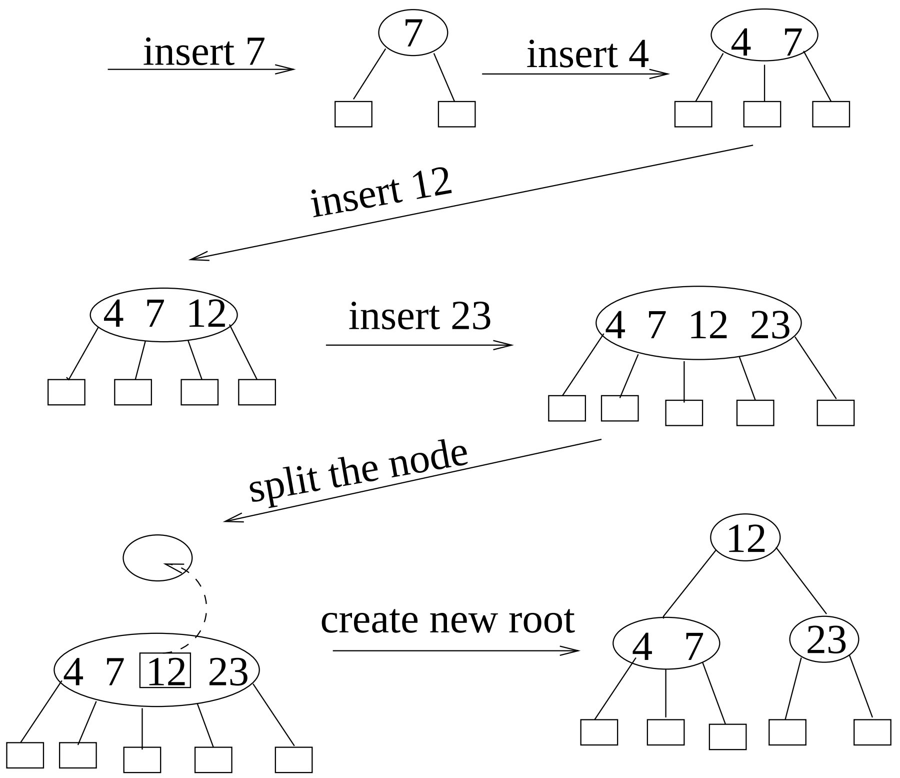
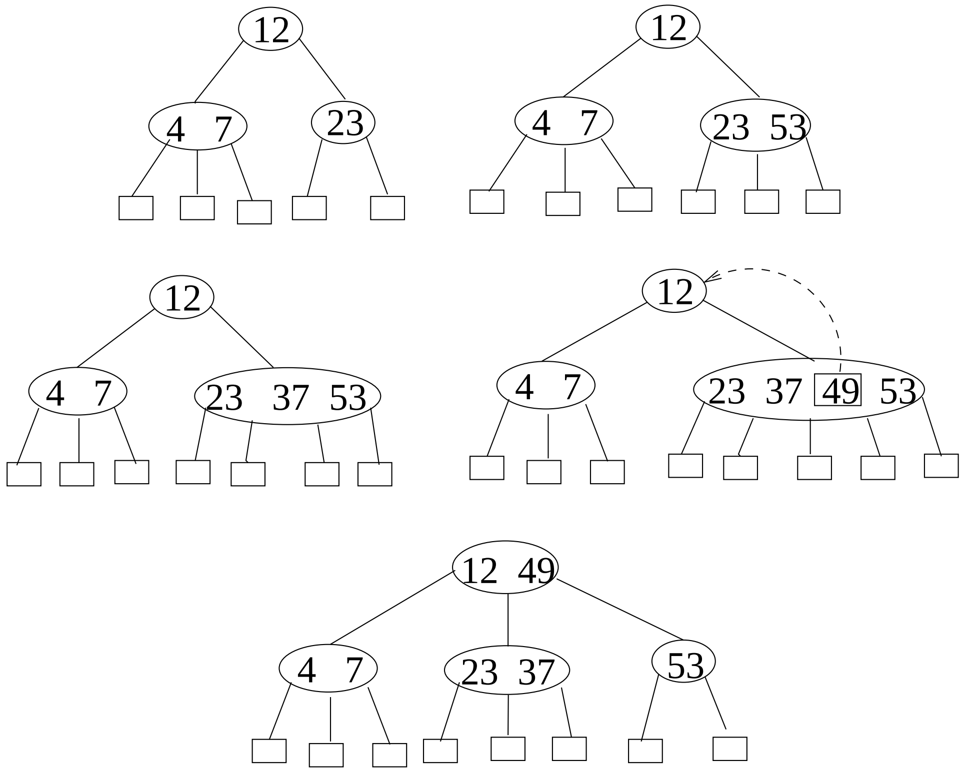
Insertion example: (2,4) tree



Insertion example: (2,4) tree



Deletion from (2,4) tree

- Proceed like BST for deletion from (2,4) tree.
- Let the item to be deleted be (k_i, x_i) from an internal node n.
- Find the right most internal node of the ith subtree of n having only external nodes as children. Let it be n_r.
- Swap the last item in n_r with (k_i, x_i).
- Now remove the last item from n_r.
- We need to see that the depth property is preserved.
- May be n_r is empty after removing the last item.
- So, n_r is now left with one child (one external node) which voilates conditions of (2, 4) tree.

Handling underflow due to deletion

- Requires transfer or merge operation.
- If deletion leaves the internal node n_r (having children which are external nodes) without an item, two case can occur:
  - n_r has an immediate sibling (left or right) that is a 3-node (2 keys) or 4-node (3 keys).
  - n_r has no immediate sibling with more than 1-key each

Handling underflow: Case 1

- Transfer the first (last) item of the right (left) immediate sibling to the parent node.
- This will force deletion of leftmost (rightmost) external child of the above right (left) sibling node.
- Next transfer the appropriate item from the parent node down the vacant node and settle the item there.
- Note the 2 transfers does not affect the parent or the node from which deletion takes place, but one of the immediate sibling of n_r gets poorer by an item and its external node.
- Since as that sibling has at least 1 key the property of (2, 4) tree is not violated.

Handling underflow: Case 2

- Suppose immediate sibling(s) of n_r has (have) only $\lceil \frac{m}{2} \rceil - 1$ keys – only 1 value in (2,4) case.
- Then it is not possible to borrow a value from the sibling.
- So, merge the values left in n_r – no values in (2, 4) case.
- Transfer a key (greater/smaller than deleted value) from parent of n_r.
- The sibling gets $2\lceil \frac{m}{2} \rceil - 2 = (m-1)$ values – 1+1+0 =2 key values in (2, 4) case.

Handling underflow: Case 2

- Thus sibling does not violate the constraint for maximum number of values at a node.
- But parent becomes poorer by one value – no value may be left in it.
- So underflow condition is propagated to one level up.

Handling underflow: General case

- Node that underflowed node contributes $\lceil \frac{m}{2} \rceil - 2$ values.
- Parent node contribute 1 value.
- The sibling of underflowed node contributes between $\lceil \frac{m}{2} \rceil - 1$ and $m - 1$ values.
- Minimum number of total contributions $= 2 \times \lceil \frac{m}{2} \rceil - 2 = m - 1$.

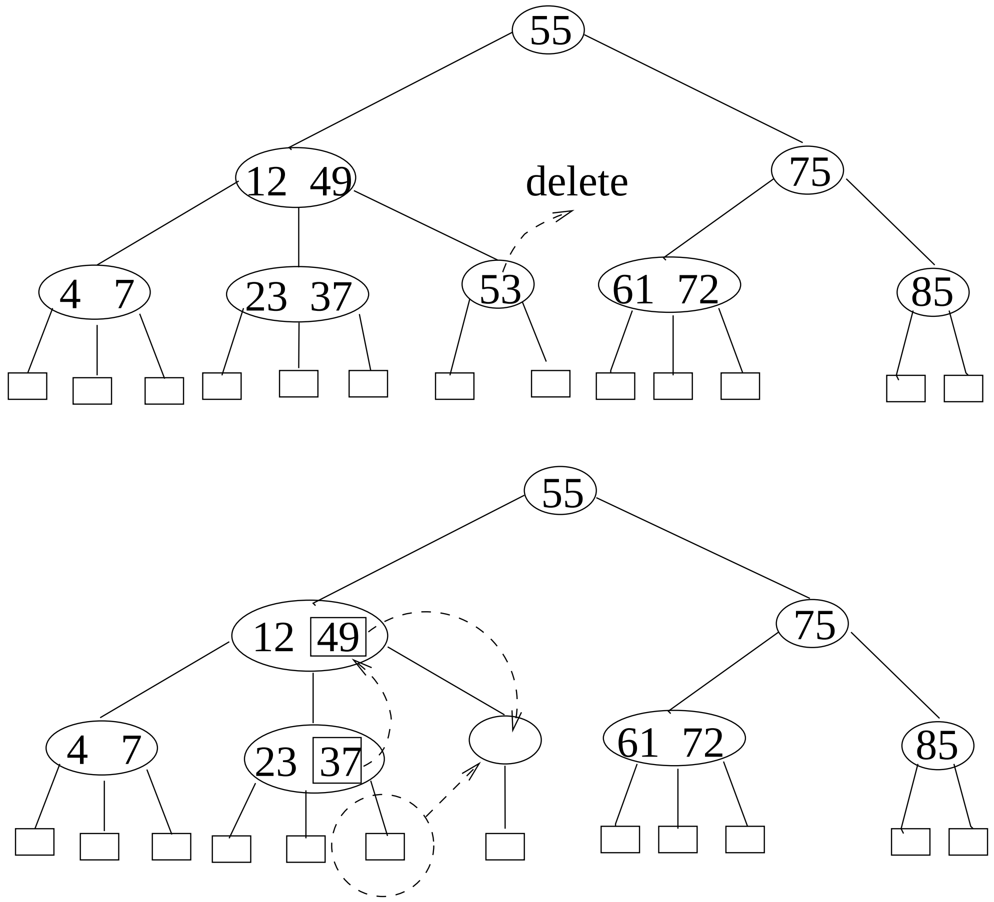
Handling underflow: General case

- If the sibling contributes more than $\lceil \frac{m}{2} \rceil - 1$ values then total number of values exceeds $m - 1$.
- Thus, we can steal at least one value from sibling, and promote that to the parent and bring down 1 value from parent down to underflowed node.
- It leave parent unaffected – the value demoted replace by the value promoted from sibling.
- Sibling does not violated the constraint – it had at least $\lceil \frac{m}{2} \rceil$ values.
- Underflowed node gets one additional value – it now has required minimum number of values.

Handling underflow: General case

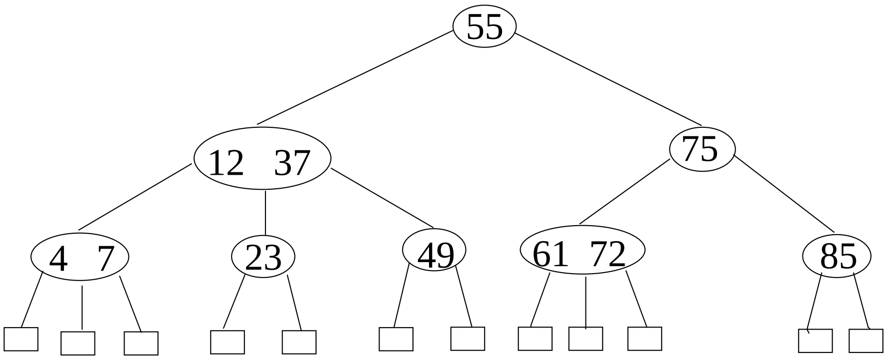
- The second case occurs when the adjacent sibling(s) has (have) only $\lceil \frac{m}{2} \rceil - 1$ values each.
- In this case merge the underflowed node to sibling and bring down one value from parent – underflowed node disappears.
- Total number of values at sibling will be $2 \times \lceil \frac{m}{2} \rceil - 2 = m - 1$.
- Parent gets poorer by 1 value – underflow can appear there.
- Therefore at most $\log n$ merge operation may be required, and
- The merge may finally result in a reduction in height of the tree.

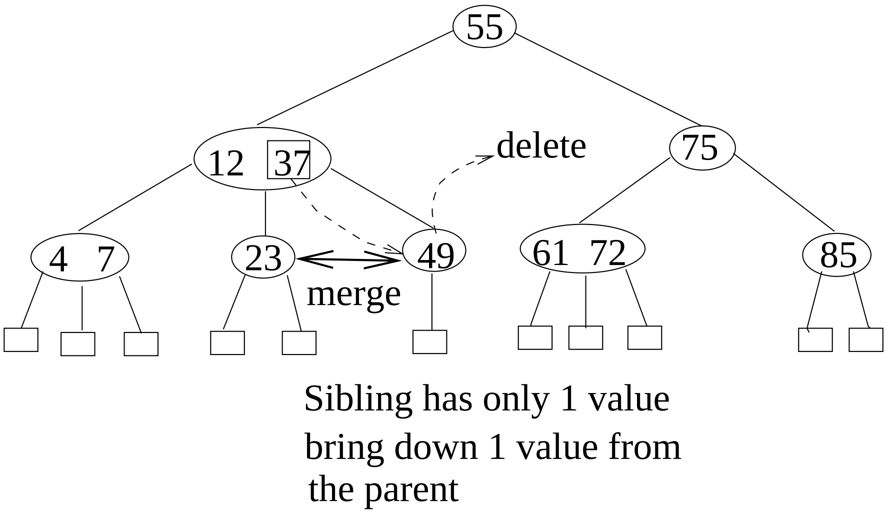
Example: deletion in (2,4) tree



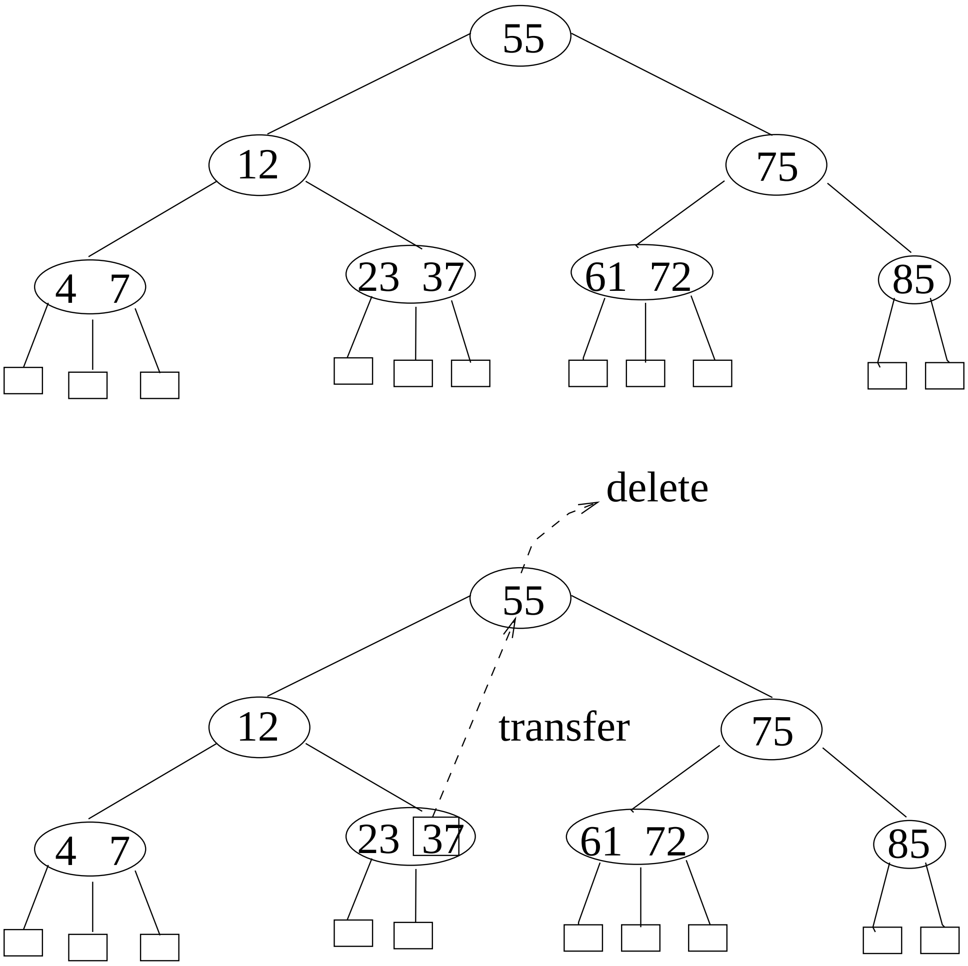
Borrow a value from sibling

Example: deletion in (2,4) tree

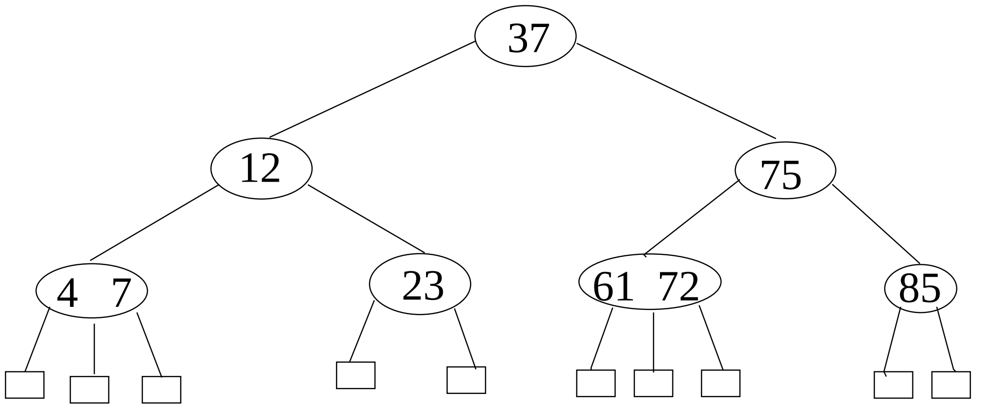


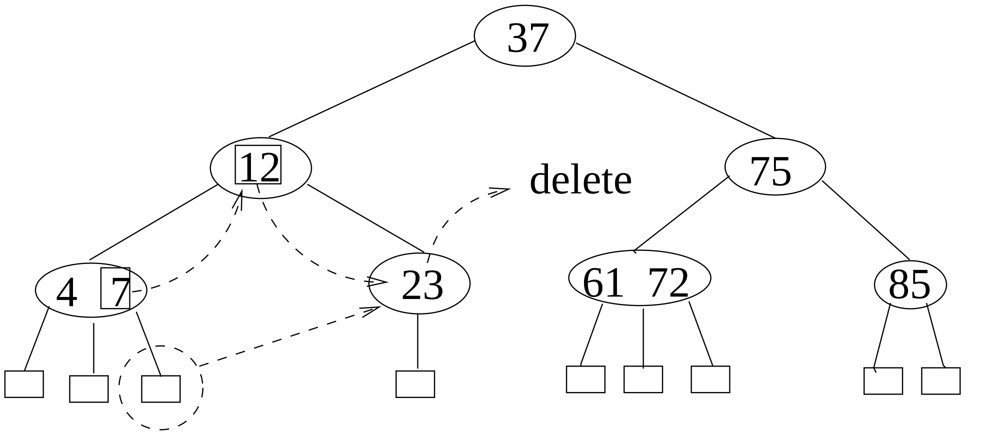


Example: deletion in (2,4) tree

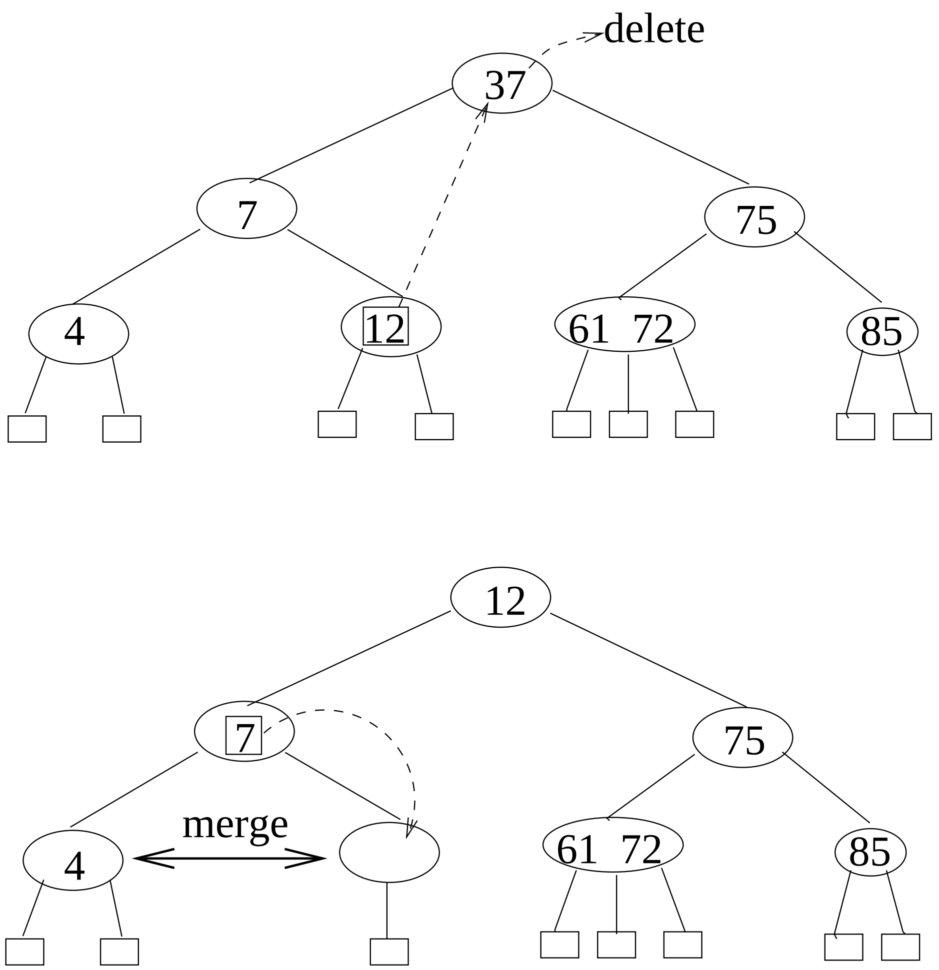


Example: deletion in (2,4) tree

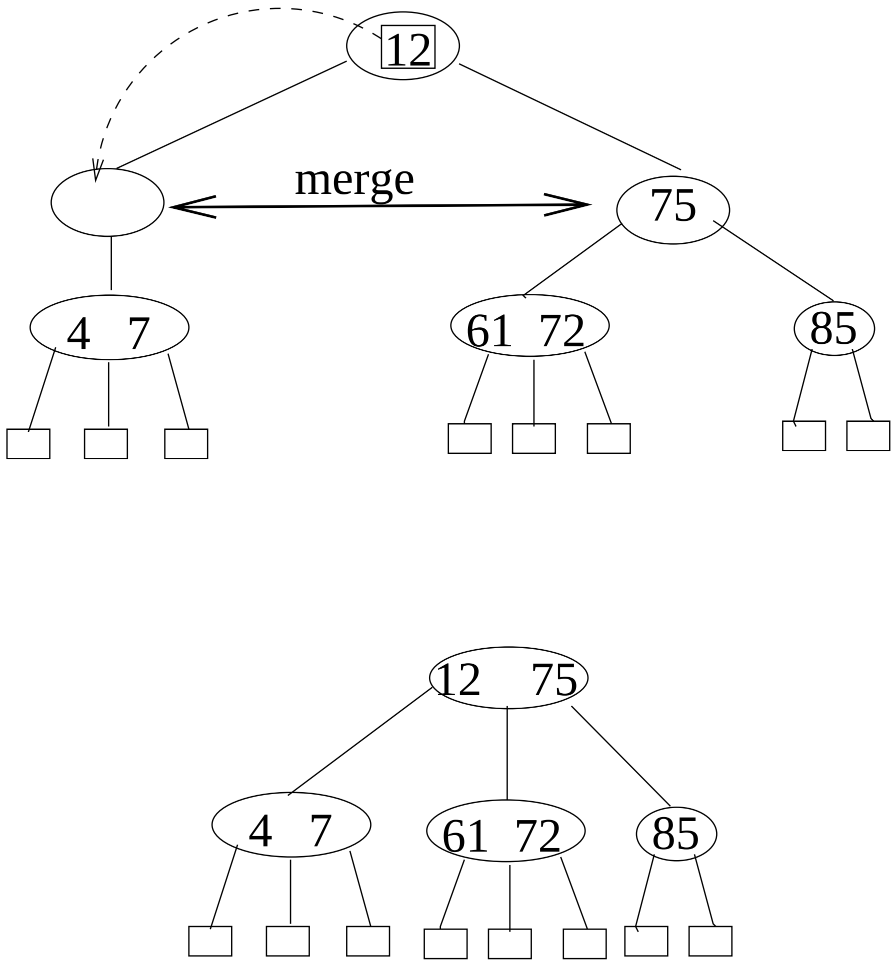




Example: deletion in (2,4) tree



Example: deletion in (2,4) tree



B-trees

- We know that membership search requires examination of at most O(1) node at each level of a (a, b) tree.
- For upper bound assume each node to have only $\lceil m/2 \rceil$ children.
- Therefore, during a search is O($\log_{\lceil d/2 \rceil} n$) nodes are examined, where n is the number of nodes in the B-tree.
- O($\log n / \log B$) disk transfers may be required in the worst case.
- An insertion we proceeds like a member search
- But it may split a node if it has $m + 1$ children after the insertion.
- In the worst case one split may be required at each level.
- So, $\log_B n$ splits may be needed.
- Same thing holds for merge too when deletion results in reduction in tree height.