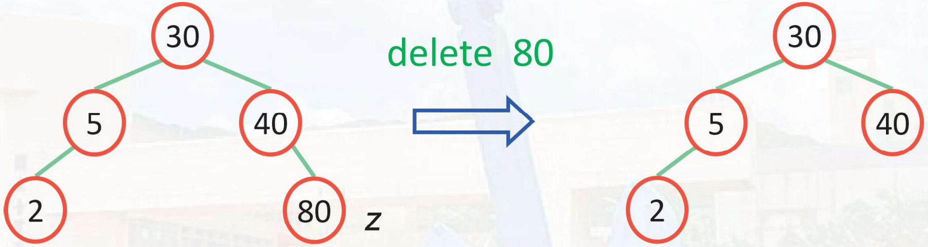
Deletion from a BST

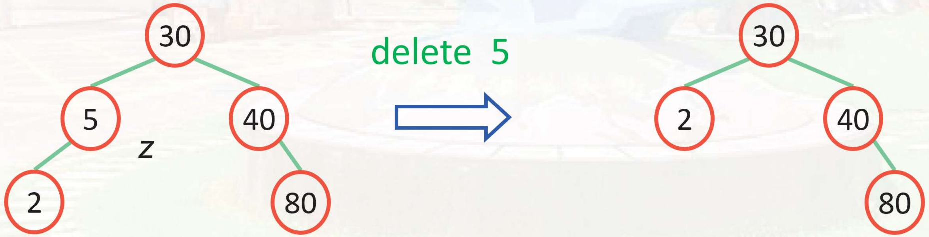
- Case 1: **leaf**
  - delete the node and set the pointer from the parent node to NULL.
- Case 2: **having only one child**:
  - delete the node and change the pointer from the parent node to the single-child node.
- Case 3: **having two children**:
  - replaced by the largest element in its left subtree, or replaced by the smallest element in its right subtree.

Illustraton (Case 1 & 2)

Case 1:

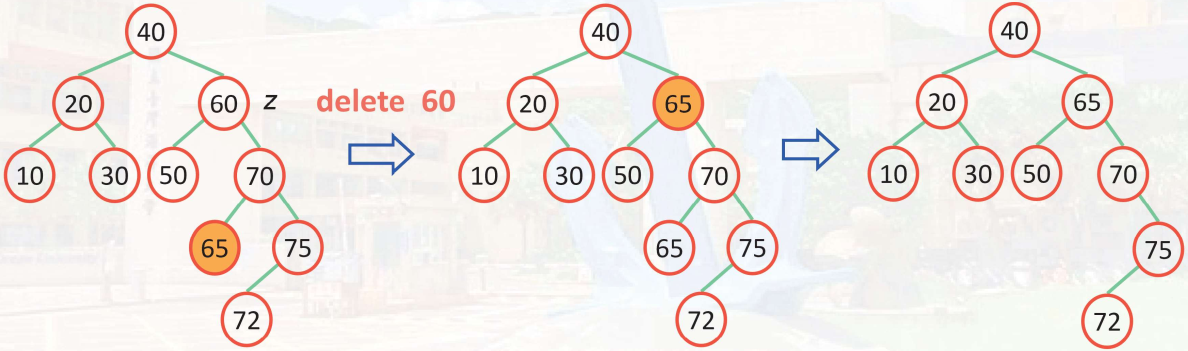


Case 2:



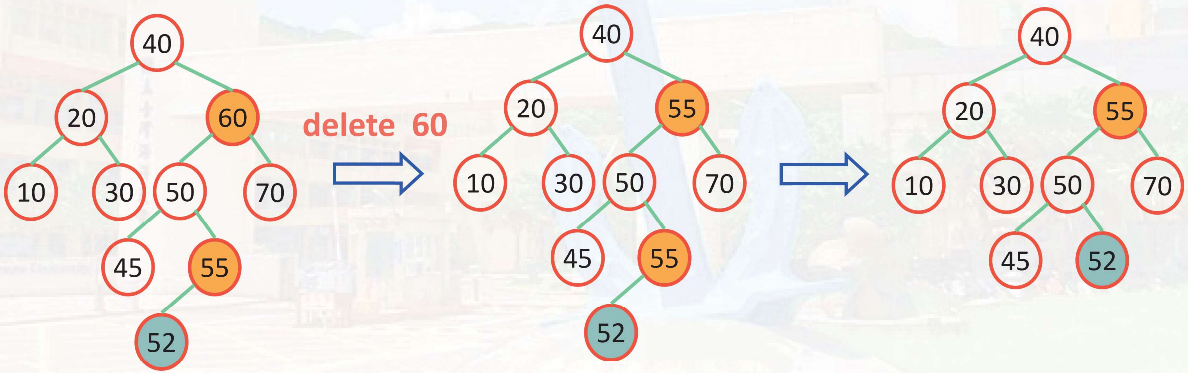
Illustraton (Case 3)

Case 3:



Illustraton (Case 3)

Case 3:



Time Complexity Analysis of Deleting a Node in a BST

- The case: Deleting a nonleaf node that has two children.
- We can verify (Exercise) that, in both ways, it is originally in a node with a degree of at most one.
    - Check the largest and smallest elements in a subtree.
- The time complexity for case 3 is $O(h)$ (h: the height of the BST).
- A deletion can be performed in $O(h)$ time.