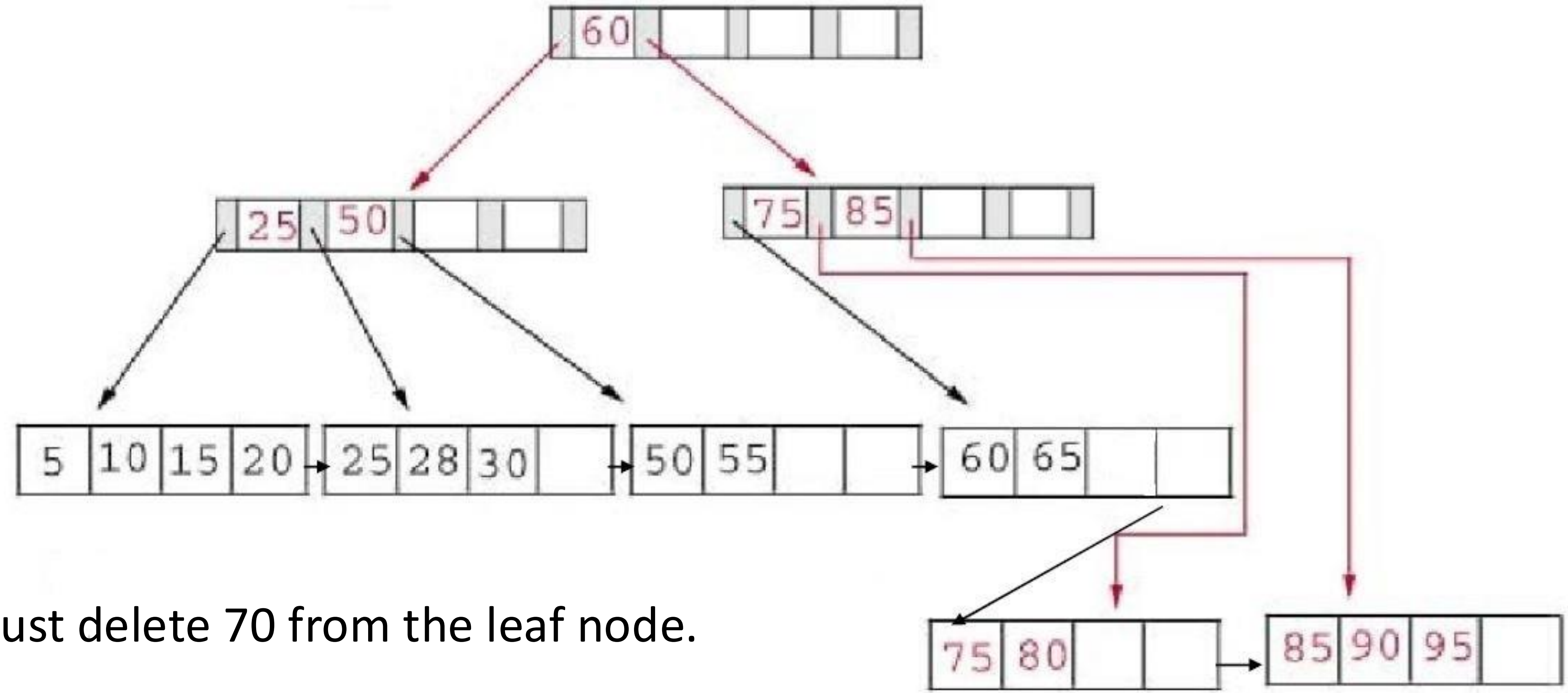
Tree data structure (16)

B+ tree index deletion

- Step1: Descend to the leaf node where the key fits
- Step 2: Remove the required key and associated reference from the node.
  - ❑ (Case 1): If the node still has half-full keys, repair the keys in parent node to reflect the change in child node if necessary and stop.
  - ❑ (Case 2): If the node is less than half-full, but left or right sibling node is more than half-full, redistribute the keys between this node and its sibling. Repair the keys in the level above to represent that these nodes now have a different “split point” between them.
  - ❑ (Case 3): If the node is less than half-full, and left and right sibling node are just half-full, merge the node with its sibling. Repeat step 2 to delete the unnecessary key in its parent.

Tree data structure (17)

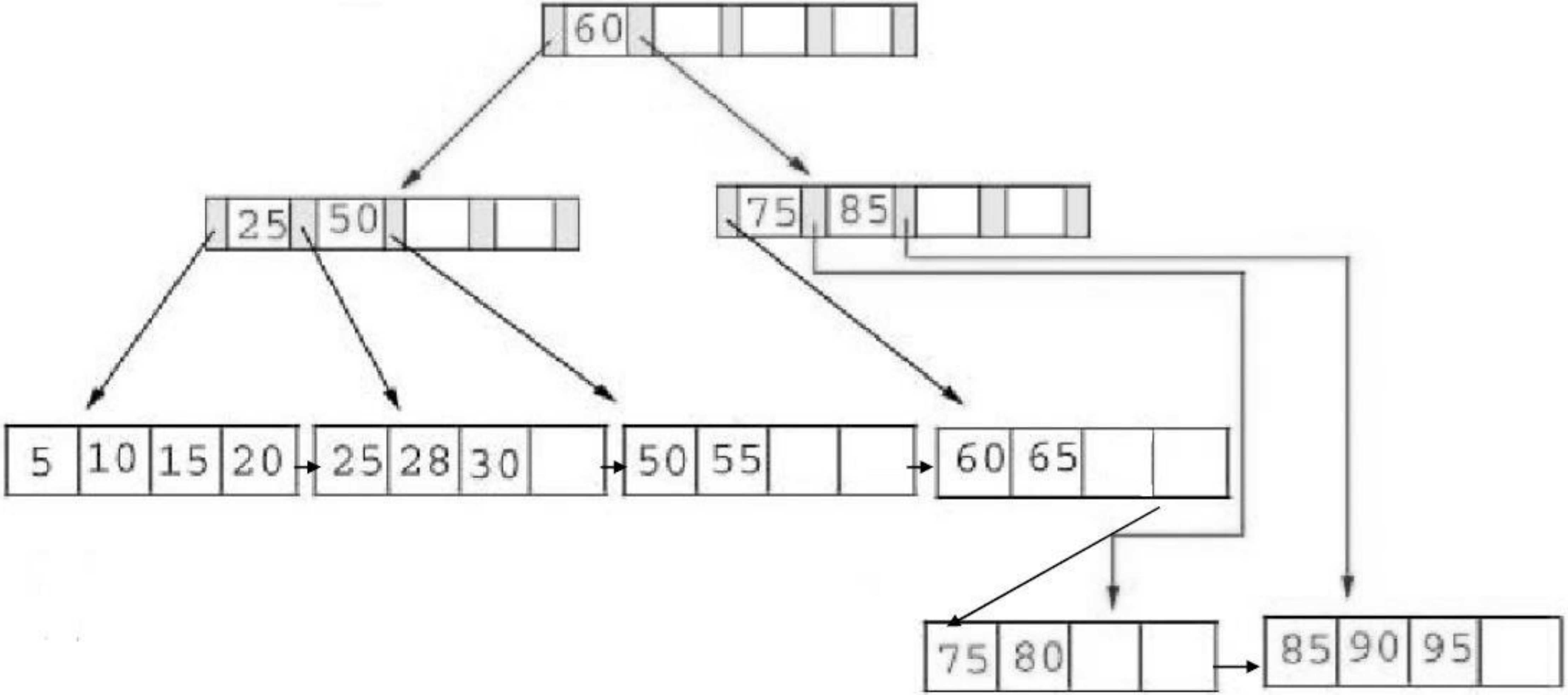
B+ tree index deletion – example 1: delete 70 from the tree below



Just delete 70 from the leaf node.

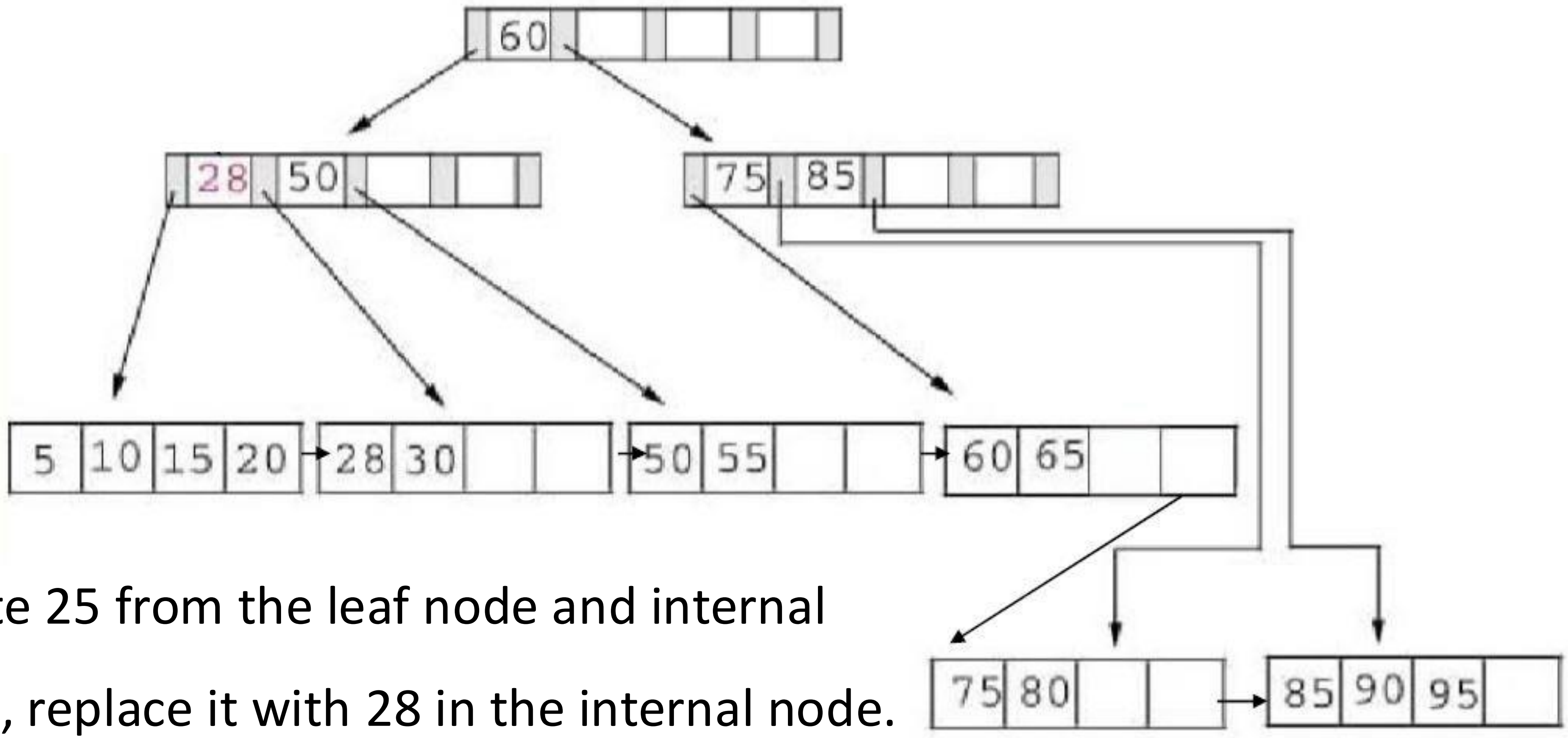
Tree data structure (18)

B+ tree index deletion – example 2: delete 25 from the tree below



Tree data structure (19)

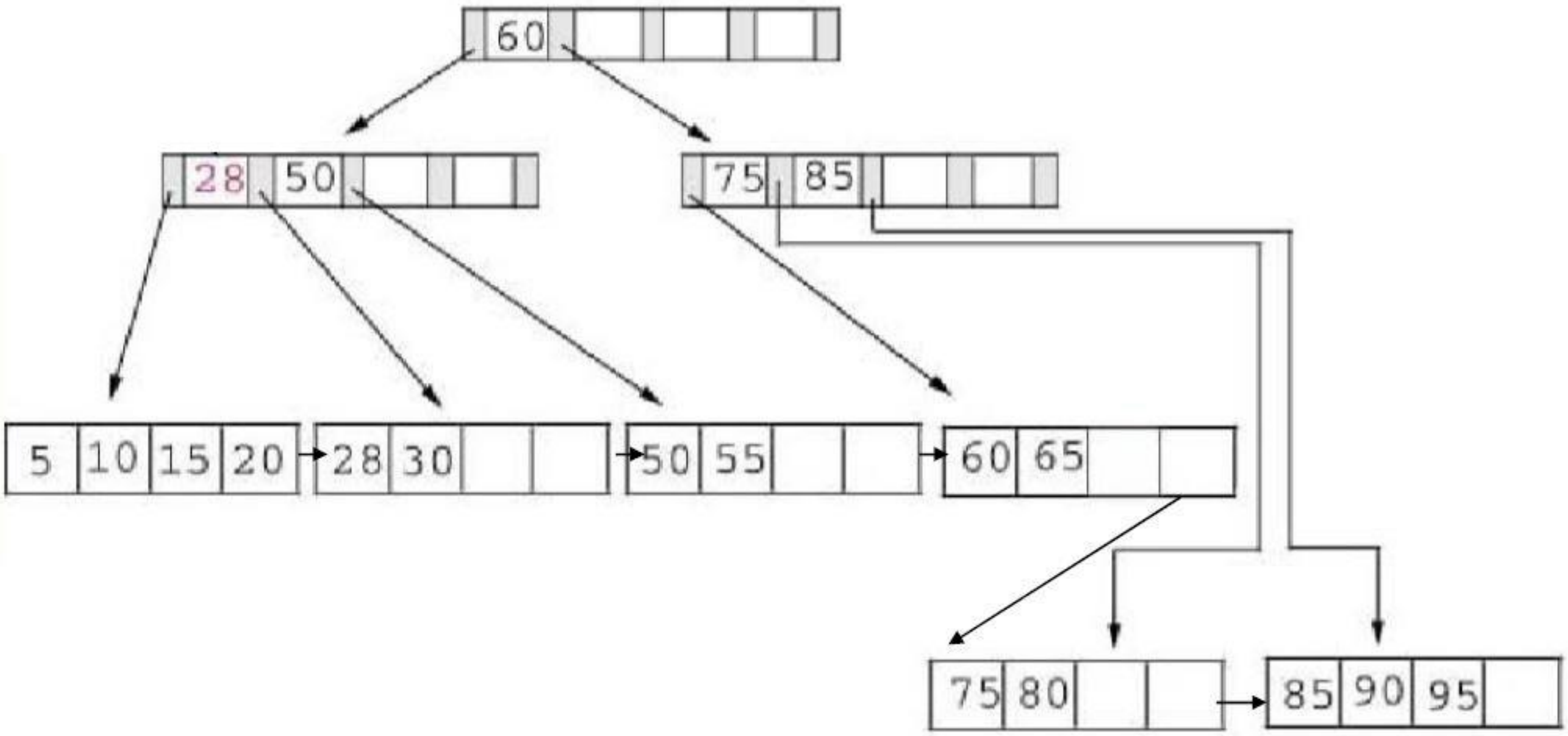
B+ tree index deletion – example 2: delete 25 from the tree below



Delete 25 from the leaf node and internal node, replace it with 28 in the internal node.

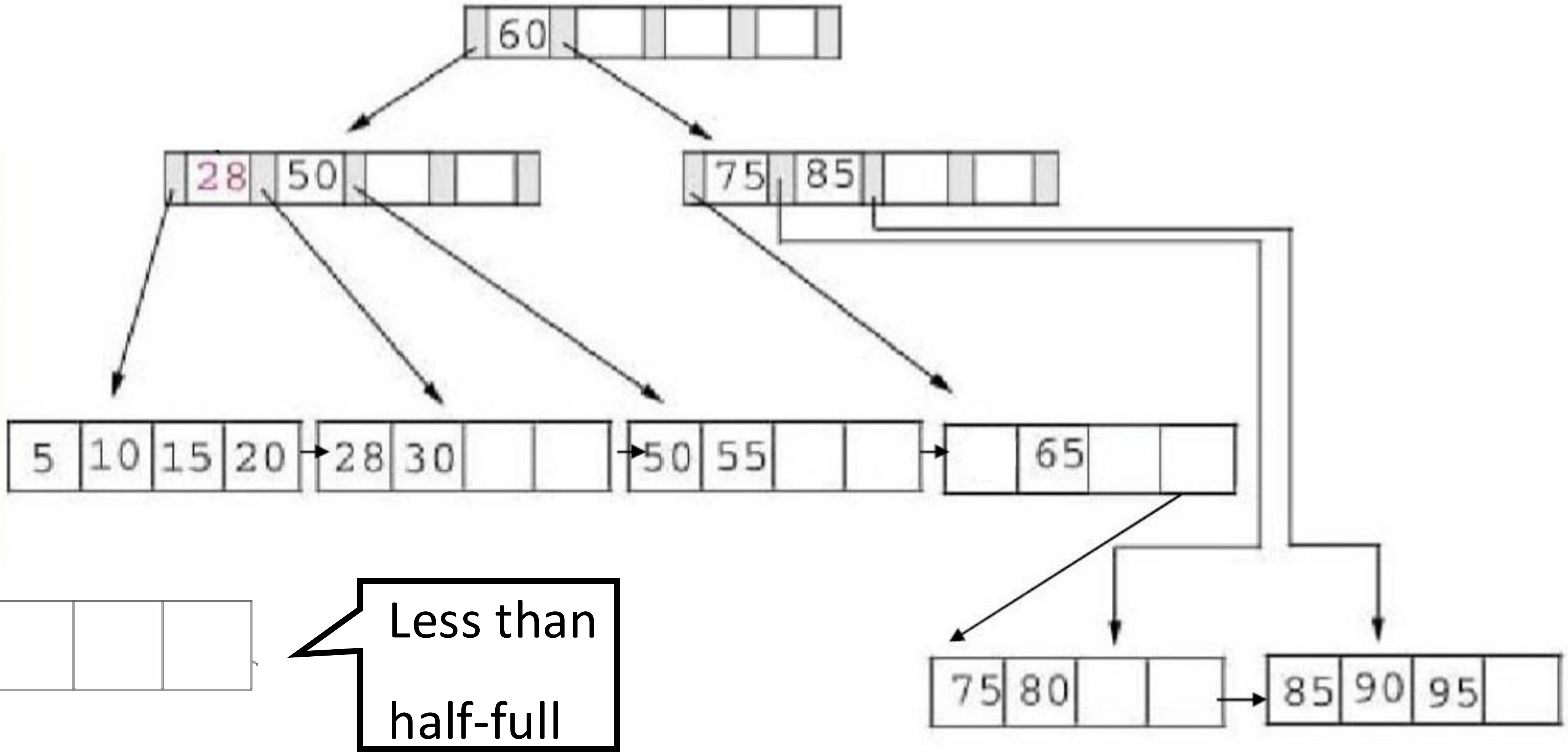
Tree data structure (20)

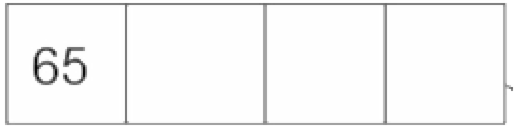
B+ tree index deletion – example 3: delete 60 from the tree below



Tree data structure (21)

B+ tree index deletion – example 3: delete 60 from the tree below

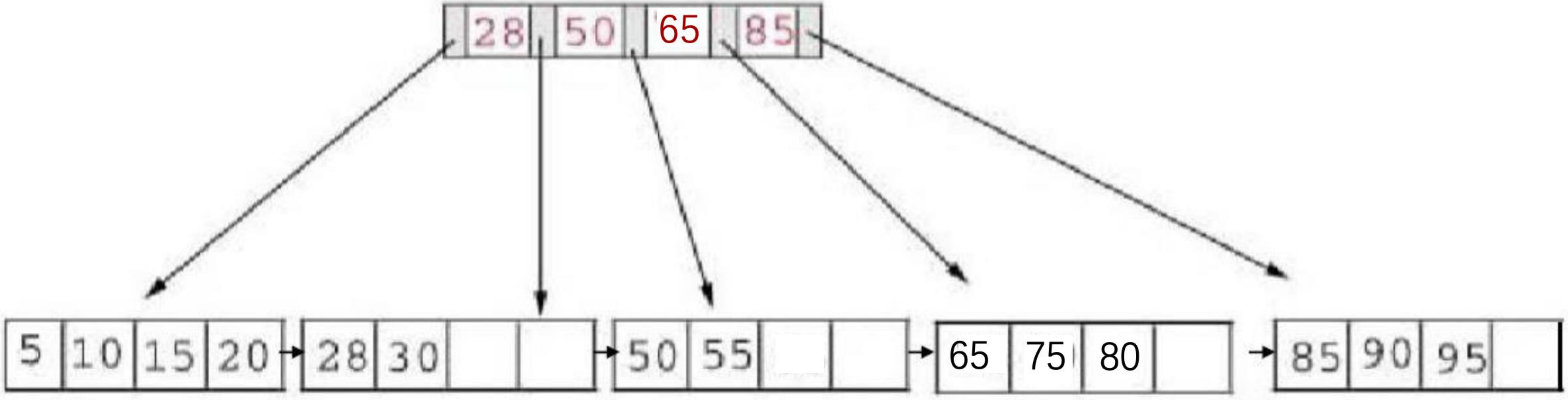




Less than half-full

Tree data structure (22)

B+ tree index deletion – example 3: delete 60 from the tree below



Delete 60 from the leaf node, combine right sibling nodes and then internal nodes.