4. Trees

4.4 AVL Trees

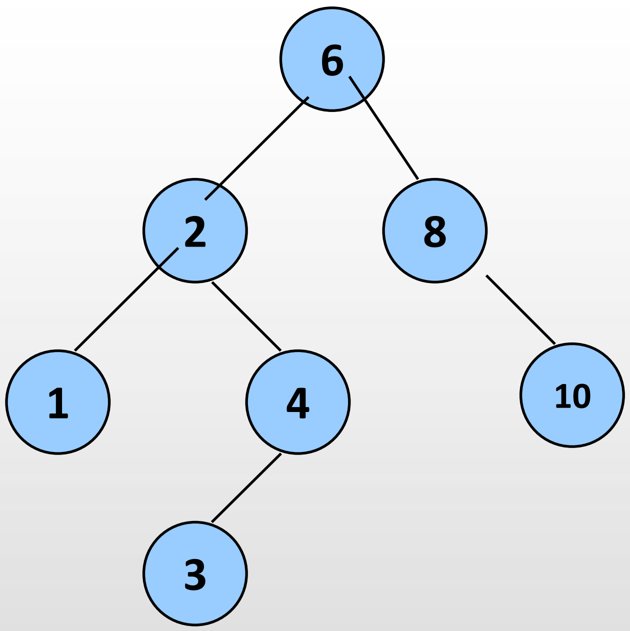
AVL (Adelson-Velskii and Landis) Tree

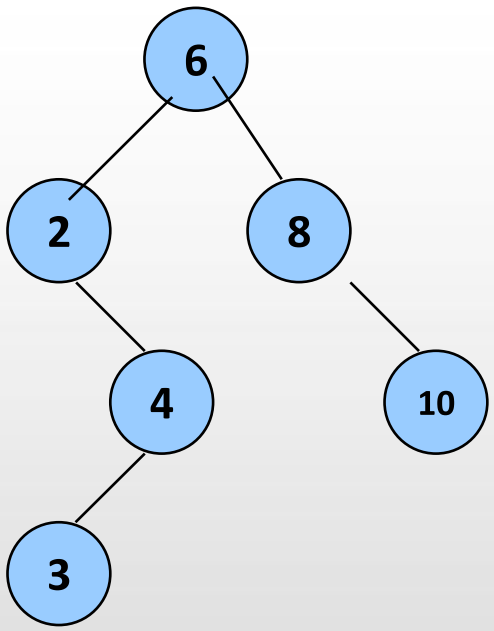
Definition: An AVL tree is a binary search tree wherein the height of the left and right subtrees of every node differs by at most one.

Operations:

- tree operations
- insertion (deletion) uses **rotation**

AVL vs non-AVL



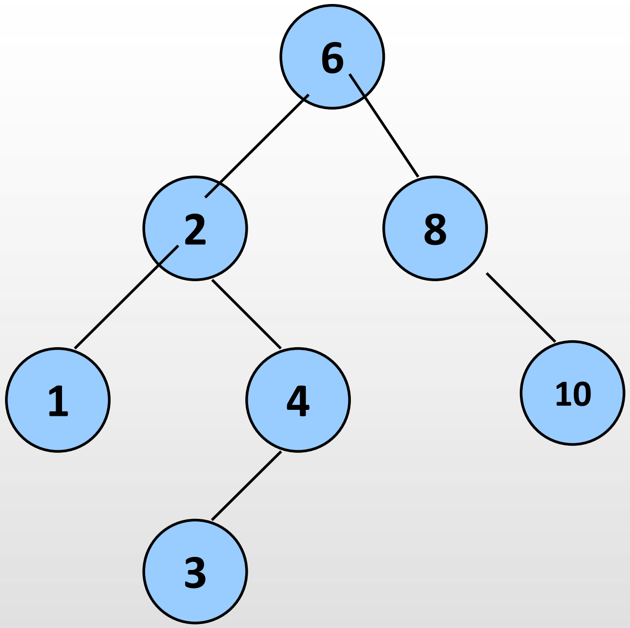


Which is (not) an AVL tree?

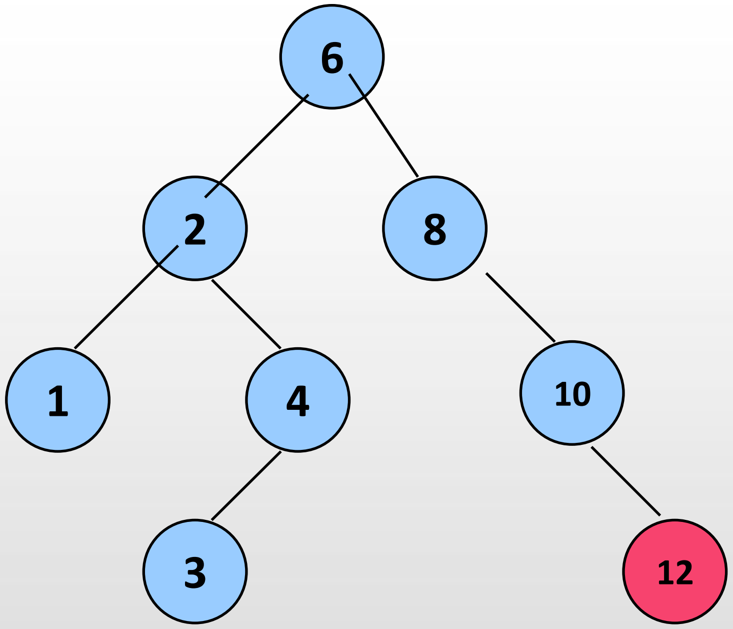
AVL Tree

```
typedef struct node{
  int value;
  int height;
  struct node *left;
  struct node *right;
  struct node *parent;
}AVL;
```

AVL Insertion



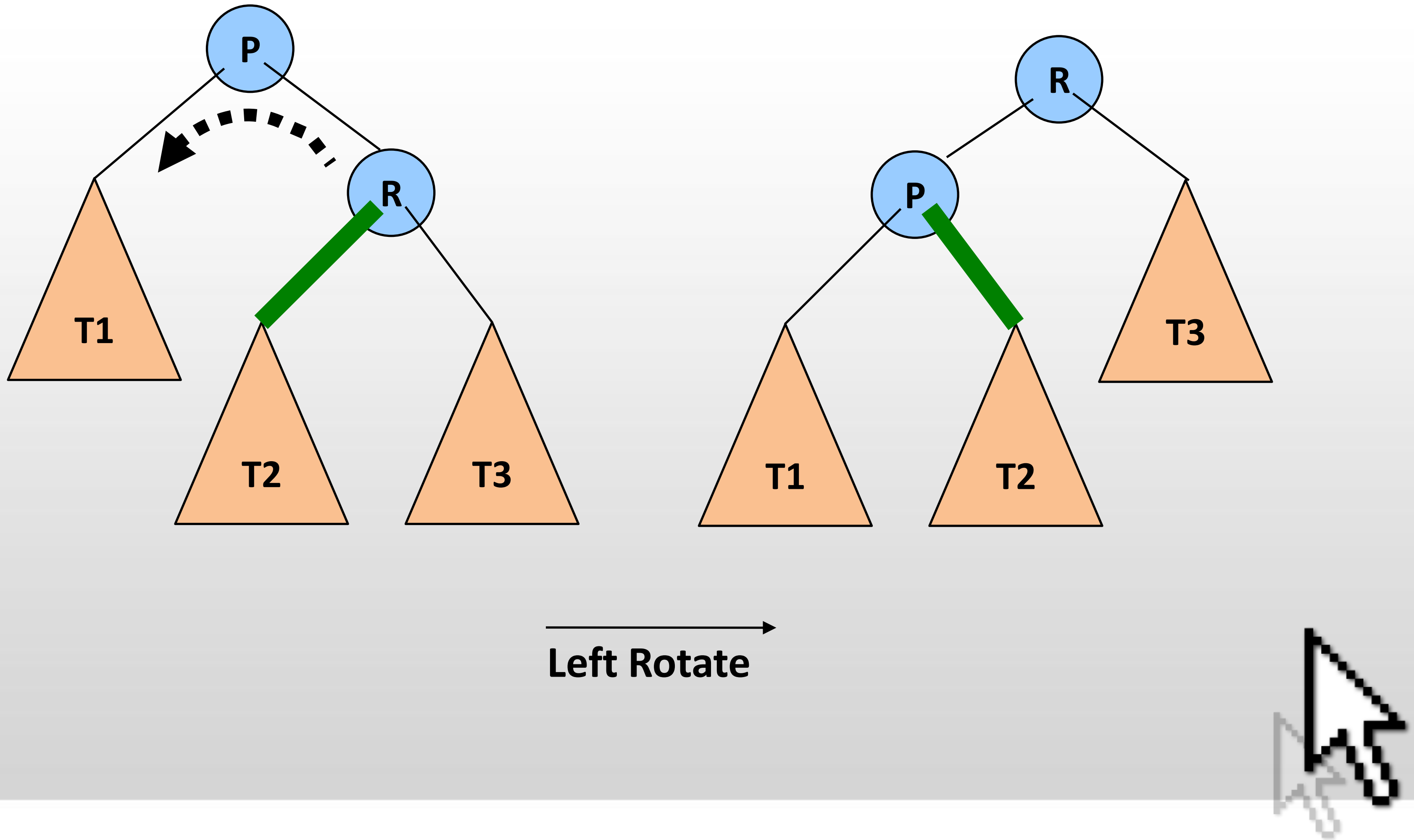
AVL Insertion



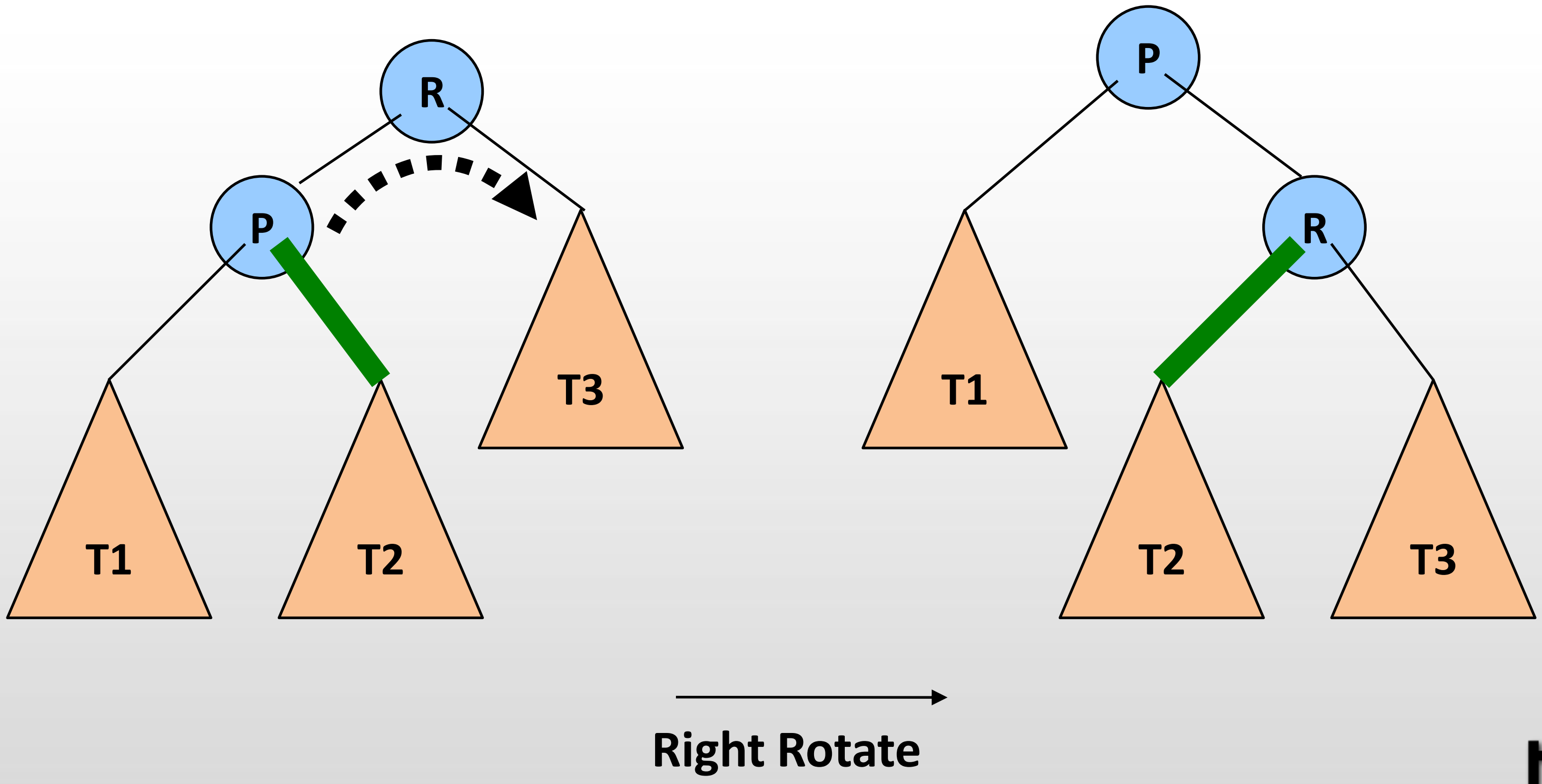
Insert Operation

- Single rotation
  - Left rotate
  - Right rotate
- Double rotation
  - Left-right rotate
  - Right-left rotate

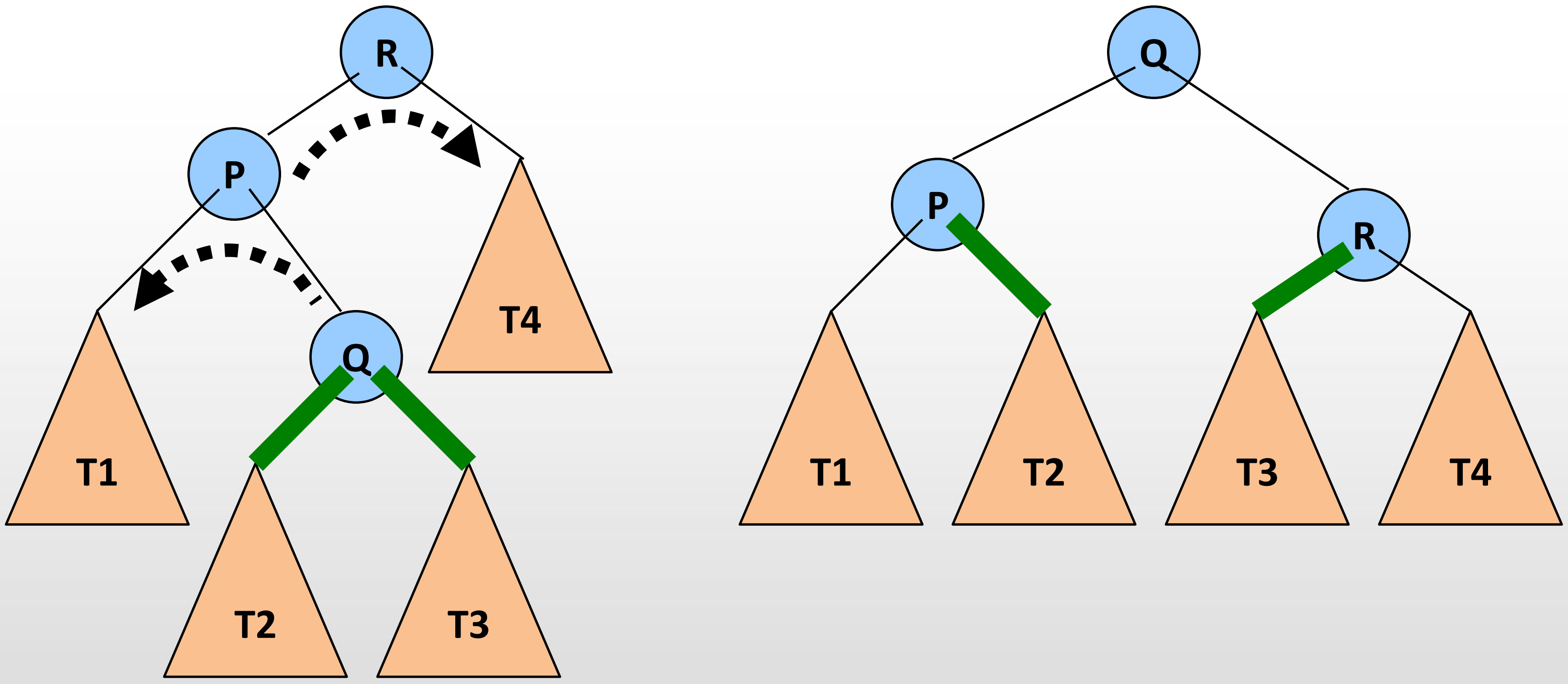
Single Rotation



Single Rotation



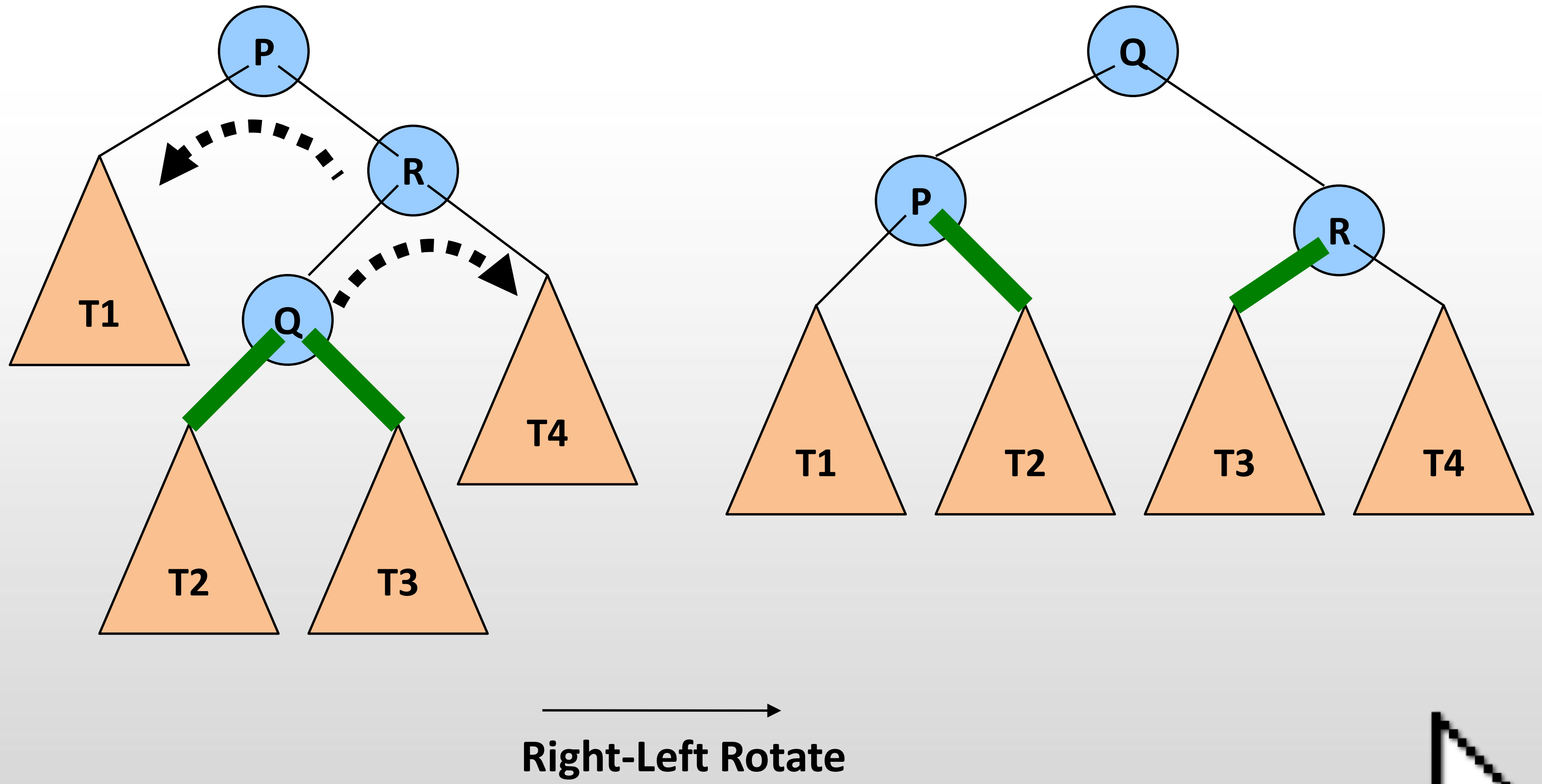
Double Rotation



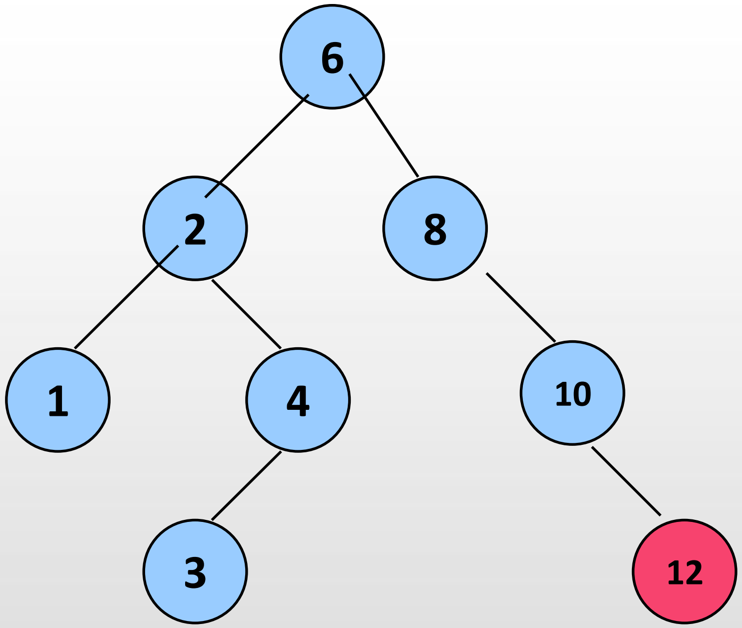
$\longrightarrow$

Left-Right Rotate

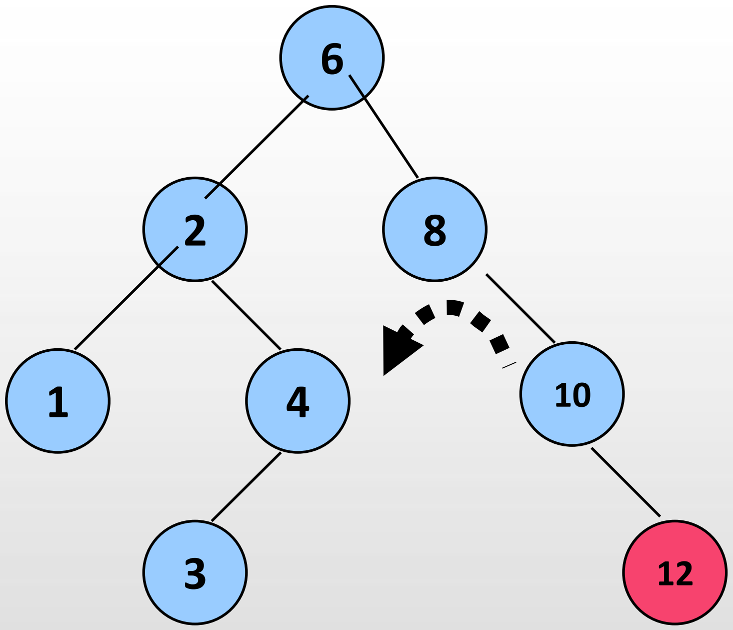
Double Rotation



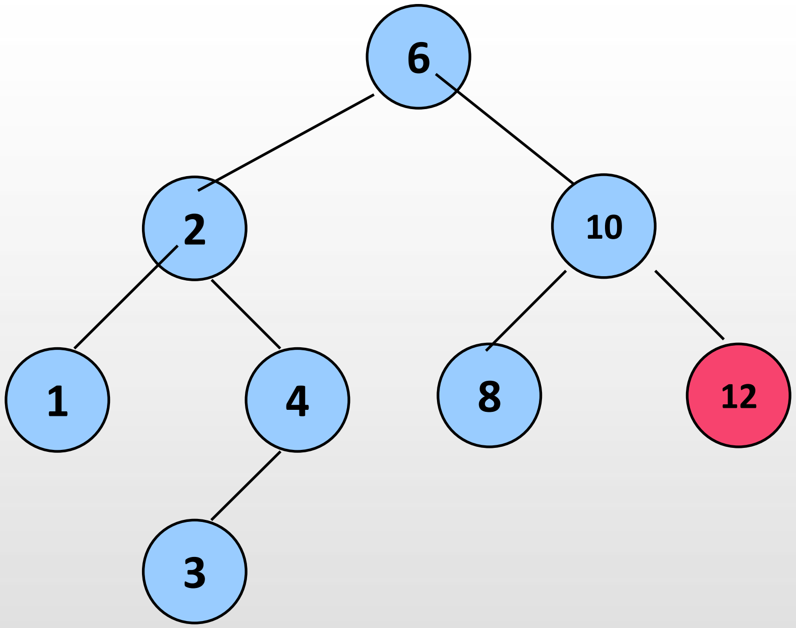
AVL Insertion



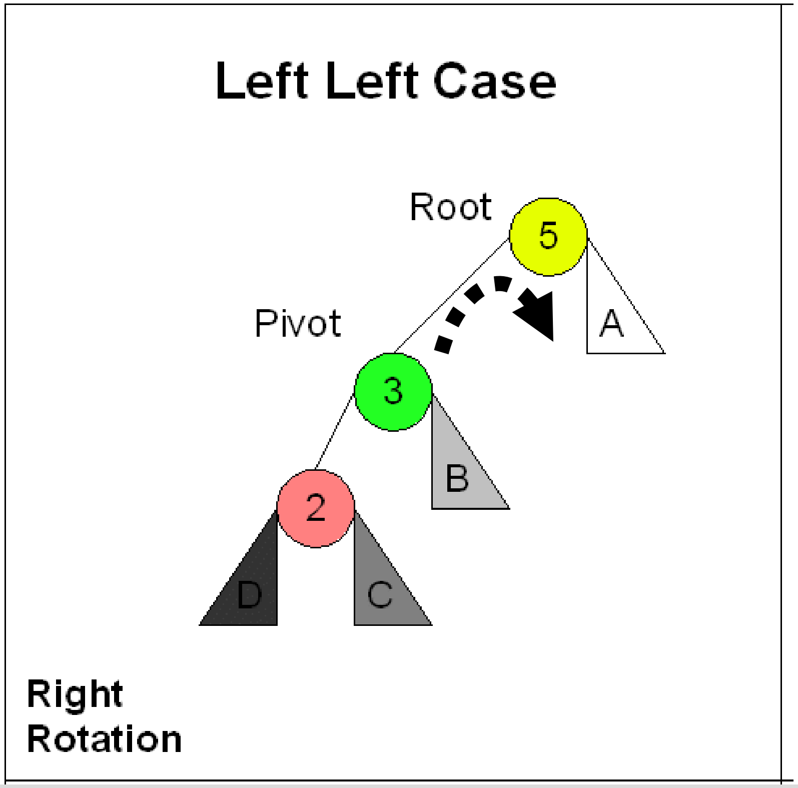
AVL Insertion

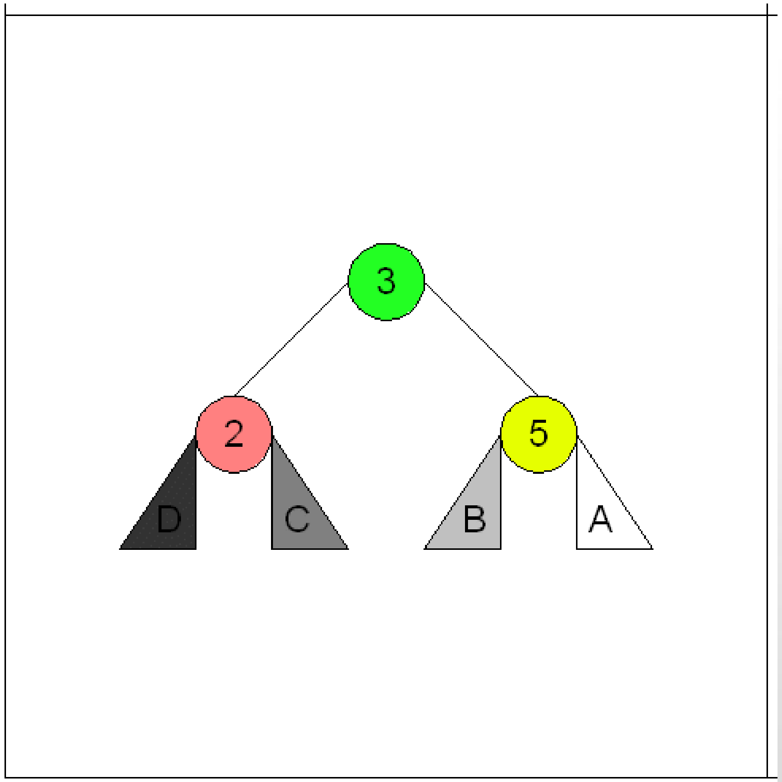


AVL Insertion

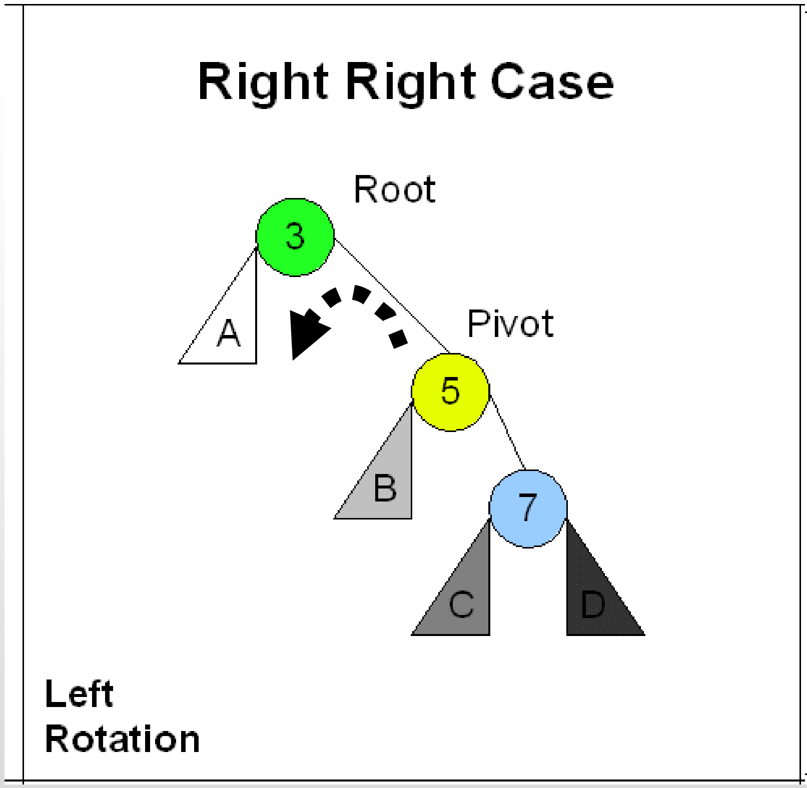


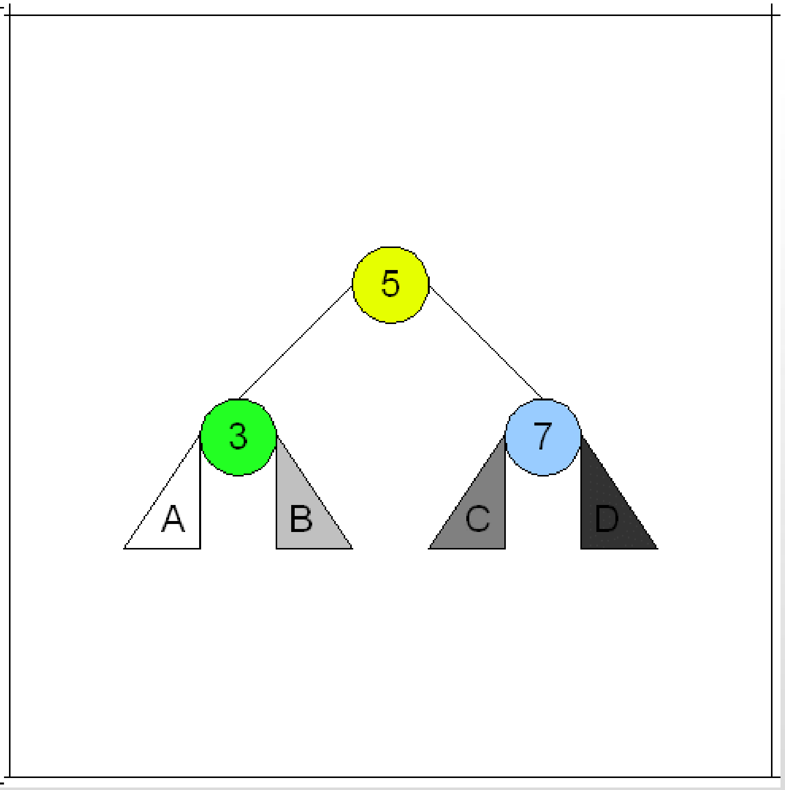
AVL Insertion Guide



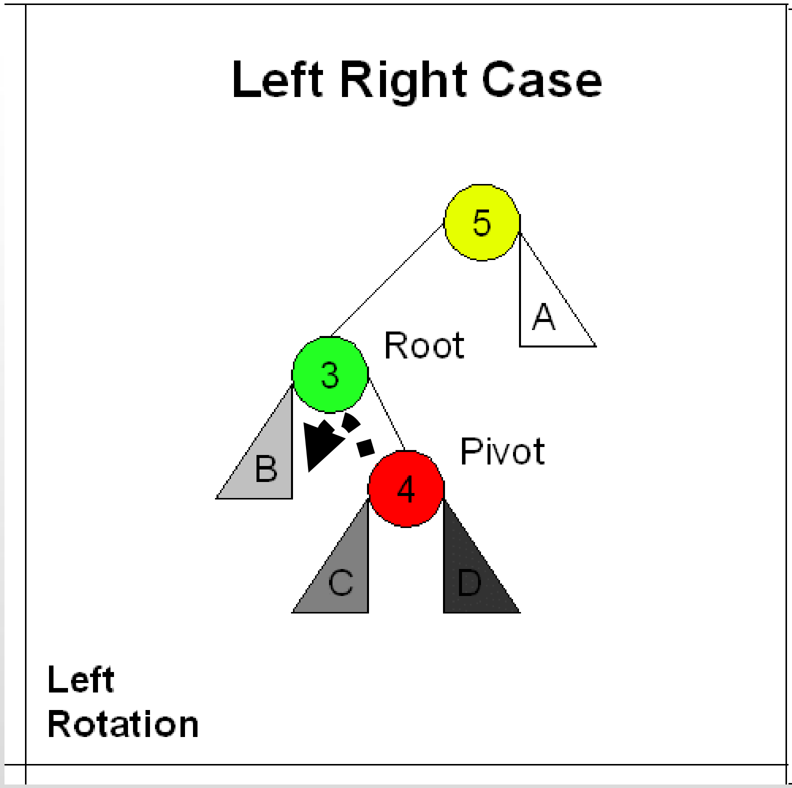


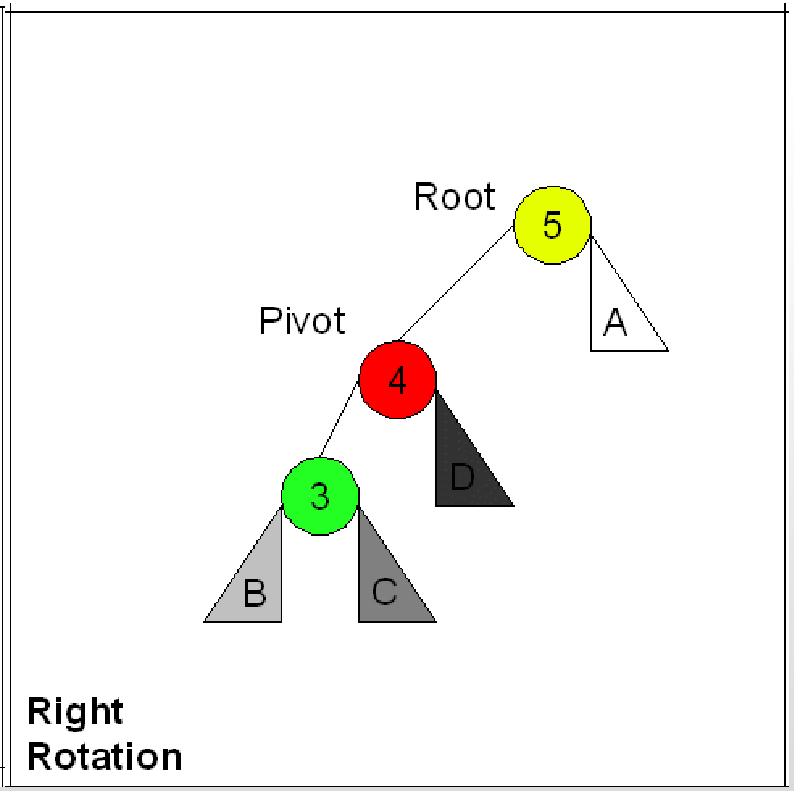
AVL Insertion Guide



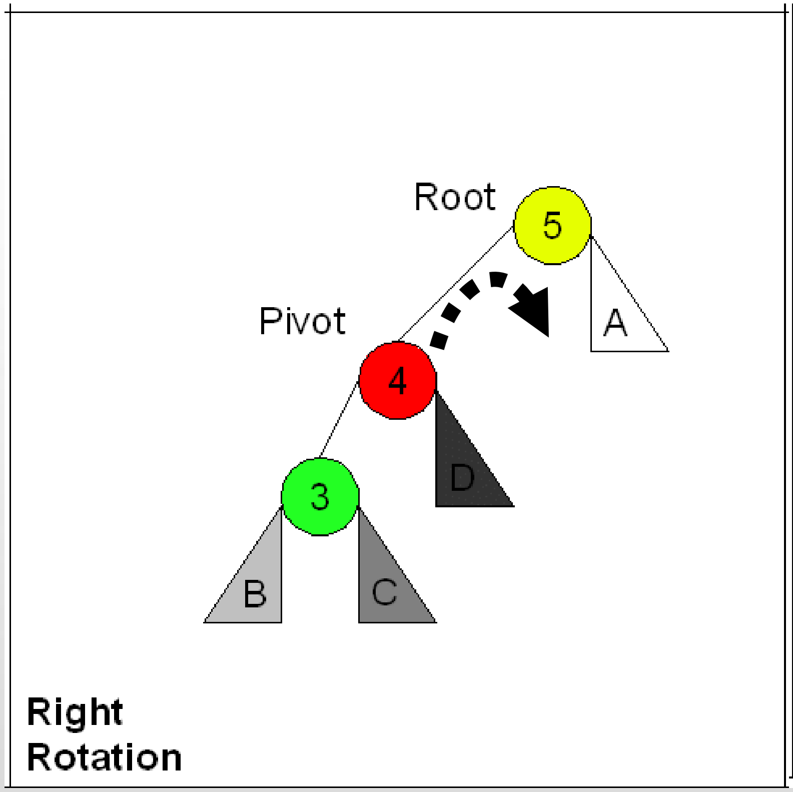


AVL Insertion Guide

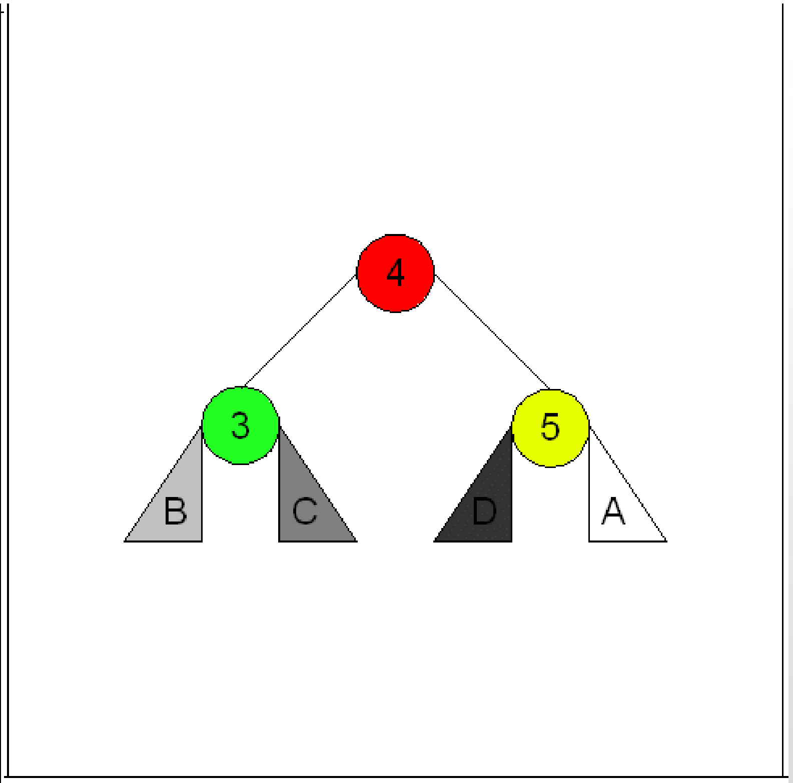




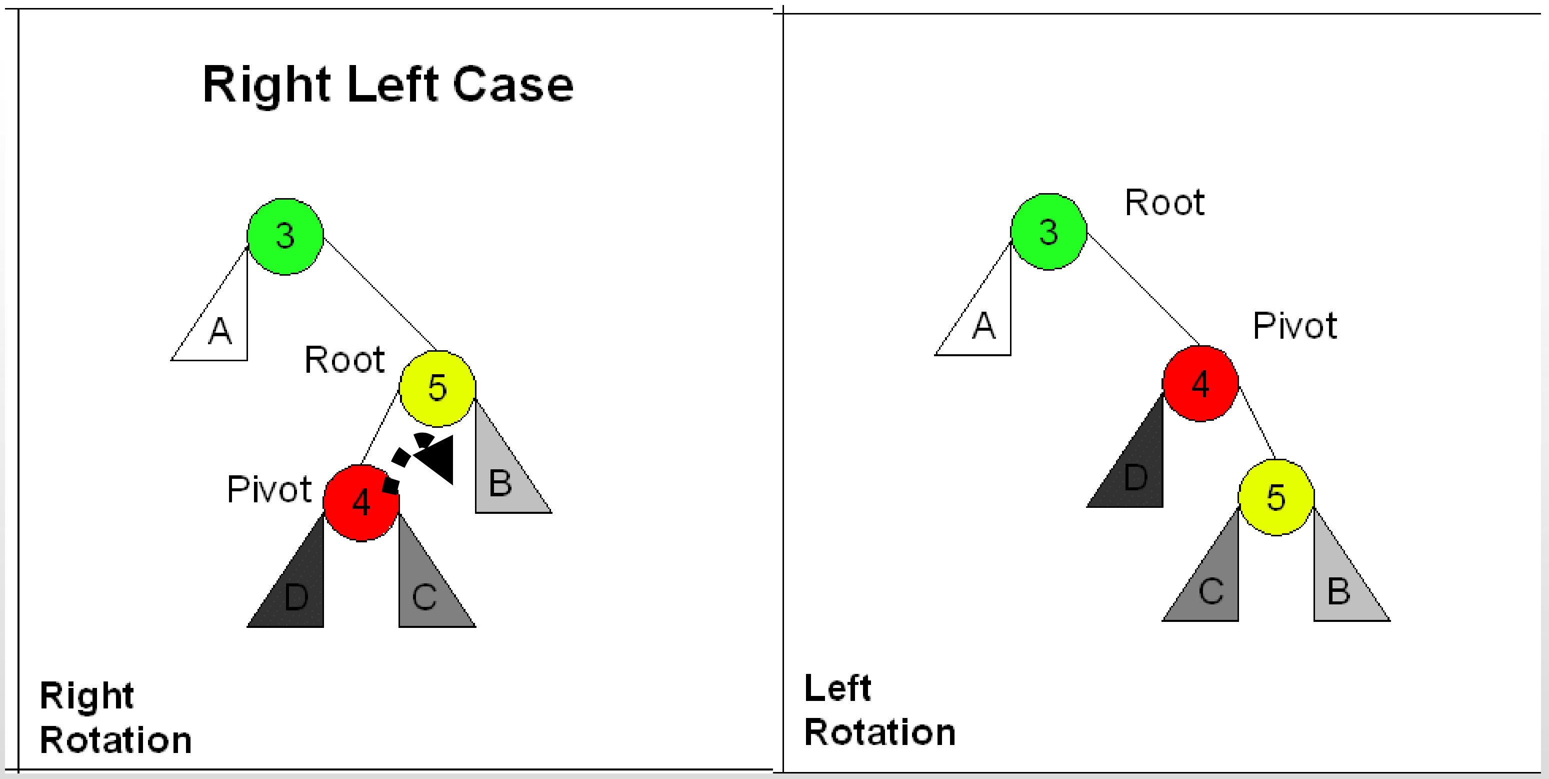
AVL Insertion Guide



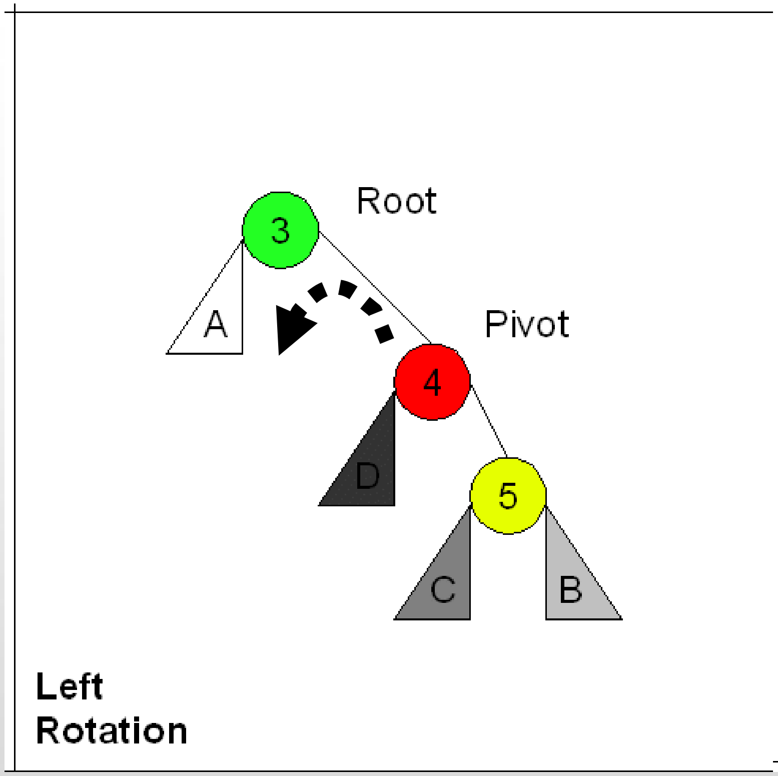
Right Rotation

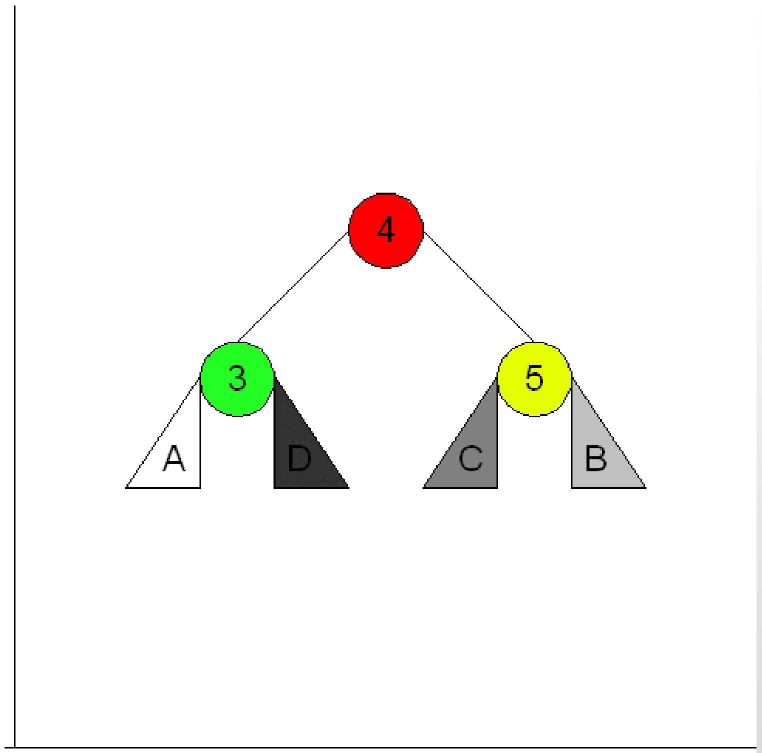


AVL Insertion Guide

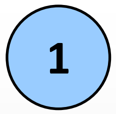


AVL Insertion Guide

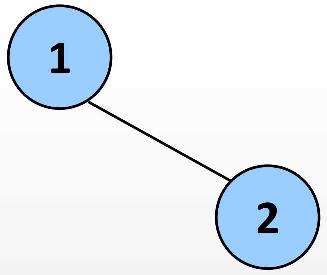




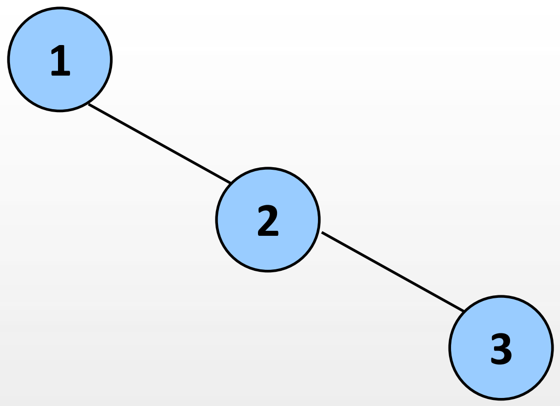
AVL Insertion



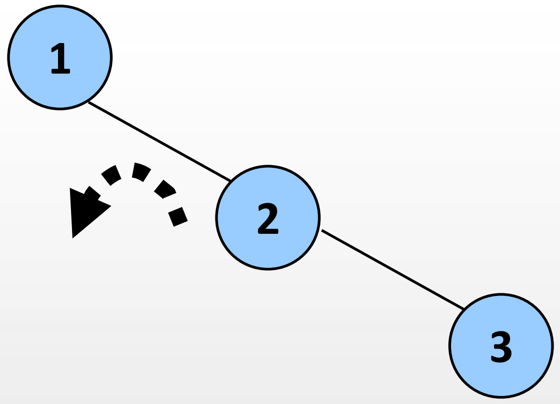
AVL Insertion



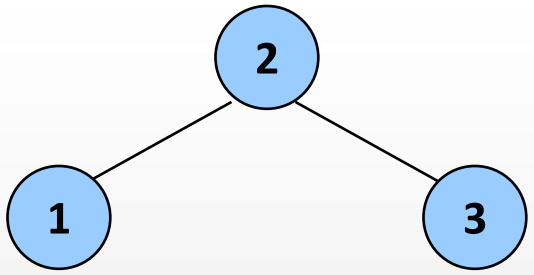
AVL Insertion



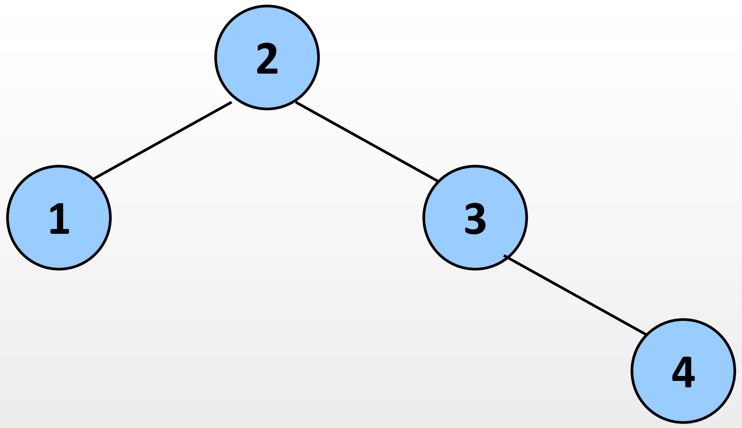
AVL Insertion



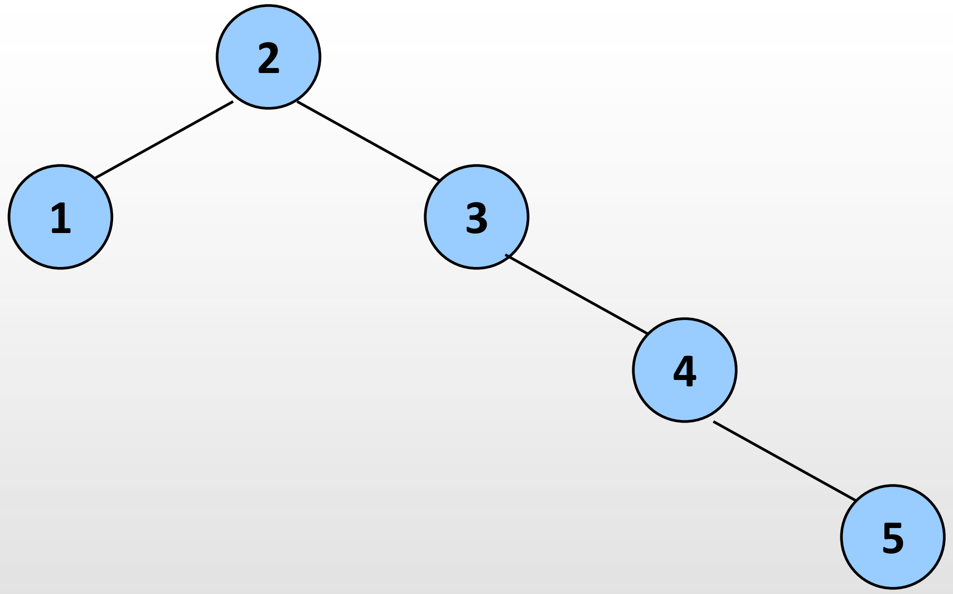
AVL Insertion



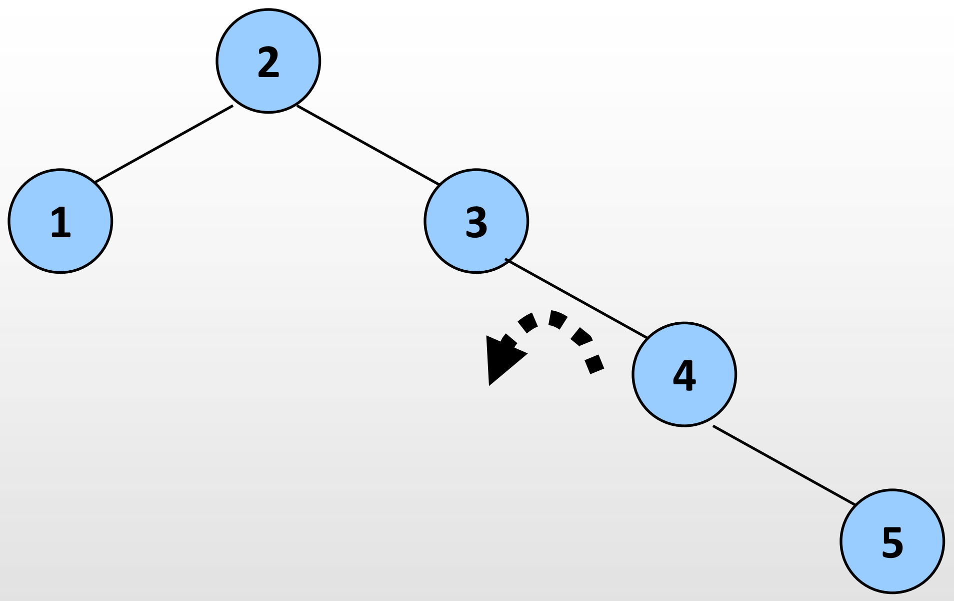
AVL Insertion



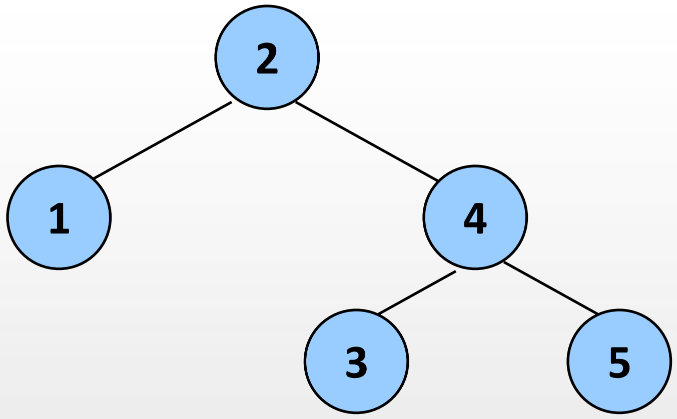
AVL Insertion



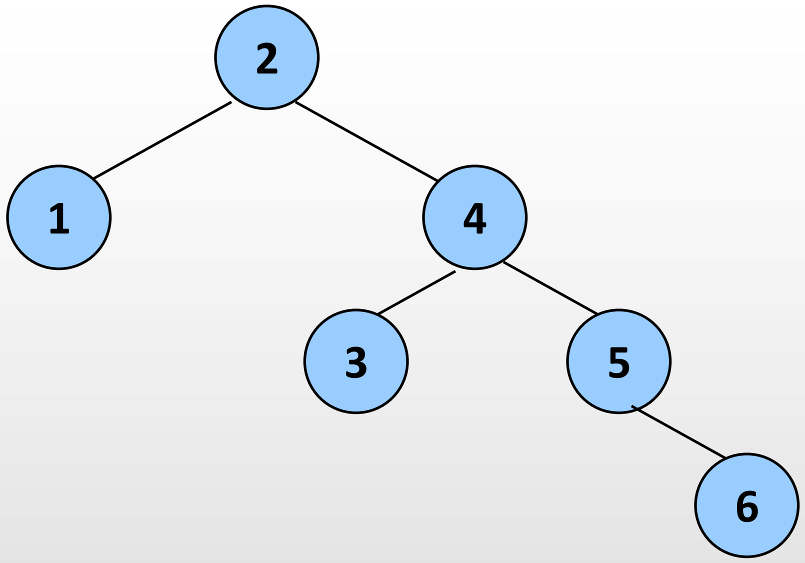
AVL Insertion



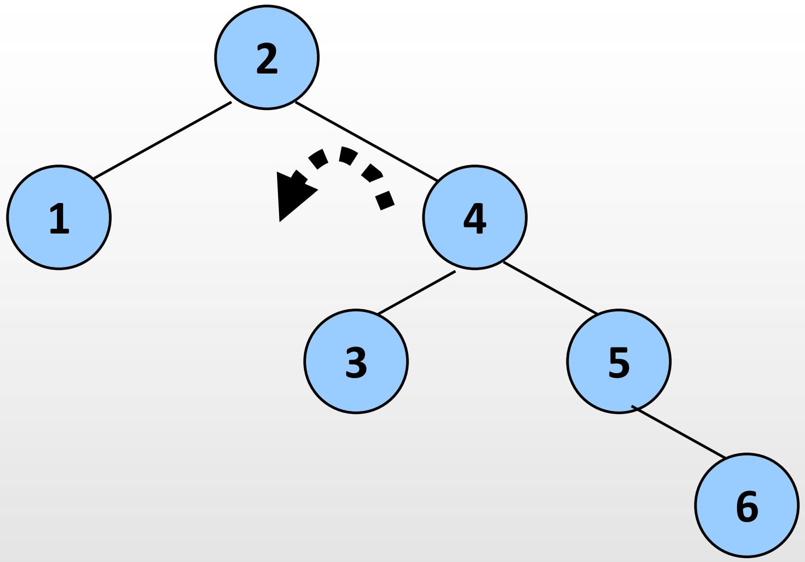
AVL Insertion



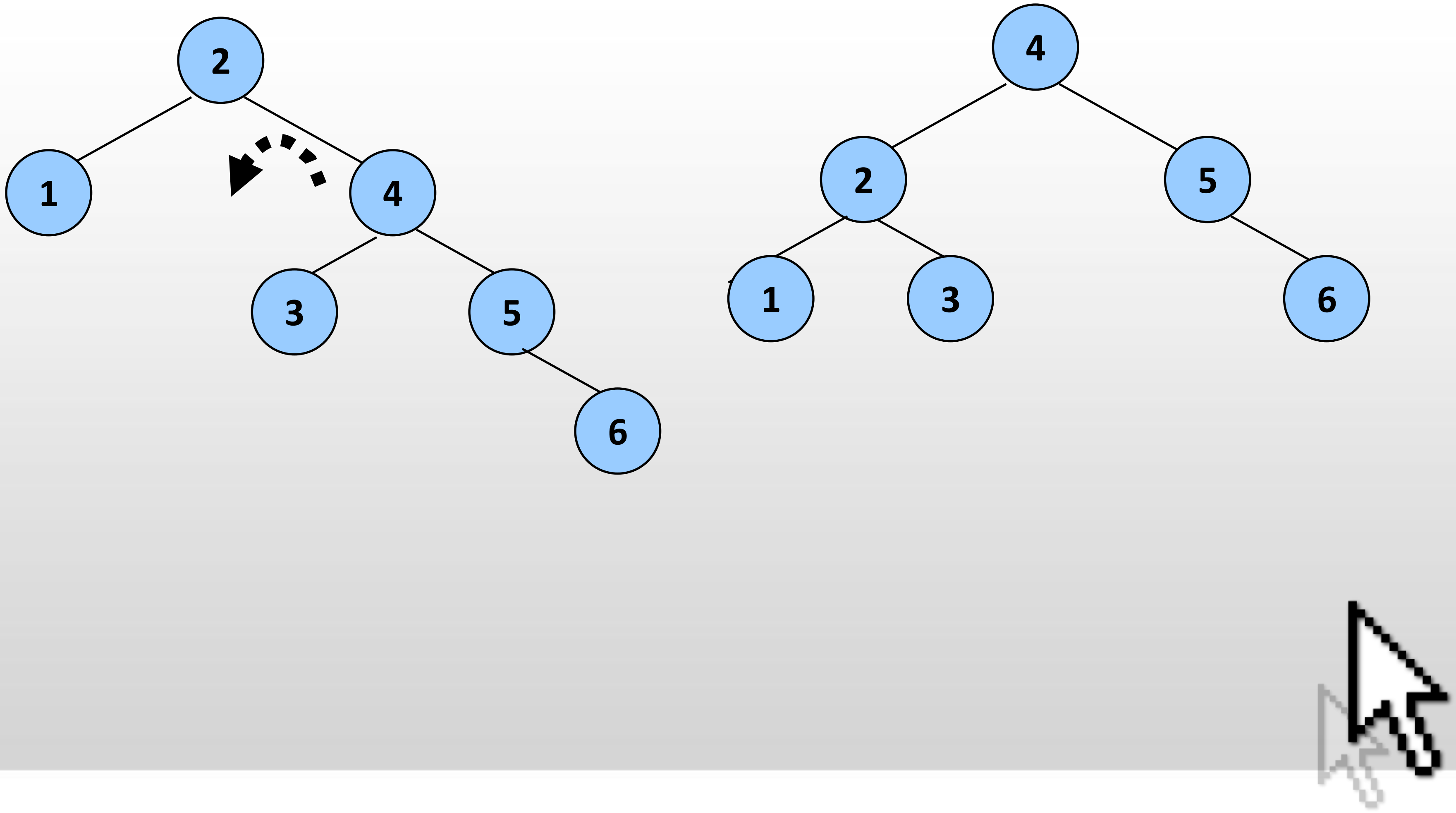
AVL Insertion



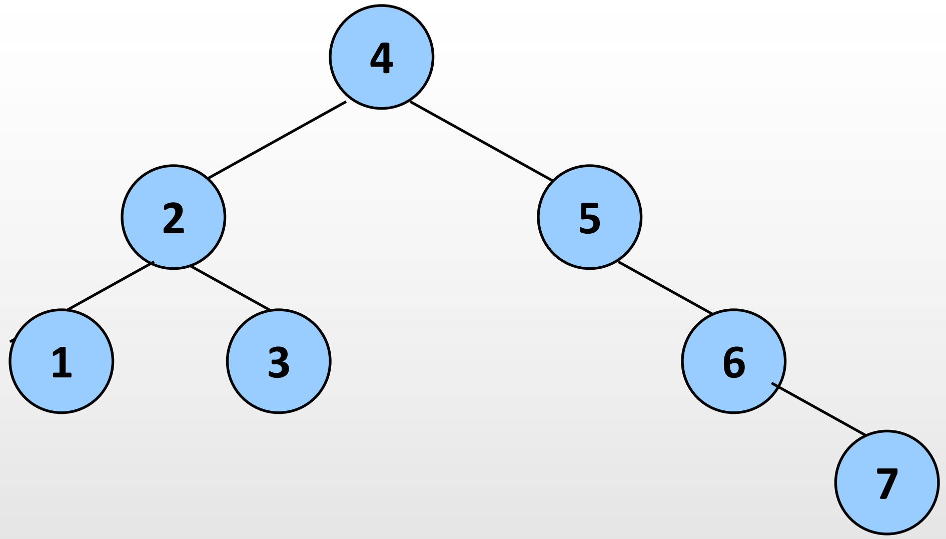
AVL Insertion



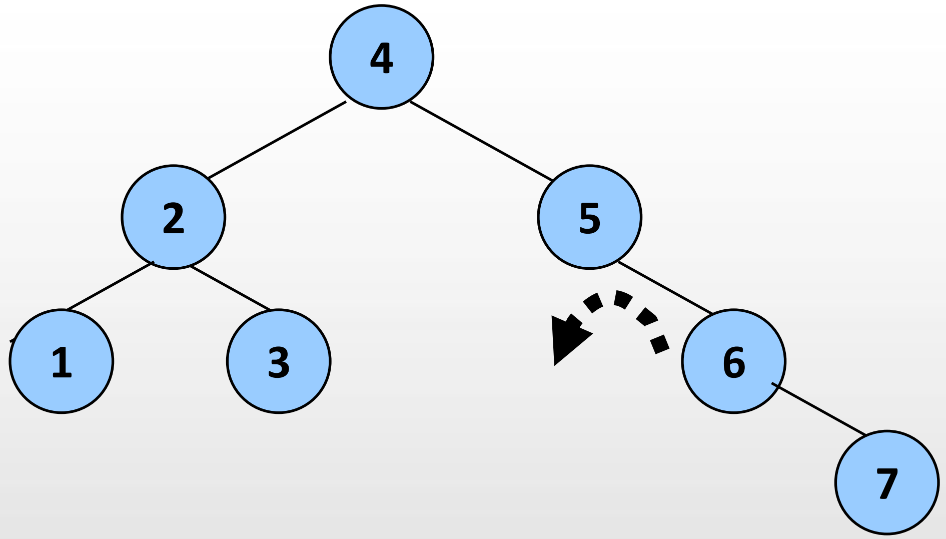
AVL Insertion



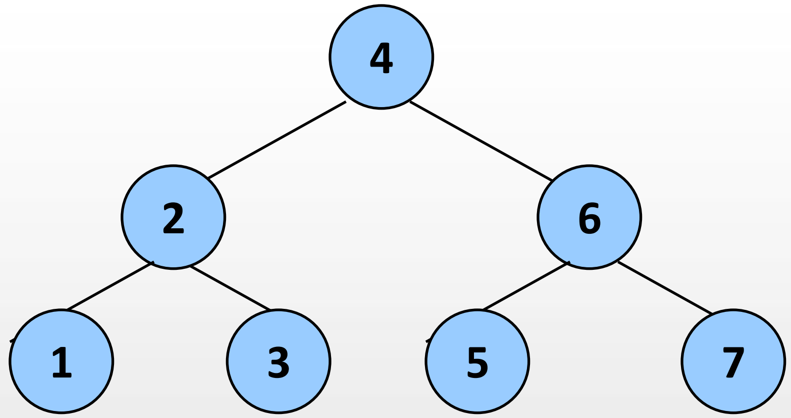
AVL Insertion



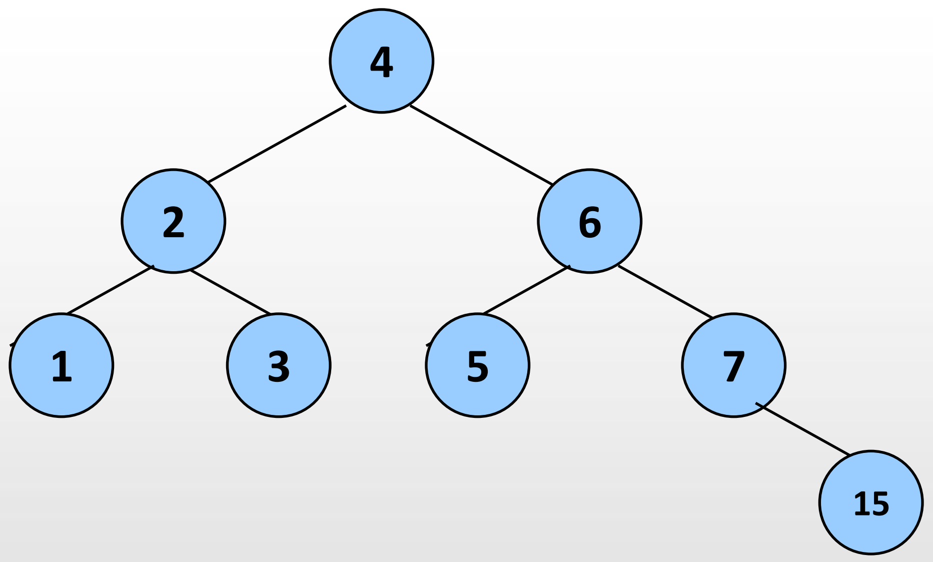
AVL Insertion



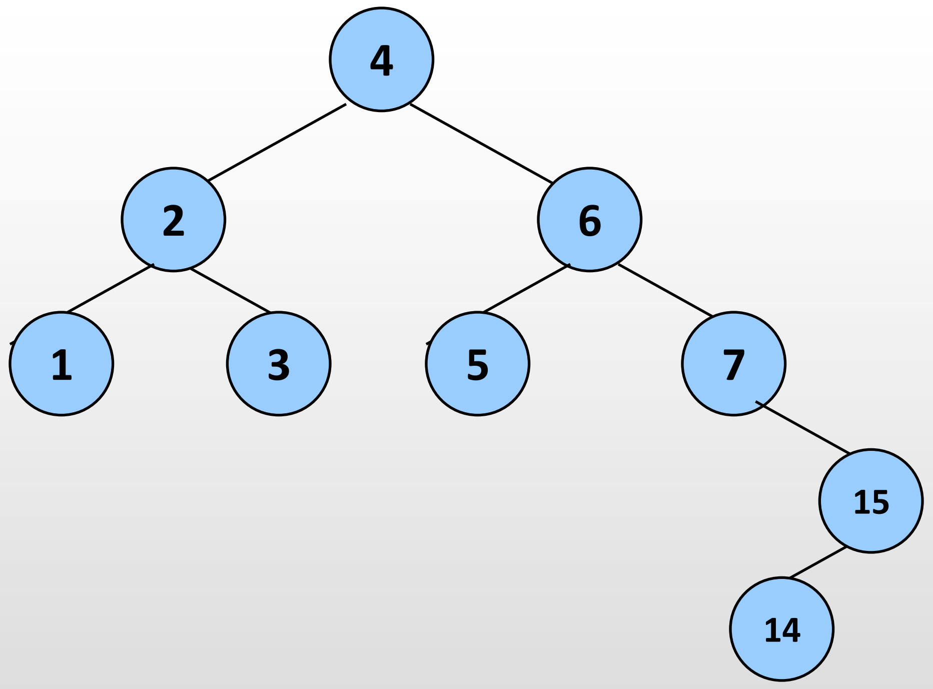
AVL Insertion



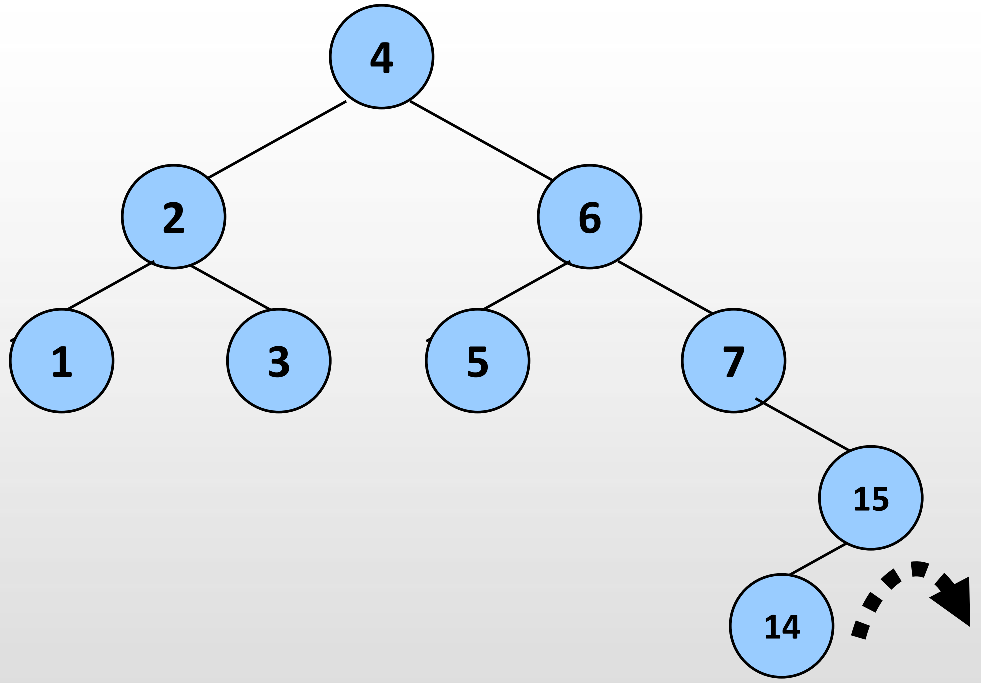
AVL Insertion



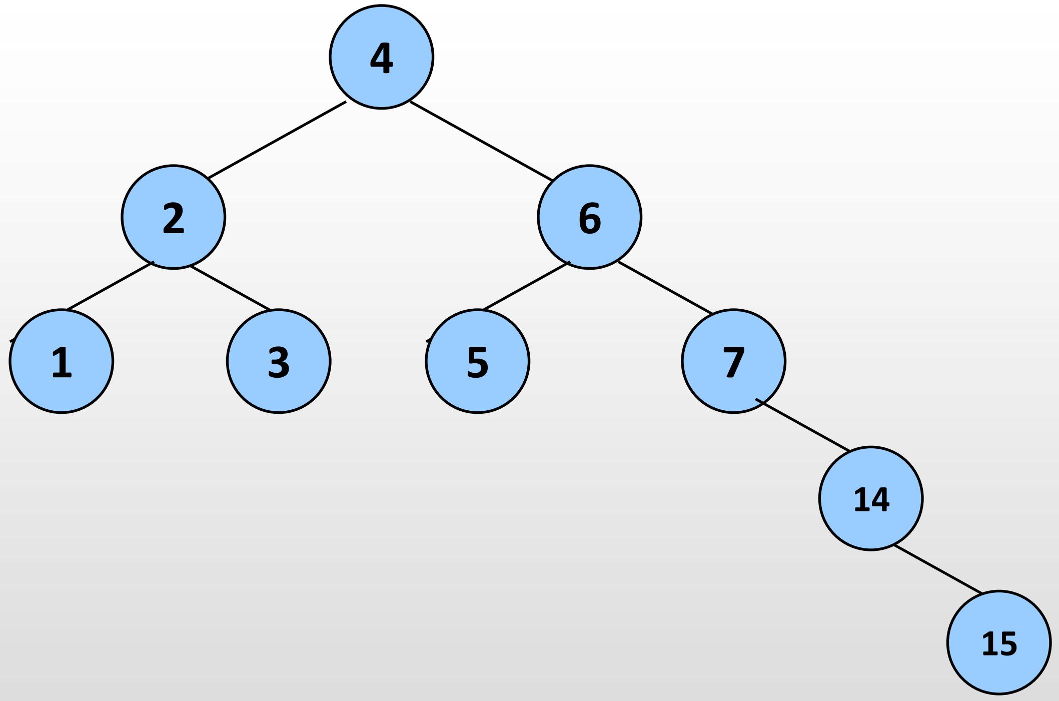
AVL Insertion



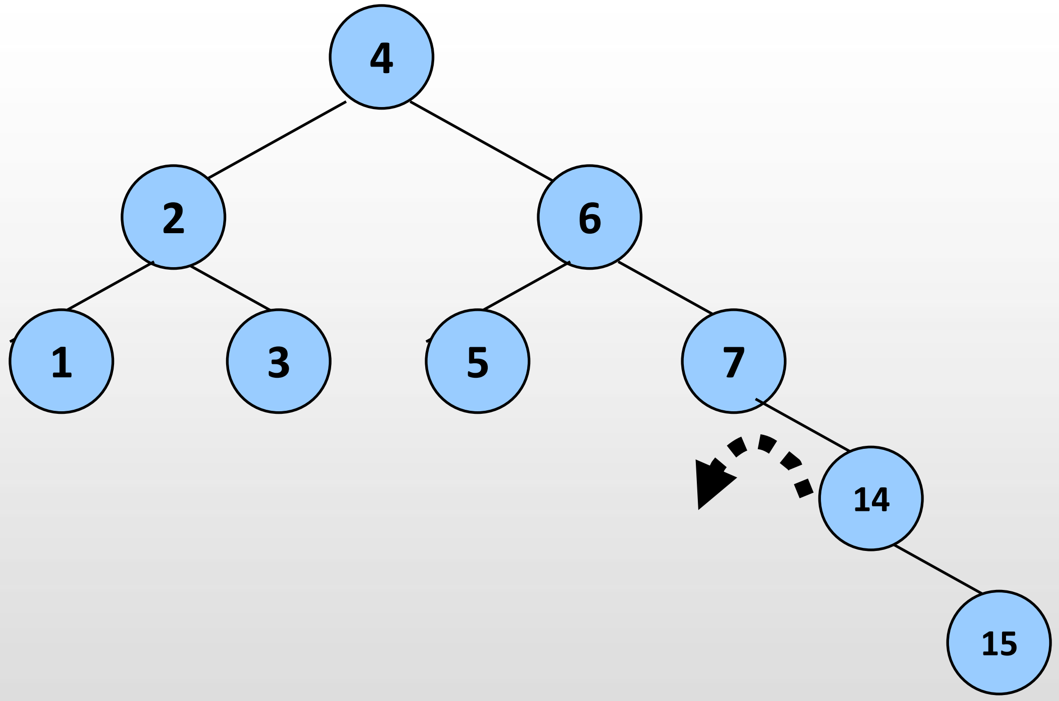
AVL Insertion



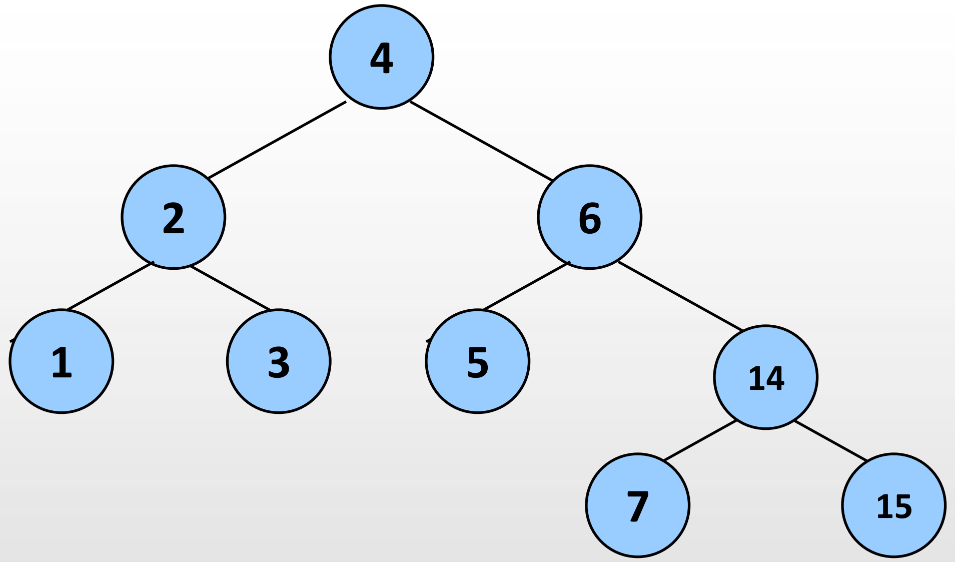
AVL Insertion



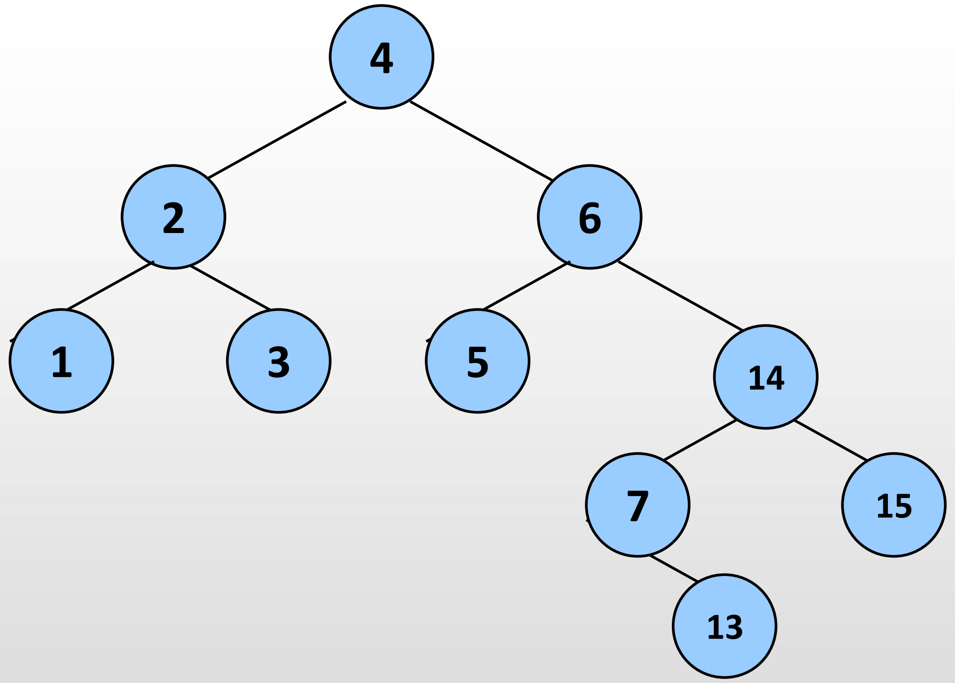
AVL Insertion



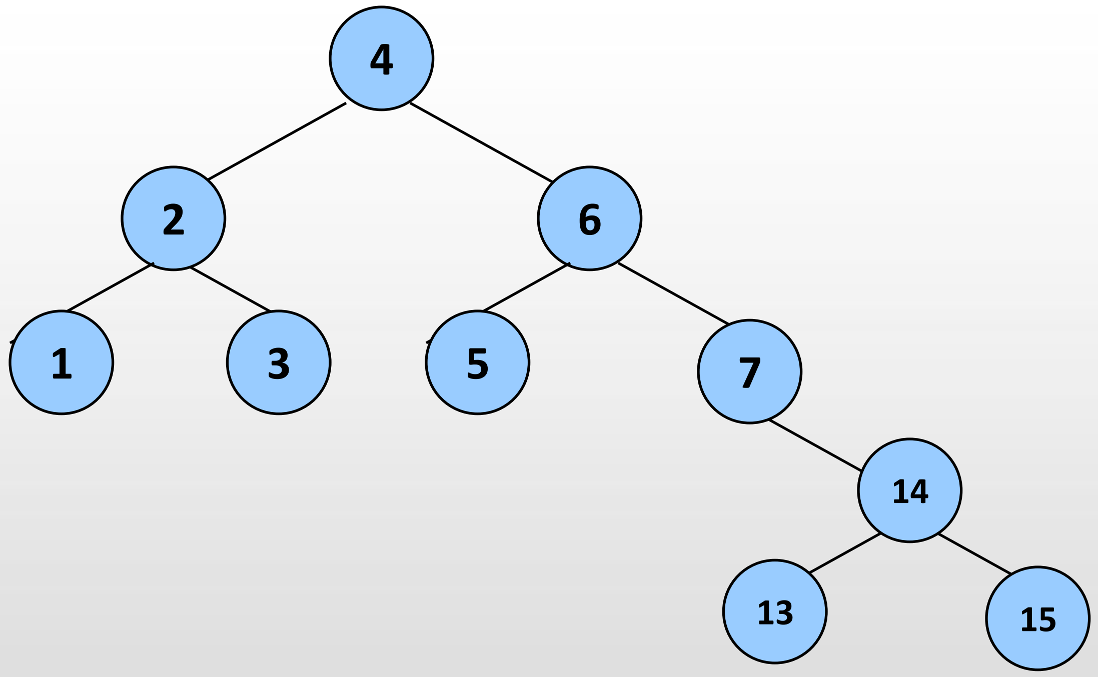
AVL Insertion



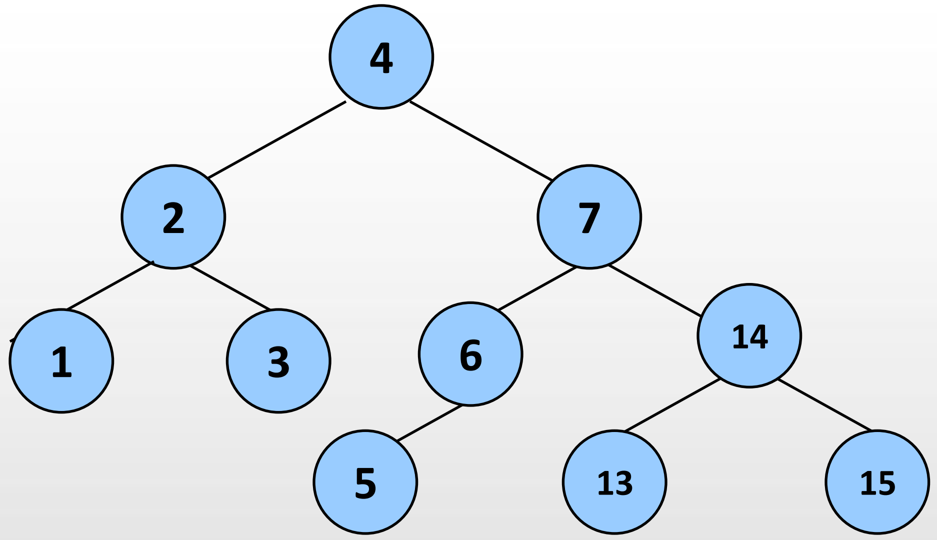
AVL Insertion



AVL Insertion



AVL Insertion



AVL Deletion

- If the node is a leaf, remove it.
- If the node is not a leaf, replace it with either the largest in its left subtree (inorder predecessor) or the smallest in its right subtree (inorder successor), and remove that node.
- The node that was found as replacement has at most one subtree.
- After deletion, retrace the path back up the tree (parent of the replacement) to the root, adjusting the balance factors as needed.