



2-3 Deletion (Case #2)

How do we delete an element present at an internal node? It will be explained in this lesson with the help of an example.

We'll cover the following



- Case 2: Element at Internal Node:
- Example

Case 2: Element at Internal Node:#

Deletion is always performed at the leaf. So whenever we need to delete a key at the internal node, we swap it with any of its in-order successors and somehow make it shift to any leaf node as deletion is always performed at the *leaf*. Shift the element at the leaf node and then delete it. The element to be deleted can be swapped by:

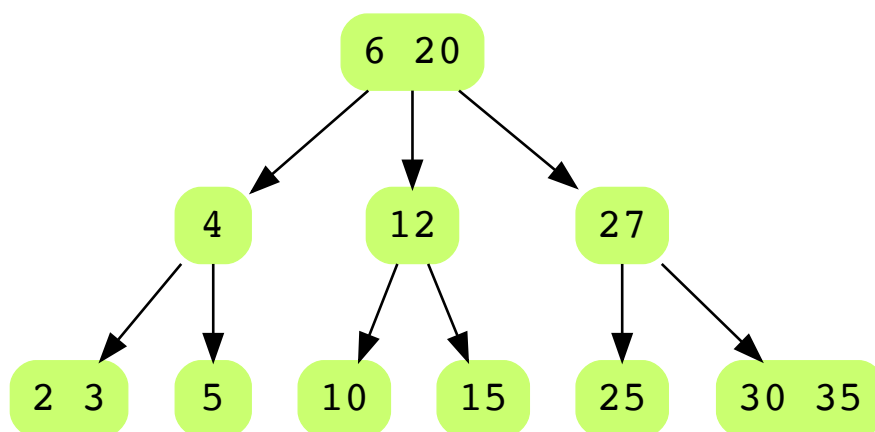
- an element with the largest key on the *left*
- an element with the smallest key on the *right*

This is applicable when the child node has more than one key stored at the node. If there is only one value at child node, then you are bound to swap the parent with whatever value child node has.

Example#



See the following illustration which covers the two scenarios discussed above:



1 of 20



That's it! We are now done with 2-3 Trees. The next (and last) lesson will discuss a more advanced version of 2-3 Trees.

Interviewing soon? We've partnered with [Hired](#) so that companies apply to you instead of you applying to them. [See how](#) ⓘ


[← Back](#)
[Next →](#)

2-3 Deletion (Case #1)

2-3-4 Trees

☒ Mark as Complete

