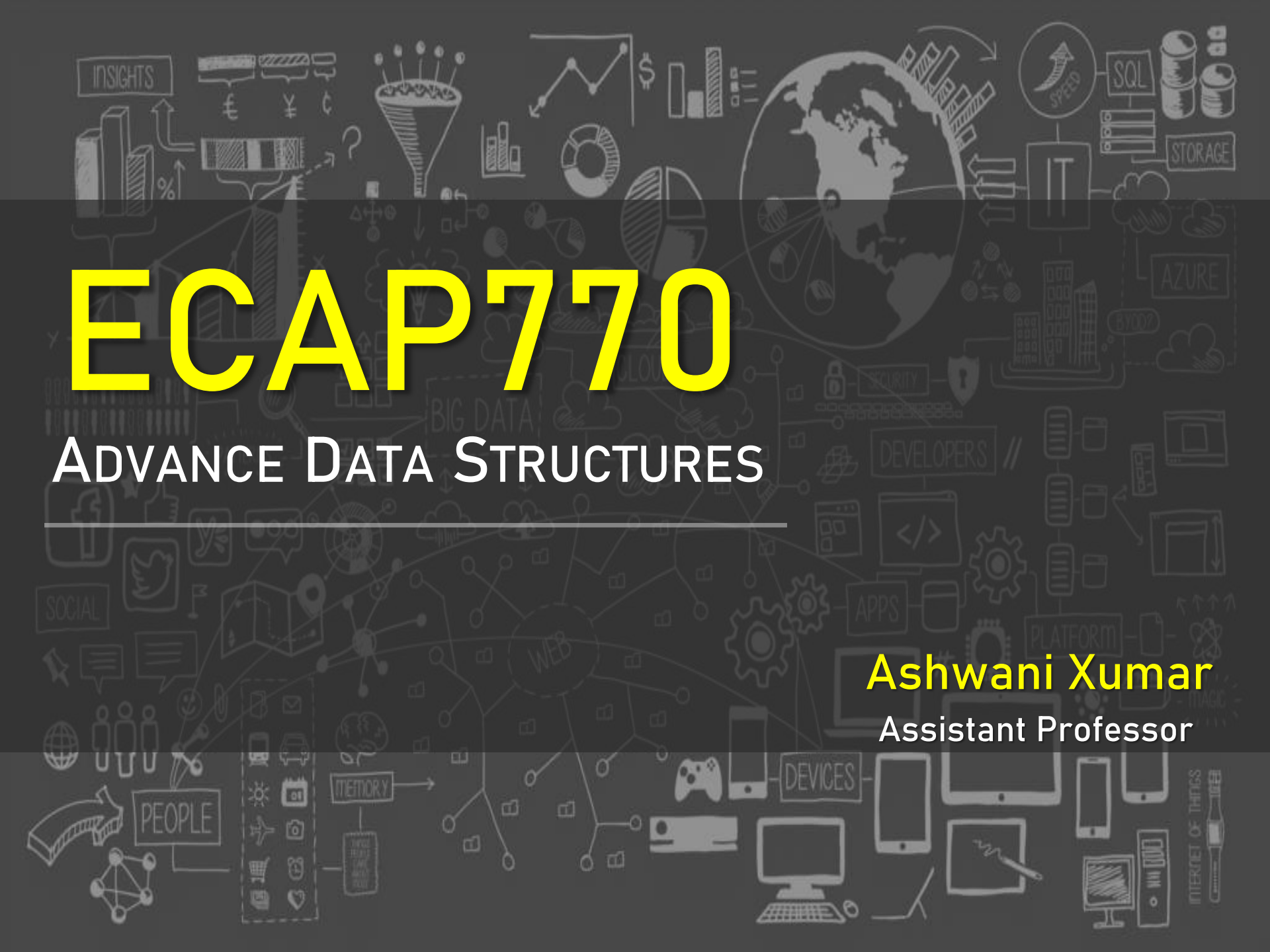


ECAP770

Advance Data Structures

Ashwani Xumar

Assistant Professor

Learning Outcomes

After this lecture, you will be able to

- Understand Red-Black Tree
- Operations on Red-Black Tree

Red-Black Tree

- A Red Black Tree is a self-balancing binary search tree with one extra attribute for each node: the colour, which is either red or black.
- It was invented in 1972 by Rudolf Bayer.
- Colours in tree are used to ensure that the tree remains balanced during insertion and deletion.

Why Red-Black Trees

- BST operations take O(h) time where h is the height of the BST.
- Cost of these operations may become O(n) for a skewed Binary tree. If we make sure that the height of the tree remains O(log n) after every insertion and deletion, then an upper bound of O(log n) for all these operations.
- The height of a Red-Black tree is always O(log n) where n is the number of nodes in the tree.

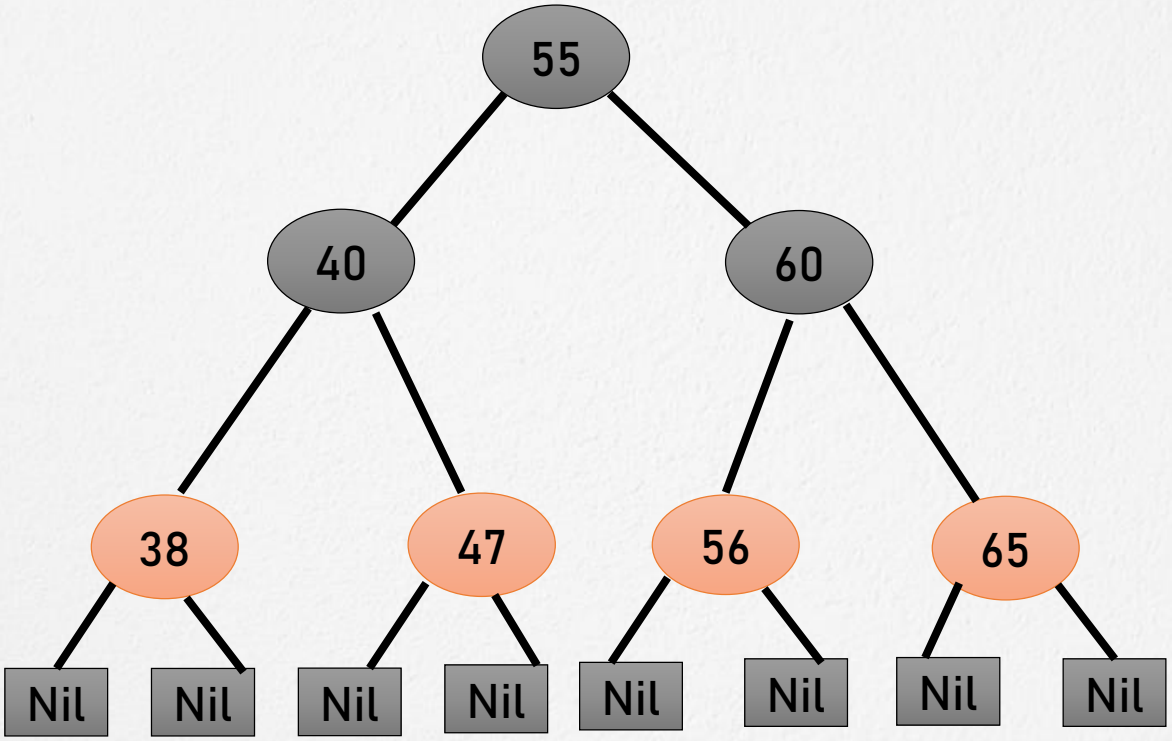
Red-black tree properties

- Red - Black Tree must be a Binary Search Tree.
- The root of the tree is always black.
- The children of Red colored node must be colored BLACK. (There should not be two consecutive RED nodes)
- Every new node must be inserted with RED color.

Red-black tree properties

- Every leaf (e.i. NULL node) must be colored BLACK.
- In all the paths of the tree, there should be same number of BLACK colored nodes.

Red-black tree



Red-black tree operations

- Insertion
- Deletion
- Search
- Recolor and Rotation performed on insertion and deletion operation as per Red-Black Tree properties.

Red-black tree insertion operations

- The insertion operation in Red Black Tree is similar to the Binary Search Tree.
- Every new node must be inserted with the color RED.
- After every insertion operation, we need to check all the properties of Red-Black Tree. If all the properties are satisfied then we go to next operation otherwise we perform the following operation to make it Red Black Tree.
- 1. Recolor
- 2. Rotation
- 3. Rotation followed by Recolor

Steps for Red-black tree insertion operations

- Check whether tree is Empty.
- If tree is Empty then insert the newNode as Root node with color Black and exit from the operation.
- If tree is not Empty then insert the newNode as leaf node with color Red.
- If the parent of newNode is Black then exit from the operation.

Steps for Red-black tree insertion operations

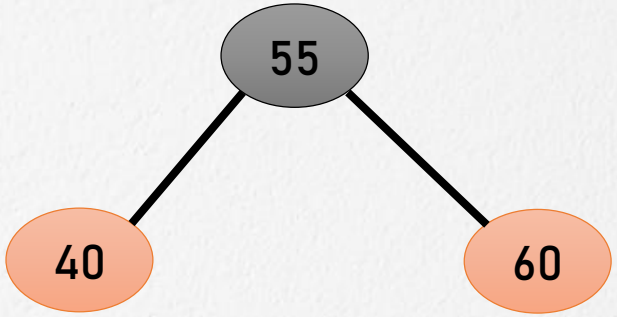
- If the parent of newNode is Red then check the color of parentnode's sibling of newNode.
- If it is colored Black or NULL then make suitable Rotation and Recolor it.
- If it is colored Red then perform Recolor. Repeat the same until tree becomes Red Black Tree.

Red-black tree insertion operations

Step 1

Inset 55

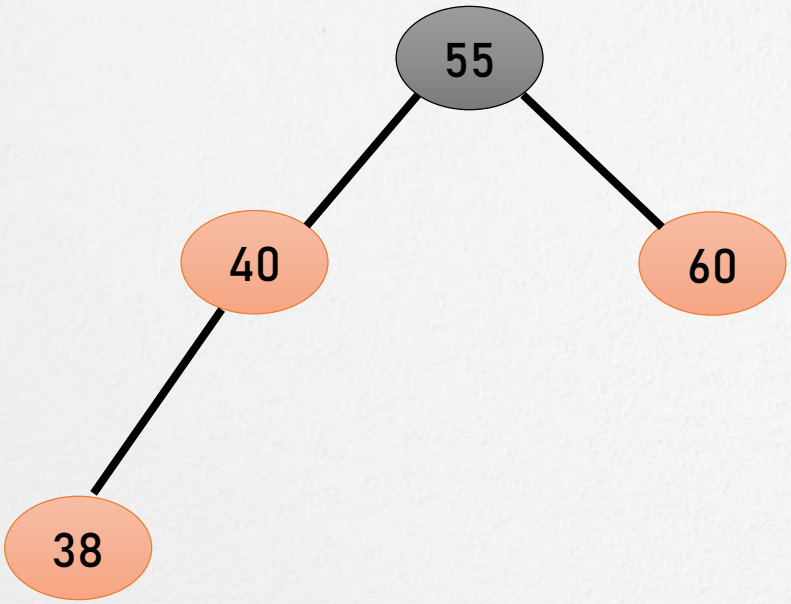
Step 2



Insert 40 60

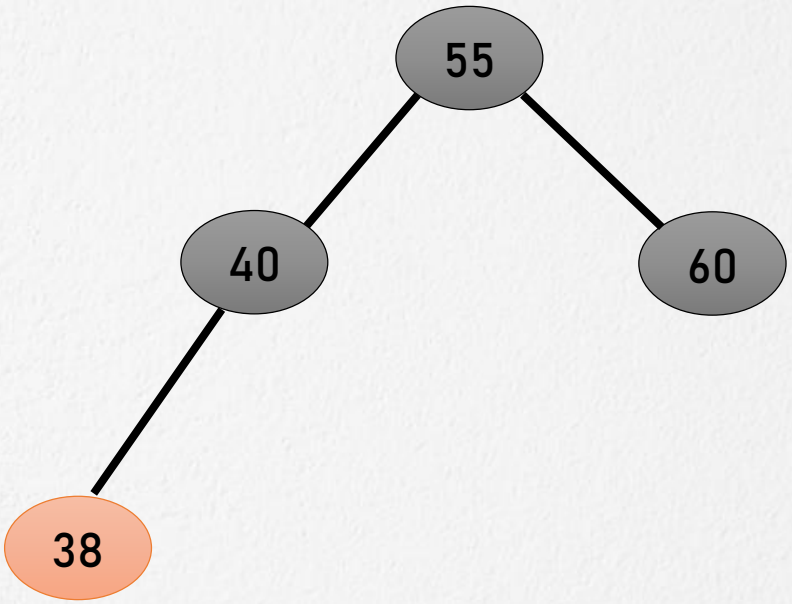
Red-black tree insertion operations

Step 3



Inset 38

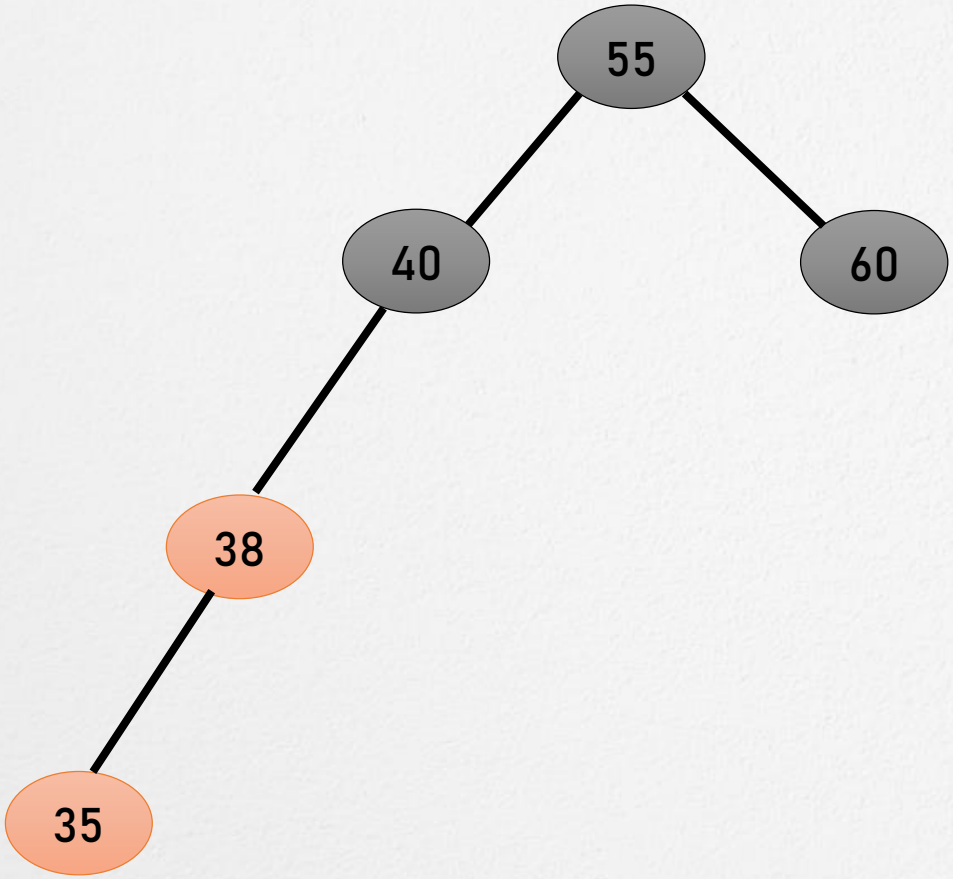
Step 4



After recolor operation

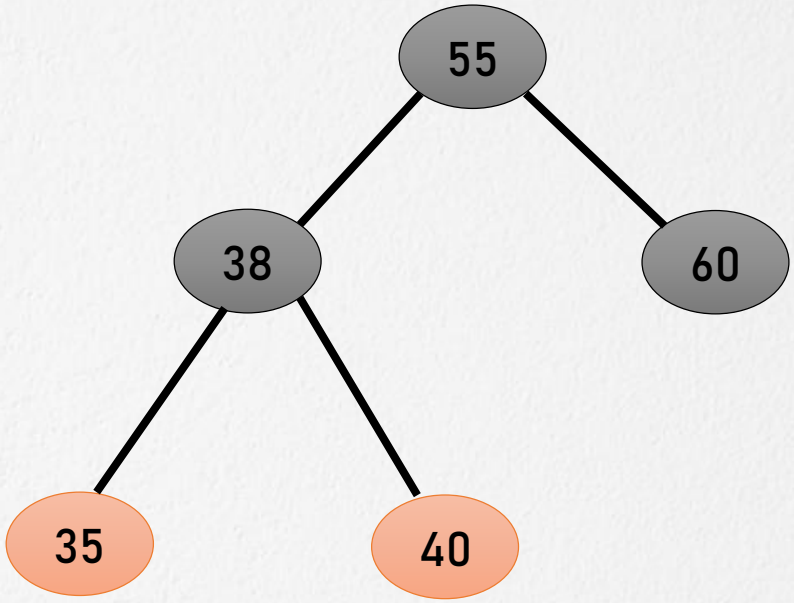
Red-black tree insertion operations

Step 5



Inset 35

Step 6



After rotation and recolor

Red-black tree deletion operations

- The deletion operation in Red-Black Tree is similar to the BST.
- In deletion operation, we need to check with the Red-Black Tree properties. If any of the properties are violated then make suitable operations like Recolor, Rotation and Rotation followed by Recolor to make it Red-Black Tree.

Algorithm: Search operation

searchElement (tree, val)

Step 1:

If tree -> data = val OR tree = NULL

Return tree

Else

If val < data

Return searchElement (tree -> left, val)

Else

Return searchElement (tree -> right, val)

[End of if]

[End of if]

Step 2: END

Time complexity in big O notation

Sr. No.	Algorithm	Time Complexity
1.	Search	O(log n)
2.	Insert	O(log n)
3.	Delete	O(log n)

Red-black tree applications

- To implement Java packages.
- To implement Standard Template Libraries (STL) in C++.
- It is used in K-mean clustering algorithm for reducing time complexity.
- To implement finite maps
- It is used to implement CPU Scheduling in Linux.
- In MySQL Red-Black tree is used for indexes on tables.

That's all for now...