Deletion Operation in BST

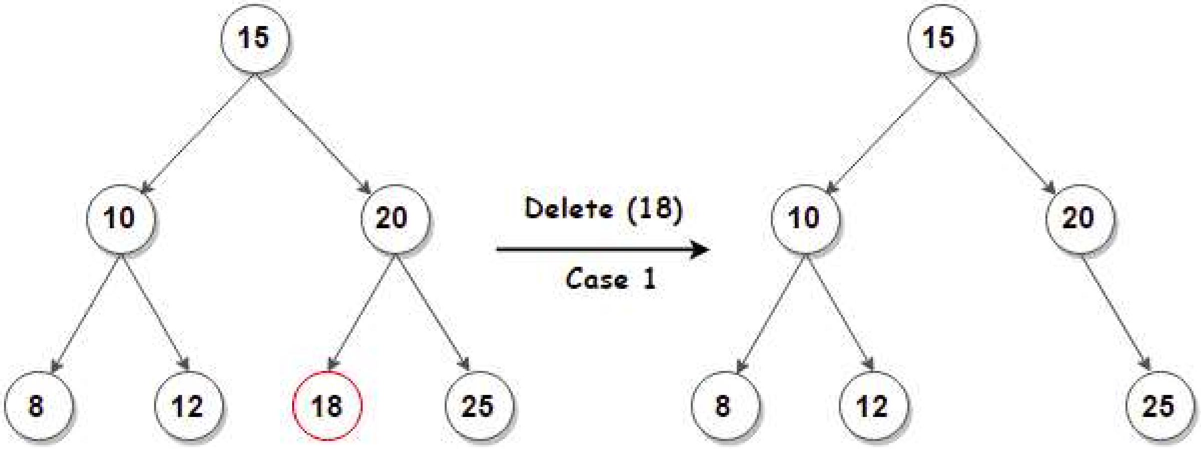
In a binary search tree, the deletion operation is performed with **O(log n)** time complexity. Deleting a node from Binary search tree includes following three cases...

- **Case 1: Deleting a Leaf node (A node with no children)**
- **Case 2: Deleting a node with one child**
- **Case 3: Deleting a node with two children**

Case 1: Deleting a leaf node

We use the following steps to delete a leaf node from BST...

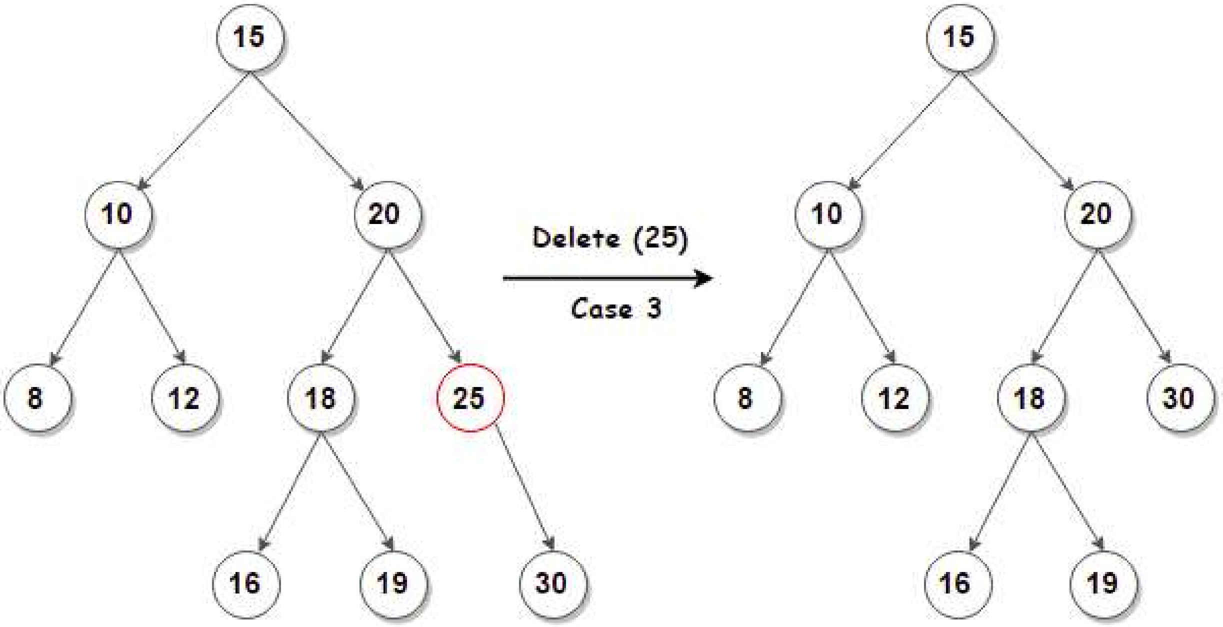
- Step 1 - **Find** the node to be deleted using **search operation**
- Step 2 - Delete the node using **free** function (If it is a leaf) and update NULL in the parent reference and terminate the function.



Case 2: Deleting a node with one child

We use the following steps to delete a node with one child from BST...

- Step 1 - **Find** the node to be deleted using **search operation**
- Step 2 - If it has only one child then create a link between its parent node and child node.
- Step 3 - Delete the node using **free** function and terminate the function.

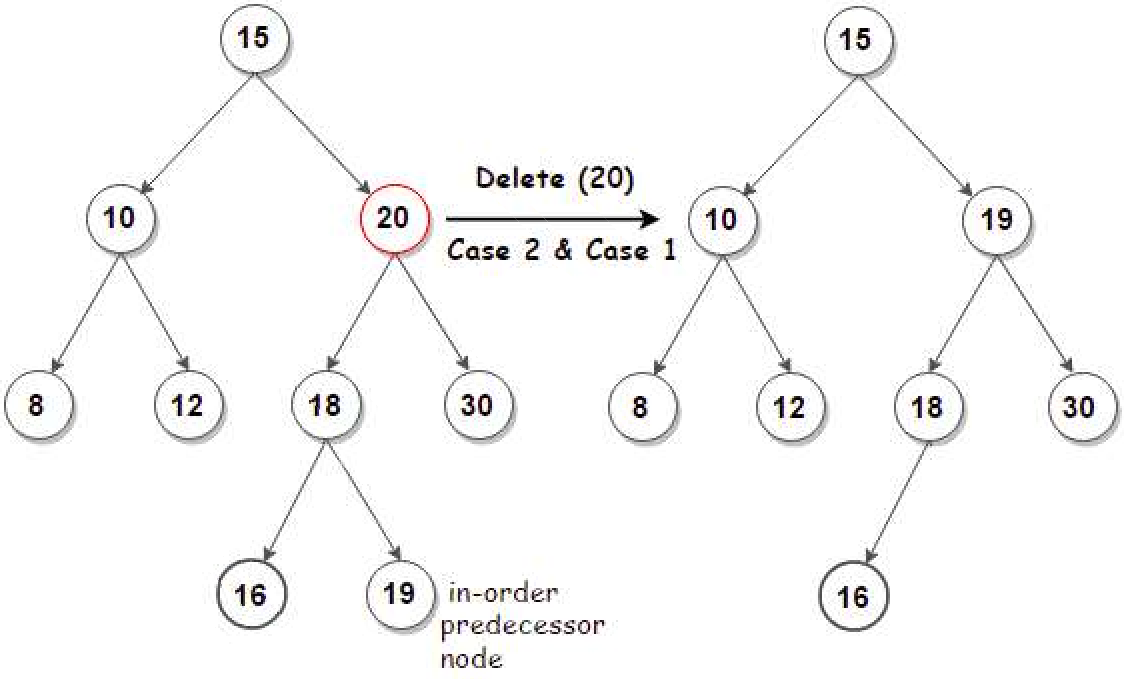


Case 3: Deleting a node with two children

We use the following steps to delete a node with two children from BST...

- Step 1 - **Find** the node to be deleted using **search operation**
- Step 2 - If it has two children, then find the **largest** node in its **left subtree** (OR) the **smallest** node in its **right subtree**.
- Step 3 - **Swap** both **deleting node** and node which is found in the above step.
- Step 4 - Then check whether deleting node came to **case 1** or **case 2** or else goto step 2

- Step 5 - If it comes to **case 1**, then delete using case 1 logic.
- Step 6- If it comes to **case 2**, then delete using case 2 logic.
- Step 7 - Repeat the same process until the node is deleted from the tree.



Note :

Reference Books : Taken contents and diagrams from various websites.