Red & Black Tree Simulation No 2 (INSERTION)

Insert: 15, 30, 20, 40, 10

Delete: 30, 15

Insert: 60, 30, 18, 25, 45

Delete: 20, 10, 40

Insert: 65, 20, 35

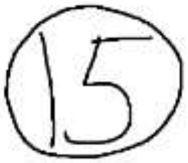
The algoritm for every insertion is:

1. Newly inserted nodes will be **RED**
2. If x is the root, change the colour of x as **BLACK** (Black height of complete tree increases by 1).
3. Do the following if the color of x's parent is not **BLACK and** x is not the root.
   a) If x's uncle is RED
   (i) Change the colour of parent and uncle as **BLACK**.
   (ii) Colour of a grandparent as **RED**.
   (iii) Change x = x's grandparent, repeat steps 2 and 3 for new x.

   b) If x's uncle is BLACK, then there can be four configurations for x, x's parent (**p**) and x's grandparent (**g**)
   (i) Left Left Case (p is left child of g and x is left child of p)
   (ii) Left Right Case (p is left child of g and x is the right child of p)
   (iii) Right Right Case (Mirror of case i)
   (iv) Right Left Case (Mirror of case ii)

INSERT 15

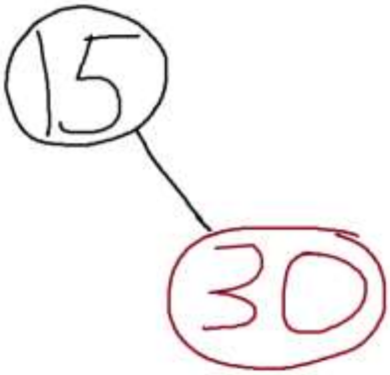
1. Newly inserted nodes will be **RED**
2. If x is the root, change the colour of x as **BLACK**



INSERT 30

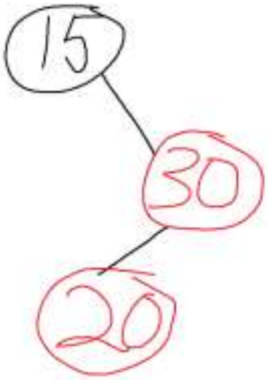
Like the standard AVL tree, for insertion

1. Newly inserted nodes will be **RED**

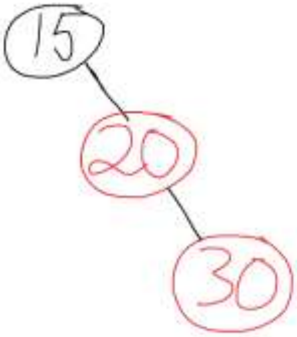


INSERT 20

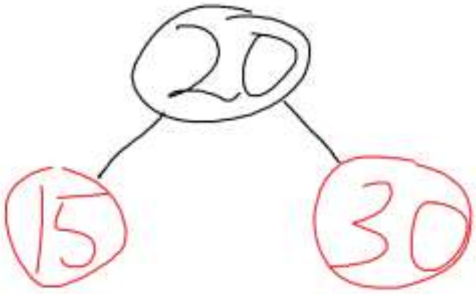
1. Newly inserted nodes will be **RED**



2. Node and parent are both red, so we rotate. This is a RIGHT-LEFT case, so we single rotate RIGHT.

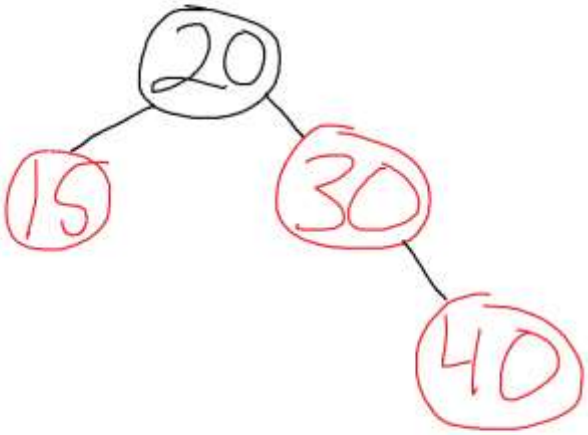


3. This is a RIGHT-RIGHT case and both parent and node is red. So we single rotate LEFT. Then SWAP the parent's color and grandparent's color.

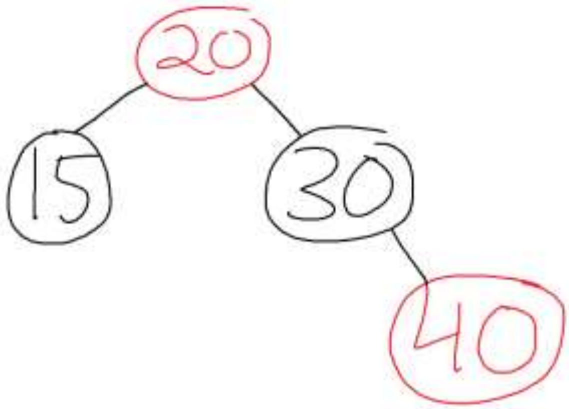


INSERT 40

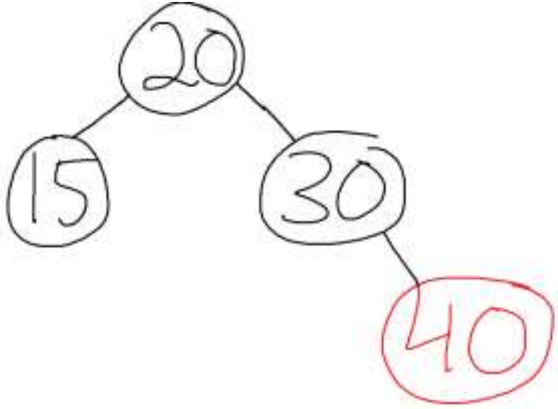
1. Newly inserted nodes will be **RED**



2. Do the following if the color of x’s parent is not **BLACK and** x is not the root.
   a) If x’s uncle is RED
   (i) Change the colour of parent and uncle as **BLACK**.
   (ii) Colour of a grandparent as **RED**.
   (iii) Change x = x’s grandparent, repeat steps 2 and 3 for new x.

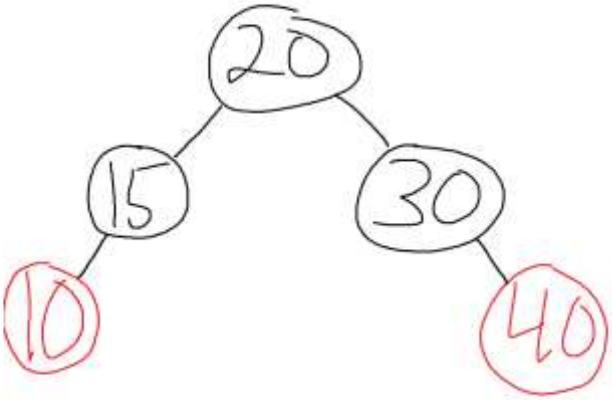


3. If x is the root, change the colour of x as **BLACK**



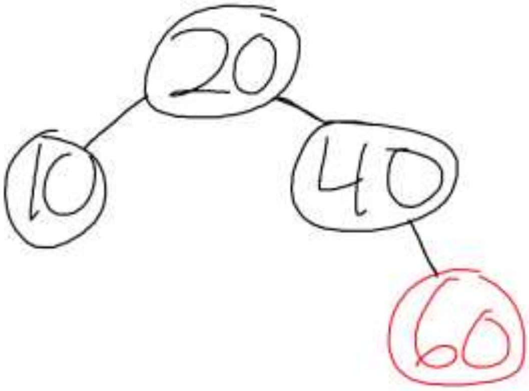
INSERT 10

1. Newly inserted nodes will be **RED**



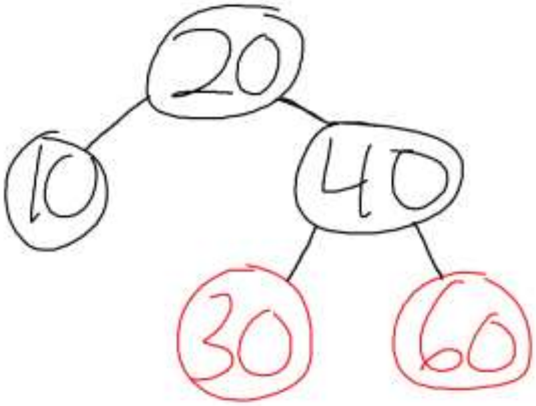
INSERT 60

1. Newly inserted nodes will be **RED**



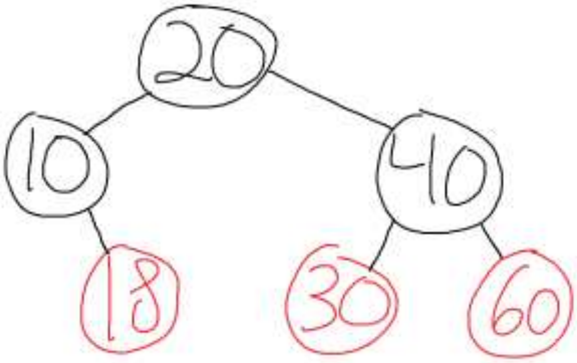
INSERT 30

1. Newly inserted nodes will be **RED**



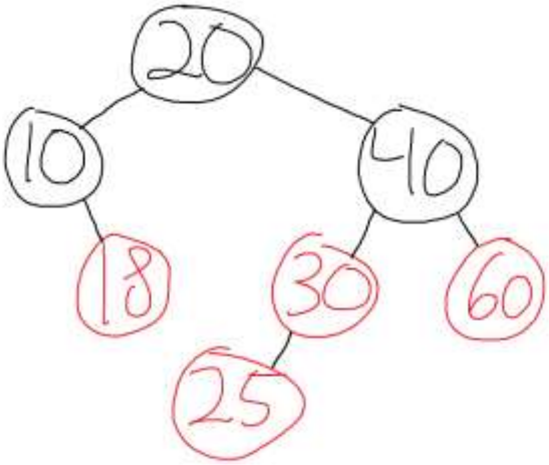
INSERT 18

1. Newly inserted nodes will be **RED**

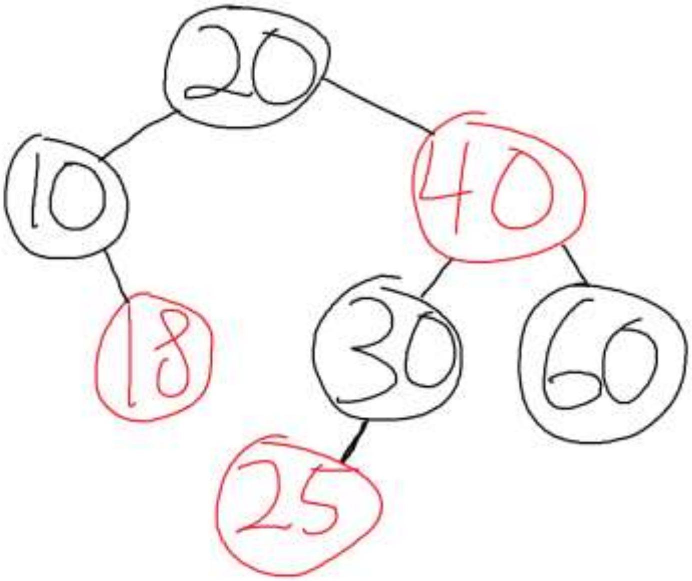


INSERT 25

1. Newly inserted nodes will be **RED**

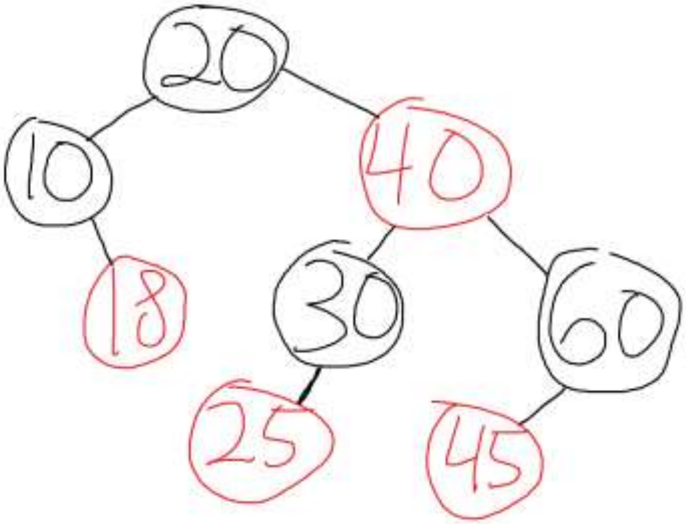


2. Do the following if the color of x's parent is not **BLACK and** x is not the root.
   a) If x's uncle is RED
   (i) Change the colour of parent and uncle as **BLACK**.
   (ii) Colour of a grandparent as **RED**.
   (iii) Change x = x's grandparent, repeat steps 2 and 3 for new x.



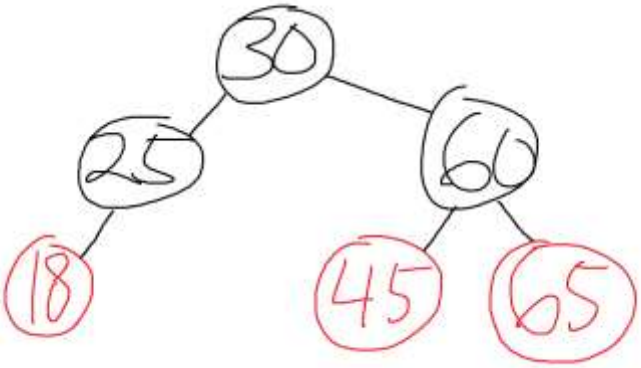
INSERT 45

1. Newly inserted nodes will be **RED**



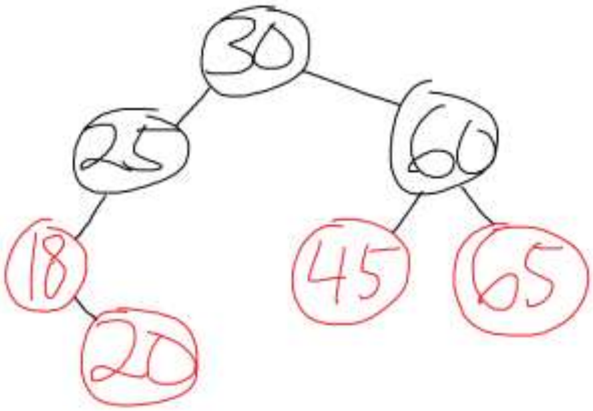
INSERT 65

1. Newly inserted nodes will be **RED**

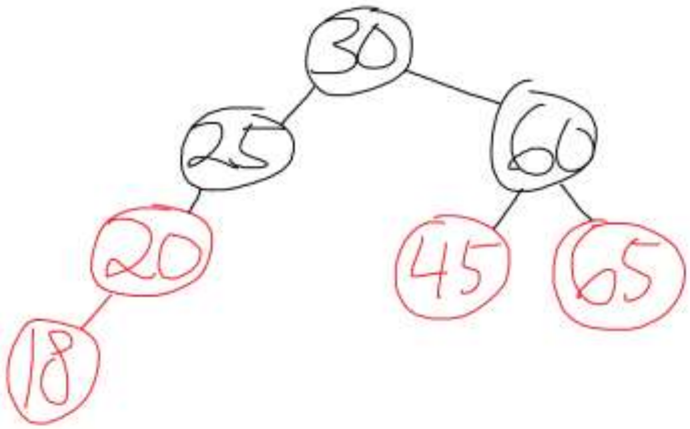


INSERT 20

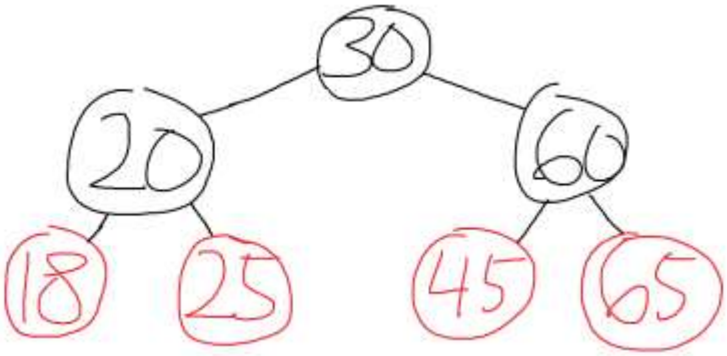
1. Newly inserted nodes will be **RED**



2. Both node and parent are red, so we rotate. This is a LEFT-RIGHT case, we single rotate LEFT.

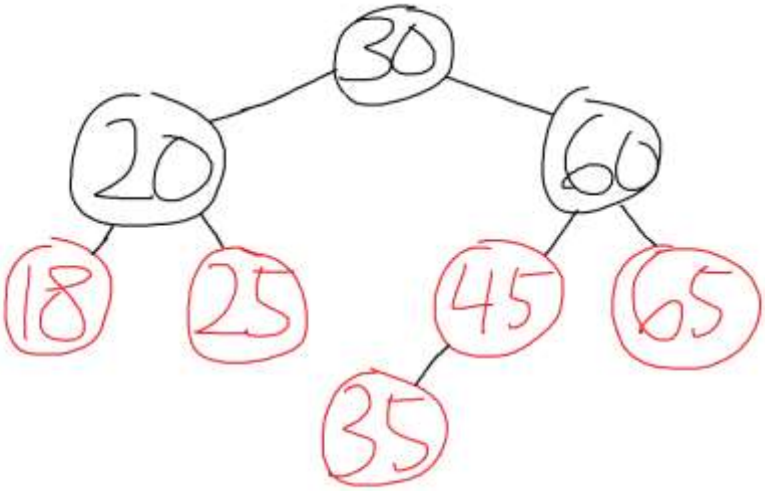


3. Both node and parent are red, so we rotate. This is a LEFT-LEFT case, we single rotate RIGHT. Then we SWAP the parent’s color and grandparent’s color.

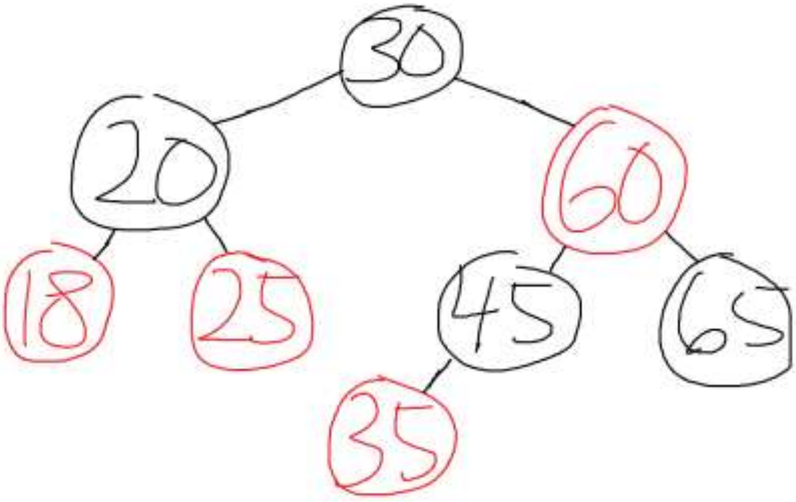


INSERT 35

1. Newly inserted nodes will be **RED**



2. Do the following if the color of x's parent is not **BLACK and** x is not the root.
   a) If x's uncle is RED
   (i) Change the colour of parent and uncle as **BLACK**.
   (ii) Colour of a grandparent as **RED**.
   (iii) Change x = x's grandparent, repeat steps 2 and 3 for new x.


30
20
60
18
25
45
65
35