

Step1: see the children condition (replace deleteNode with x)

- ▶ 1. 2 NIL children, replace deleteNode with NIL
- ▶ 2. 1 NIL child and 1 regular child, replace deleteNode with regular child
- ▶ 3. 2 regular children, replace deleteNode with the minimum node of its right branch

Step2: see the color condition

- ▶ 1. deleteNode == red && replacement == red / NIL
-> DONE
- ▶ 2. deleteNode == red && replacement == black
-> color the replacement red -> go to rearrange
- ▶ 3. deleteNode == black && replacement == red
-> color the replacement black -> DONE
- ▶ 4. deleteNode == black && replacement == black / NIL
-> go to rearrange

Step3: rearrange the nodes (if needed) (x is the replacement node)

- ▶ Case0: `x == red`
- ▶ Case1: `x == black && sibling == red`
- ▶ Case2: `x == black && sibling == black && sibling.left == black && sibling.right == black`
- ▶ Case3: `x == black && sibling == black &&`
 - `(x == x.p.left) sibling.left == red && sibling.right == black`
 - `(x == x.p.right) sibling.right == red && sibling.left == black`
- ▶ Case4: `x == black && sibling == black &&`
 - `(x == x.p.left) sibling.right == red`
 - `(x == x.p.right) sibling.left == red`

Case0

► color x black

Case1

- ▶ 1. color sibling black
- ▶ 2. color x.p red
- ▶ 3. rotate x.p:
 - if $x == x.p.left$, do left-rotation
 - if $x == x.p.right$, do right-rotation
- ▶ 4. change sibling:
 - if $x == x.p.left$, set sibling = x.p.right
 - if $x == x.p.right$, set sibling = x.p.left
- ▶ 5. with new x and sibling, go to Case2 / 3 / 4

Case2

- ▶ 1. color sibling red
- ▶ 2. set $x = x.p$
- ▶ 3. if new $x == \text{red}$, color x black
if new $x == \text{black}$, **get new sibling** and go to Case1 / 2 / 3 / 4

Case3

- ▶ 1. color sibling's child black:
 - if $x == x.p.left$, color sibling.left black
 - if $x == x.p.right$, color sibling.right black
- ▶ 2. color sibling red
- ▶ 3. rotate sibling:
 - if $x == x.p.left$, do right-rotation
 - if $x == x.p.right$, do left-rotation
- ▶ 4. change sibling:
 - if $x == x.p.left$, set sibling = $x.p.right$
 - if $x == x.p.right$, set sibling = $x.p.left$
- ▶ 5. go to Case4

Case4

- ▶ 1. color sibling the same color as x.p
- ▶ 2. color x.p black
- ▶ 3. color sibling's child black
 - if $x == x.p.left$, color sibling.right black
 - if $x == x.p.right$, color sibling.left black
- ▶ 4. rotate x.p
 - if $x == x.p.left$, do left-rotation
 - if $x == x.p.right$, do right-rotation