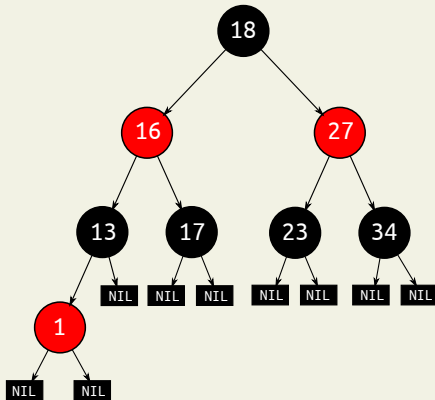# Lecture 5

Instructor: Karteek Sreenivasaiah

21st August 2018

Recap - Red-Black Trees

1. Every node is colored either Red or Black.
2. The root node is black.
3. The leaf nodes (NIL) are black.
4. Both children of a red node are colored black.
5. For any node, all paths from the node to the descendant leaves have the same number of black nodes.

Recap

Claim

A red-black tree with black-height β has height at most 2β.

Theorem

A red-black tree with *n internal* nodes has height at most $2\log(n+1)$.

Insert procedure

Insert(x) – Insert value x into the red-back tree.
High level strategy:

- ▶ Create a node X with value x and color red.
- ▶ Insert node X just like inserting into a Binary Search Tree.
- ▶ Call procedure FixInsert at node X.

FixInsert procedure

Only properties 2 and 4 could be broken after inserting a red node:

1. All nodes are colored either Red or Black.
2. The root node is black.
3. The leaf nodes (NIL) are black.
4. Both children of a red node are black.
5. For any node, all paths from the node to the descendant leaves have the same number of black nodes.

Property 2
The root node is black.

Some invariants when FixInsert is called on a node Z:

- ▶ Z is colored Red.
- ▶ If Property 2 is violated, then node Z itself is the root.

Property 2
The root node is black.

Some invariants when FixInsert is called on a node Z:

- ▶ Z is colored Red.
- ▶ If Property 2 is violated, then node Z itself is the root.

Resolution: Simply color Z black.

Property 4
A red node has black children.

Property 4
A red node has black children.

Some invariants when FixInsert is called on a node Z:

► Z is colored Red.

► If Property 4 is violated, it is violated only by the node Z and its parent.

Property 4
A red node has black children.

Some invariants when FixInsert is called on a node Z:

- Z is colored Red.
- If Property 4 is violated, it is violated only by the node Z and its parent.

There are three cases when FixInsert is called on a node Z.

<div align="center">

Case 1

Uncle of Z is Red.

</div>

Resolution:
- ▶ Recolor parent, uncle and grandparent.
- ▶ Call FixInsert(grandparent)

Example 1

Insert 9 to the following RBT:
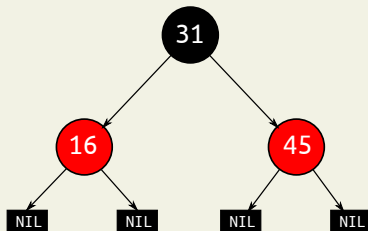
Example 1

Find the position where 9 should be inserted
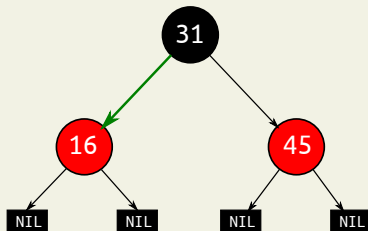
Example 1

Find the position where 9 should be inserted
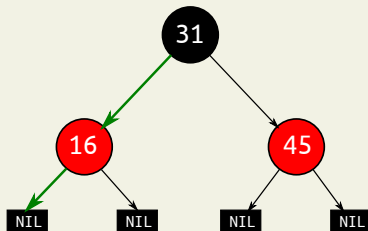
Example 1

Insert 9 as a new node with color red
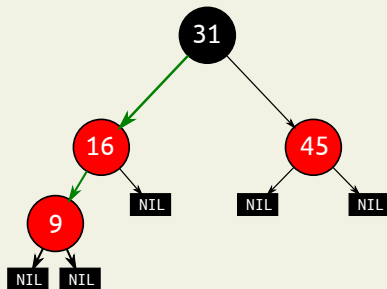
Example 1

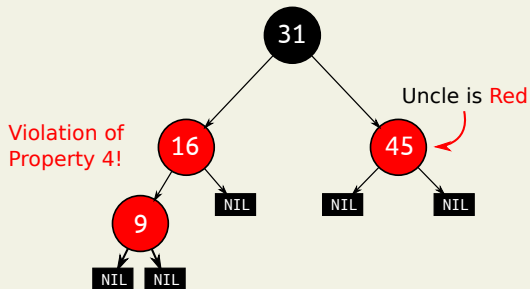Call FixInsert at the inserted location.
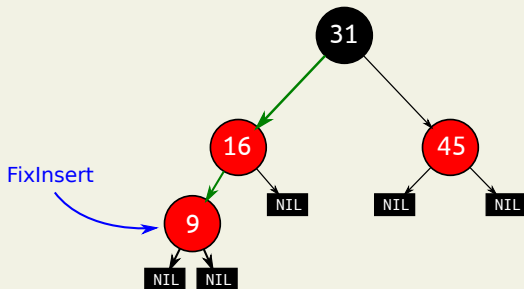
Example 1

Property 4 is violated. Check color of the uncle to determine case.
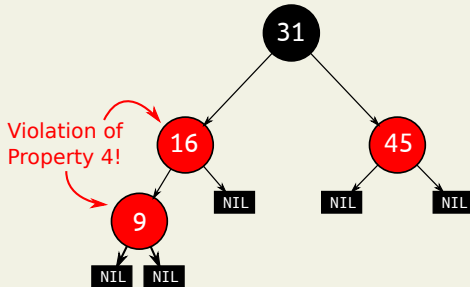
Example 1

We are in Case 1.
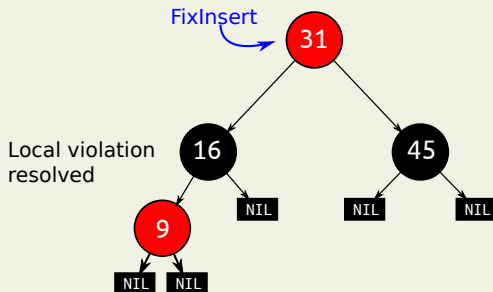
Example 1

Case 1 is resolved by recoloring.
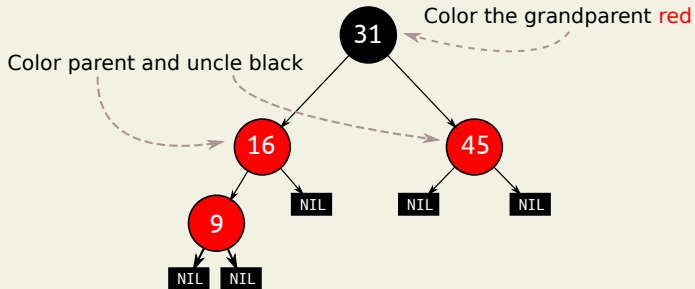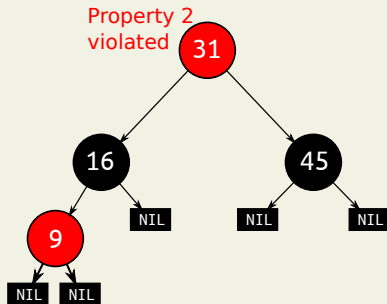
Example 1

Case 1 is resolved by recoloring.
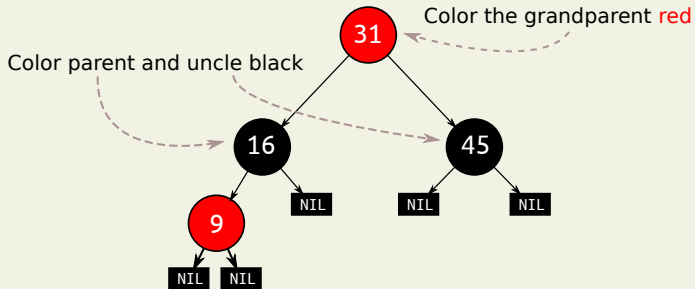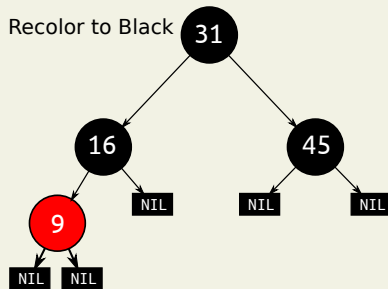
Example 1

Now call FIXINSERT on the grandparent.

Example 1

Root node is not black.

Example 1

Simply recolor root to black.

Case 3

Uncle of Z is Black and Z is left child of a left child.

Resolution:

- ▶ Recolor parent and grandparent.
- ▶ Rotate right at grandparent.

Case 3

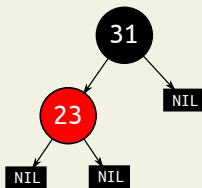## Case 3

Uncle of Z is Black and Z is left child of a left child.

Resolution:

- ▶ Recolor parent and grandparent.
- ▶ Rotate right at grandparent.
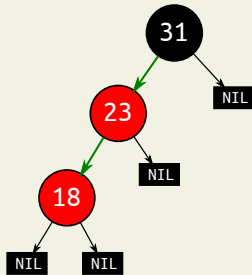
Symmetric case: Z is right child of a right child.

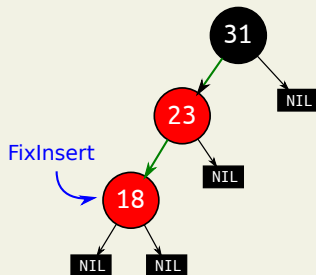# Example 2

Want to insert 18 into this RBT.

Example 2

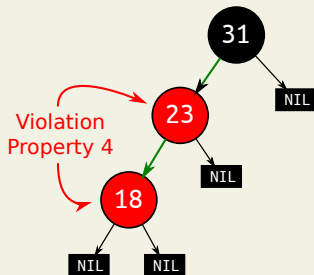Insert 18 as a red node according to BST property.

Example 2

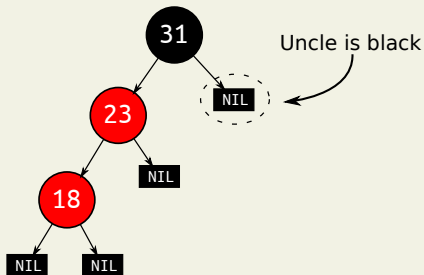Call FixInsert on the new node.

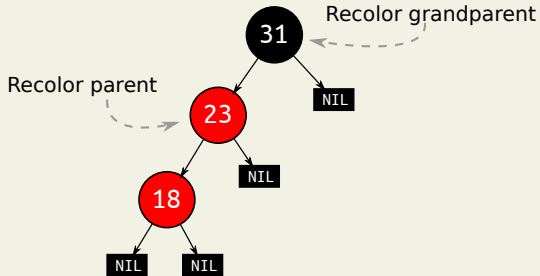# Example 2

FIXINSERT has to fix property 4.

Example 2

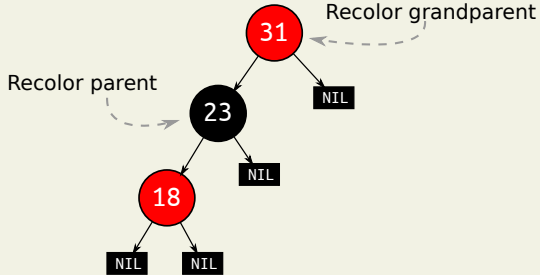Determine the case by looking at uncle.

Example 2

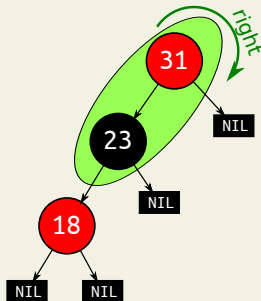Node 18 and its parent 23 are both left children. This is case 3.

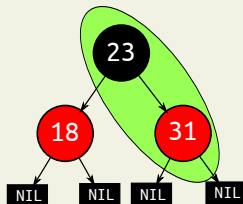# Example 2

Recolor parent to black and grandparent to red.

Example 2

Rotate right at grandparent.

Example 2

The resulting tree has no violations.

Case 2

Uncle of Z is Black and Z is right child of a left child.

Resolution:

- ▶ Assign parent to Z.
- ▶ Rotate left at Z.
- ▶ Call FixInsert at Z.

The above procedure results in case 3.

Case 2

Case 2
Uncle of Z is Black and Z is right child of a left child.
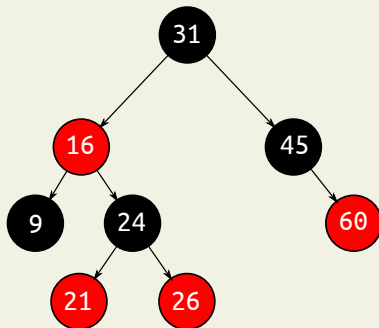
Resolution:
- ▶ Assign parent to Z.
- ▶ Rotate left at Z.
- ▶ Call FixInsert at Z.

The above procedure results in case 3.
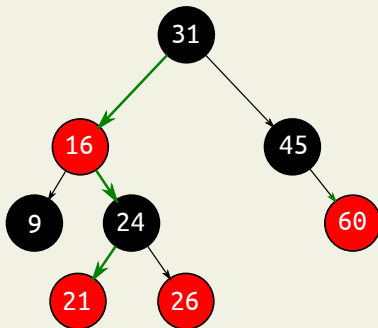Symmetric case: Z is left child of a right child.
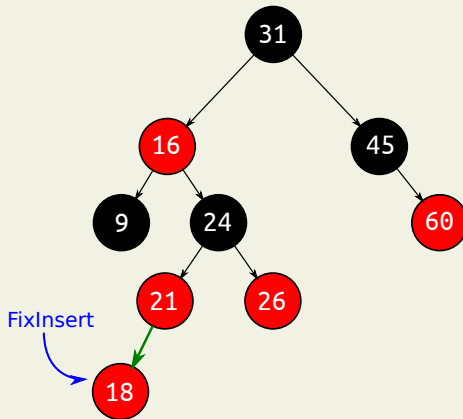
Example 3

Want to insert 18 into the following:
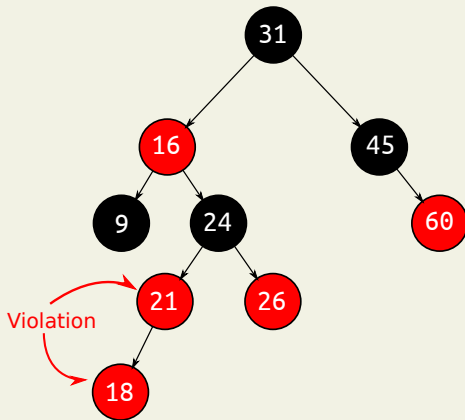
Example 3

Find the position for 18:

Example 3

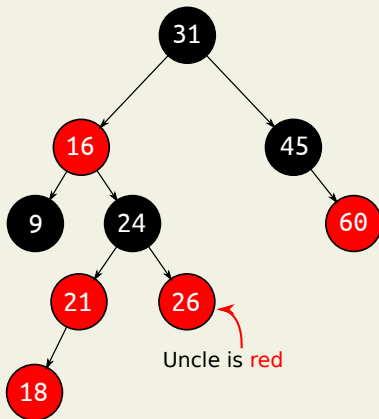Insert a new node with value 18 and color red. Call FixInsert at the inserted location.

Example 3

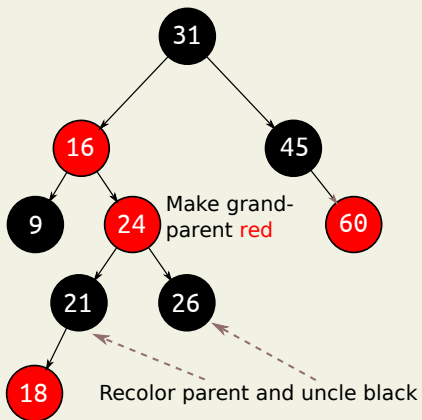Property 4 is violated. Check color of uncle to determine the case.

Example 3

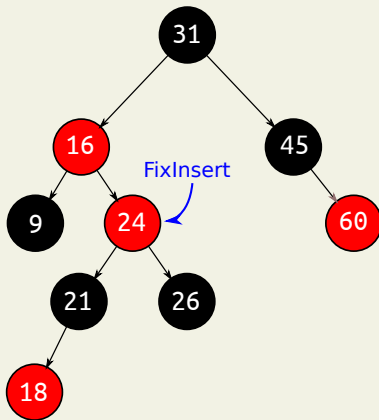Uncle is red. So we are in case 1.

Uncle is red

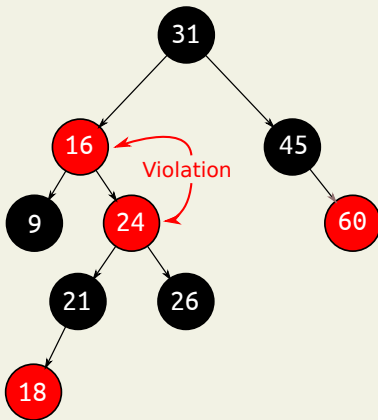# Example 3

Recolor as done earlier.
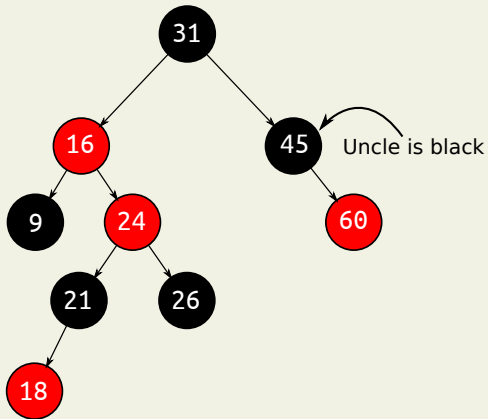
Example 3

Call FixInsert on grandparent.

Example 3

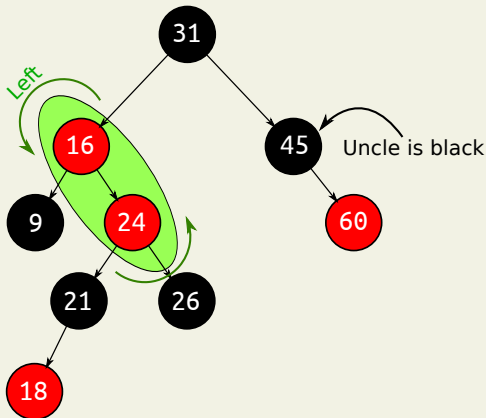Property 4 does not hold. Check color of Uncle to determine case.

Example 3

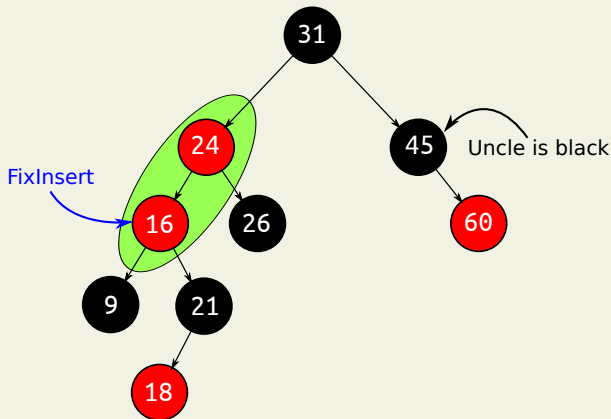Uncle is black and 24 is right child of a left child. So we are in case 2.

Example 3

Set Z as node with 16. Rotate left at Z and call FIXINSERT on Z.
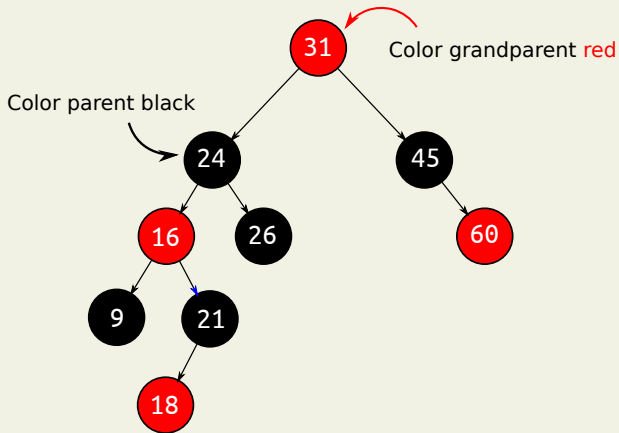
Example 3

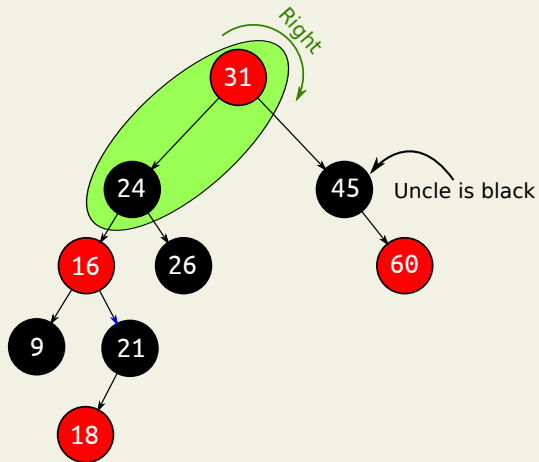Now, Uncle is black and 16 is left child of a left child. This is case 3.
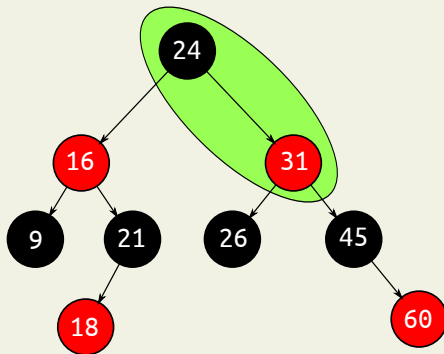
Example 3

Recolor granparent red, parent black.

Example 3
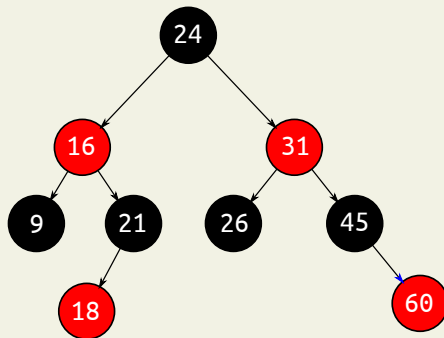
Right rotate at grandparent.

Example 3

Right rotate at grandparent.

Example 3

Done!

FixInsert psudeocode

Algorithm 1 FixInsert called on node Z

1: **while** color(parent(Z)) = red **do**
2:     $U \leftarrow$ Uncle(Z)
3:     **if** parent(Z) is the left child of the grandparent **then**
4:        **if** color(U) = red **then**
5:           Recolor parent, uncle and grandparent.
6:           $Z \leftarrow$ grandparent(Z).
7:        **else**
8:           **if** Z is the right child **then**
9:              $Z \leftarrow$ parent(Z); Left rotate at (Z)
10:           **end if**
11:           Recolor parent and grandparent.
12:           Right rotate at grandparent(Z).
13:        **end if**
14:     **end if**
15: **end while**
16: color(root)$\leftarrow$ black.