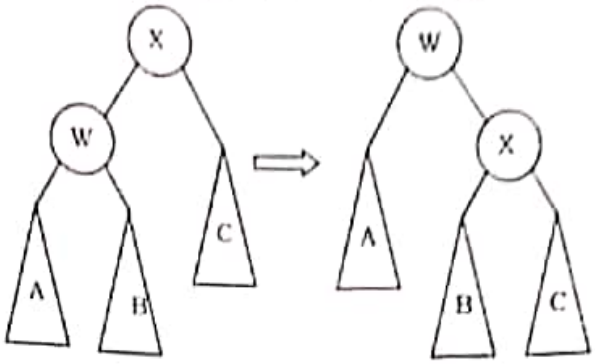
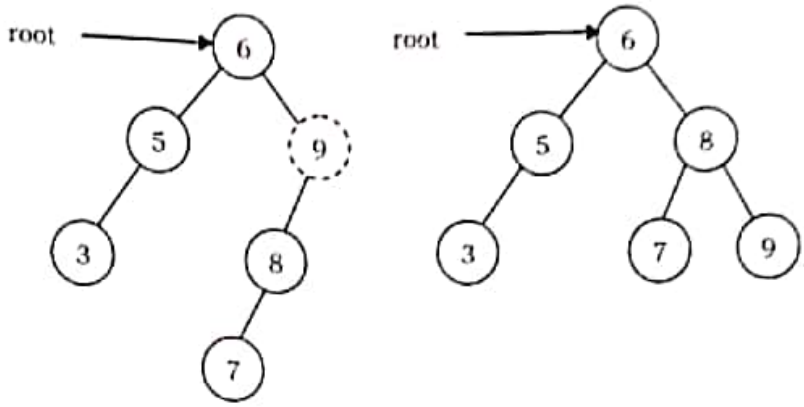
Single Rotations

Left Left Rotation (LL Rotation) [Case-1]: In the case below, at node *X*, the AVL tree property is not satisfying. As discussed earlier, rotation does not have to be done at the root of a tree. In general, we start at the node inserted and travel up the tree, updating the balance information at every node on the path.





For example, in the figure above, after the insertion of 7 in the original AVL tree on the left, node 9 becomes unbalanced. So, we do a single left-left rotation at 9. As a result we get the tree on the right.

```
struct AVLTreeNode *SingleRotateLeft(struct AVLTreeNode *X ){
    struct AVLTreeNode *W = X→left;
    X→left = W→right;
    W→right = X;
    X→height = max( Height(X→left), Height(X→right) ) + 1;
    W→height = max( Height(W→left), X→height ) + 1;
    return W;  /* New root */
}
```

Time Complexity: O(1).
Space Complexity: O(1).

Right Right Rotation (RR Rotation) [Case-4]: In this case, the node *X* is not satisfying the AVL tree property.