

PSEUDO-CODE FOR SINGLE AND DOUBLE BST ROTATIONS (AVL tree)

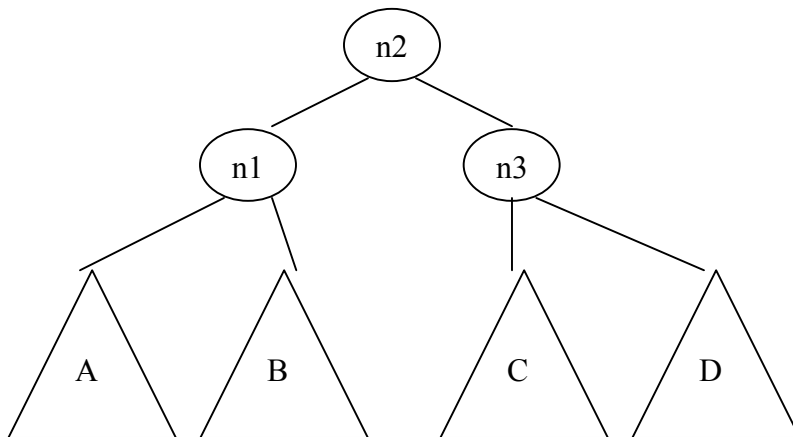
Single Rotations. See picture below single rotation code.

```
// rotate n2 with (left child) n1, i.e., clockwise
Node rotateRight (Node n2) {
    Node k = n2.left;
    n2.left = k.right;
    k.right = n2;

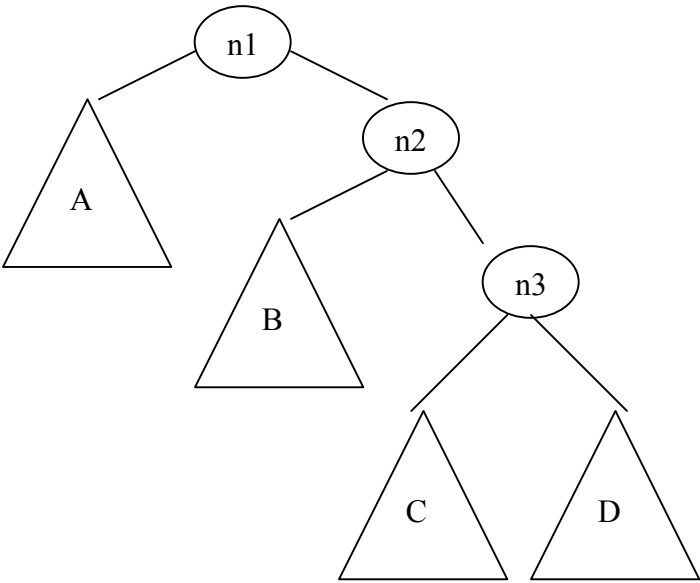
    // update heights here if needed
    // ...

    return k;
}
```

```
// rotate n2 with (right child) n3, i.e., counter clockwise
Node rotateLeft (Node n2) {
    Node k = n2.right;
    n2.right = k.left;
    k.left = n2;
    // heights ...
    return k;
}
```



Initial tree



After a single right rotation (rotate n2 and n1)

Double Rotations

```
/* rotate n3's left child (n1) with n1's right child (n2)
 * i.e., -> rotate left
 * followed by rotate n3 with n3's (new) left child (n2)
 * i.e., -> rotate right
 * see picture below
 */
```

```
Node doubleRotateLeftRight (Node n3) {
    n3.left = rotateLeft(n3.left);
    Node k = rotateRight (n3);
    return k;
}
```

```
/* rotate n1's right child(n3) with n3's left child (n2)
 * -> rotate right,
 * followed by rotate n1 with n1's (new) right child (n2)
 * -> rotate left
 * see picture below
 */
```

```
Node doubleRotateRightLeft (Node n1) {
    N1.right = rotateRight(n1.right);
    Node k = rotateLeft(n1);
    Return k;
}
```

