

Insertion in Red-black tree

To insert a Node k into a red-black tree T :

Case 1 T is empty

make k the root node & make its color as black.

Case 2 P is black & k 's parent node z .

if P is black, none of the properties are violated. Don't do anything.

Case 3 P is red & adjacent nodes are red z .

* if u & P 's sibling is also red z .

change color of P and u to black.

change color of grandparent to red.

& if its not the root node z .

* if u & P 's sibling z is black or NULL

• 1. P is right of G & k is right of P

perform left rotation at G and

interchange color of G & P .

• 2. P is right of G & k is left of P .

right rotate P and it reduces to

1. follow same steps.

Akanksha

• 3. P is left of G & R is left of P.
right rotation at G & interchange
color of G & P.

• 4. P is left of G & R is right of P.
left rotation at P and it reduces to
3. follow same steps.

insertion (T, K)
{

bst insertion (T, K) // normal bst
insertion

while K.parent.color == RED

if K.parent == K.parent.parent.right

u = K.parent.parent.left

if u.color == RED

u.color = BLACK

K.parent.color = BLACK

K.parent.parent.color = RED

K = K.parent.parent

else if K == K.parent.left

K = K.parent

leftrotate (T, K)

K.parent.color = BLACK

K.parent.parent.color = RED

Rightrotate (T, K.parent.
parent)

else (same as then clause with left, right
exchanged)

T.root.color = BLACK.

Akanksha