RBT Deletion (Transplant)

- $RB-TRANSPLANT(T,u,v)$

```
if $u.p == T.nil$
    $T.root = v$
elseif $u == u.p.left$
    $u.p.left = v$
else
    $u.p.right = v$
$v.p = u.p$
```

BST transplant 과 다른점은
NIL을 T.nil로 한 것

BST deletion 이랑 똑같음

- $RB-DELETE(T,z)$

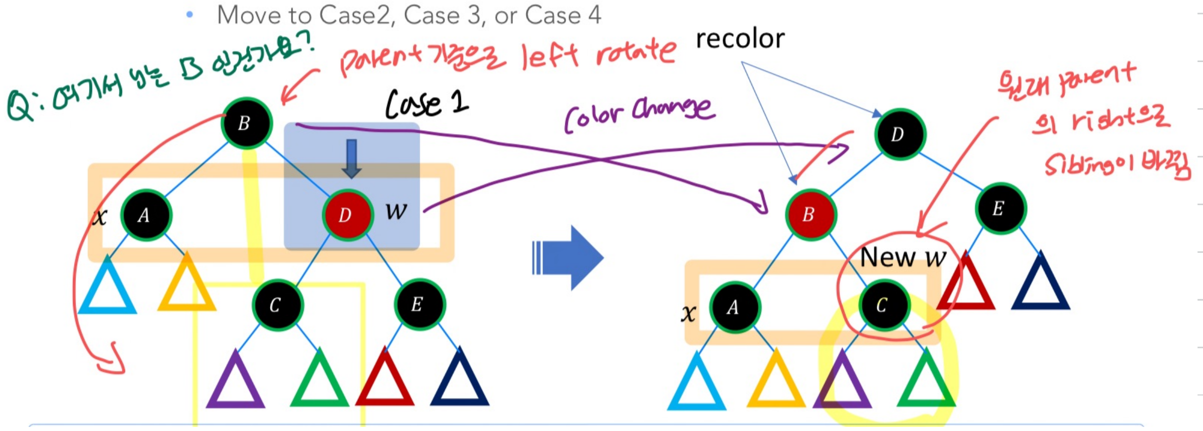
```
$y = z$
$y-original-color = y.color$
Case2
if $z.left == T.nil$
    $x = z.right$
    $RB-TRANSPLANT(T,z,z.right)$
    // replace $z$ by its right child
elseif $z.right == T.nil$
    $x = z.left$
    $RB-TRANSPLANT(T,z,z.left)$
    // replace $z$ by its left child
else   Case 3
    $y = TREE-MINIMUM(z.right)$
        // $y$ is $z$'s successor
    $y-original-color = y.color$
    $x = y.right$
    if $y \neq z.right$ // is y farther down the tree?
        $RB-TRANSPLANT(T,y,y.right)$
            // replace $y$ by its right child
        $y.right = z.right$
            // $z$'s right child becomes $y$'s right child
        $y.right.p = y$
    else
        $x.p = y$ // in case $x$ is $T.nil$
    $RB-TRANSPLANT(T,z,y)$
        // replace $z$ by its successor $y$
    $y.left = z.left$ // and give $z$'s left child to $y$,
    $y.left.p = y$    // which had no left child
    $y.color = z.color$
if $y-original-color == BLACK$
    // if any red-black violations occurred, correct them
    $RB-DELETE-FIXUP(T,x)$
```

delete 후 옮기고자 하는 y가 Black일때
RBT properties 따르게 수정 필요

fixup case ①

- $RB-DELETE-FIXUP(T,x)$
  - Case 1: the sibling w of x is RED
    - Left rotate around the $x.p$
    - Recolor B and D
    - The new sibling is BLACK ($x.p$ is RED)
    - Move to Case2, Case 3, or Case 4

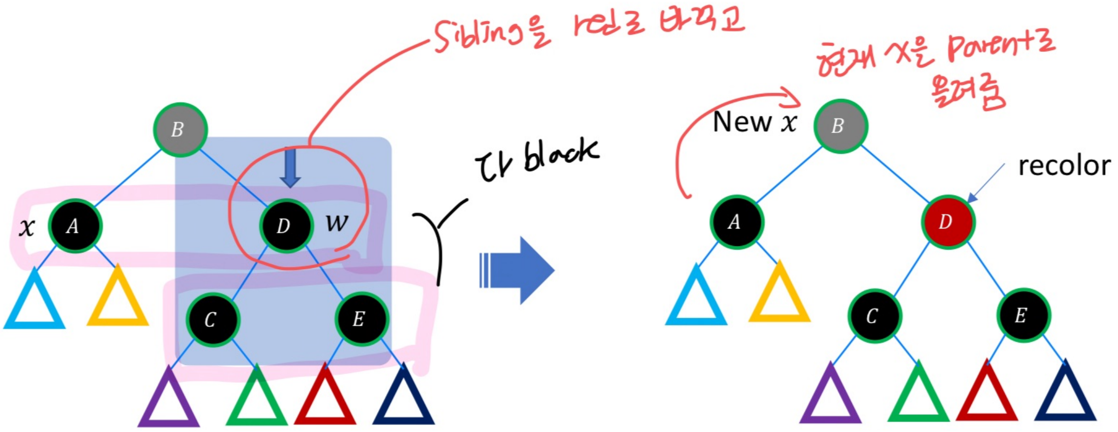
Sibling이 red일 때


Q: 여기서 b는 B 인건가요?
Parent 기준으로 left rotate
recolor
Case 1
Color change
원래 parent 의 right으로 sibling이 바뀜
B
x
A
D
w
C
E
D
B
E
New w
x
A
C

case ②

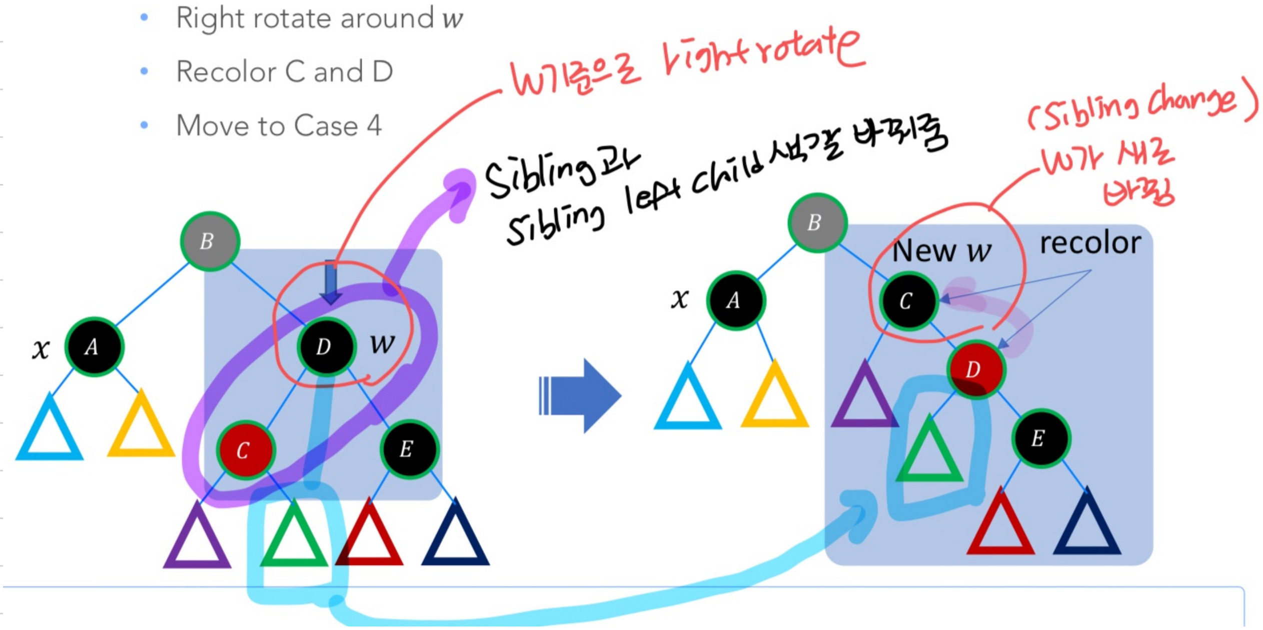
- $RB-DELETE-FIXUP(T,x)$
  - Case 2: the sibling w of x is BLACK with two BLACK children
    - Recolor w
    - Move x to point to B
    - Fix it again

Sibling이 Black이고
children도 둘다 black일때



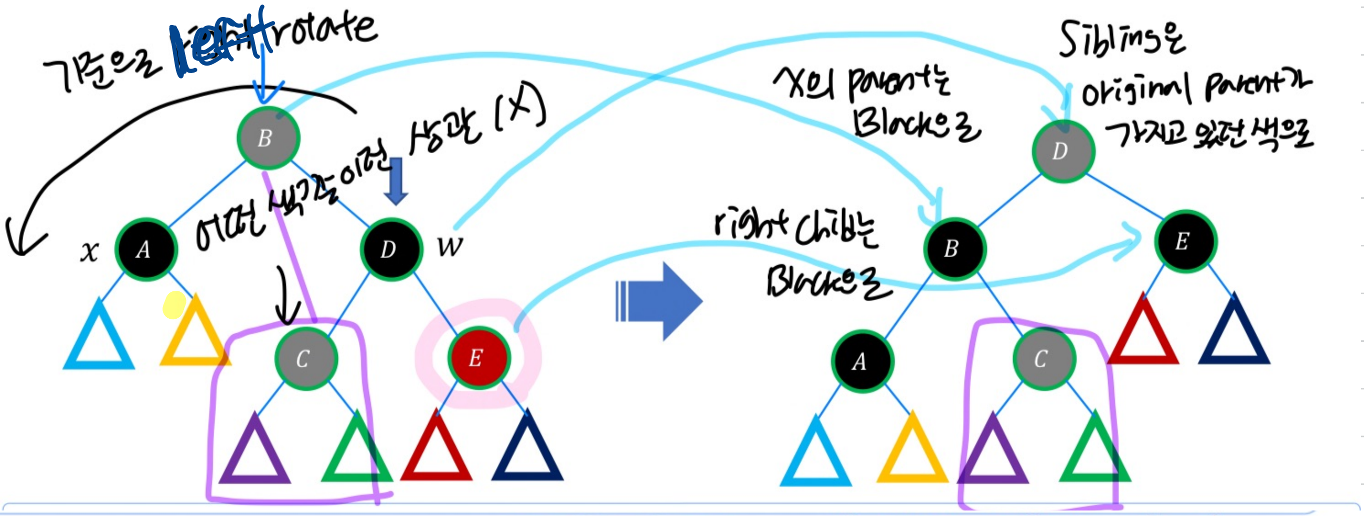
case ③

- Case 3: the sibling w of x is BLACK with w's RED left child and w's BLACK right child
  - Right rotate around w
  - Recolor C and D
  - Move to Case 4



Case ④

- $RB-DELETE-FIXUP(T,x)$
  - Case 4: the sibling w of x is BLACK, and w's right child is RED
    - Recolor B(BLACK), D(B color), and E(BLACK)
    - Left rotate around B
    - A valid RBT (terminate)



right 냐 left냐에 차이

```
• RB − DELETE − FIXUP(T, x)
1.      while x ≠ T.root and x.color == BLACK
2.        if x == x.p.left  // is x a left child?
3.          w = x.p.right // w is x's sibling
4.          if w.color == RED
5. case1      w.color = BLACK          ] parent와 sibling
6.            x.p.color = RED          ]   color change
              LEFT-ROTATE(T, x.p)
   원래 parent right →
              w = x.p.right            → new sibling 지정
9.          if w.left.color == BLACK and w.right.color == BLACK
10. case2     w.color = RED            → sibling red로 바꿈
11.           x = x.p                  → x을 parent로 올림
12.         else
13.           if w.right.color == BLACK    → right만 black일때
14.             w.left.color = BLACK       ] sibling과 left child
15. case3       w.color = RED              ]   color change
16.             RIGHT-ROTATE(T, w)         → sibling 기준 right rotate
17.             w = x.p.right              → sibling 바꿈
18.           w.color = x.p.color          }
19.           x.p.color = BLACK            } color change
20. case4     w.right.color = BLACK        }
21.           LEFT-ROTATE(T, x.p)          → left rotate
22.           x = T.root                   → x을 root로 지정
23.     else // same as lines 3–22, but with "right" and "left" exchanged
24.       w = x.p.left
25.       if w.color == RED
26.         w.color = BLACK
27.         x.p.color = RED
28.         RIGHT-ROTATE(T, x.p)
29.         w = x.p.left
30.       if w.right.color == BLACK and w.left.color == BLACK
31.         w.color = RED
32.         x = x.p
33.       else
34.         if w.left.color == BLACK
35.           w.right.color = BLACK
36.           w.color = RED
37.           LEFT-ROTATE(T, w)
38.           w = x.p.left
39.         w.color = x.p.color
40.         x.p.color = BLACK
41.         w.left.color = BLACK
42.         RIGHT-ROTATE(T, x.p)
43.         x = T.root
44.   x.color = BLACK
```