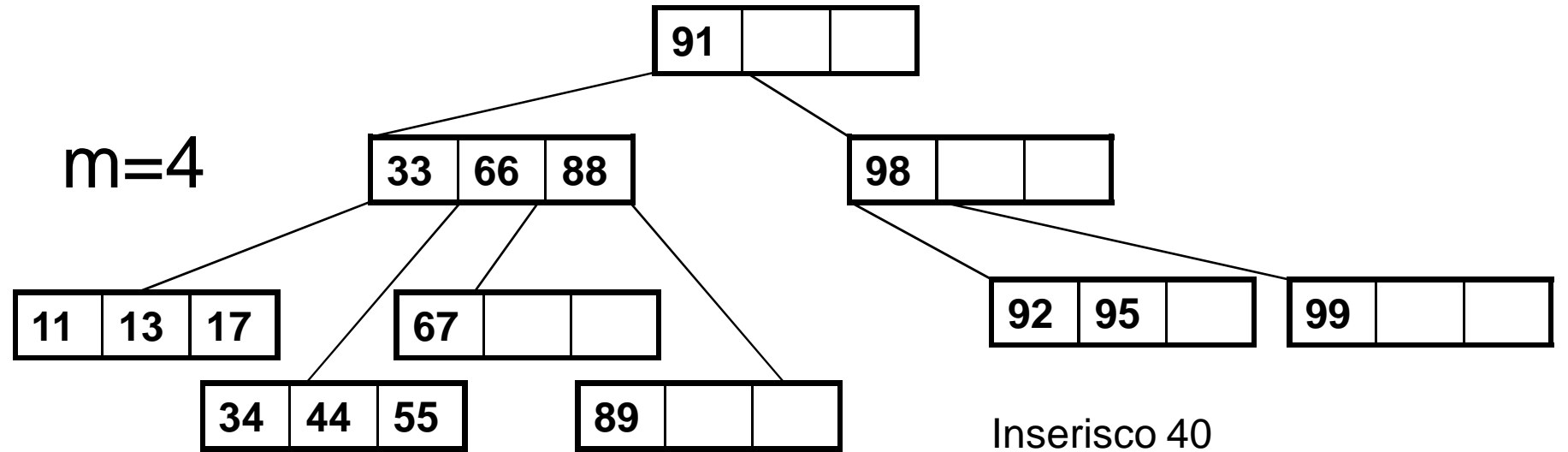
Il procedimento è identico ai 2-3 tree

22	33	55	66
----	----	----	----

Inserisco 44



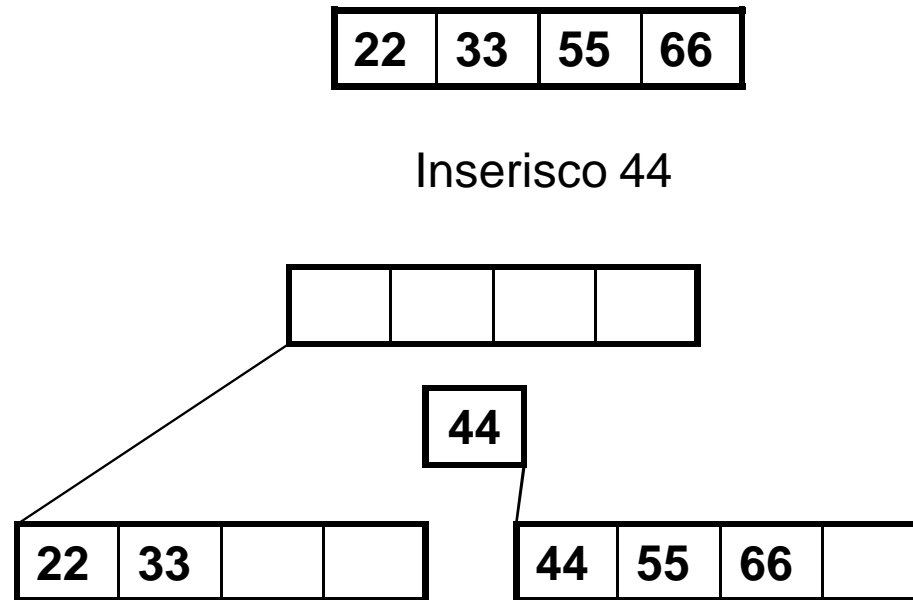
B tree: splitting a catena



B+ tree

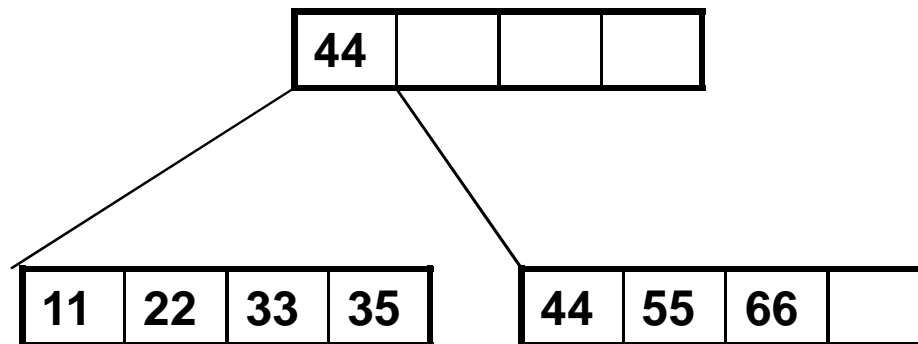
- Come i B tree ma i valori solo nelle foglie
 - I valori dei nodi intermedi servono solo a guidare le ricerche
 - Lo SPLITTING e' UGUALE ai B tree per i nodi INTERMEDI ma è DIVERSO PER LE FOGLIE: (valore che viene promosso resta anche nella foglia)
- Dettagli
 - Tutte le foglie sono linkate tra di loro
 - Le foglie potrebbero ospitare un numero di valori diverso dal numero di chiavi dei nodi intermedi (se $m=5$ i nodi intermedi hanno al max 4 chiavi, ma le foglie possono contenere ad esempio 10 valori in quel caso devono avere almeno 5 elementi)
- Cancellazione
 - Vediamo l'algoritmo di cancellazione. In particolare c'è il problema di mantenere i nodi almeno metà pieni ($m/2$ figli per i nodi intermedi, la metà per le foglie)

B+ tree: splitting di foglie (1)

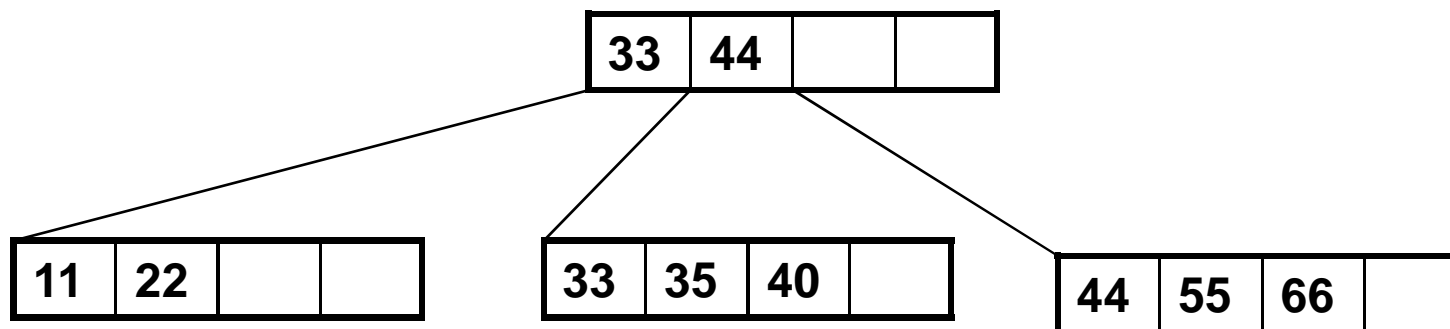


Attenzione: l'elemento resta anche nelle foglie!!!

B+ tree: splitting di foglie (2)



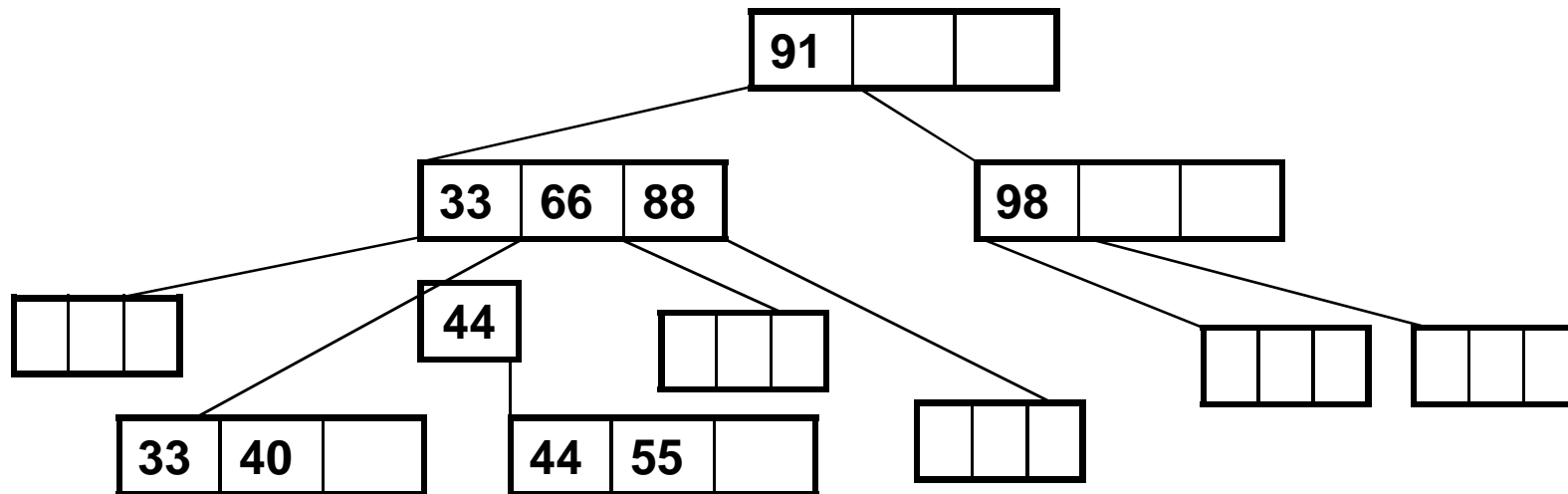
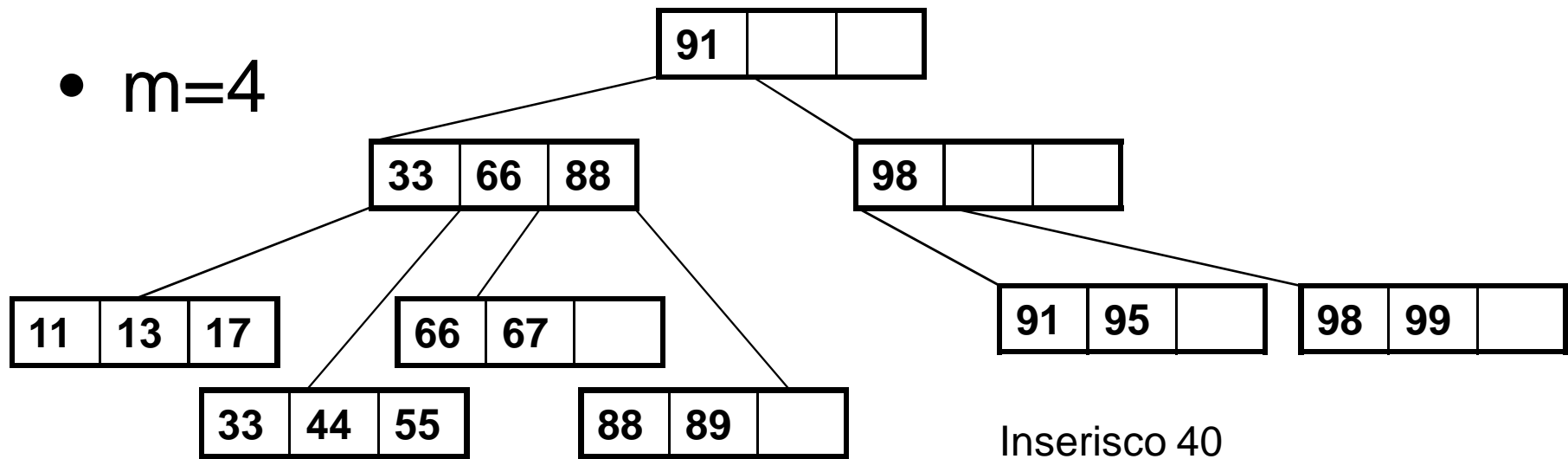
Inserisco 40



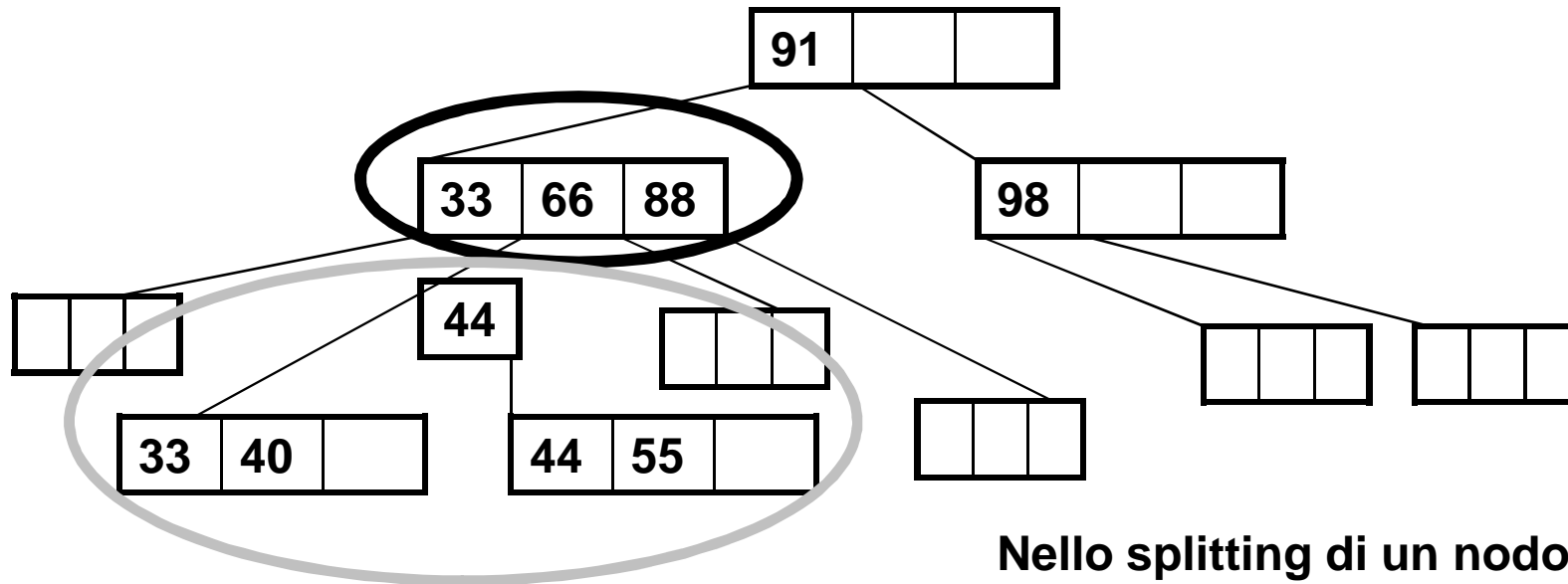
Attenzione: l'elemento resta anche nelle foglie!!!

B+ tree: splitting di nodi intermedi

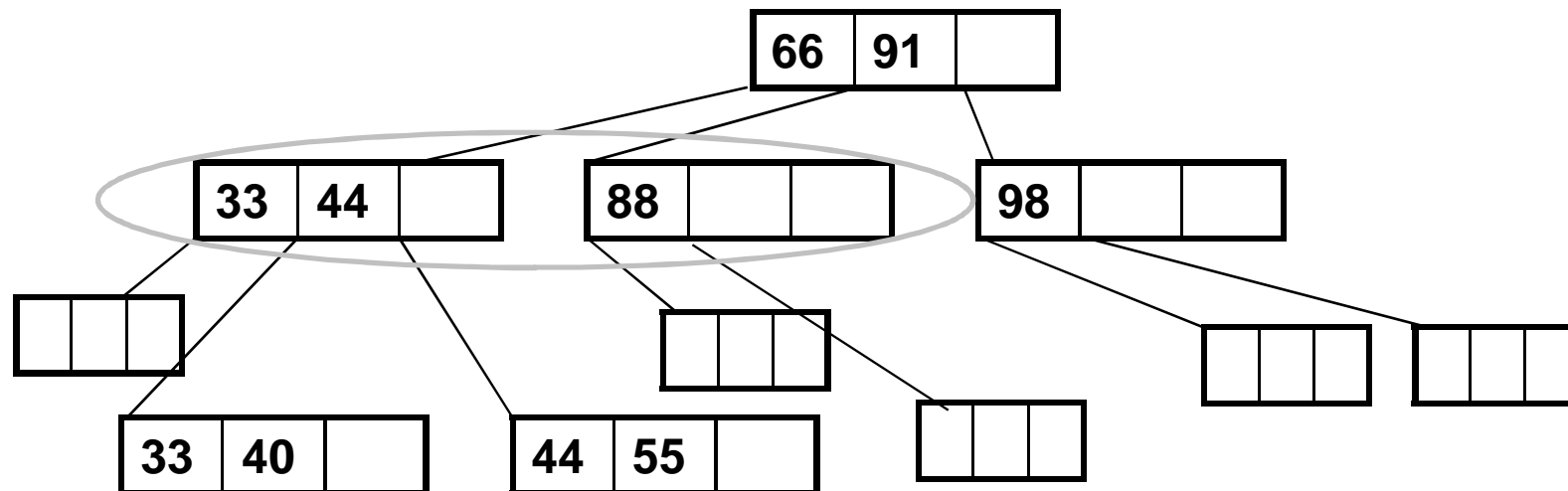
- $m=4$



B+ tree: splitting di nodi intermedi



**Nello splitting di un nodo intermedio
Invece la regola e' la stessa dei B tree**



B+ tree: cancellazione

- Se cancello un elemento e non provo underflow => OK
 - Se cancello l'elemento che è stato copiato nel padre non c'è problema, non sono obbligato a fare l'update!
- Se provo underflow
 - Chiedo in prestito elementi dai miei fratelli ricchi (ricco = puo farmi un prestito senza andare in underflow!)
 - Se non ci sono fratelli ricchi mi fondo con un mio vicino

B+ tree: risoluzione dell'underflow

- Chiedere in prestito elementi
 - Tra foglie semplice. Basta aggiornare il padre
 - Tra nodi intermedi è più complicato bisogna fare il “giro delle chiavi”
- Fusione di due nodi
 - Nel padre dei due nodi che si fondono elimino la chiave che puntava al nodo di destra (assieme al punt subito seguente)

