Design and Analysis of Algorithms

Lecture-15

Dharmendra Kumar (Associate Professor)
Department of Computer Science and Engineering
United College of Engineering and Research,
Prayagraj

Unit-2

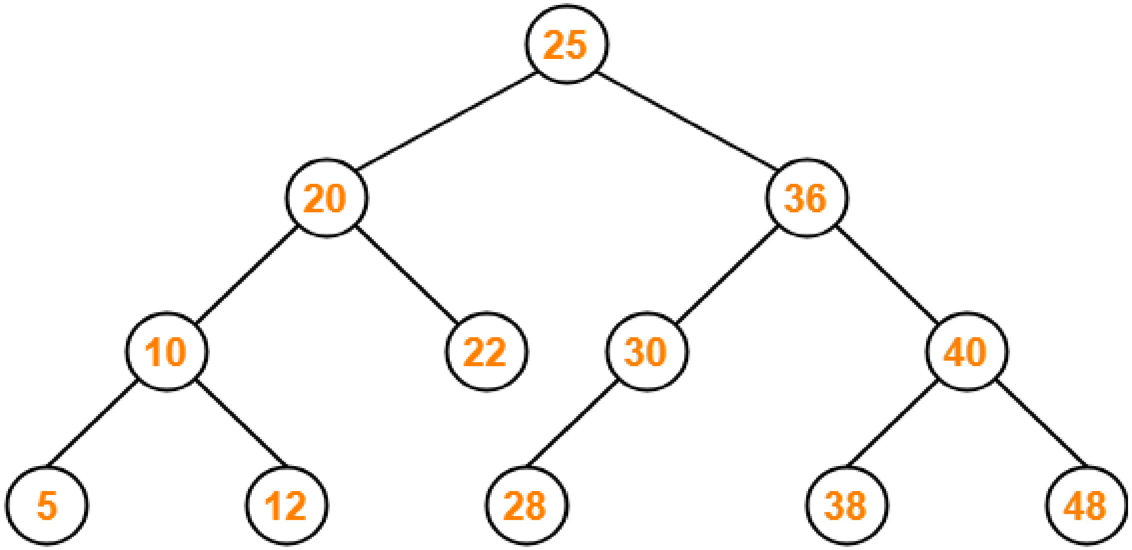
Red-Black Tree

Red-Black Tree

Binary Search Tree(BST)

A binary tree is said to be binary search tree if the value at the left child is less than value at the parent node and value at the right child is greater or equal than value at the parent node.

Example:



Binary Search Tree

Red-Black Tree

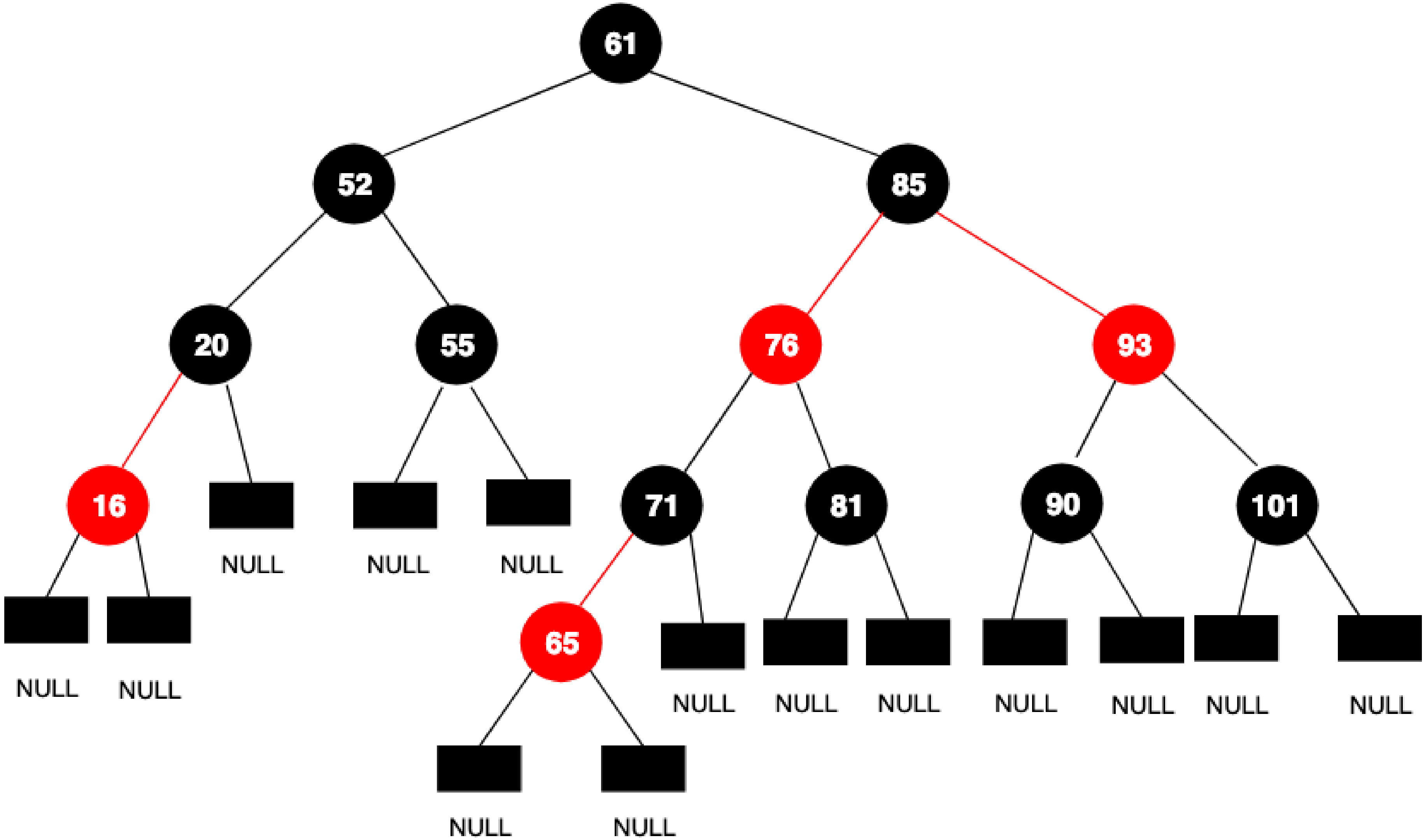
Definition

A red-black tree is a binary search tree that satisfies the following ***red-black properties***:

1. Every node is either red or black.
2. The root is black.
3. The color of every leaf node is black and its value is always NIL.
4. If a node is red, then both its children are black.
5. For each node, all simple paths from the node to descendant leaves contain the same number of black nodes.

Red-Black Tree

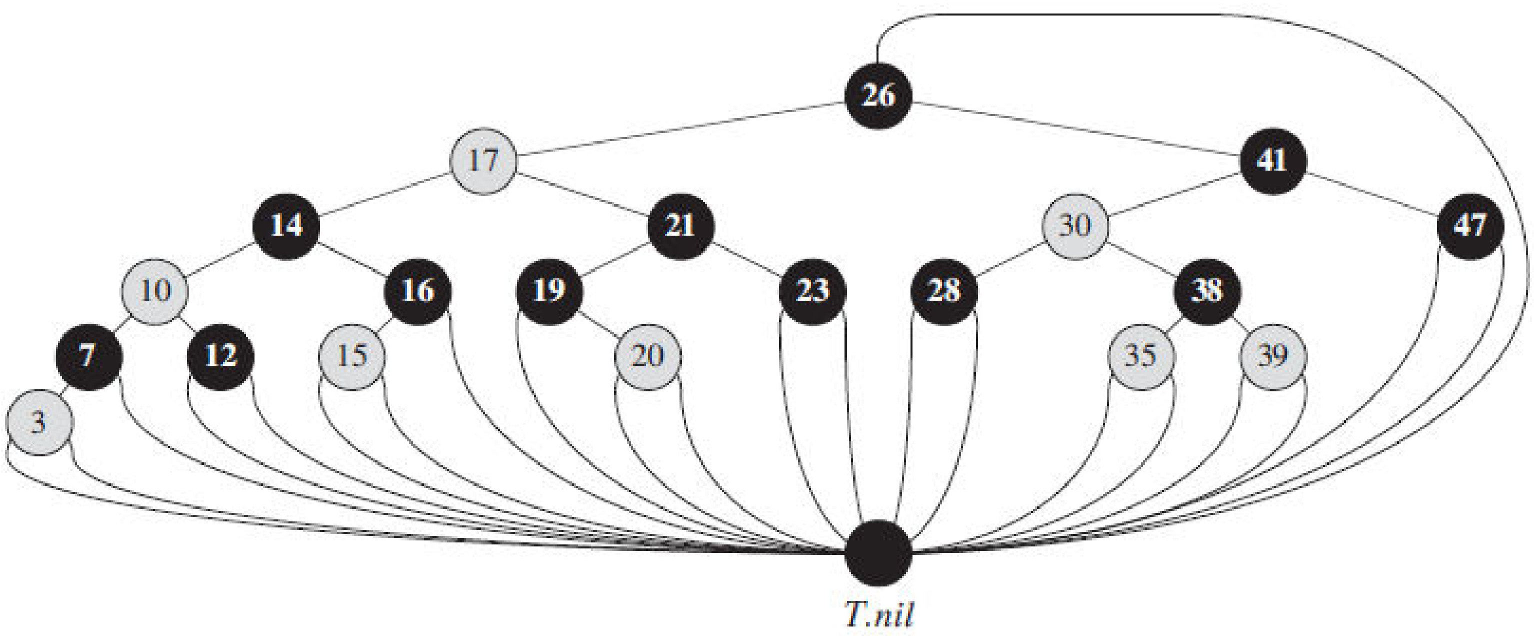
Example:



Red-Black Tree

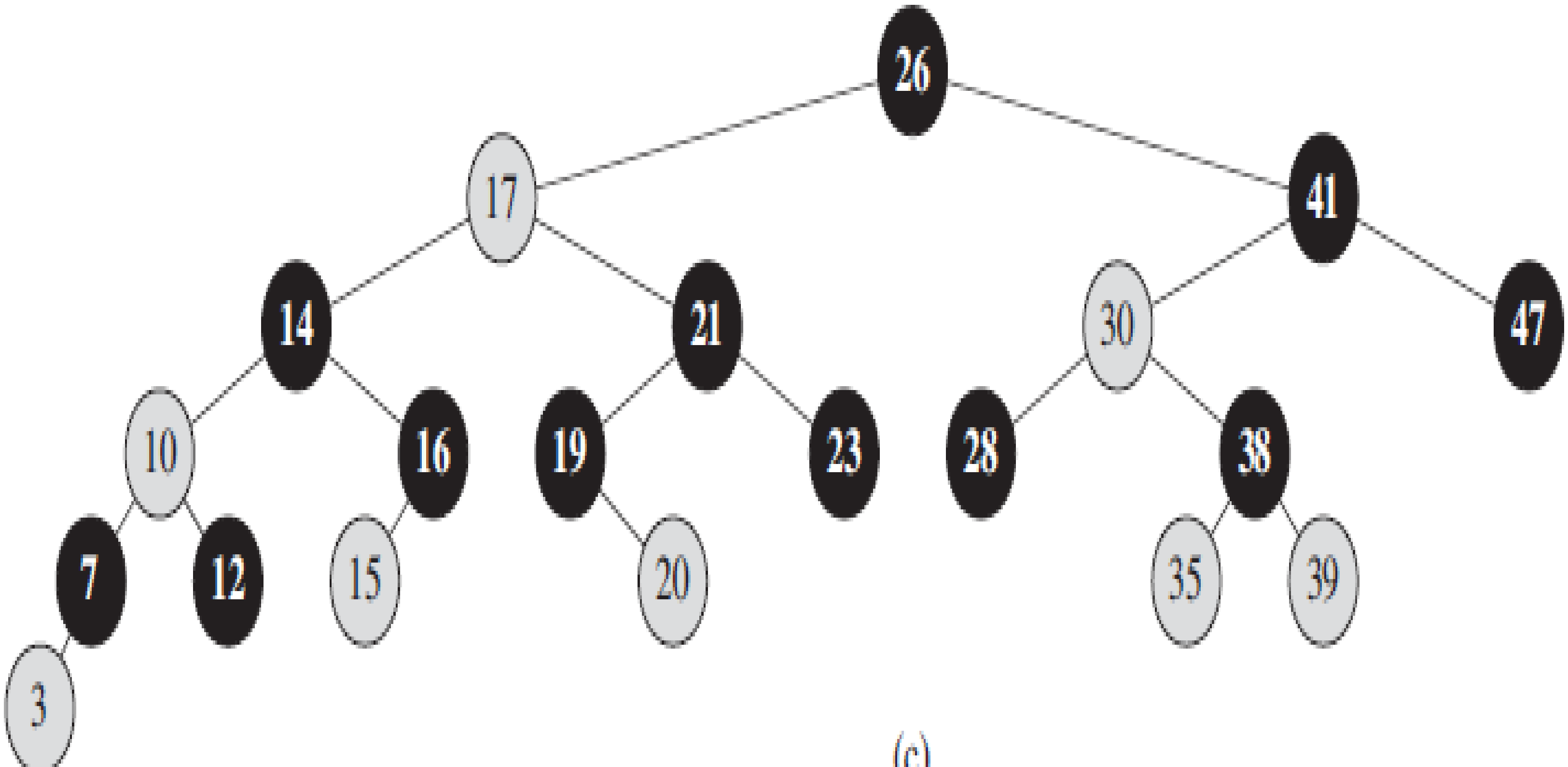
Sentinel

For a red-black tree T, the sentinel is an object with the same attributes as an ordinary node in the tree. Its color is black and its value is nil. It is represented by T.nil or nil[T].



Red-Black Tree

Red-Black tree without leaf node



(c)

Red-Black Tree

Black height of a node

Black height of a node x in the red-black tree is the number of black nodes on any downward path from node x to a leaf node but not including node x. It is denoted by bh(x).

Black height of a Red-Black tree

Black height of a red-black tree is equal to the black height of the root node.

Red-Black Tree

Theorem: A red-black tree with n internal nodes has height at most 2lg(n+1).

Proof: Before proving the theorem, first we will prove the following statement.

"The subtree rooted at any node x contains at least $2^{bh(x)}-1$ internal nodes." .... (1)

We will prove the statement (1) using induction method. We will use induction parameter as the height of the red-black tree.

For height, h = 0.

Red-black tree of height 0 is only single node i.e. it will be following :-

x 

Clearly, minimum number of internal nodes in this tree = 0

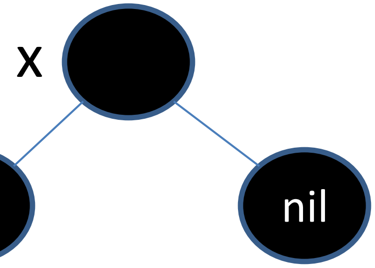
and $2^{bh(x)}-1 = 2^{0}-1 = 1-1 = 0$

Therefore, statement (1) is true for height h=0.

Red-Black Tree

For height, h = 1.

Red-black tree of height h = 1 will be the following:-



Clearly, minimum number of internal nodes in this tree = 1

And $2^{bh(x)}-1 = 2^1-1 = 2-1 = 1$

Therefore, statement (1) is also true for height h=1.

Now, we assume statement (1) is true for children of node x.

We will prove the statement for node x.

Red-Black Tree

Now,

The minimum number of internal nodes in the subtree rooted at node x

= Minimum number of internal nodes in the subtree rooted at left child of node x + Minimum number of internal nodes in the subtree rooted at right child of node x + 1

Since black height of each child of node x has either bh(x) or bh(x)-1, depending on the color of child. If the color of the child is red then black height of child is bh(x) but if the color of the child is black then black height of child is bh(x) -1. Therefore,

The minimum number of internal nodes in the subtree rooted at node x

$$= 2^{(bh(x) -1)} -1 + 2^{(bh(x) -1)} -1 + 1$$

$$= 2 \cdot 2^{(bh(x) -1)} -1$$

$$= 2^{bh(x)} -1$$

Therefore, statement (1) is also proved for node x of any height.

Red-Black Tree

Now, we will prove the given theorem using statement (1).

Let h is the height of the red-black tree.

Using property (4) of the red-black tree, minimum number of black nodes on any path from root node to the leaf node will be h/2. Therefore, minimum black height of red-black tree of height h will be h/2.

Now, the minimum number of internal nodes in red-black tree of height h $= 2^{h/2} - 1$

Since the gien number of internal nodes in red-black tree is n, therefore

$$2^{h/2} - 1 \le n \quad \Rightarrow 2^{h/2} \le n + 1$$

$$\Rightarrow h/2 \le \lg(n+1) \quad \Rightarrow \quad h \le 2\lg(n+1)$$

Therefore, the maximum height of red-black tree will be 2lg(n+1).

Now, it is proved.