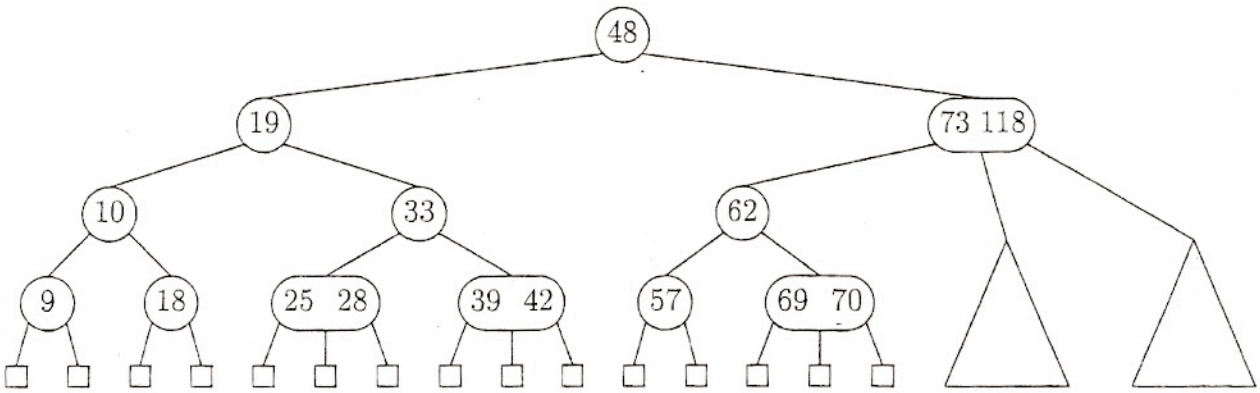
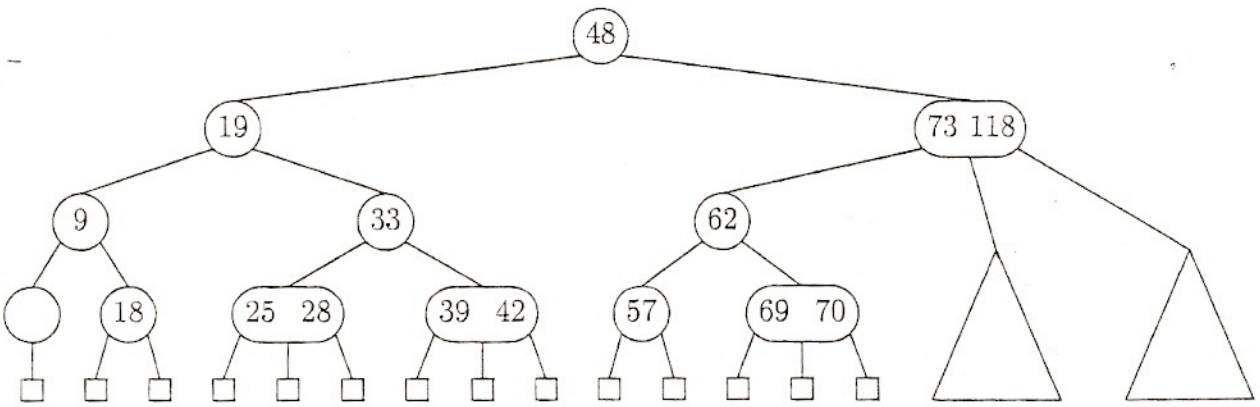
An example of a deletion from a 2-3 tree

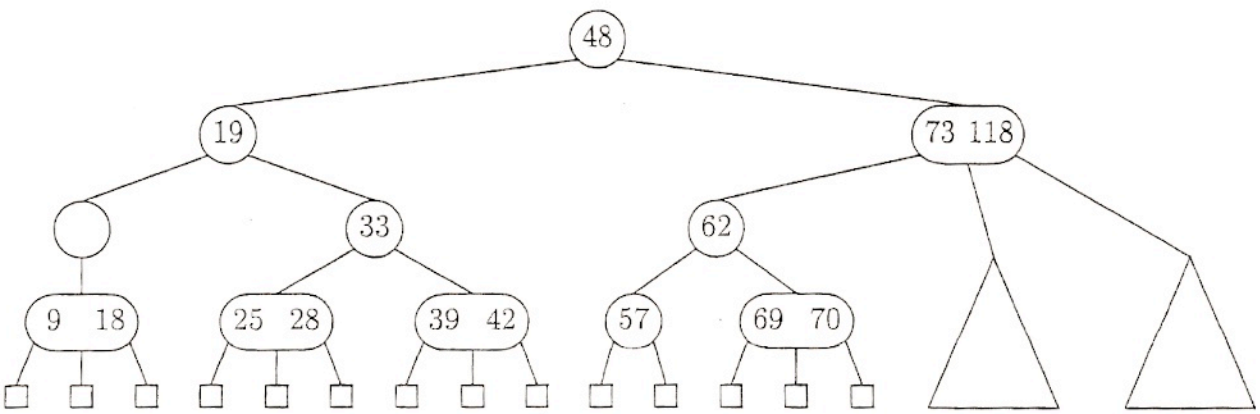
Below is a picture of a 2-3 tree; for simplicity a couple of parts of the tree that don't change during the deletion to be performed are represented by triangles.



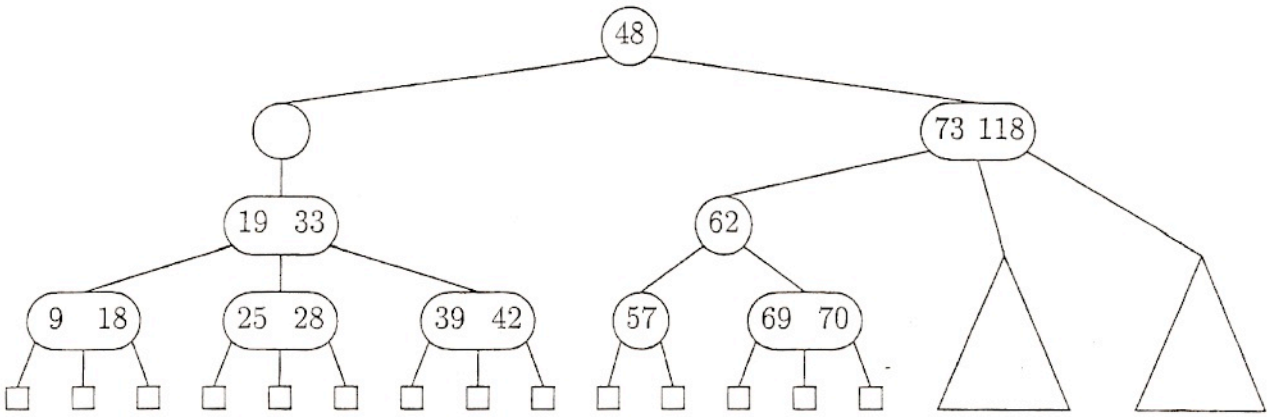
Suppose want to remove the 10. We find that it is not in a bottommost internal node, so we use the same trick as is used for deletion in binary search trees: we move the inorder predecesessor (9) into the position occupied by the key to be deleted (10), to get



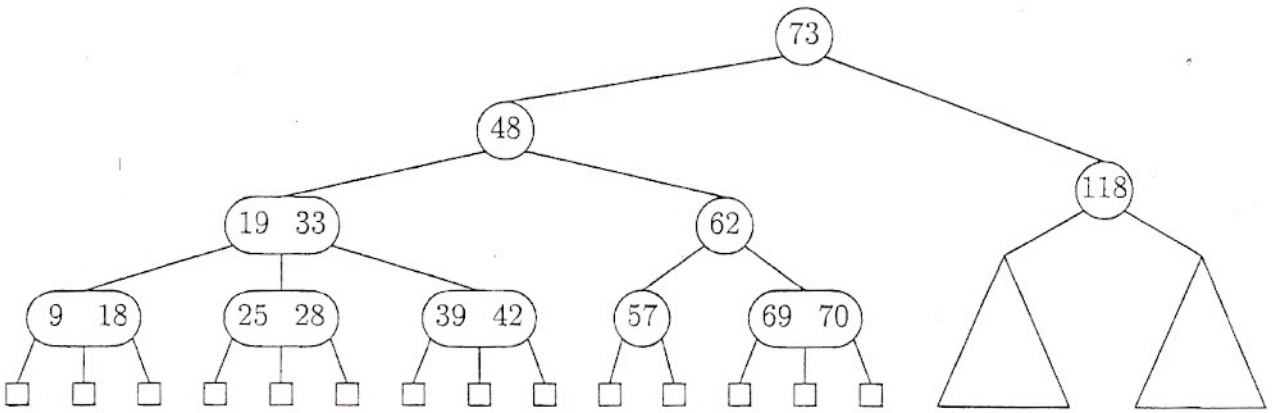
This causes an underflow: there is now an internal node with no keys and hence only one child. It does not have a neighbor with any children to spare, so to fix this we do a fusion of the empty node and its sibling. This brings down the key 9, so we have



Now we have a different node with no keys and hence only one child, but we've made progress since we've moved the problem up the tree. Again, no neighbor of the node with no keys has any children to spare. Thus we merge the node with no keys with its sibling, bringing the 19 down, to obtain



We've moved the problem another step higher in the tree! This time the node with no keys does have a neighbor with spare children: The node containing 73 and 118 has three children and thus can lose one and still be within the constraints on 2-3 trees. So, we do a transfer to obtain



Now at last we have a valid 2-3 tree and are done!