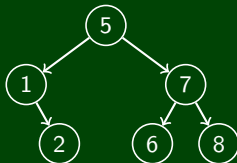


# **BST Remove Notes**



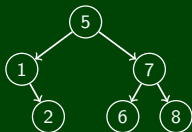
Consider the following tree:



Let's try the following removals:

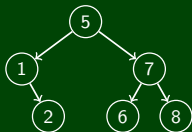
- `tree.remove(2)`
- `tree.remove(1)`
- `tree.remove(7)`
- `tree.remove(5)`

`tree.remove(2)`



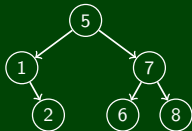
→

`tree.remove(1)`



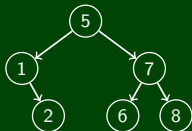
→

`tree.remove(7)`

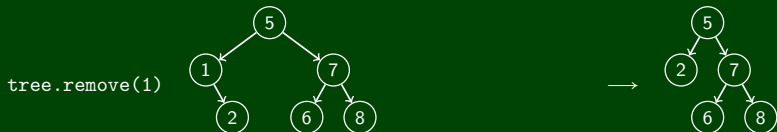
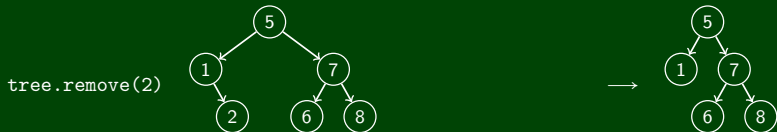


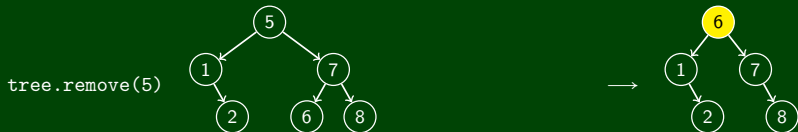
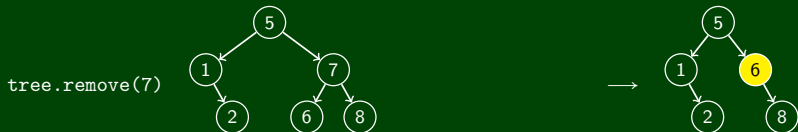
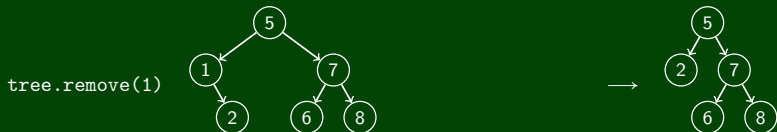
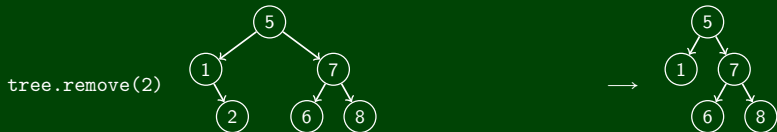
→

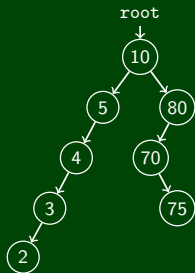
`tree.remove(5)`



→







### remove( $x$ )

- Case 1:  $x$  is a leaf
  - Just remove  $x$
- Case 2:  $x$  has one child
  - Replace  $x$  with its child
- Case 3:  $x$  has two children
  - Replace  $x$  with the **successor** or **predecessor** of  $x$

The tricky case is when  $x$  has two children. If we think of the BST in sorted array form, to get the successor, we `findMin(right subtree)` (or predecessor is `findMax(left subtree)`)