

Data Structure

(Java programming)

Chapter 13.



Contents

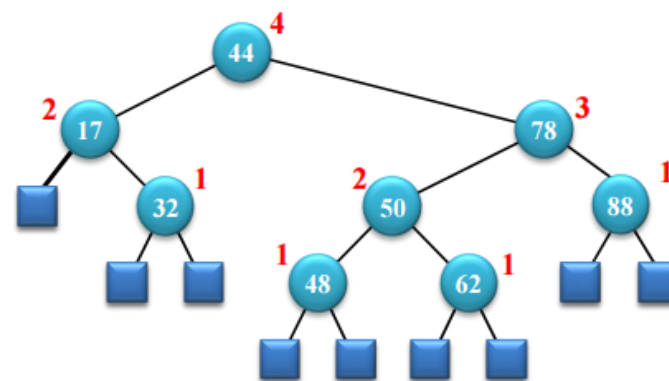
- **AVL Tree**

- *AVL Tree*

• AVL Tree

AVL Tree

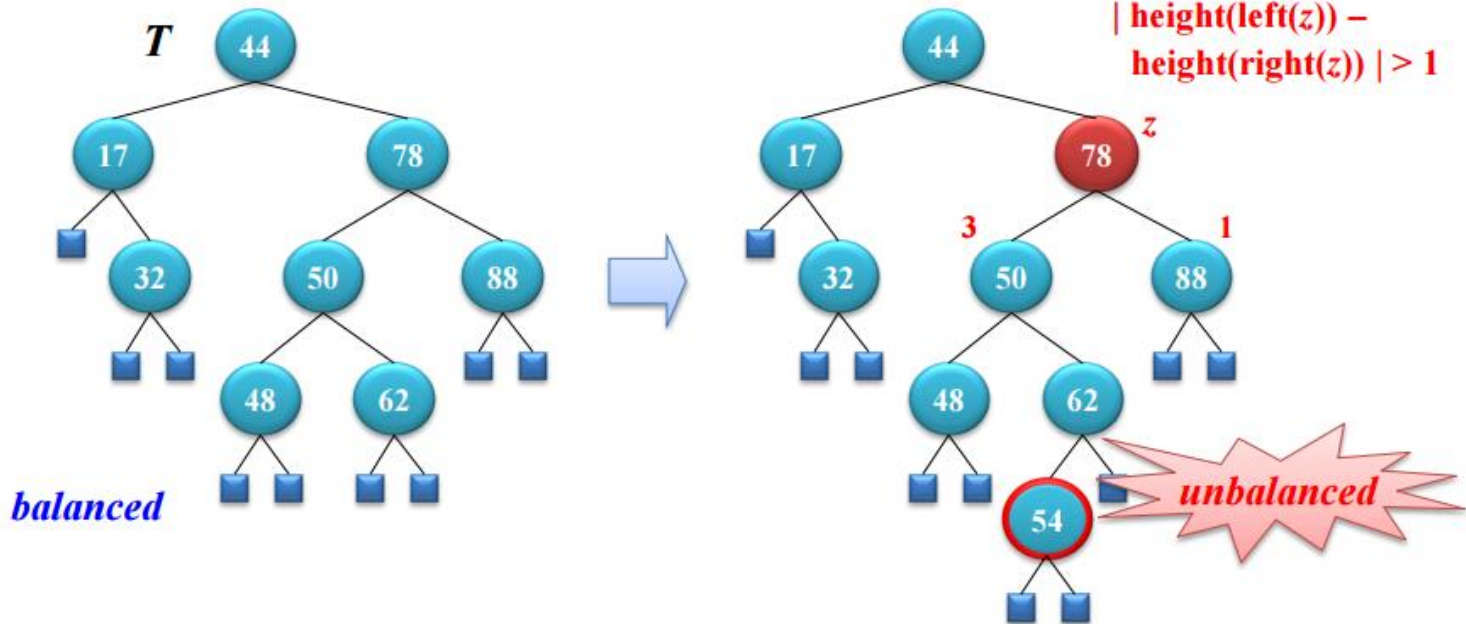
- *AVL tree* is a BST that satisfies the following property
 - **Height-balance property**
for every internal node v of an AVL tree, the heights of its children can differ by 1 at most, i.e.,
 $|\text{height}(\text{left}(v)) - \text{height}(\text{right}(v))| \leq 1$
 - i.e., a subtree of an AVL tree is also an AVL tree
 - ❖ *AVL tree is named after the initial of its inventors, Adelson-Velskii and Landis*



[An example of an AVL tree]

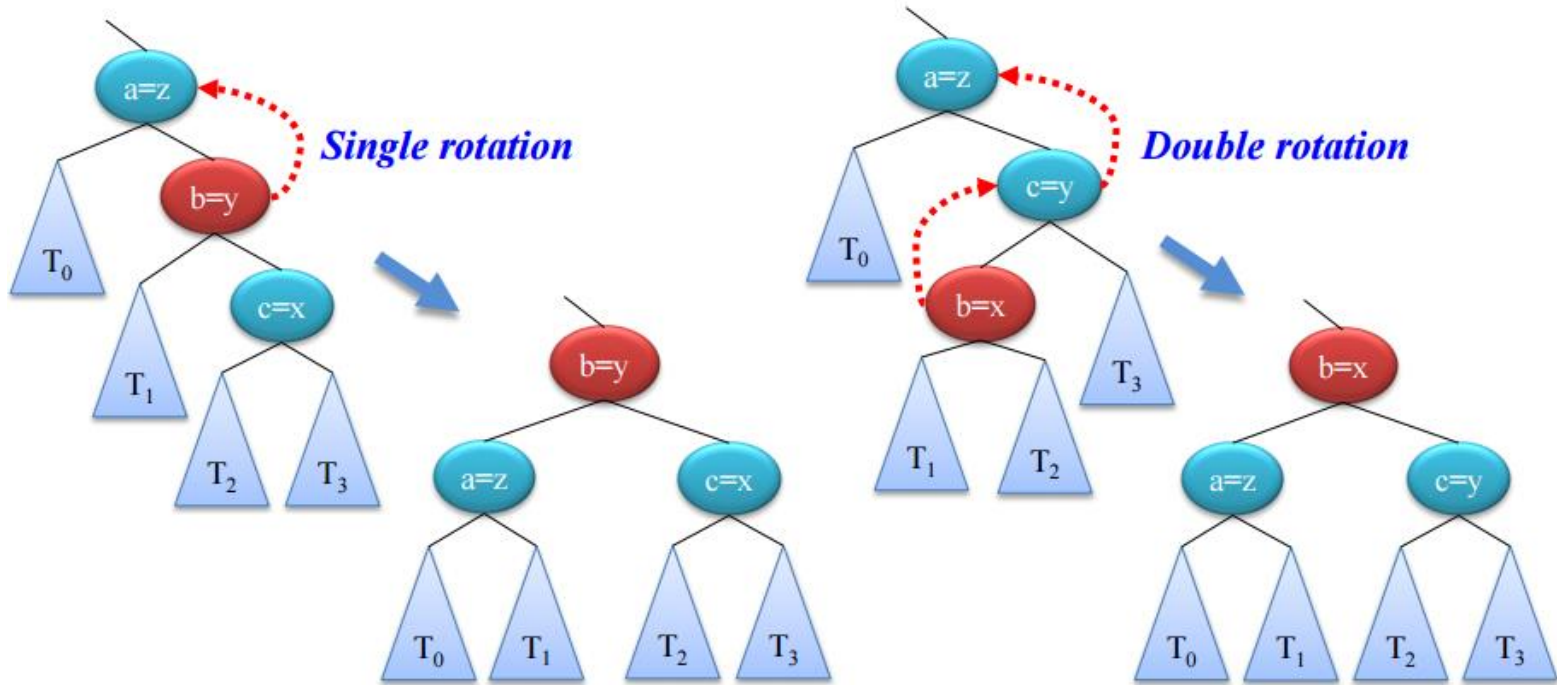
- AVL Tree

AVL Tree



AVL Tree

Trinode Restructuring



- **AVL Tree**

Trinode Restructuring Algorithm

Algorithm $\text{restructure}(x)$:

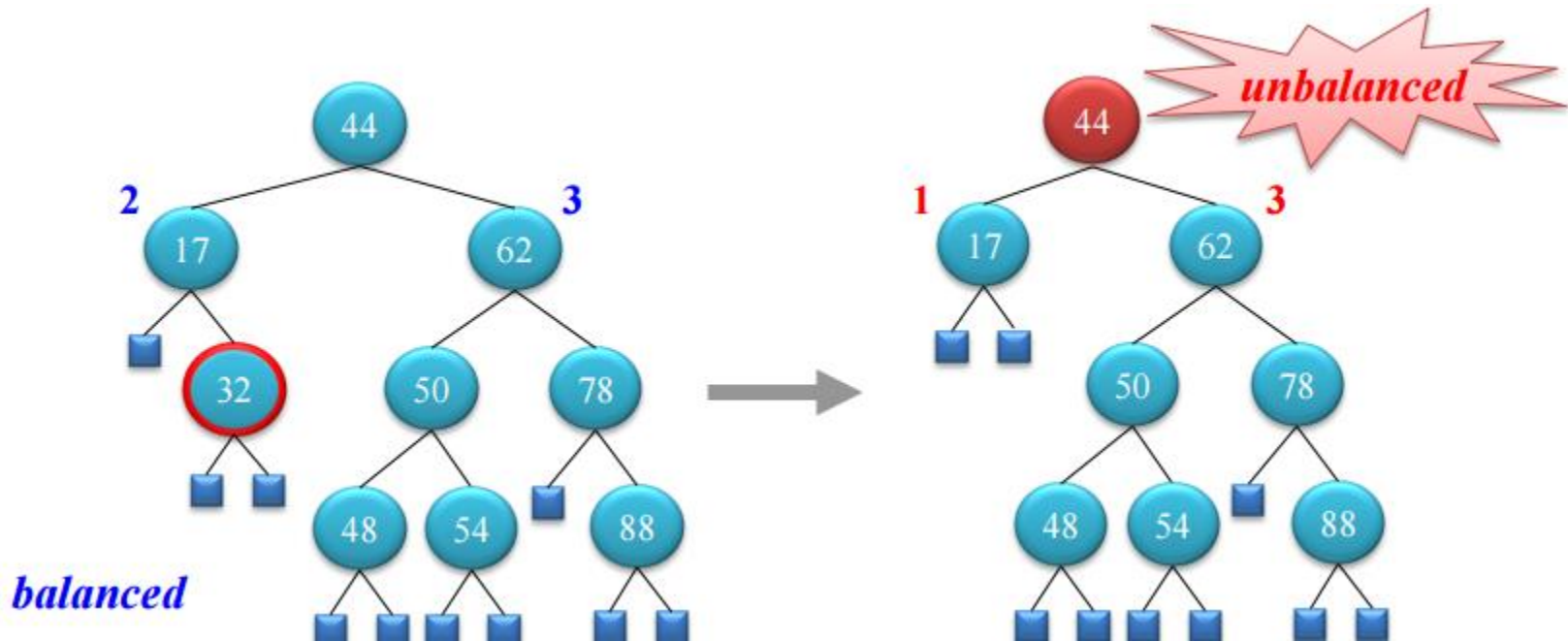
Input: A node x of a binary search tree T that has both a parent y and a grandparent z

Output: Tree T after a trinode restructuring (which corresponds to a single or double rotation) involving nodes x , y , and z

- 1: Let (a, b, c) be a left-to-right (inorder) listing of the nodes x , y , and z , and let (T_0, T_1, T_2, T_3) be a left-to-right (inorder) listing of the four subtrees of x , y , and z not rooted at x , y , or z .
- 2: Replace the subtree rooted at z with a new subtree rooted at b .
- 3: Let a be the left child of b and let T_0 and T_1 be the left and right subtrees of a , respectively.
- 4: Let c be the right child of b and let T_2 and T_3 be the left and right subtrees of c , respectively.

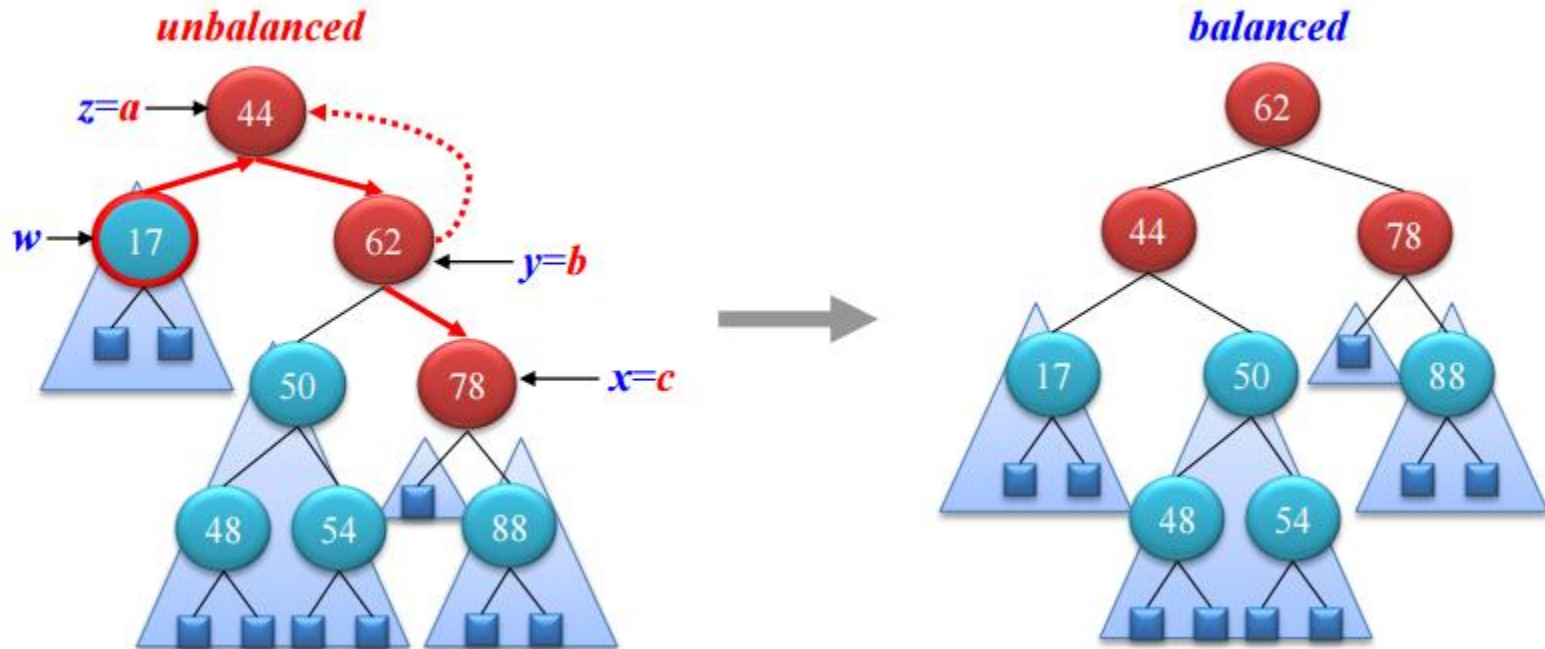
AVL Tree

Removal



AVL Tree

Removal - rebalancing



- **Notice <FINAL EXAM>**

When

2017/06/16 18:00 PM

Where

새천년관302호 A반

새천년관402호/502호 B반

새천년관302호 C반

How

오픈북 실습 시험 (수업 강의자료 / 필기자료만 지참 가능)