CIS 263 Introduction to Data Structures and Algorithms

AVL Tree

AVL Tree

General idea/rules:

- Your BST should always maintain a balanced state

AVL Tree

General idea/rules:

- Your BST should always maintain a balanced state
- While performing operations such as insertion and deletion, make sure that the tree is blanched by performing the following operations/rotations:
    - LL, RR, LR, and RL

AVL Tree

General idea/rules:

- Your BST should always maintain a balanced state
- While performing operations such as insertion and deletion, make sure that the tree is blanched by performing the following operations/rotations:
  - LL, RR, LR, and RL
- The balance factor of a node is defined as:
  "The absolute difference between the length of the left and right subtrees"

AVL Tree

General idea/rules:

- Your BST should always maintain a balanced state
- While performing operations such as insertion and deletion, make sure that the tree is blanched by performing the following operations/rotations:
  - LL, RR, LR, and RL
- The balance factor of a node is defined as:
  "The absolute difference between the length of the left and right subtrees"
- A tree is imbalanced if t
  he balance factor of any node is >=2

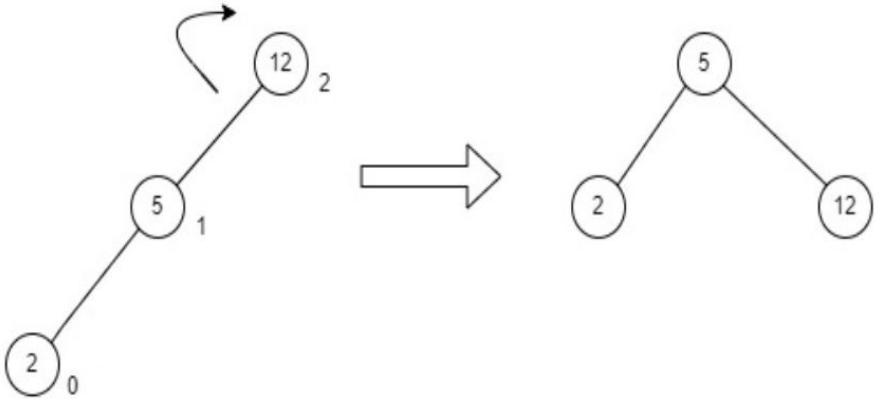
AVL Tree

General idea/rules:

- Your BST should always maintain a balanced state
- While performing operations such as insertion and deletion, make sure that the tree is blanched by performing the following operations/rotations:
    - LL, RR, LR, and RL
- The balance factor of a node is defined as:
  "The absolute difference between the length of the left and right subtrees"
- A tree is imbalanced if t
  he balance factor of any node is >=2
- **If both parent and child become imbalanced at the same time, balance the child first before balancing a parent**

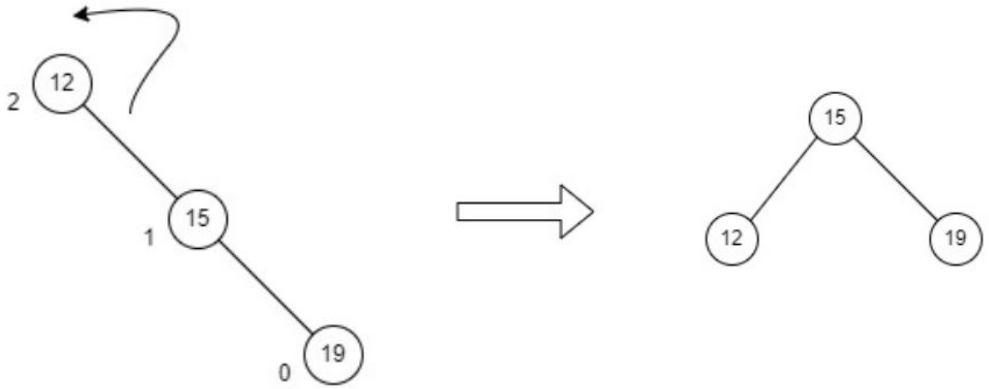
AVL Tree

LL Rotations



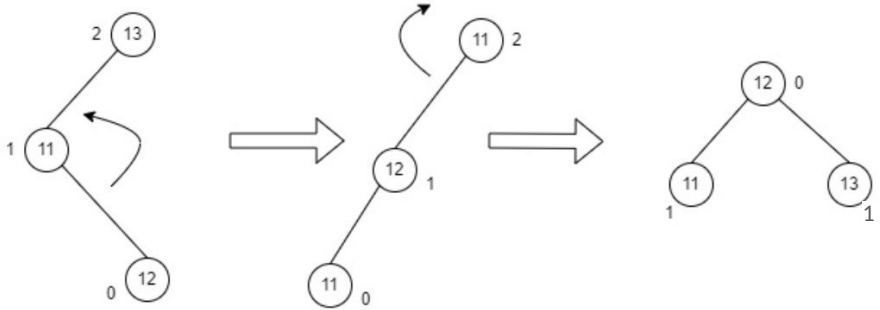
AVL Tree

RR Rotations



AVL Tree

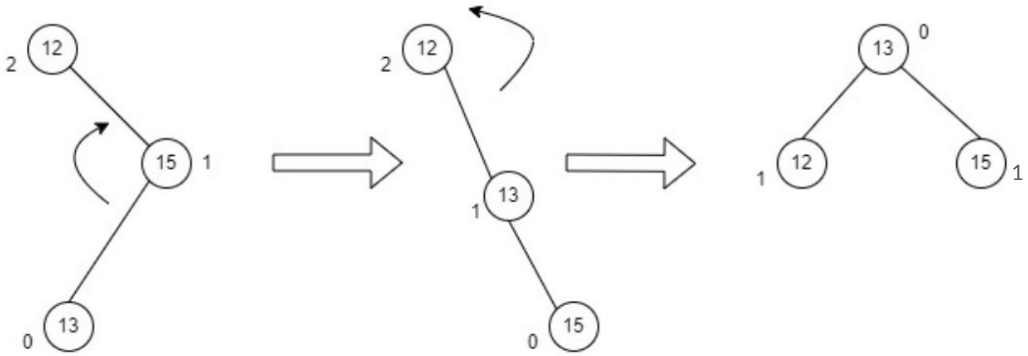
LR Rotations



Alternative way: copy leaf to grandparent, and adjust the grandparent

AVL Tree

RL Rotations



Alternative way: copy leaf to grandparent, and adjust the grandparent

AVL Tree

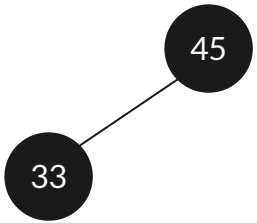
General idea/rules:

- Your BST should always maintain a balanced state
- While performing operations such as insertion and deletion, make sure that the tree is blanched by performing the following operations/rotations:
    - LL, RR, LR, and RL
- The balance factor of a node is defined as:
  "The absolute difference between the length of the left and right subtrees"
- A tree is imbalanced if t
  he balance factor of any node is >=2
- **If both parent and child become imbalanced at the same time, balance the child first before balancing a parent**

Insertion in AVL Tree

- Tree is balanced

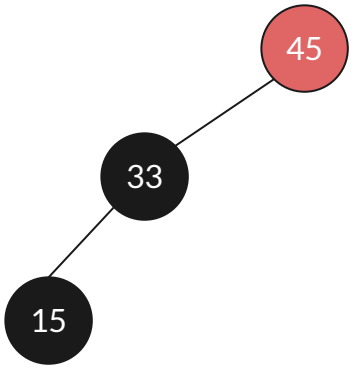
Insertion in AVL Tree



- Tree is balanced

45	33						

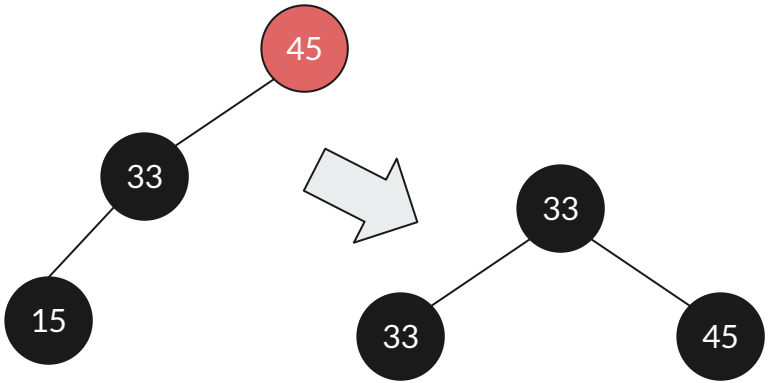
Insertion in AVL Tree



- Node 45 became imbalanced (Balance factor = 2) and so does the Tree

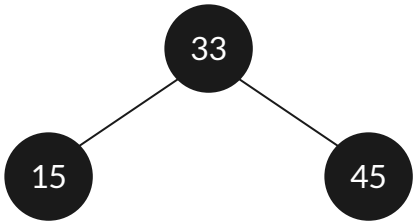
45	33	15					

Insertion in AVL Tree



- Perform LL rotation
- Tree became balanced

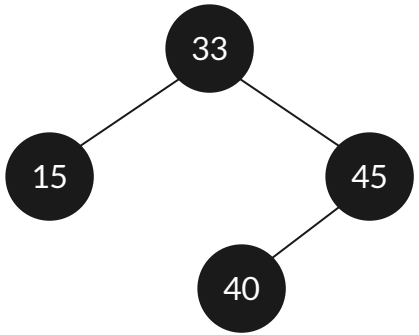
Insertion in AVL Tree



- Tree is Balanced

45	33	15					

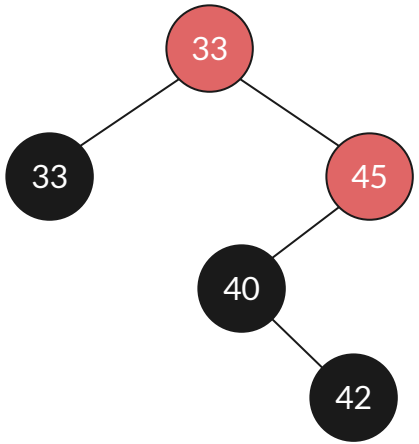
Insertion in AVL Tree



- Tree is Balanced

45	33	15	40				

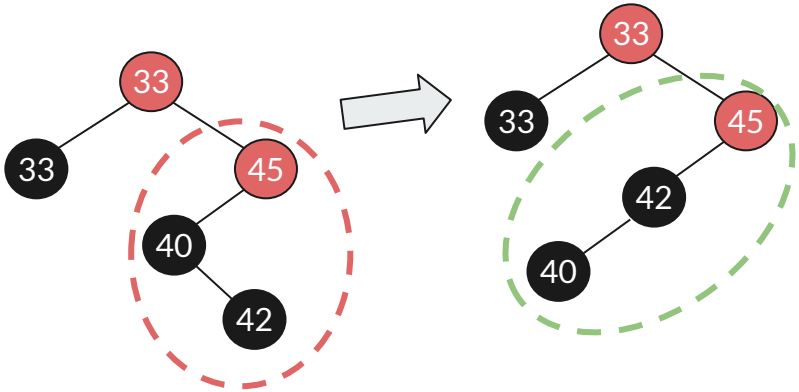
Insertion in AVL Tree



- Both node 33 and 45 became imbalanced and so does the Tree

45	33	15	40	42			

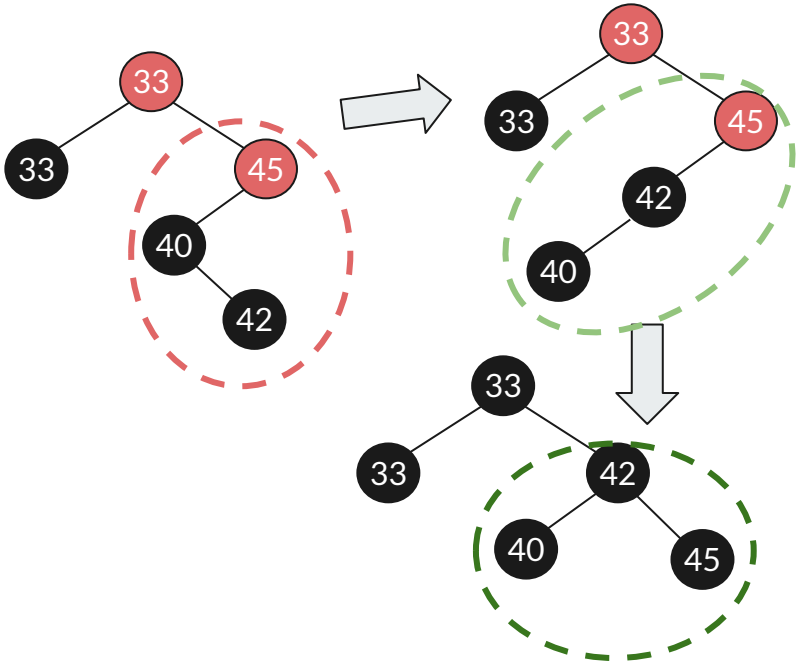
Insertion in AVL Tree



- Both node 33 and 45 became imbalanced and so does the Tree
- We fix node 45 by a LR rotation followed by by a LL rotation

45	33	15	40	42			

Insertion in AVL Tree



- Both node 33 and 45 became imbalanced and so does the Tree
- We fix node 45 by a LR rotation followed by by a LL rotation
- Tree is balanced now

45	33	15	40	42			

Same process if we delete nodes from AVL Tree