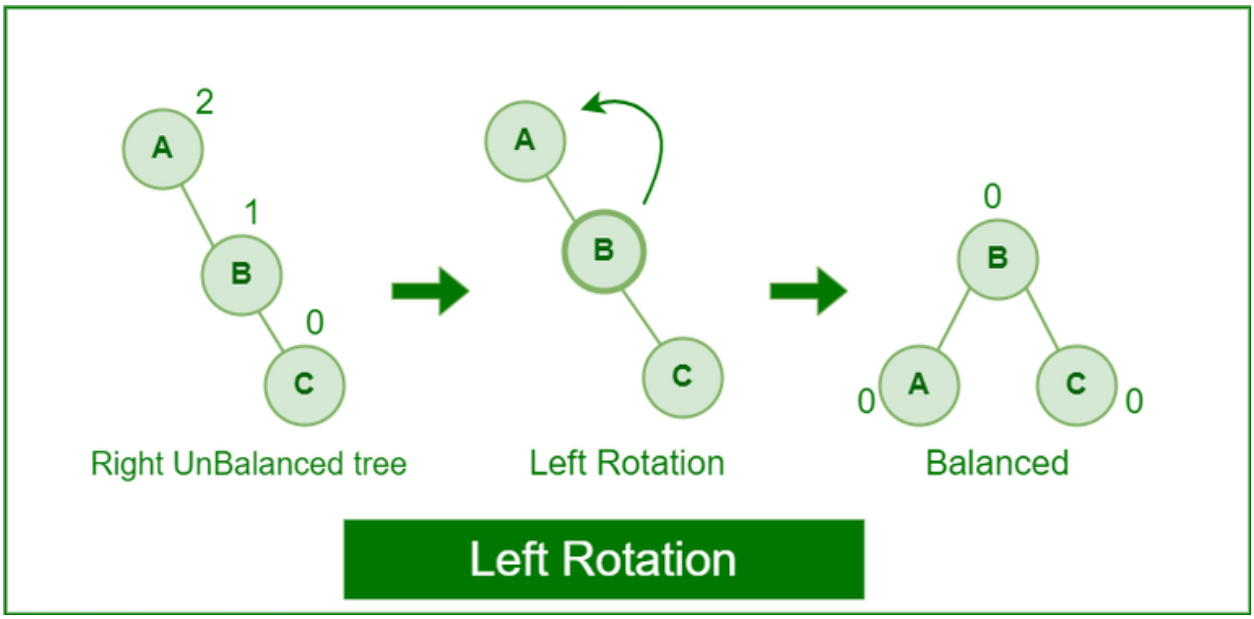


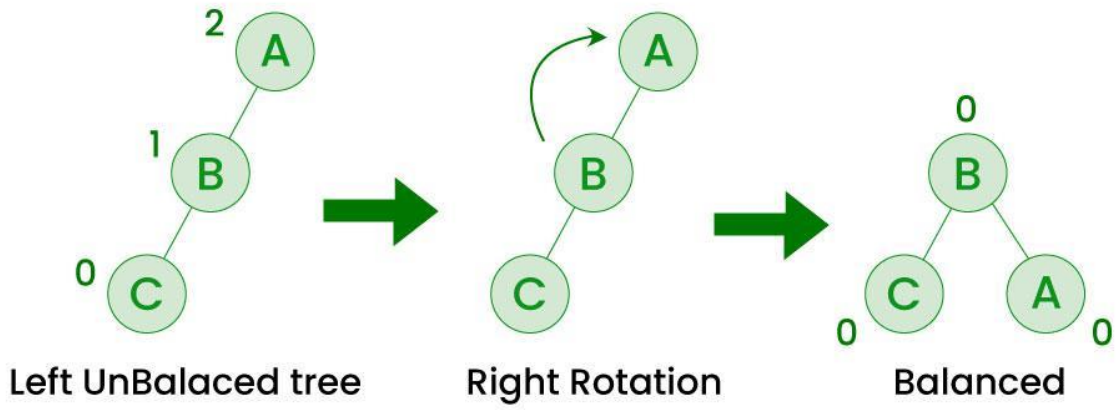
Left Rotation:

Used when the balance factor of a node becomes greater than 1 and the left subtree is heavier.

The purpose is to balance the tree by rotating the node to the left.

Pseudocode:

```
LeftRotate(x)
    y = x.right
    T2 = y.left
    y.left = x
    x.right = T2
    UpdateHeight(x)
    UpdateHeight(y)
    return y
```



AVL Tree

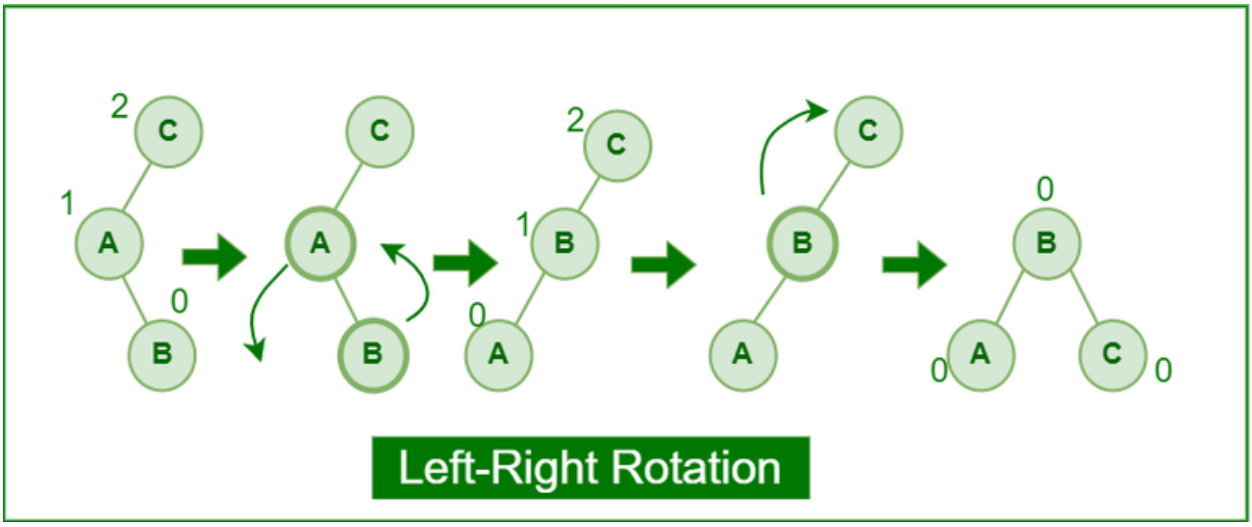
Right Rotation:

Used when the balance factor of a node becomes less than -1 and the right subtree is heavier.

The purpose is to balance the tree by rotating the node to the right.

Pseudocode:

```
RightRotate(y)
  x = y.left
  T2 = x.right
  x.right = y
  y.left = T2
  UpdateHeight(y)
  UpdateHeight(x)
  return x
```



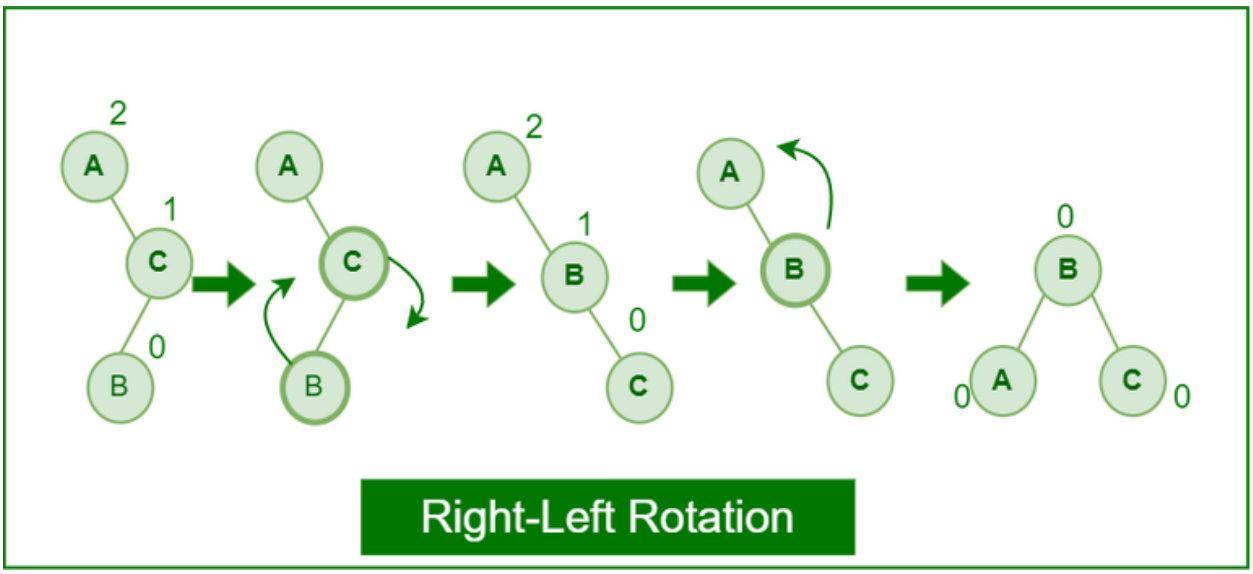
Left-Right Rotation (LR Rotation):

Used when the balance factor of a node becomes greater than 1 and the right subtree of its left child is heavier.

The purpose is to balance the tree by performing a left rotation on the left child followed by a right rotation on the node.

Pseudocode:

```
LeftRightRotate(x)
    x.left = LeftRotate(x.left)
    return RightRotate(x)
```



Right-Left Rotation (RL Rotation):

Used when the balance factor of a node becomes less than -1 and the left subtree of its right child is heavier.

The purpose is to balance the tree by performing a right rotation on the right child followed by a left rotation on the node.

Pseudocode:

```
RightLeftRotate(y)
  y.right = RightRotate(y.right)
  return LeftRotate(y)
```

Tester cpp output without rotate:

Add a new node:100

Print AVL Tree structure:

100

Add a new node:250

Print AVL Tree structure:

  250

100

Add a new node:200

Print AVL Tree structure:

  250

    200

100

Add a new node:300

Print AVL Tree structure:

```
      300
   250
      200
100
```

Add a new node:400

Print AVL Tree structure:

```
         400
      300
   250
      200
100
```

Add a new node:500

Print AVL Tree structure:

```
            500
```

400

300

250

200

100

In-order traversal of AVL tree after insertions:

100

200

250

300

400

500

Add a new node:111

Print AVL Tree structure:

500

400

300

250

200

111

100

Add a new node:211

Print AVL Tree structure:

```
            500
         400
      300
   250
         211
      200
         111
100
```

Add a new node:311

Print AVL Tree structure:

```
            500
         400
            311
      300
   250
```

211

200

111

100

In-order traversal of AVL tree after insertions:

100

111

200

211

250

300

311

400

500

C:\Workspace\C++\BTP-500\C++-CODES\AVL-INSERT\x64\Debug\AVL-INSERT.exe (process 38196) exited with code 0.

To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.

Press any key to close this window . . .