

빅 데이터 혁신 공유 대학

파이썬으로 배우는 데이터 구조

한동대학교 전산전자공학부

김영섭 교수



교육부



한국연구재단



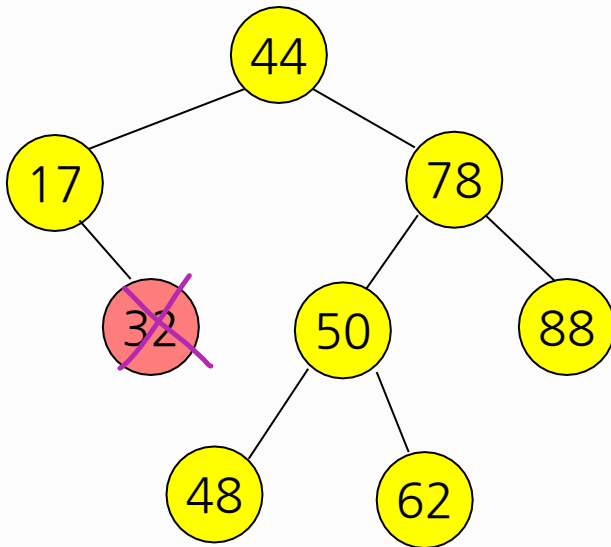
Data Structures in Python

Chapter 7 - 2

- Binary Search Tree(BST)
- BST Algorithms
- **AVL Tree**
- AVL Algorithms

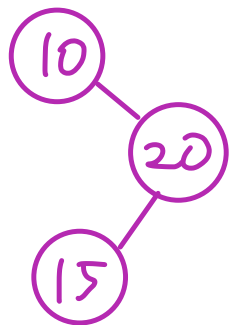
Final - 다음 결과를 PPT혹은 그림파일로 Piazza에 올립니다.

- Draw AVL trees whenever the tree changes its shape by insertion and deletion. Include trees before and after its rotation and the type of rotation.
 - Tree가 모양을 **바꿀 때마다** AVL tree들을 그리고, 각 단계별로 LL, RR, LR, RL case를 표시하고, Double Rotation일 경우에는 이를 표시하고, 또한 동시에 LL/RR로 경우를 나누어 나타내십시오.
- (1) [2.0점] Insert the sequence of elements (10, 20, 15, 25, 30, 16, 18, 19) into an AVL tree. / ①
Delete 30 in the AVL tree that you got above and rebalance it. / ②
- (2) [0.5점] Delete 32 in **the AVL tree shown below** and rebalance it.

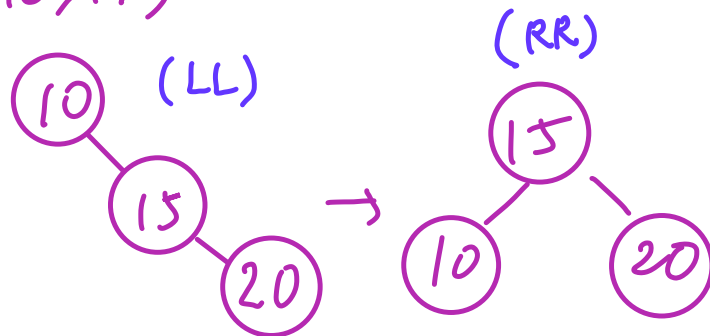


(1) AVL (10, 20, 15, 25, 30, 16, 18, 19)

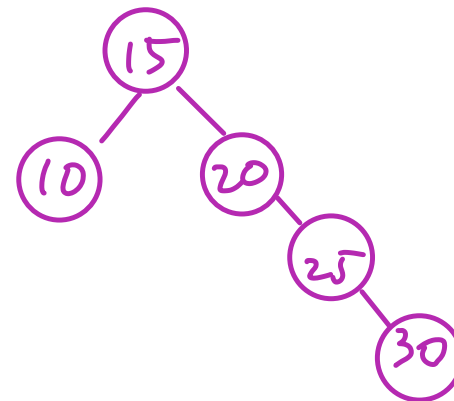
Insert
10, 20, 15



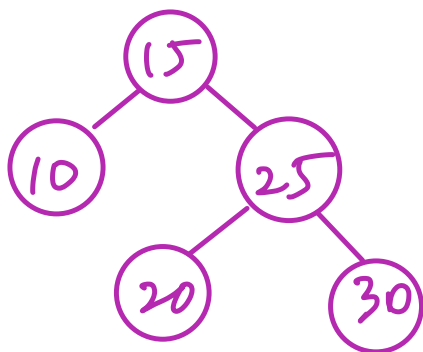
RL(10)



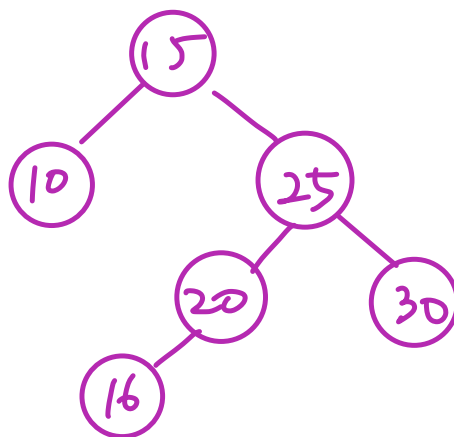
Insert
25, 30



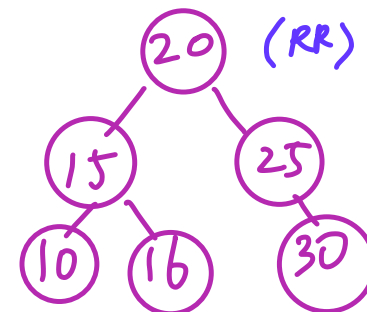
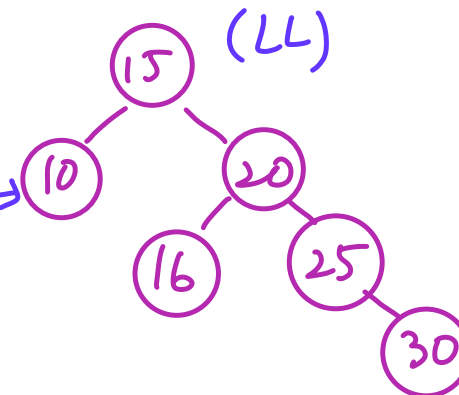
RR(20)



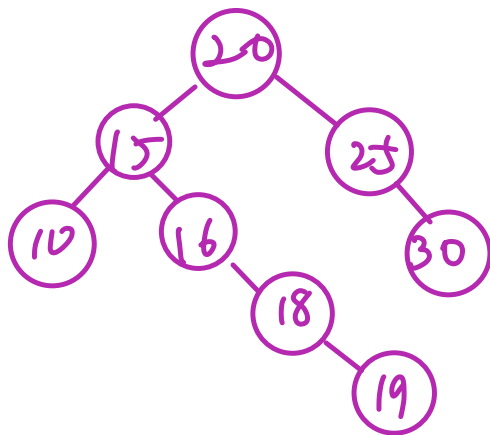
Insert
16



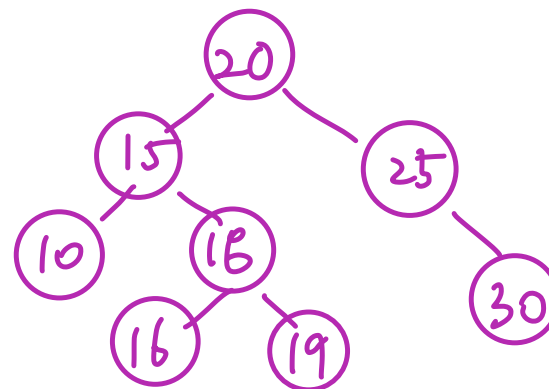
RL(15)



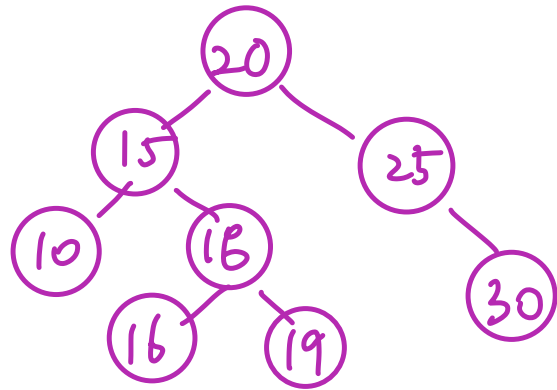
Insert
18, 19



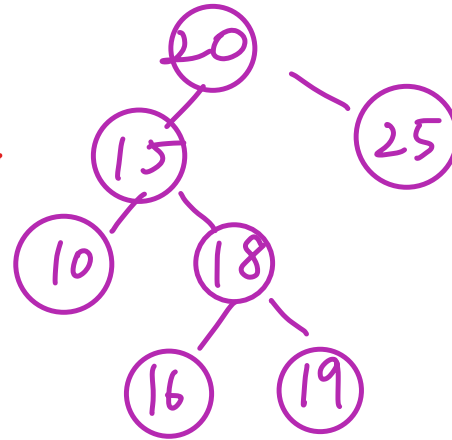
RR(16)



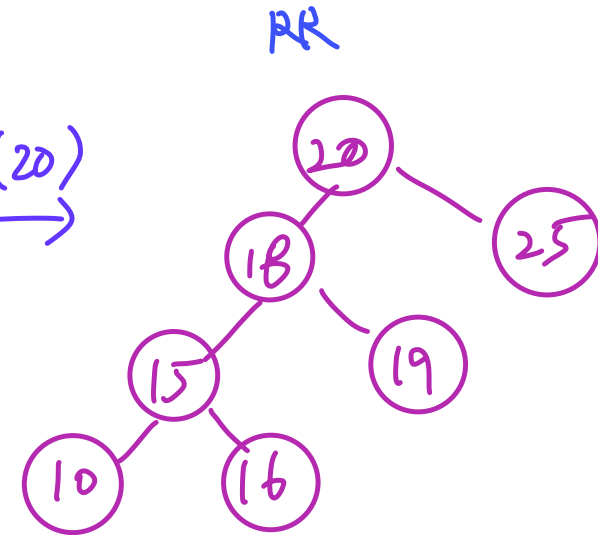
(1) Delete 30



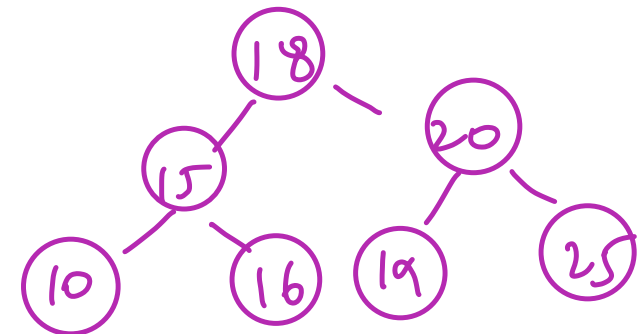
Delete
30



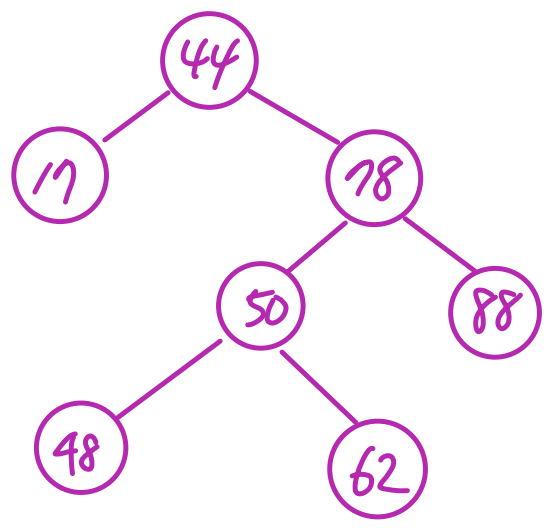
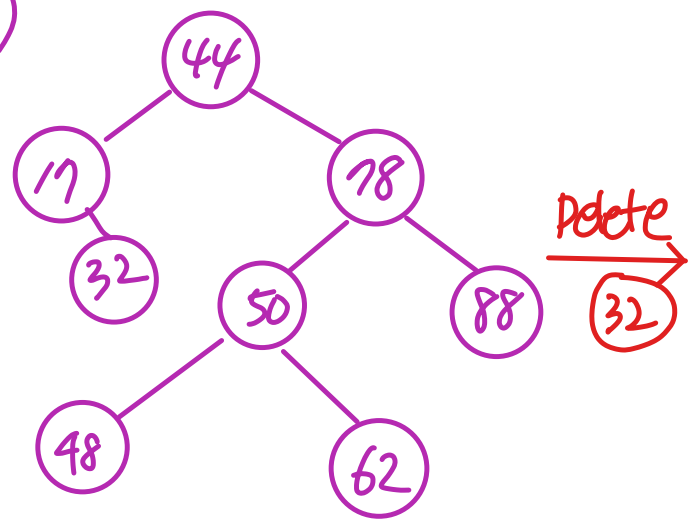
LR(20)



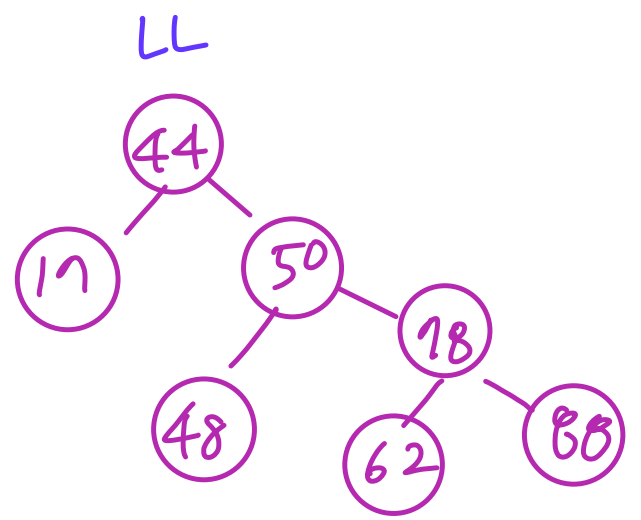
LL



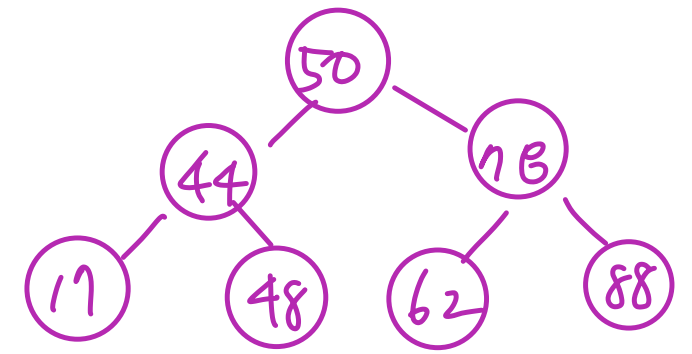
(2)



RL(44)
→



↓
RR



Data Structures in Python

Chapter 7 - 2

- Binary Search Tree(BST)
- BST Algorithms
- **AVL Tree**
- AVL Algorithms