Red-Black Tree- Deletion

Prepared by

Dr Annushree Bablani

Red Black Tree- Deletion

- To perform operation, we first execute the deletion algorithm for binary search trees
- Thus, the node which is deleted is the parent of an external node.
- If node is red, it won't violate any property
- If node is a leaf, it won't violate any property
- Otherwise, if node is black and has a child, it will violate property 2, 3, and 4
- For property 2, set the color of root to black after deletion

Red Black Tree- Deletion

- To fix property 3 and 4:
  - From now on, lets call the deleted node to be z
  - If z's child x (which is the replacing node) is red, set x to black. Done!
  - If x is black, add another black to x, so that x will be a doubly black node, and property 3 and 4 are fixed. But property 1 is violated

Red Black Tree- Deletion

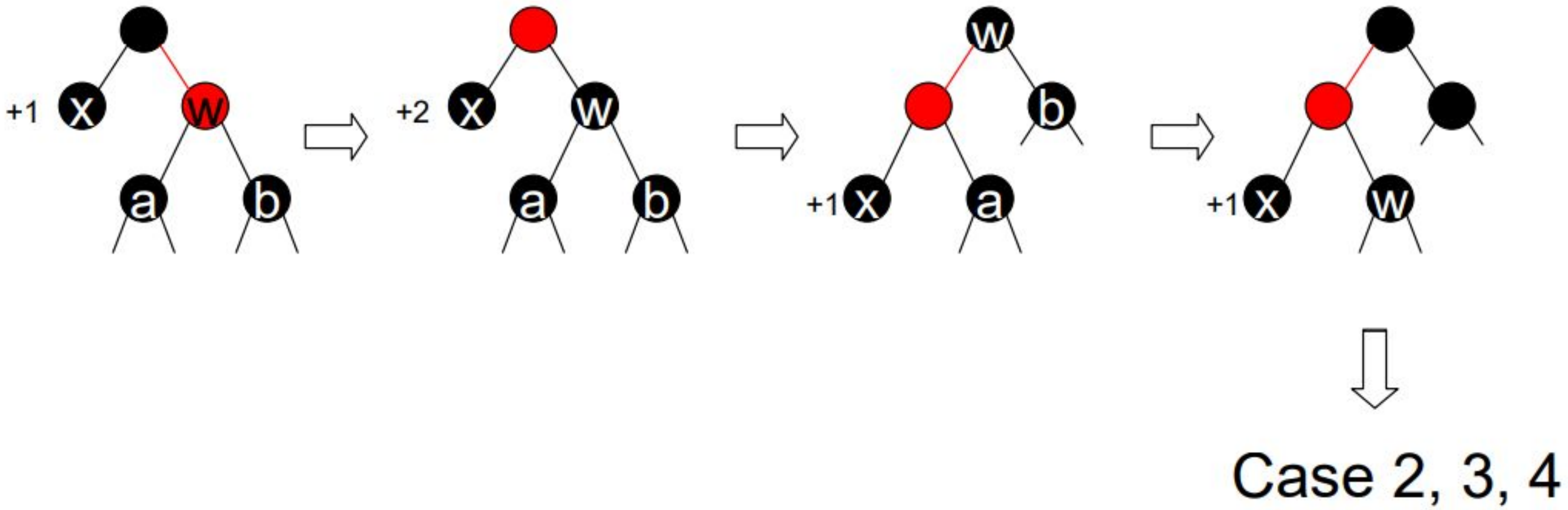
- To fix property 1, we will consider if
  - x is a left child or right child
  - The color of x's sibling w is red or black
  - The colors of w's children
- We consider x is a left child first, the other case can be done by symmetric operation

Red Black Tree- Deletion

- There are 4 cases:
  - Case 1: w is red
  - Case 2: w is black, both w's children are black
  - Case 3: w is black, w's left child is red, w's right child is black
  - Case 4: w is black, w's right child is red

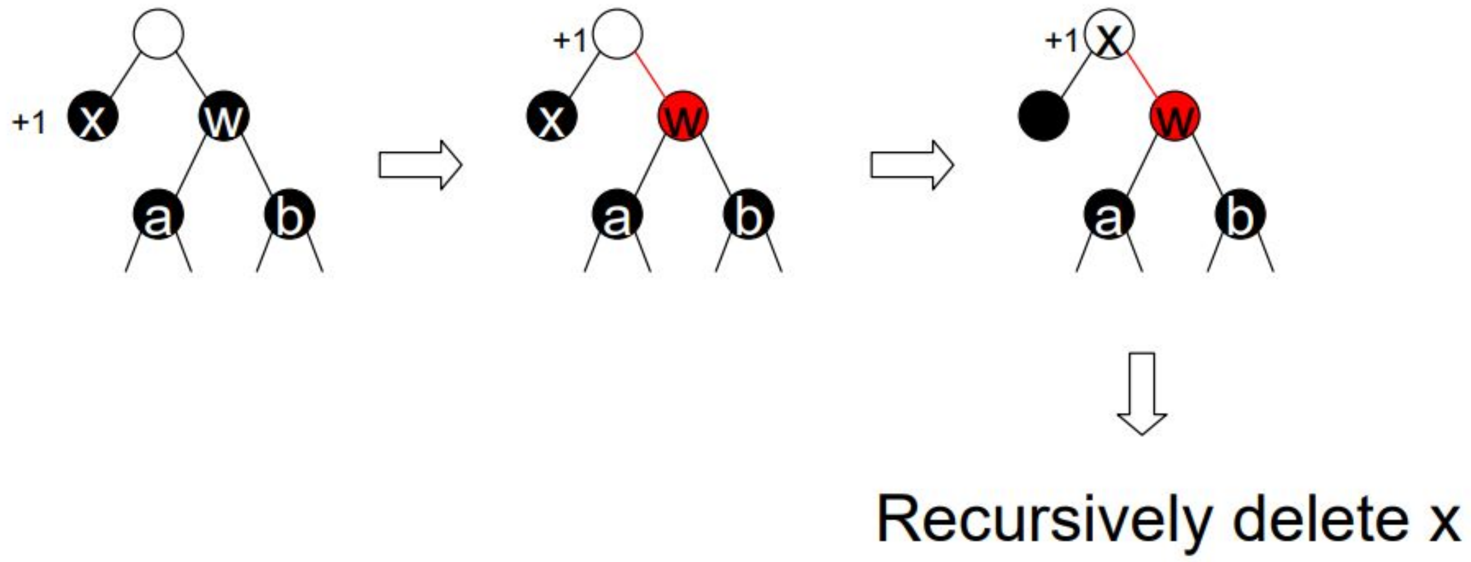
Red Black Tree - Deletion

- Case 1: w is red



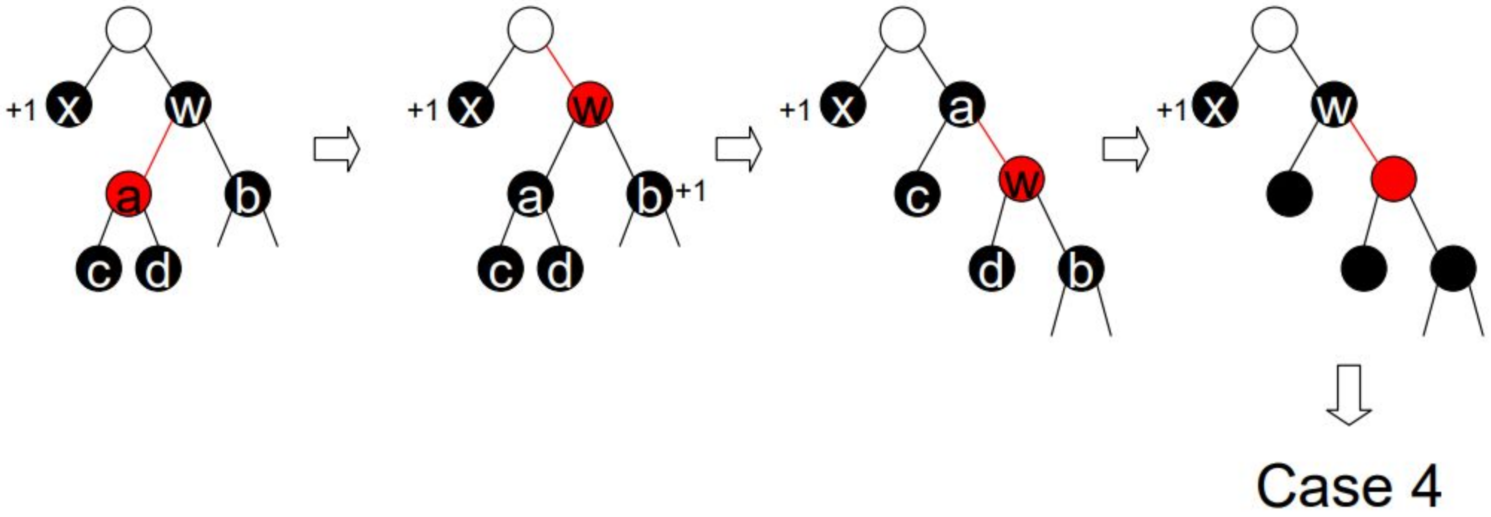
Red Black Tree - Deletion

- Case 2: w is black, both w’s children are black



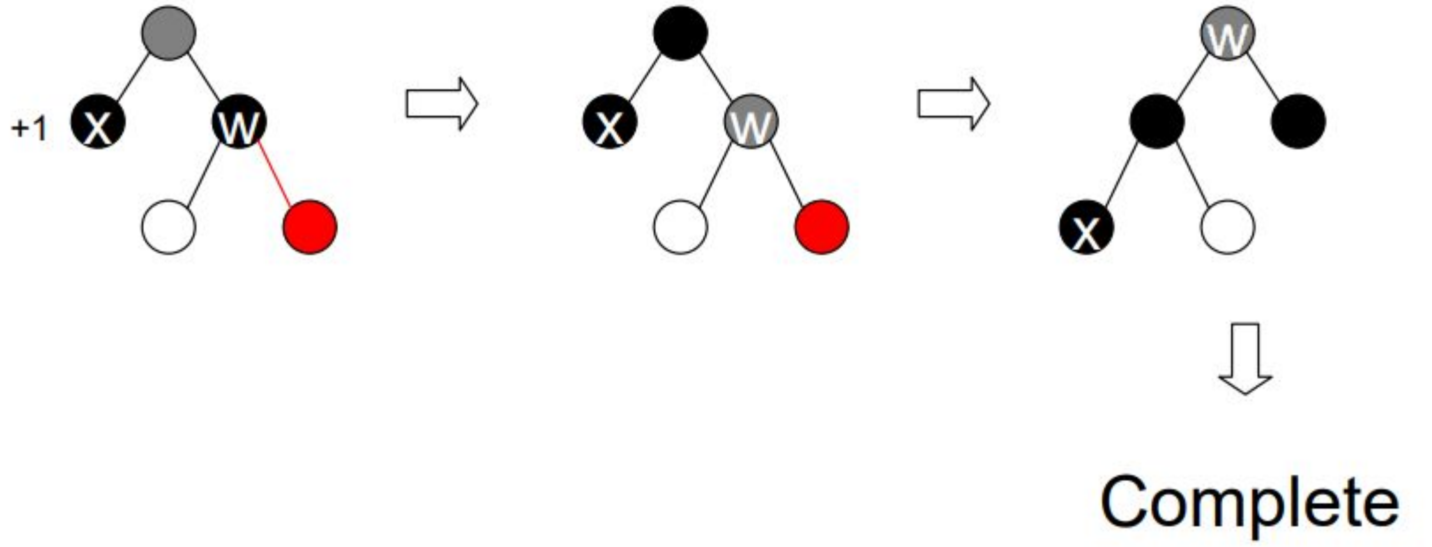
Red Black Tree - Deletion

- Case 3: w is black, w's left child is red, w's right child is black



Red Black Tree - Deletion

- Case 4: w is black, w's right child is red



Red-Black Tree Deletion - Pseudocode

- RB-Delete(T, z)

```
if z->left = null or z->right = null
    then y ← z
else y ← TREE-SUCCESSOR(z)
if y->left ≠ null
    then x ← y->left
else x ← y->right
x->p ← y->p
if y->p = null
    then T->root ← x
else if y = y->p->left
    then y->p->left ← x
else y->p->right ← x
if y ≠ z
    then z->key ← y->key
copy y's data into z
if y->color = BLACK
    then RB-DELETE-FIXUP(T, x)
return y
```

Red-Black Tree Deletion - Pseudocode

- **RB-DELETE-FIXUP(T, x)**

```
while x ≠ T->root and x->color = BLACK
    do if x = x->p->left
        then w ← x->p->right
        if w->color = RED
            then w->color ← BLACK Case 1
        x->p->color ← RED Case 1
        LEFT-ROTATE(T, x->p) Case 1
        w ← x->p->right Case 1
        if w->left->color = BLACK and w->right->color = BLACK
            then w->color ← RED Case 2
    x ← x->p Case 2
        else if w->right->color = BLACK
            then w->left->color ← BLACK Case 3
        w->color ← RED Case 3
        RIGHT-ROTATE(T, w) Case 3
        w ← x->p->right Case 3
        w->color ← x->p->color Case 4
        x->p->color ← BLACK Case 4
        w->right->color ← BLACK Case 4
        LEFT-ROTATE(T, x->p) Case 4
        x ← T->root Case 4
        else (same as then clause with "right" and "left" exchanged)
            x->color ← BLACK
```

Red Black Tree – Deletion Summary

In all cases, except 2, deletion can be completed by a simple rotation/ recoloring.

In case 2, the height of the subtree reduces and so we need to proceed up the tree

- If we proceed up the tree, we only need to recolor/rotate.

Complexity- O(log n)

Red-Black Trees to 2-4 Trees

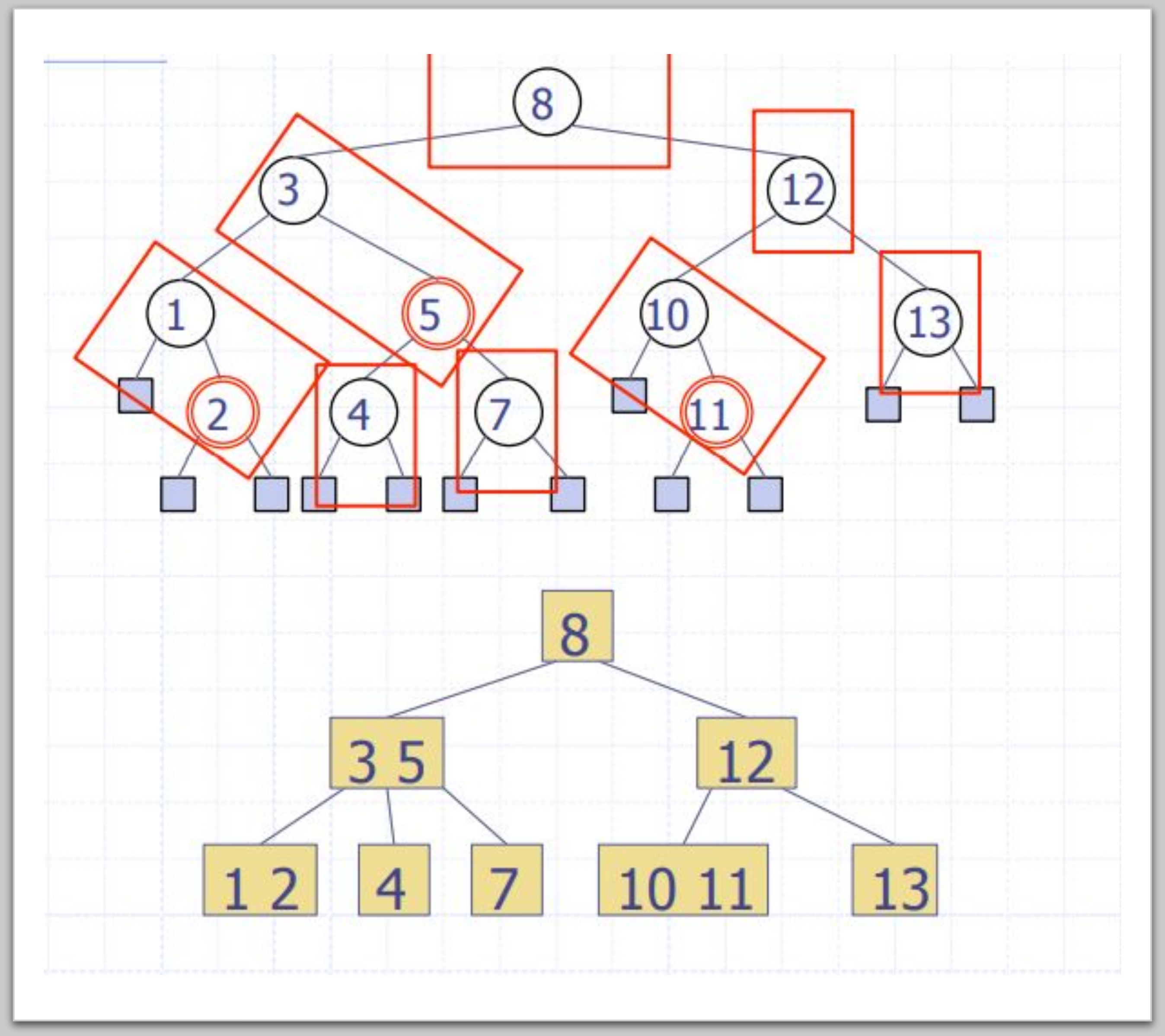
Any red-black tree can be converted into a 2-4 tree

Take a black node and its red children (at most 2) and combine them into one node of a 2-4 tree.

Each node thus formed has at least 1 and at most 3 keys

Since black depth of all external nodes is the same, in the resulting 2-4 tree all the external nodes will be at the same level.

Red-Black Tree to 2-4 Tree - Example



2-4 Trees to Red-Black Trees

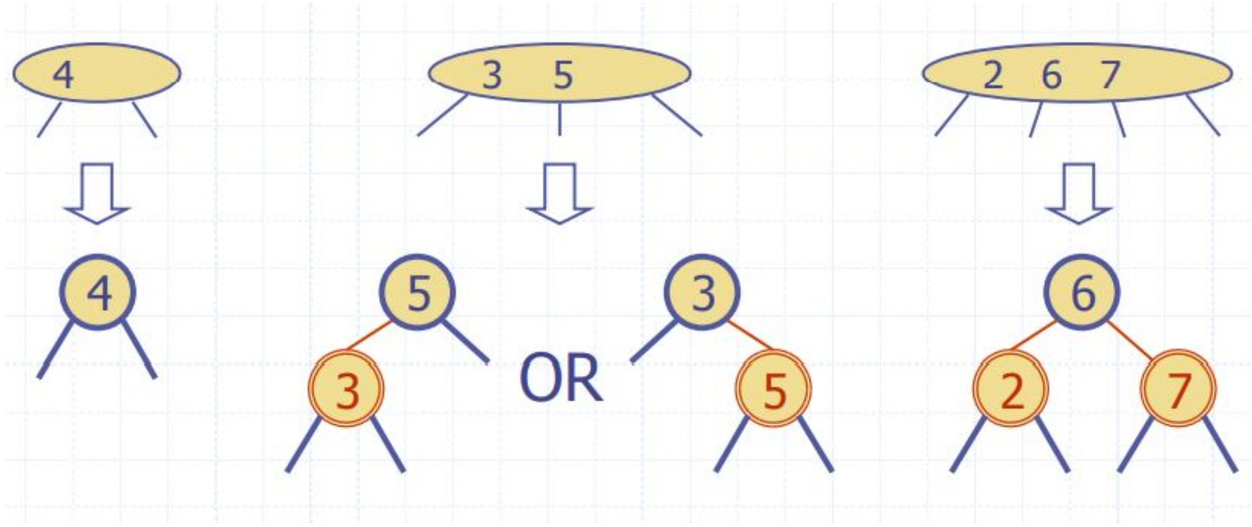
Any 2-4 tree can be converted into a red-black tree

We replace a node of the 2-4 tree with one black node and 0/1/2 red nodes which are children of the black node.

The height of 2-4 tree is the black depth of the red-black tree created.

Every red node has a black child.

2-4 Tree to Red-Black Trees



2-4 Trees to Red-Black Trees - Example