

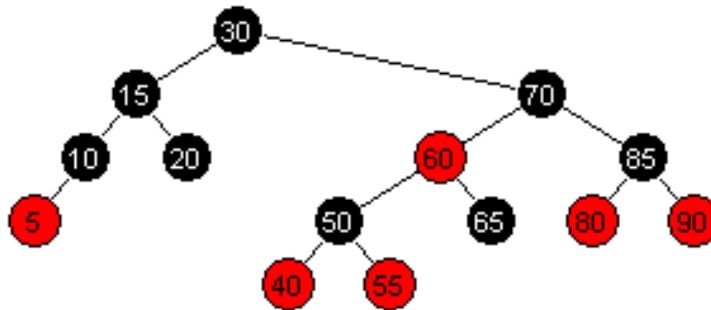
Red Black Trees

- Definition
- Bottom-up Insertion
- Top-Down Insertion

Definition of Red Black Trees

- A Red Black tree is a BST with the following properties:
 1. Every node is either colored **red** or black.
 2. The root is black.
 3. No two successive nodes are **red**.
 4. Every path from the root to a null node has the same number of black nodes.

Example



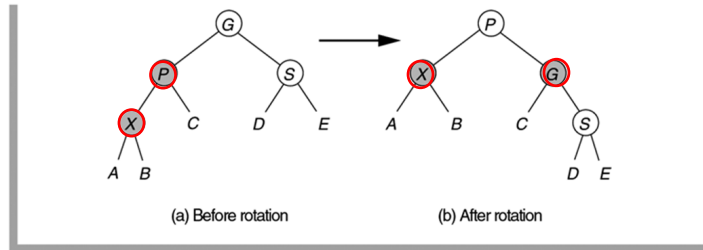
Bottom-Up Insertion Strategy

- Insertion is always done as a leaf (as in ordinary BST insertion) and the new node is red.
- In Bottom-Up insertion, we first insert the node.
- On the recursive travel back up the tree, we balance the tree.
- Rotations preserve red-black tree properties.

Bottom-Up Insertion Strategy

figure 19.35

If *S* is black, a single rotation between parent and grandparent, with appropriate color changes, restores property 3 if *X* is an outside grandchild.

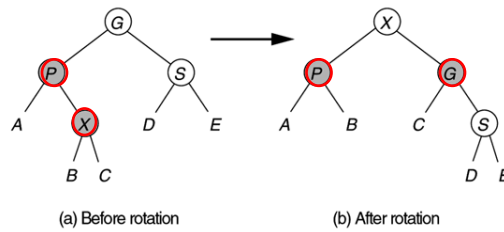


Copyright © 2010 Pearson Education

Bottom-Up Insertion Strategy

figure 19.36

If *S* is black, a double rotation involving *X*, the parent, and the grandparent, with appropriate color changes, restores property 3 if *X* is an inside grandchild.



Copyright © 2010 Pearson Education

Bottom-Up Insertion Strategy

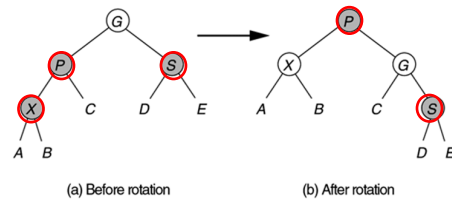


figure 19.37

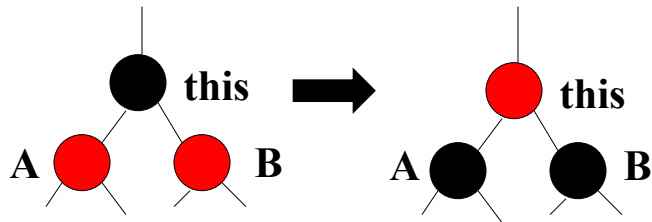
If *S* is red, a single rotation between parent and grandparent, with appropriate color changes, restores property 3 between *X* and *P*.

Copyright © 2010 Pearson Education

Top-Down Insertion Strategy

- Insertion is always done as a leaf (as in ordinary BST insertion) and the new node is red.
- In Top-Down insertion, the rotations are done while traversing down the tree to the insertion point.
- Top-Down insertion can be done iteratively.

Top-Down Traversal



Situation: A black node with two red children.

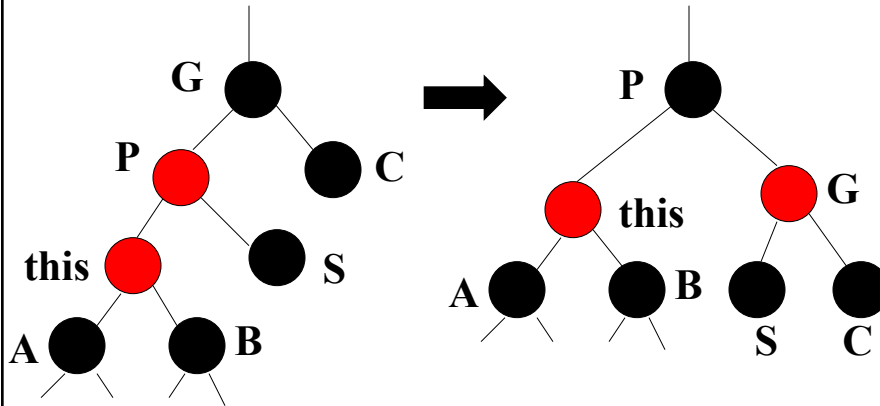
Action:

- Recolor the node **red** and the children black.
- If the parent is **red**, perform rotations, otherwise continue down the tree

Rotations

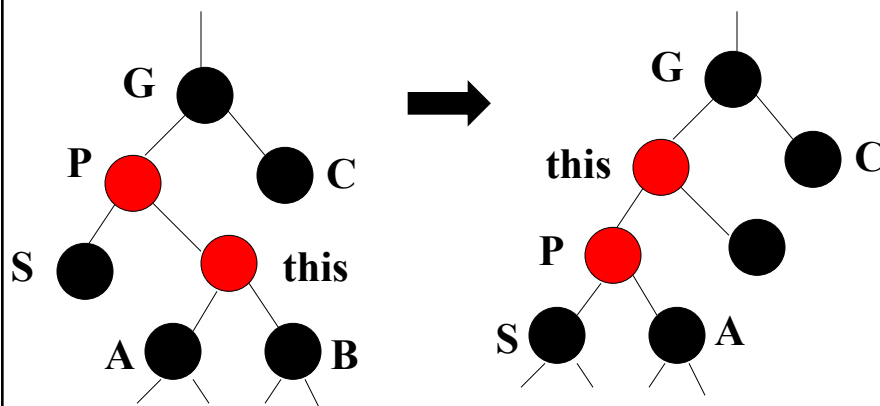
- If the color flip produced two successive red nodes, perform either a single or a double rotation.
- The rotations are just like those for AVL trees.
- If the two red nodes are both left children or both right children, perform a *single rotation*.
- Otherwise, perform a *double rotation*.
- The only difference to AVL tree rotations are that we recolor nodes rather than adjust their heights.

Single Rotation on Left Child



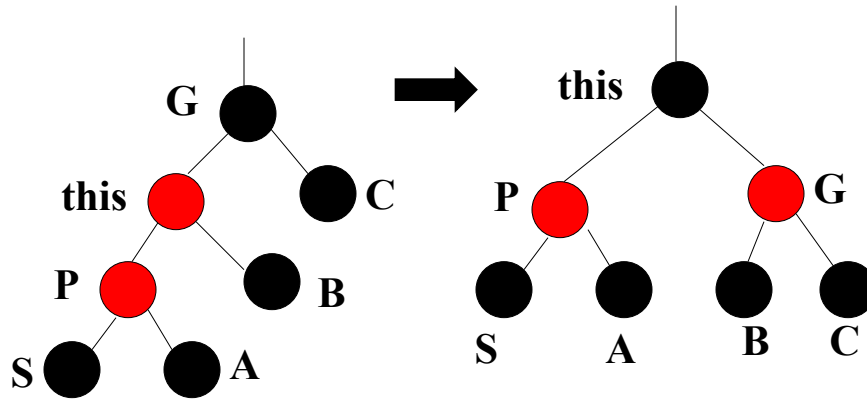
- The rotation is done on G, the grandparent of this.
- The colors of P and G are flipped.

Double Rotation on Left Child



- Again, the rotation is done on G, the grandparent of this.

Double Rotation on Left Child (cont'd)



- Recolor **this** and **G**

Testing

- Insert: 1, 2, 3, 4, 5, 6, 7, 7
- Insert: 7, 6, 5, 4, 3, 2, 1, 1
- Insert: 10, 85, 15, 70, 20, 60, 30, 50, 65, 80, 90, 40, 5, 55