

**Habib University**  
shaping futures

# CS 201 Data Structure II (L2 / L5)

## Delete in Red-Black Tree

**Muhammad Qasim Pasta**

[qasim.pasta@sse.habib.edu.pk](mailto:qasim.pasta@sse.habib.edu.pk)

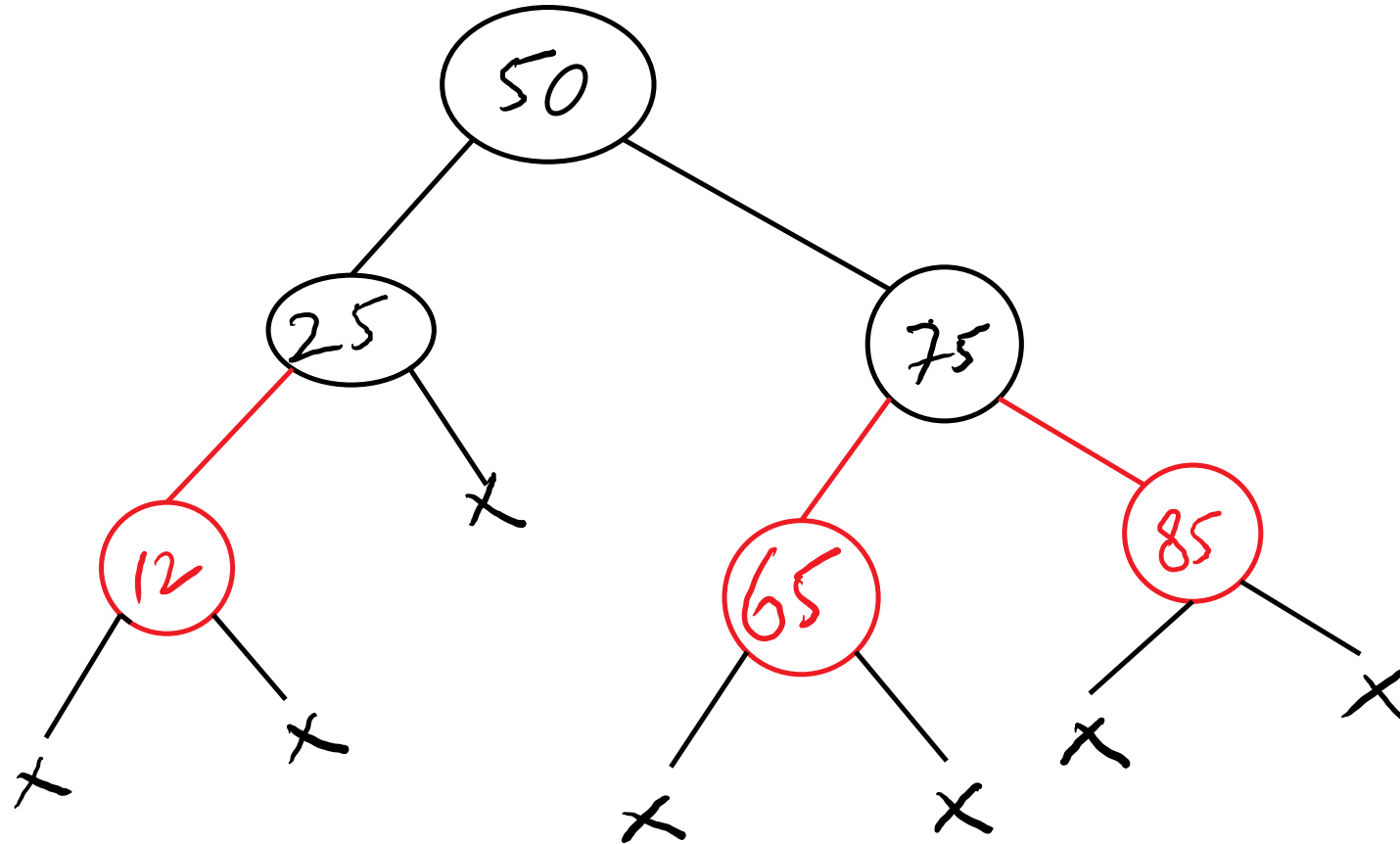


# Steps to delete a node

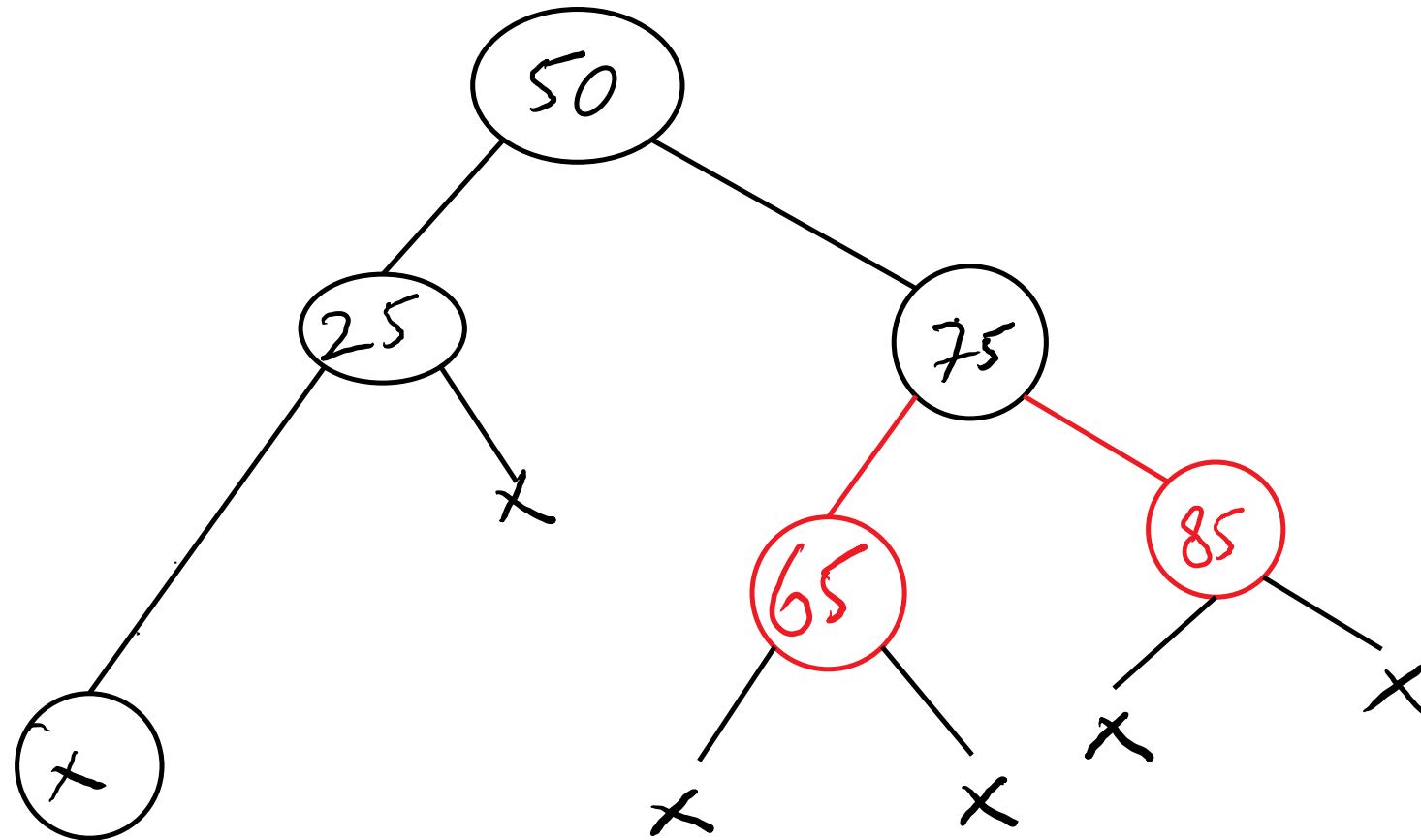
1. Perform BST Delete
  - BST Delete has three cases: 1. Leaf, 2. One Child, and 3. Two children.
  - But actual deletion occurs in cases of 1 and 2
  - Case 3 leads to either Case 1 or 2
2. Let x is the node to be deleted and y is the child of x (if x is a leaf then its child is NIL)
  - Replace x with y and delete y
3. If x or y is RED then (one must be black)
  - Change the color of x to black – we are done (no change in black height)
4. Otherwise: apply cases (life is not easy!)

# Leaf (Simple Case)

- Delete 12

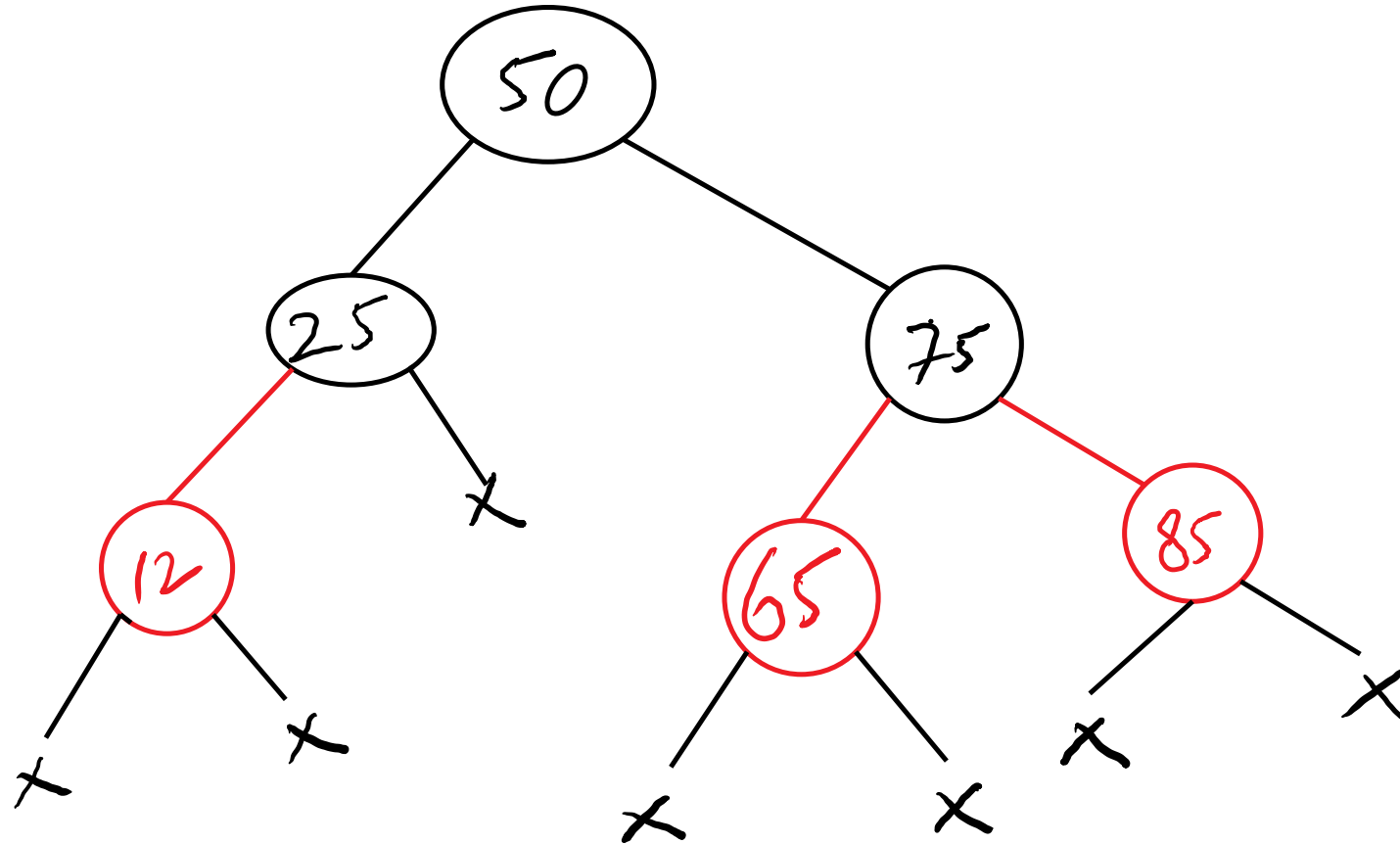


# Leaf (Simple Case)



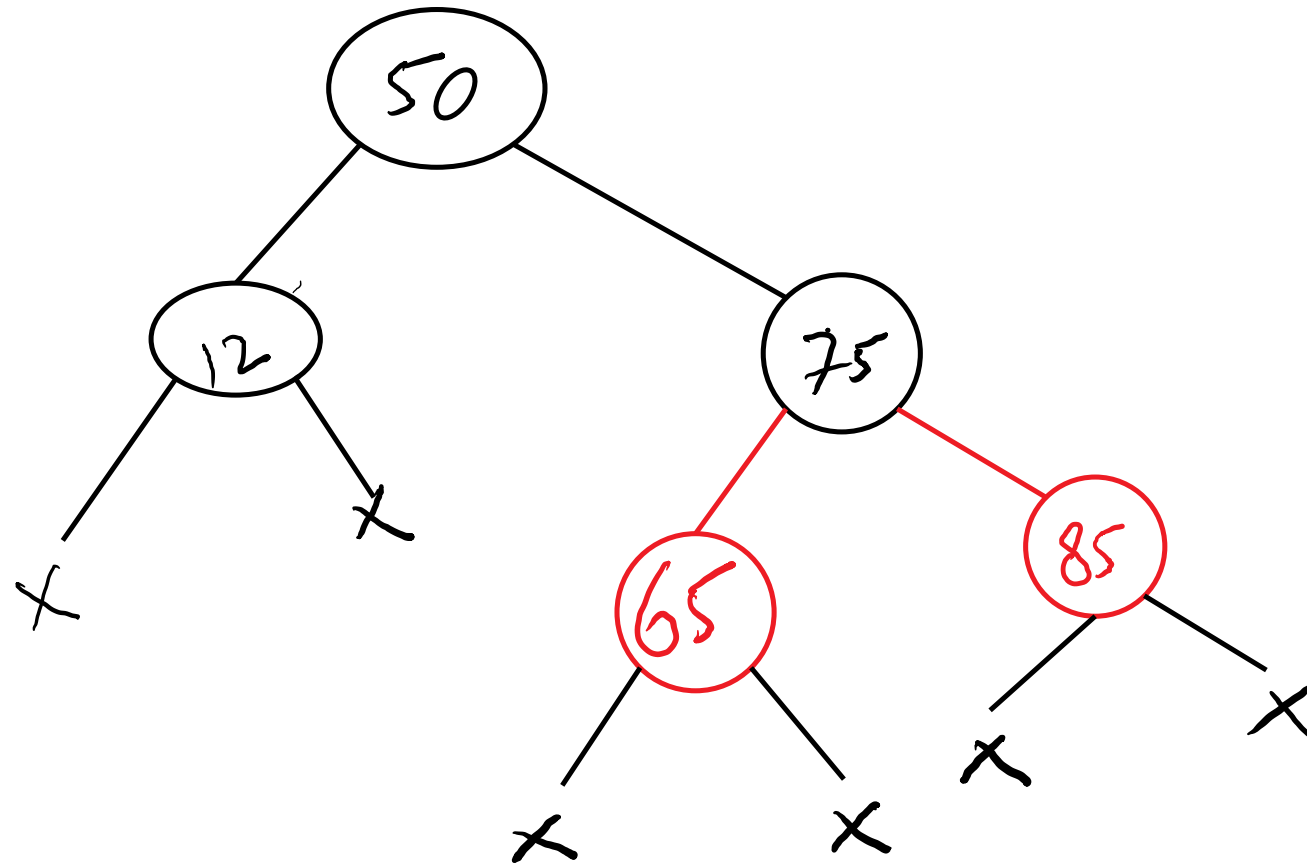
# One Child (Simple Case)

- Delete 25



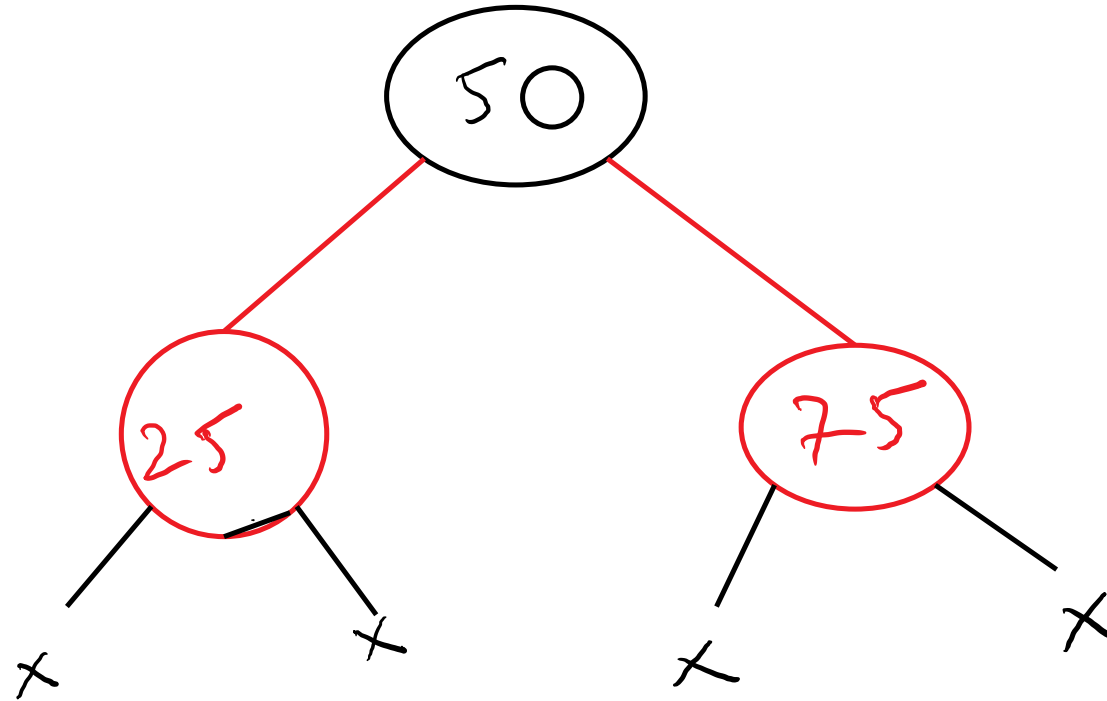
# One Child (Simple Case)

- Delete 25



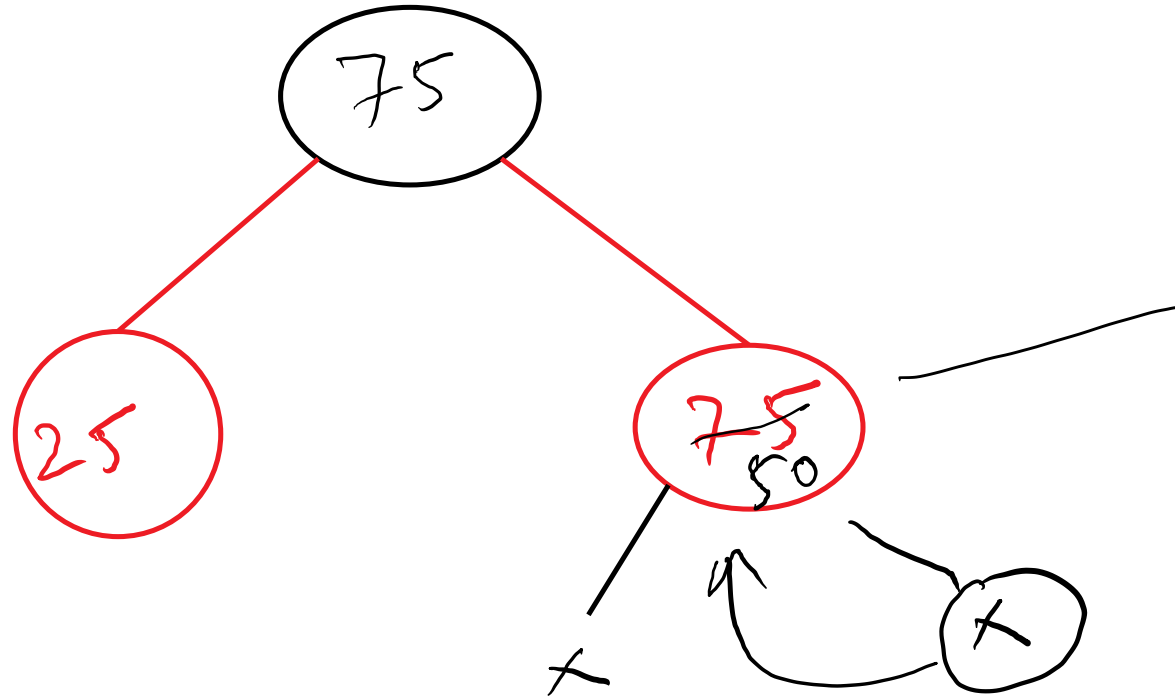
# Two Children (Simple Case)

- Delete 50



# Two Children (Simple Case)

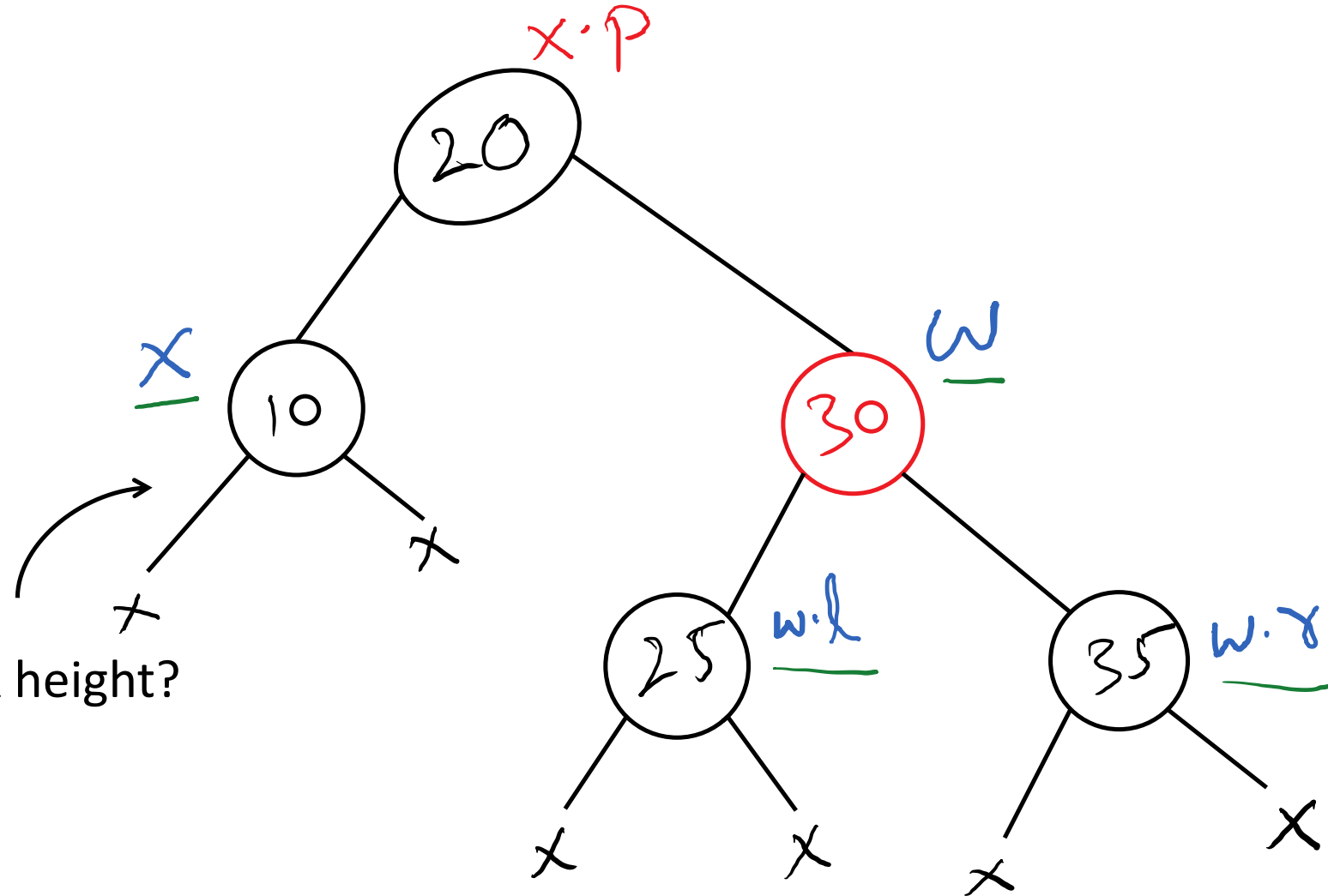
- Delete 50





## Other cases:

- Delete 10???



- Impact on black height?

# Cases:

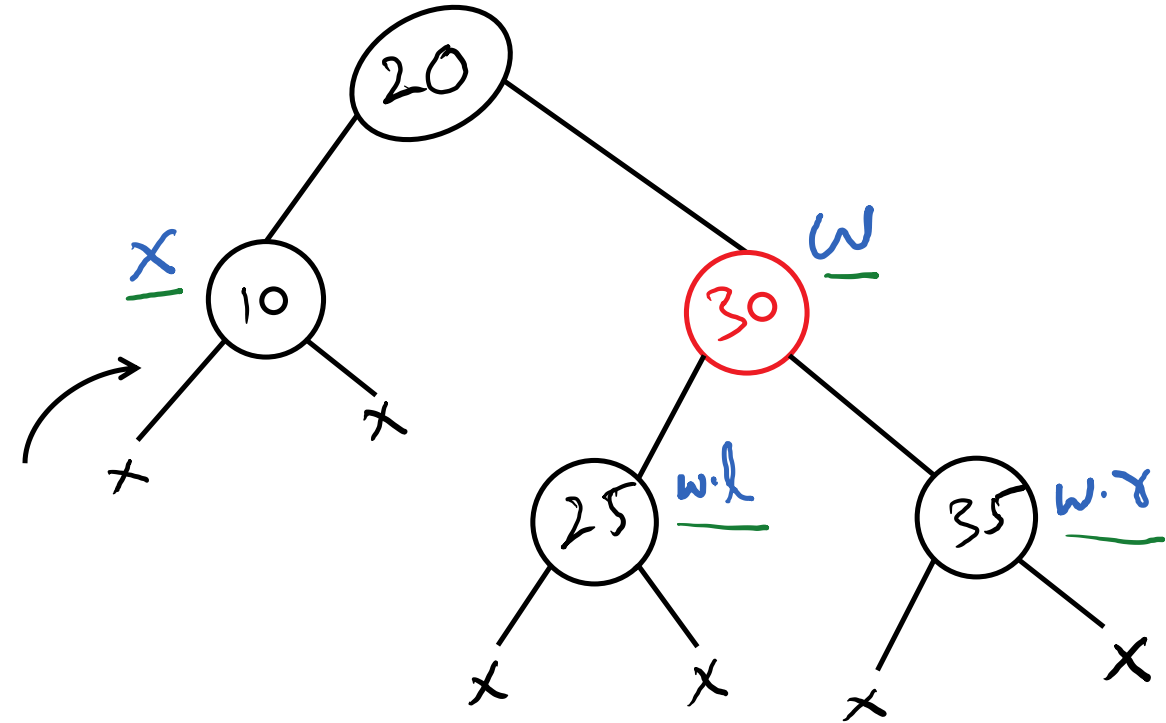


Case #	x	w	w.l	w.r
1	B	R		
2	B	B	B	B
3.1	B (x is left)	B	R	B
3.2	B( x is right)	B	B	R
4.1	B (x is left)	B		R
4.2	B (x is right)	B	R	

## Case 1:

Case #	x	w	w.l	w.r
1	B	R		

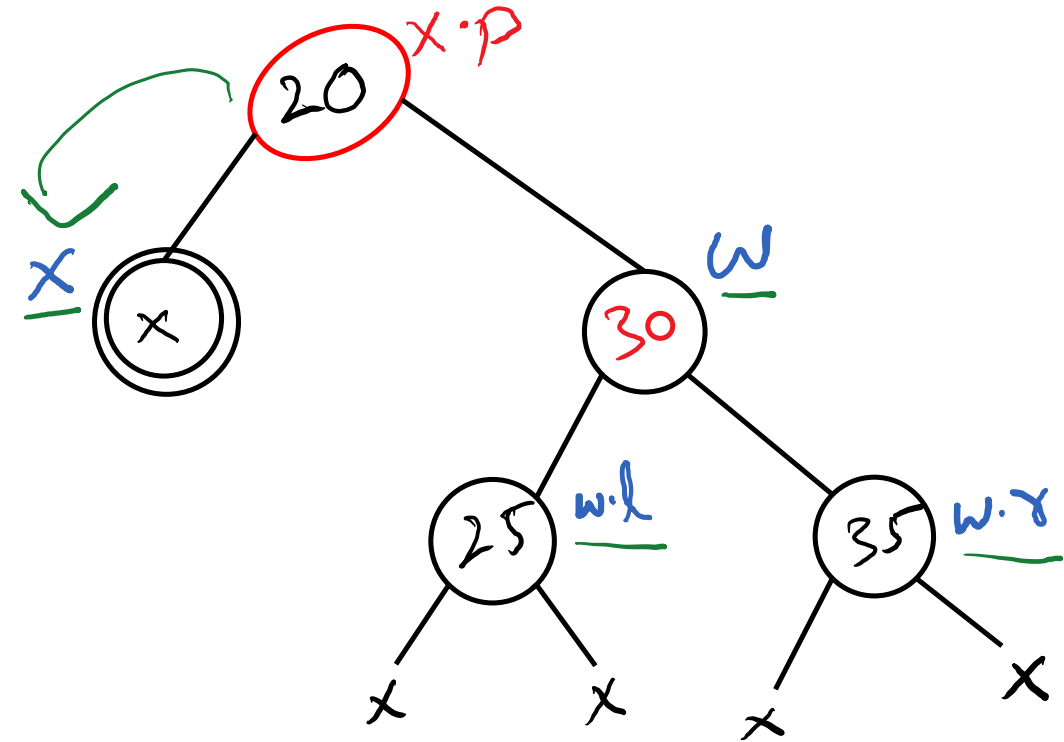
1.  $x.p = \text{RED}$
2.  $w = \text{BLACK}$
3. Rotate  $x.p$ 
  - Rotate Left (if  $x$  is a left child)
  - Rotate Right (if  $x$  is a right child)
4. Update pointer  $w$ :
  - $w = x.p.\text{right}$  (if  $x$  is a left child)
  - $w = x.p.\text{left}$  (if  $x$  is a right child)
5. Move to Case 2,3, OR 4



## Case 1:

Case #	x	w	w.l	w.r
1	B	R		

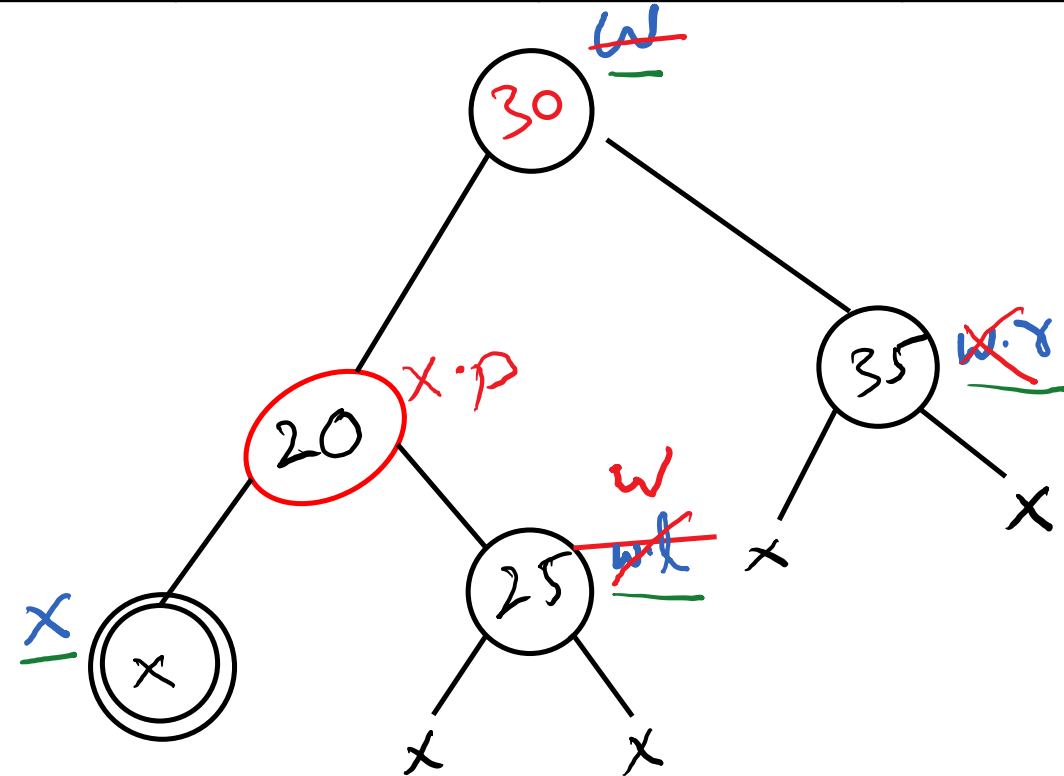
1.  $x.p = \text{RED}$
2.  $w = \text{BLACK}$
3. Rotate  $x.p$ 
  - Rotate Left (if  $x$  is a left child)
  - Rotate Right (if  $x$  is a right child)
4. Update pointer  $w$ :
  - $w = x.p.\text{right}$  (if  $x$  is a left child)
  - $w = x.p.\text{left}$  (if  $x$  is a right child)
5. Move to Case 2,3, OR 4



## Case 1:

Case #	x	w	w.l	w.r
1	B	R		

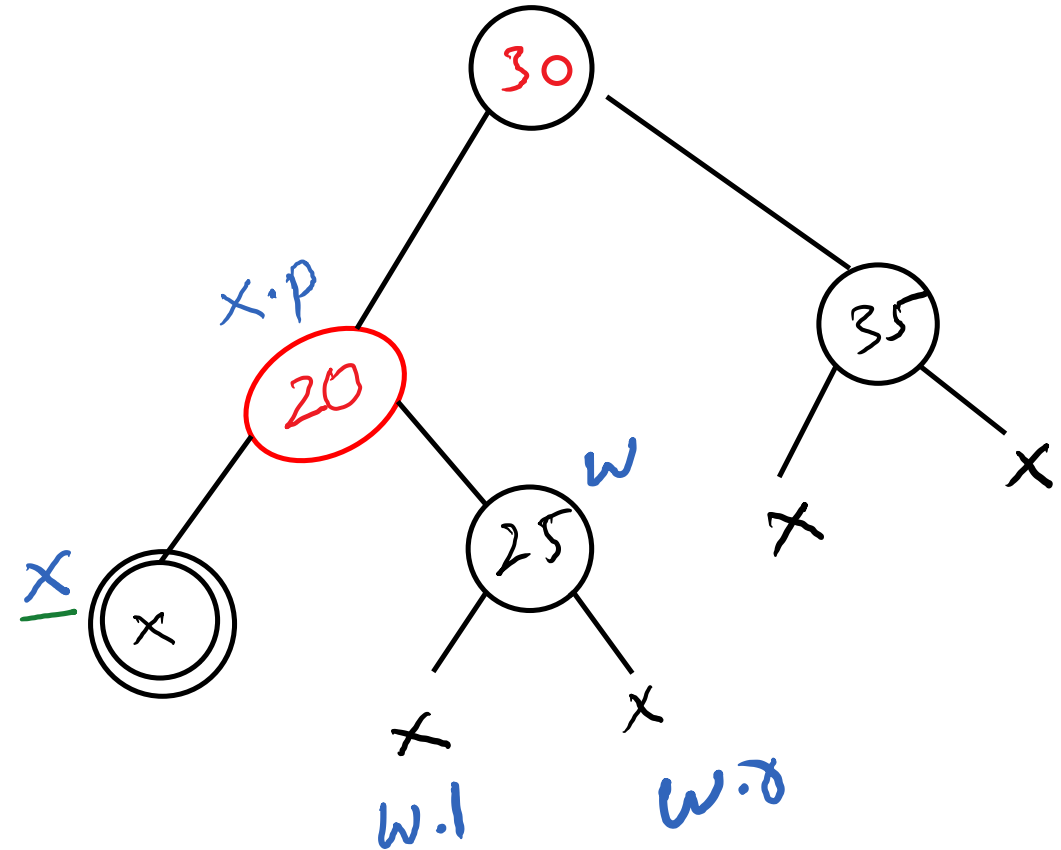
1.  $x.p = \text{RED}$
2.  $w = \text{BLACK}$
3. Rotate  $x.p$ 
  - Rotate Left (if  $x$  is a left child)
  - Rotate Right (if  $x$  is a right child)
4. Update pointer  $w$ :
  - $w = x.p.\text{right}$  (if  $x$  is a left child)
  - $w = x.p.\text{left}$  (if  $x$  is a right child)
5. Move to Case 2,3, OR 4



## Case 2:

Case #	x	w	w.l	w.r
2	B	B	B	B

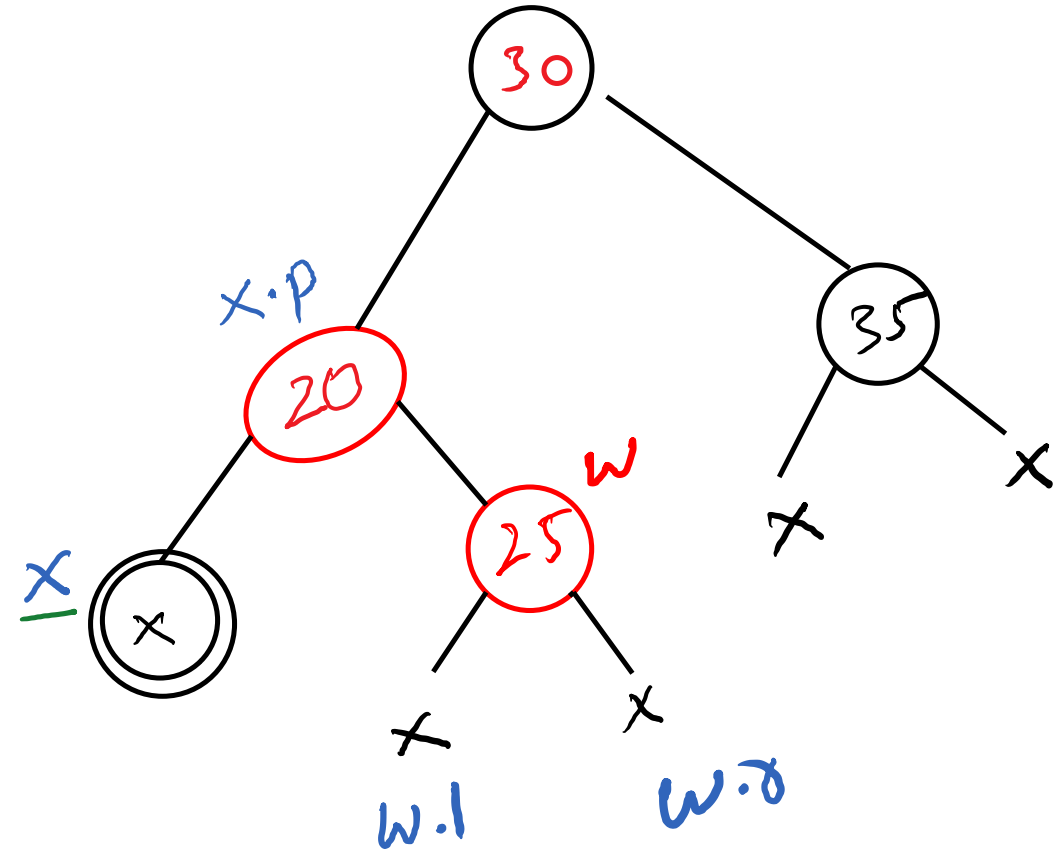
1.  $w = \text{RED}$
2.  $x = x.p$
3.  $x = \text{BLACK}$  (if  $x$  is RED)
  - If  $x$  is already BLACK then check cases



## Case 2:

Case #	x	w	w.l	w.r
2	B	B	B	B

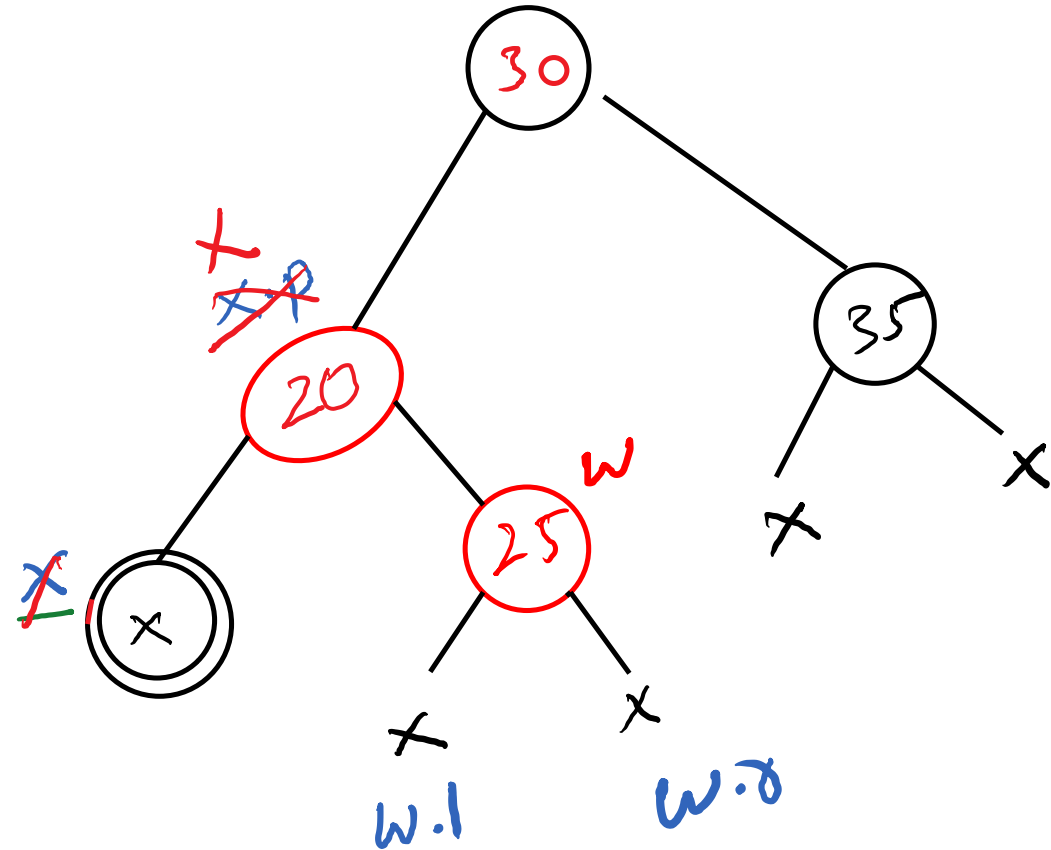
1.  $w = \text{RED}$
2.  $x = x.p$
3.  $x = \text{BLACK}$  (if  $x$  is RED)
  - If  $x$  is already BLACK then check cases



## Case 2:

Case #	x	w	w.l	w.r
2	B	B	B	B

1.  $w = \text{RED}$
2.  $x = x.p$
3.  $x = \text{BLACK}$  (if  $x$  is RED)
  - If  $x$  is already BLACK then check cases

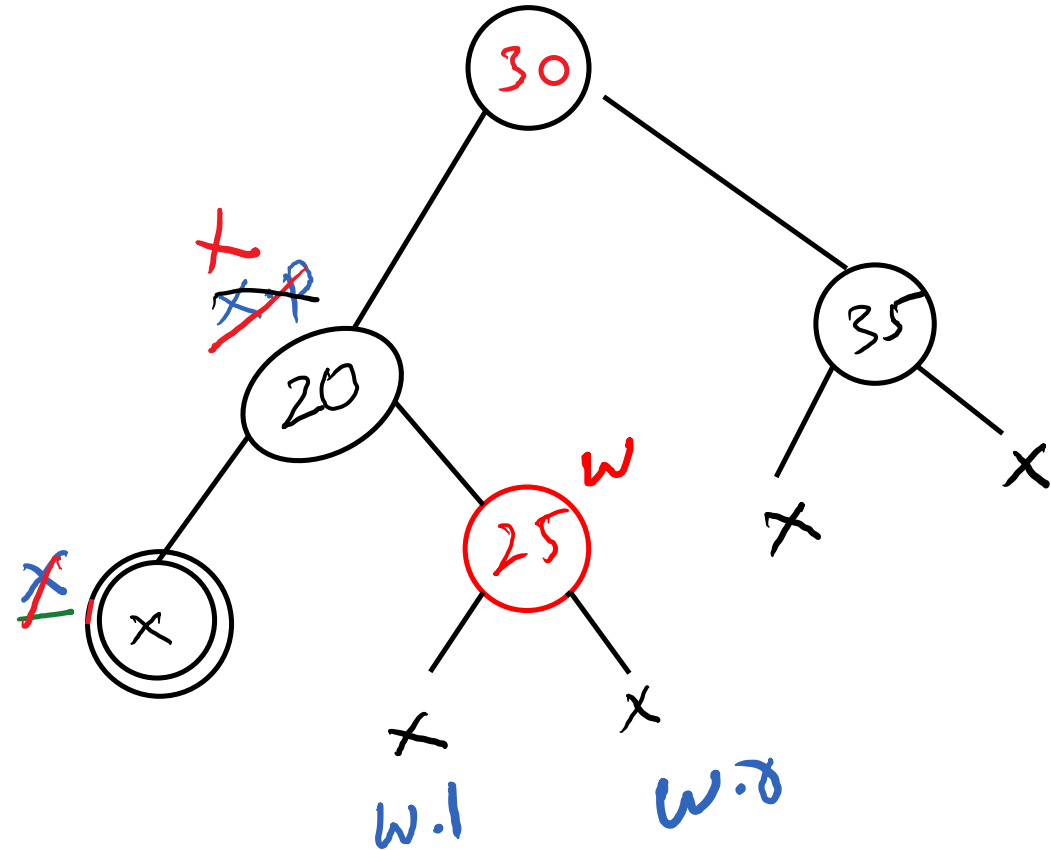




## Case 2:

Case #	x	w	w.l	w.r
2	B	B	B	B

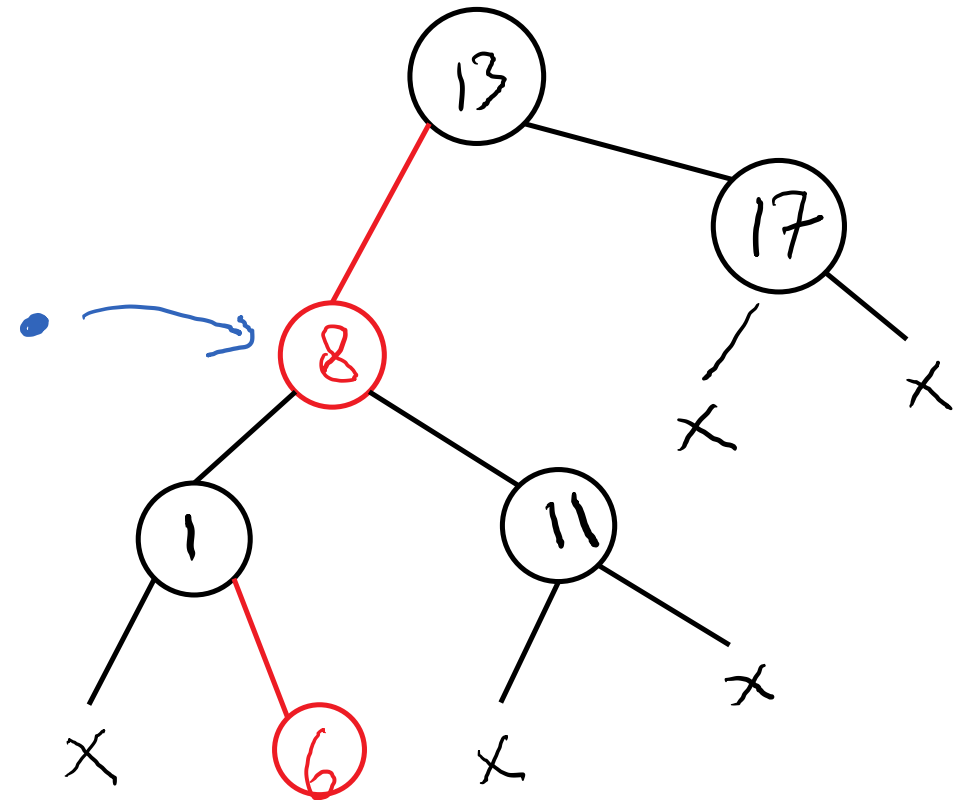
1.  $w = \text{RED}$
2.  $x = x.p$
3.  $x = \text{BLACK}$  (if  $x$  is RED)
  - If  $x$  is already BLACK then check cases



## Case 3:

Case #	x	w	w.l	w.r
3.1	B (x is a left child)	B	R	B
3.2	B (x is right)	B	B	R

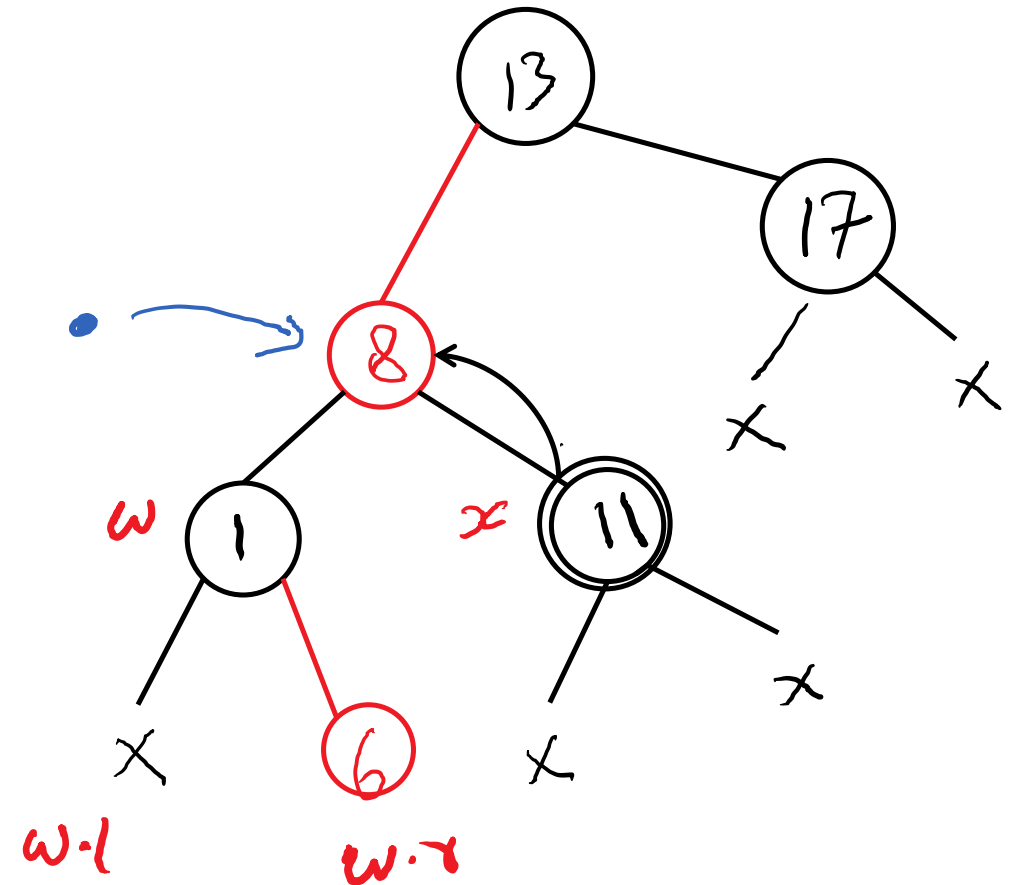
1.  $w.l = \text{BLACK}$  ( $w.r = \text{BLACK}$ )
2.  $w = \text{RED}$
3. Rotate Right at W (Rotate Left at w)
4. Update pointer w  
–  $w = x.p.\text{right}$  ( $w = x.p.\text{left}$ )
5. Proceed to Case 4



## Case 3:

Case #	x	w	w.l	w.r
3.1	B (x is a left child)	B	R	B
3.2	B (x is right)	B	B	R

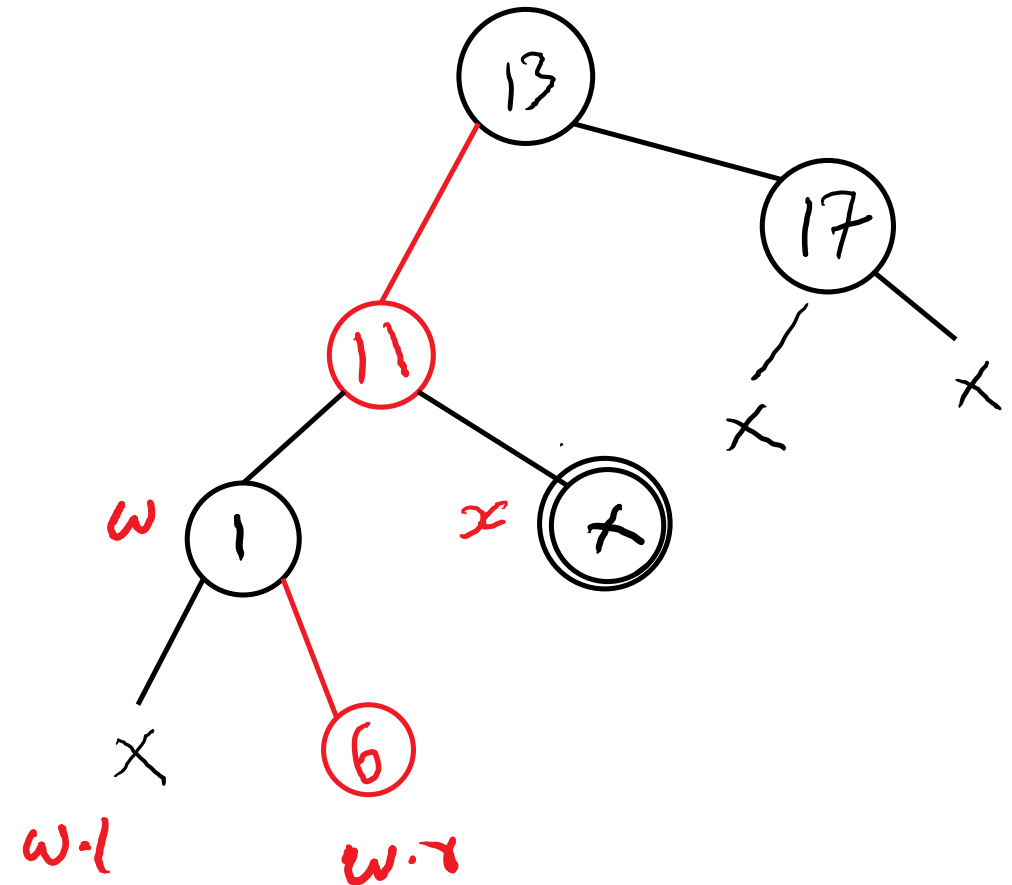
1.  $w.l = \text{BLACK}$  ( $w.r = \text{BLACK}$ )
2.  $w = \text{RED}$
3. Rotate Right at W (Rotate Left at w)
4. Update pointer w  
–  $w = x.p.\text{right}$  ( $w = x.p.\text{left}$ )
5. Proceed to Case 4



## Case 3:

Case #	x	w	w.l	w.r
3.1	B (x is a left child)	B	R	B
3.2	B(x is a right child)	B	B	R

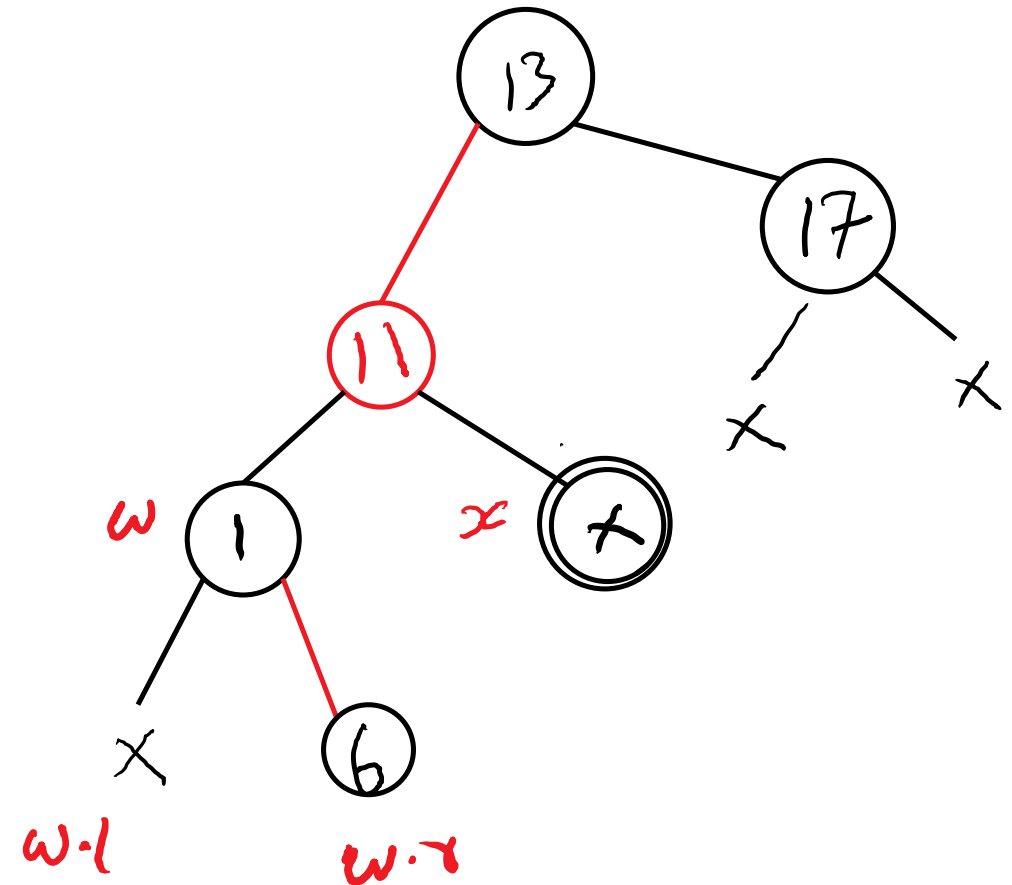
1.  $w.l = \text{BLACK}$  ( $w.r = \text{BLACK}$ )
2.  $w = \text{RED}$
3. Rotate Right at W (Rotate Left at w)
4. Update pointer w  
–  $w = x.p.\text{right}$  ( $w = x.p.\text{left}$ )
5. Proceed to Case 4



## Case 3:

Case #	x	w	w.l	w.r
3.1	B (x is a left child)	B	R	B
3.2	B(x is a right child)	B	B	R

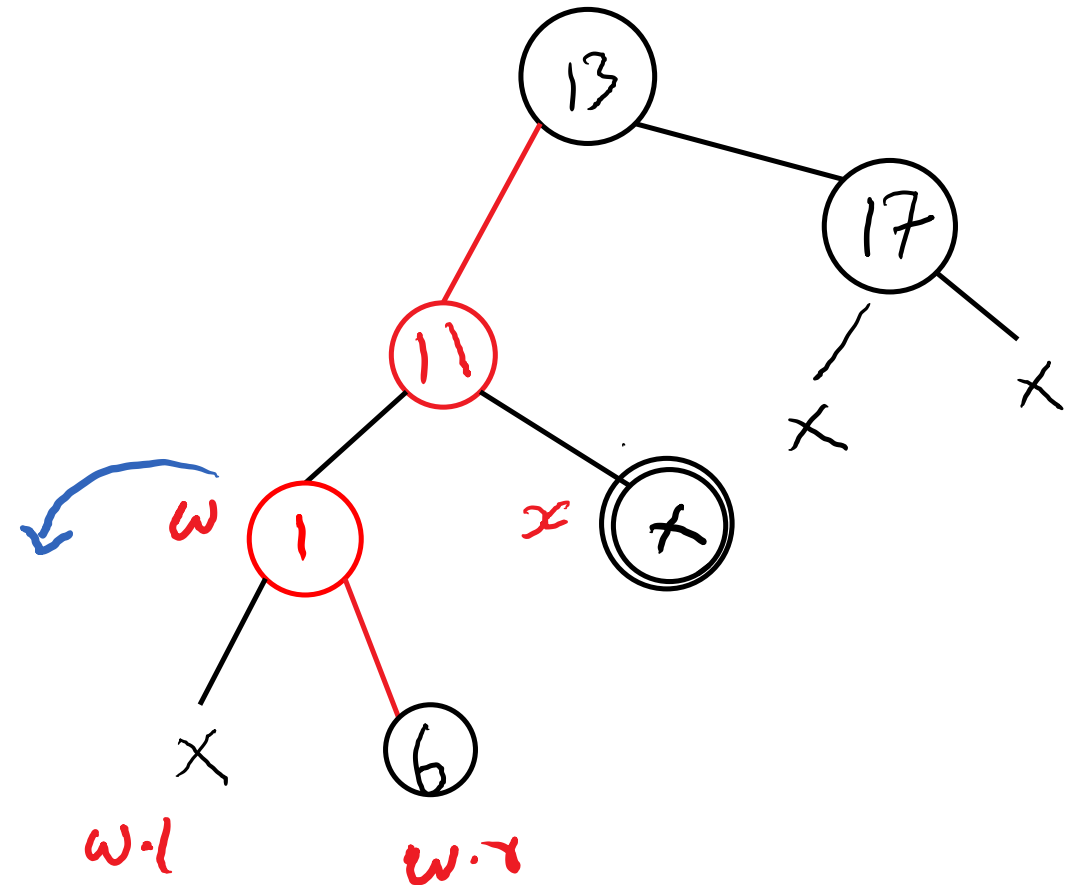
1.  $w.l = \text{BLACK}$  ( $w.r = \text{BLACK}$ )
2.  $w = \text{RED}$
3. Rotate Right at W (Rotate Left at w)
4. Update pointer w  
–  $w = x.p.\text{right}$  ( $w = x.p.\text{left}$ )
5. Proceed to Case 4



## Case 3:

Case #	x	w	w.l	w.r
3.1	B (x is a left child)	B	R	B
3.2	B(x is a right child)	B	B	R

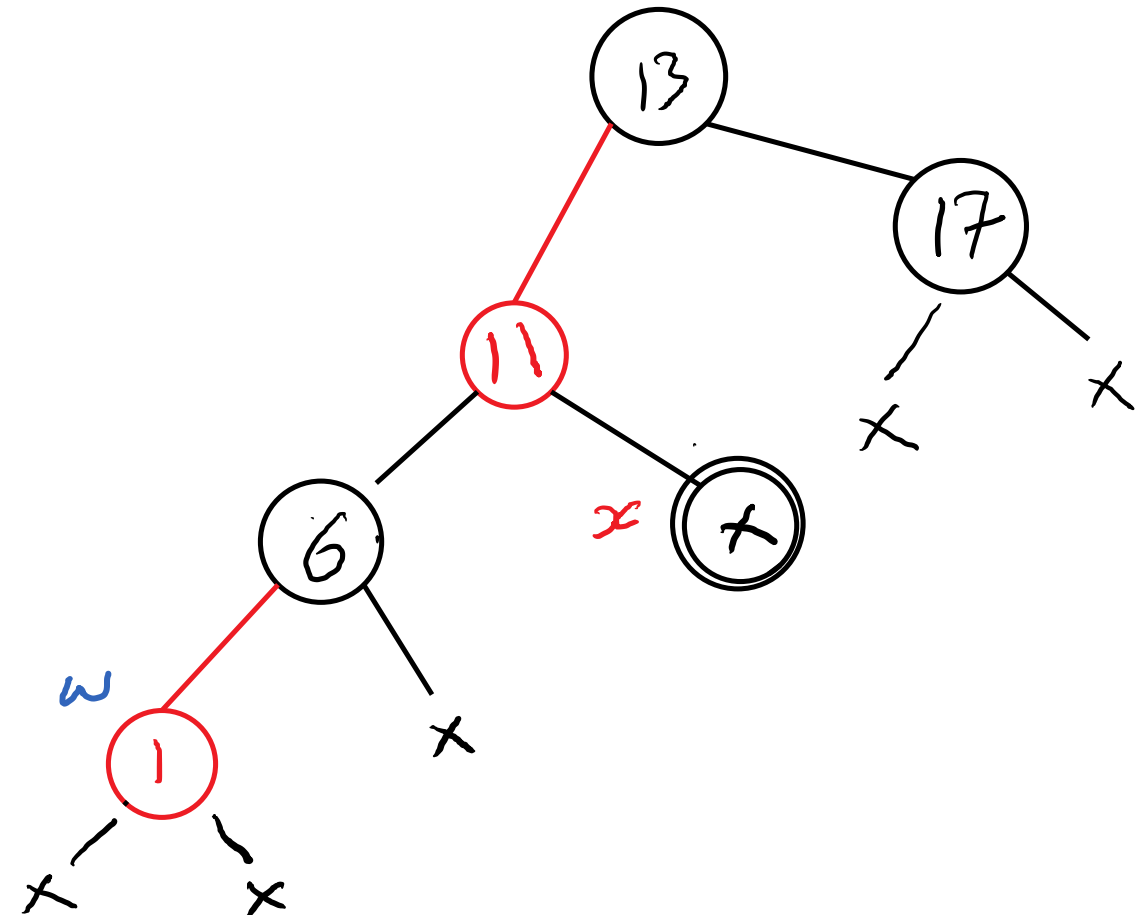
1.  $w.l = \text{BLACK}$  ( $w.r = \text{BLACK}$ )
2.  $w = \text{RED}$
3. Rotate Right at W (Rotate Left at w)
4. Update pointer w  
–  $w = x.p.\text{right}$  ( $w = x.p.\text{left}$ )
5. Proceed to Case 4



## Case 3:

Case #	x	w	w.l	w.r
3.1	B (x is a left child)	B	R	B
3.2	B(x is a right child)	B	B	R

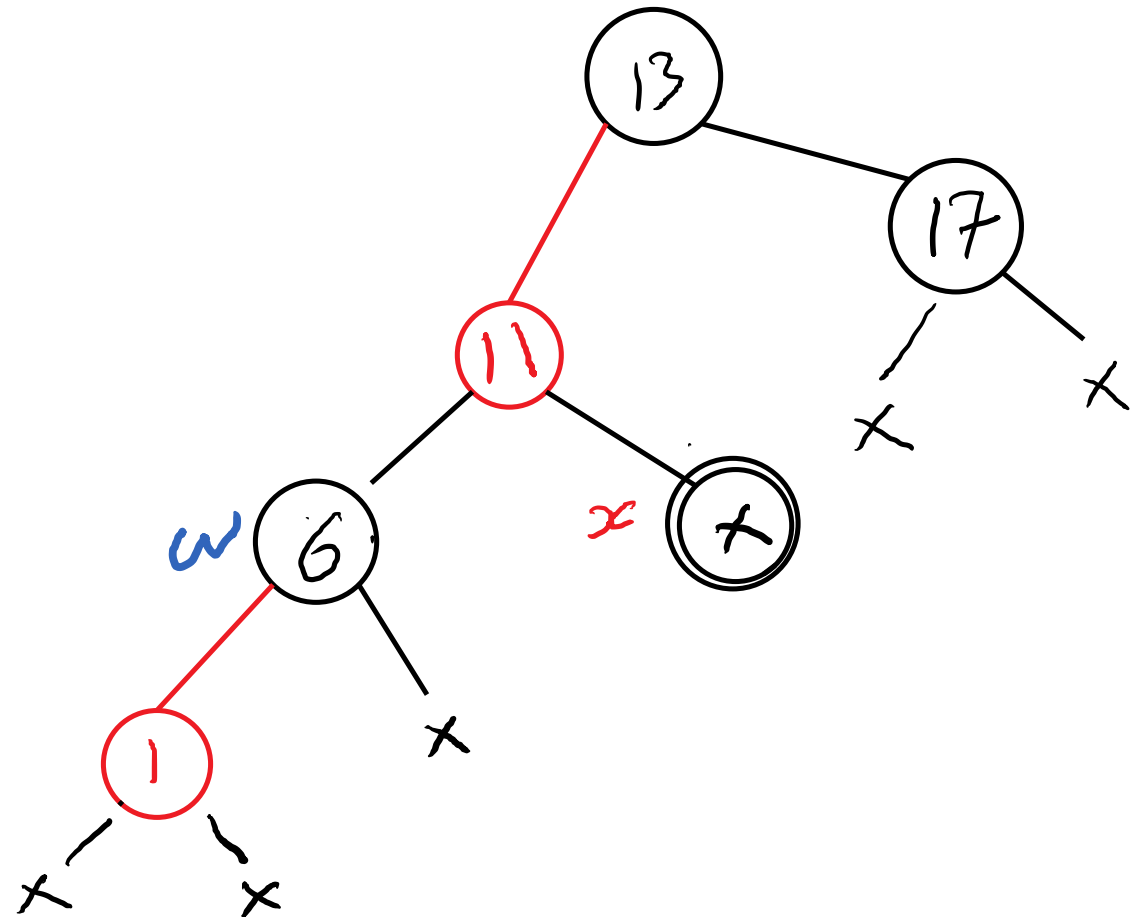
1.  $w.l = \text{BLACK}$  ( $w.r = \text{BLACK}$ )
2.  $w = \text{RED}$
3. Rotate Right at W (Rotate Left at w)
4. Update pointer w  
–  $w = x.p.\text{right}$  ( $w = x.p.\text{left}$ )
5. Proceed to Case 4



## Case 3:

Case #	x	w	w.l	w.r
3.1	B (x is a left child)	B	R	B
3.2	B(x is a right child)	B	B	R

1.  $w.l = \text{BLACK}$  ( $w.r = \text{BLACK}$ )
2.  $w = \text{RED}$
3. Rotate Right at W (Rotate Left at w)
4. Update pointer w  
–  $w = x.p.\text{right}$  ( $w = x.p.\text{left}$ )
5. Proceed to Case 4

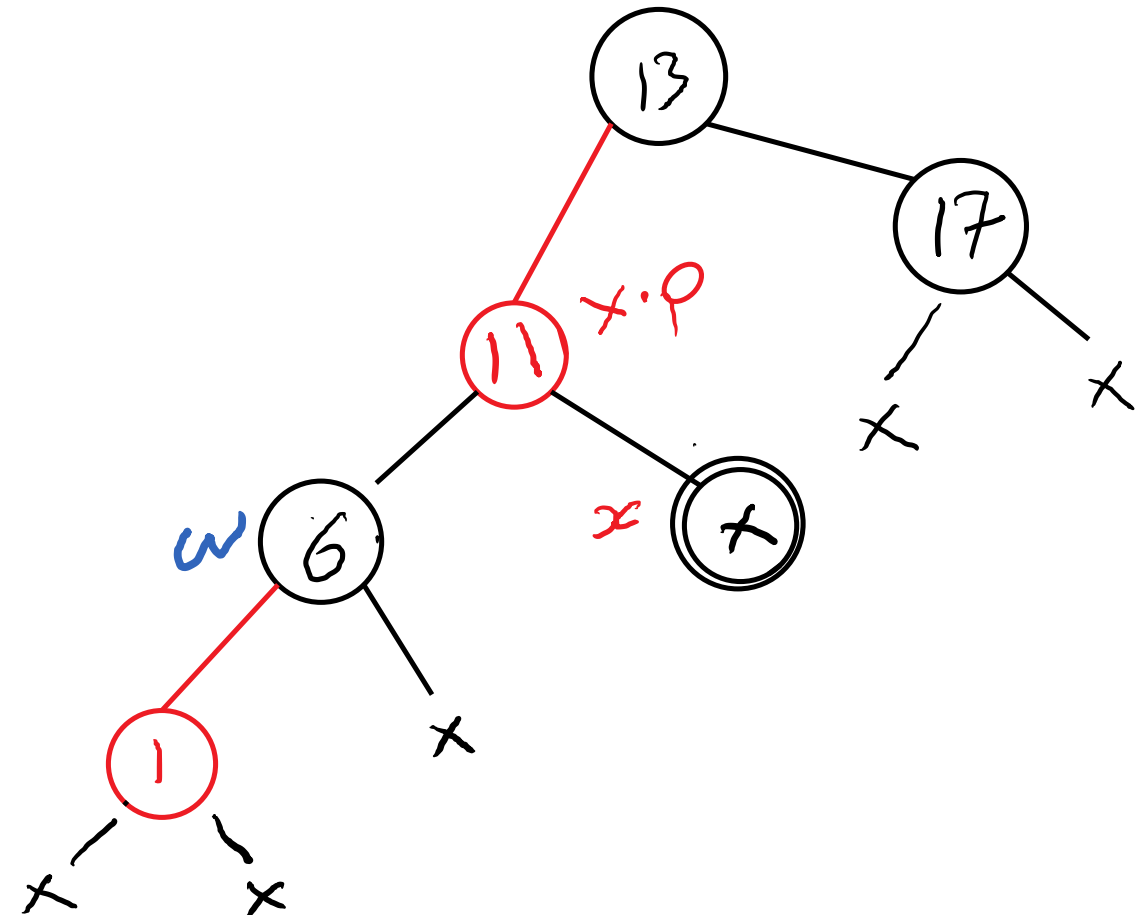




## Case 4:

Case #	x	w	w.l	w.r
4.1	B (x is left)	B		R
4.2	B (x is right)	B	R	

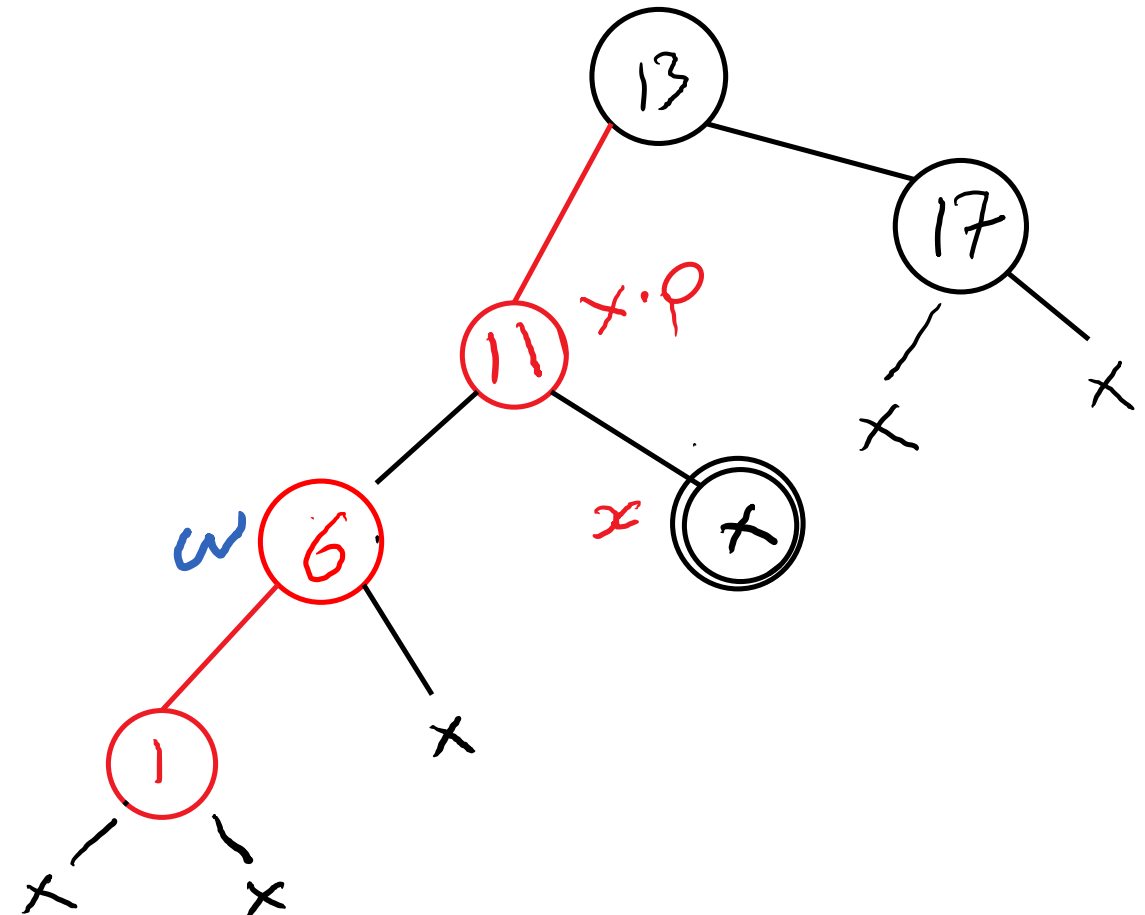
1.  $\text{color}(w) = \text{color}(x.p)$
2.  $x.p = \text{BLACK}$
3.  $w.\text{right} = \text{BLACK}$  ( $w.\text{left} = \text{BLACK}$ )
4. Rotate Left at  $x.p$  (Rotate Right at  $x.p$ )



## Case 4:

Case #	x	w	w.l	w.r
4.1	B (x is left)	B		R
4.2	B (x is right)	B	R	

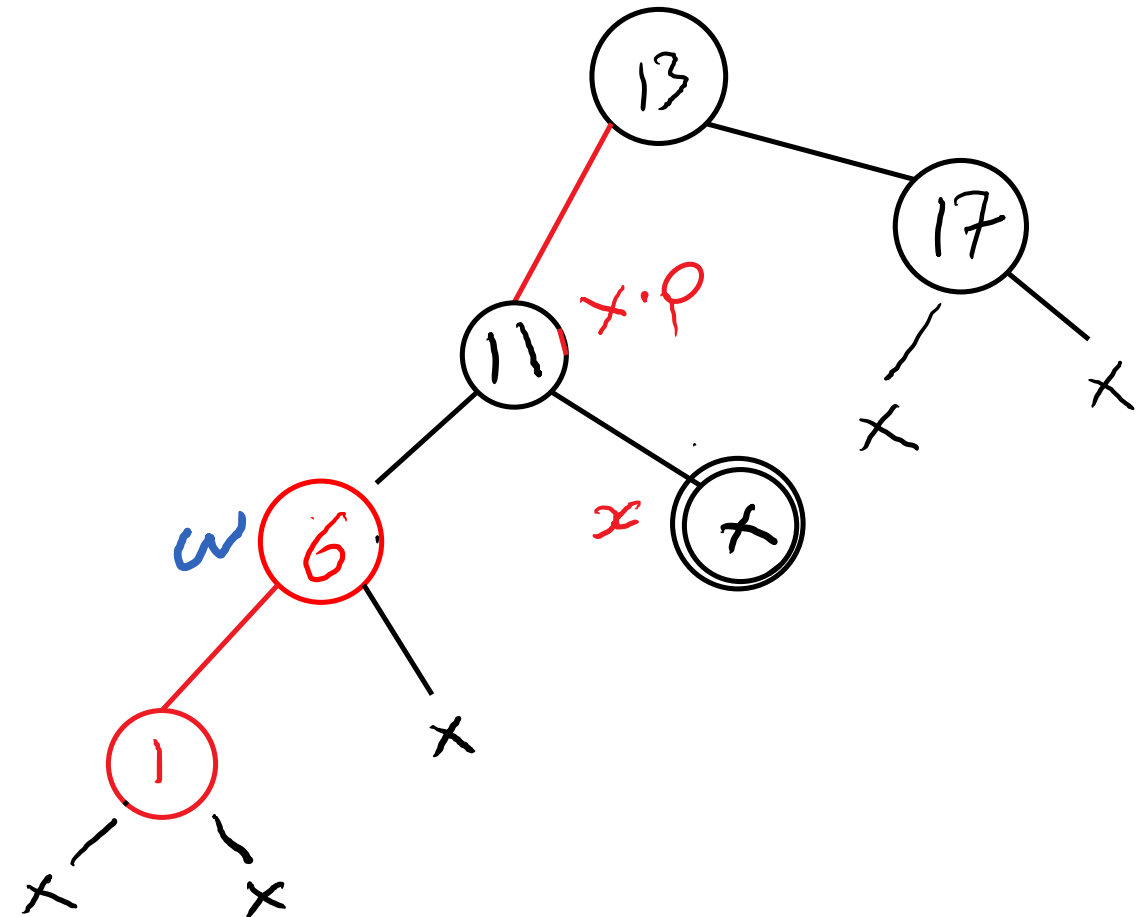
1.  $\text{color}(w) = \text{color}(x.p)$
2.  $x.p = \text{BLACK}$
3.  $w.\text{right} = \text{BLACK}$  ( $w.\text{left} = \text{BLACK}$ )
4. Rotate Left at  $x.p$  (Rotate Right at  $x.p$ )



## Case 4:

Case #	x	w	w.l	w.r
4.1	B (x is left)	B		R
4.2	B (x is right)	B	R	

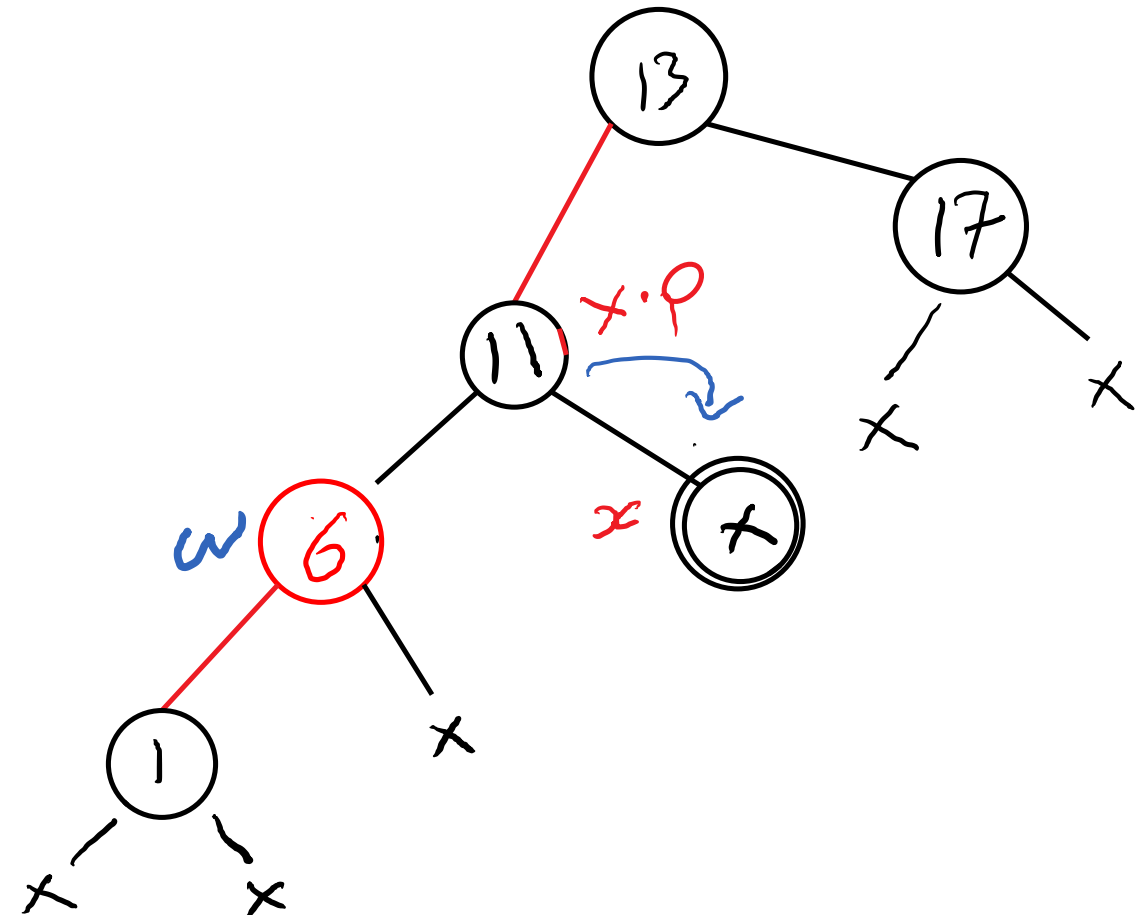
1.  $\text{color}(w) = \text{color}(x.p)$
2.  $x.p = \text{BLACK}$
3.  $w.\text{right} = \text{BLACK}$  ( $w.\text{left} = \text{BLACK}$ )
4. Rotate Left at  $x.p$  (Rotate Right at  $x.p$ )



## Case 4:

Case #	x	w	w.l	w.r
4.1	B (x is left)	B		R
4.2	B (x is right)	B	R	

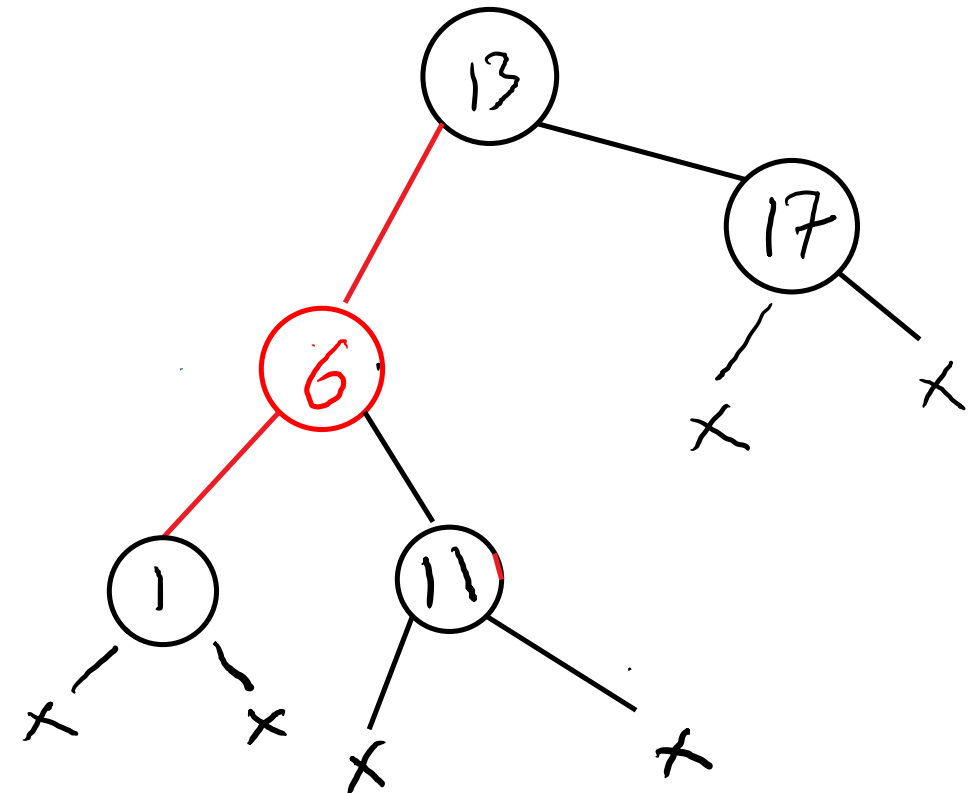
1.  $\text{color}(w) = \text{color}(x.p)$
2.  $x.p = \text{BLACK}$
3.  $w.\text{right} = \text{BLACK}$  ( $w.\text{left} = \text{BLACK}$ )
4. Rotate Left at  $x.p$  (Rotate Right at  $x.p$ )



## Case 4:

Case #	x	w	w.l	w.r
4.1	B (x is left)	B		R
4.2	B (x is right)	B	R	

1.  $\text{color}(w) = \text{color}(x.p)$
2.  $x.p = \text{BLACK}$
3.  $w.\text{right} = \text{BLACK}$  ( $w.\text{left} = \text{BLACK}$ )
4. Rotate Left at  $x.p$  (Rotate Right at  $x.p$ )



Example:

Delete 18

