4.7 RED BLACK TREES

A red - black tree is a data structure which is a type of self-balancing binary search tree with the following colouring properties.

R1 : A node is either red or black

R2 : The root is black

R3 : All NULL pointer are black

R4 : Children of the red node must be black.

R5 : Every path from a given node to any of its NULL pointer contains the same number of black nodes.

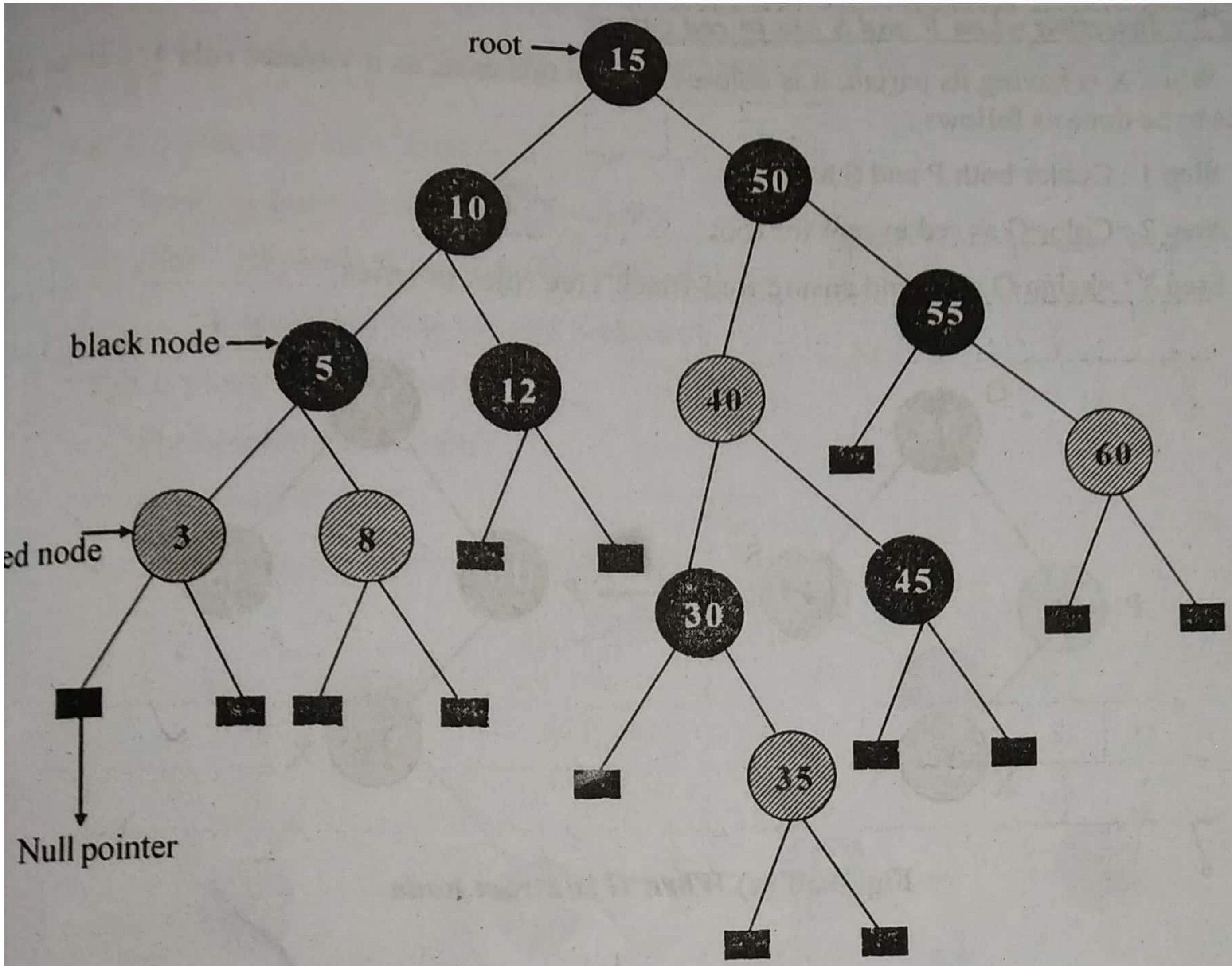


Fig. 4.29 *Red Black Tree*

7.1 Insertion on Red - Black Trees

7.1.1 Bottom up Insertion

Inserting a new element in a red-black tree follows binary search tree property and it should coloured as red.

Let X → new element to be inserted.

P → parent of X

G → grandparent of X

S → Sibling of P

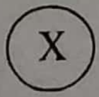
Zag → left rotation

Zig → right rotation

Various cases of insertion in red-black - Tree

Case 1 : Insertion in empty red black tree

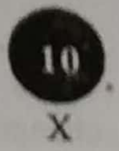
When X is inserted as a root node, as per rule 2 it is coloured black.



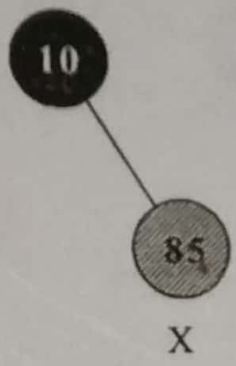
Example 1 :

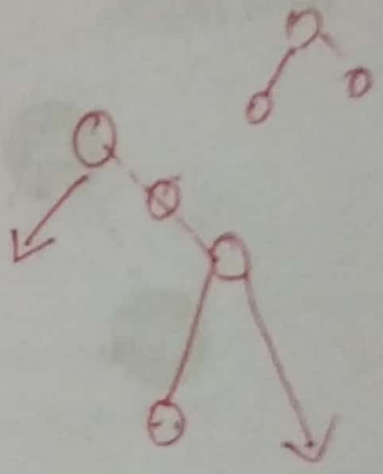
Insert 10, 85, 15, 70, 20, 60, 30, 50, 65, 80, 90, 40, 5, 55 into an empty Red-Black Tree.

Insert 10 :

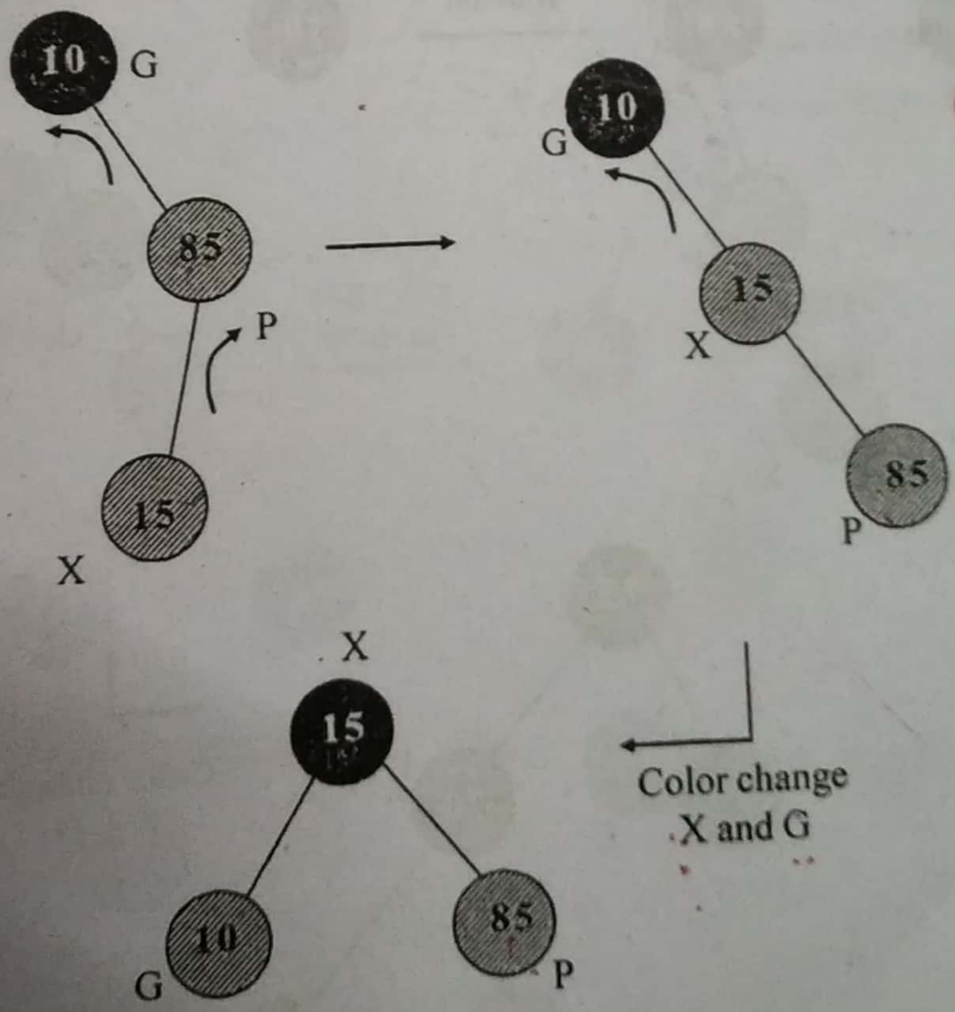


Insert 85 :

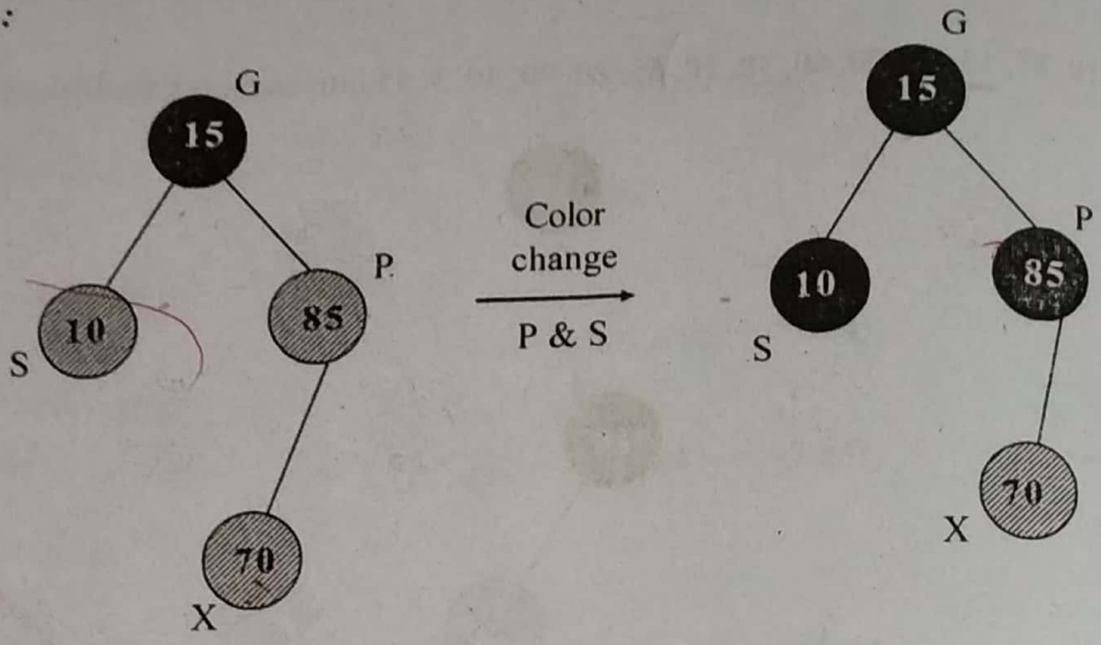




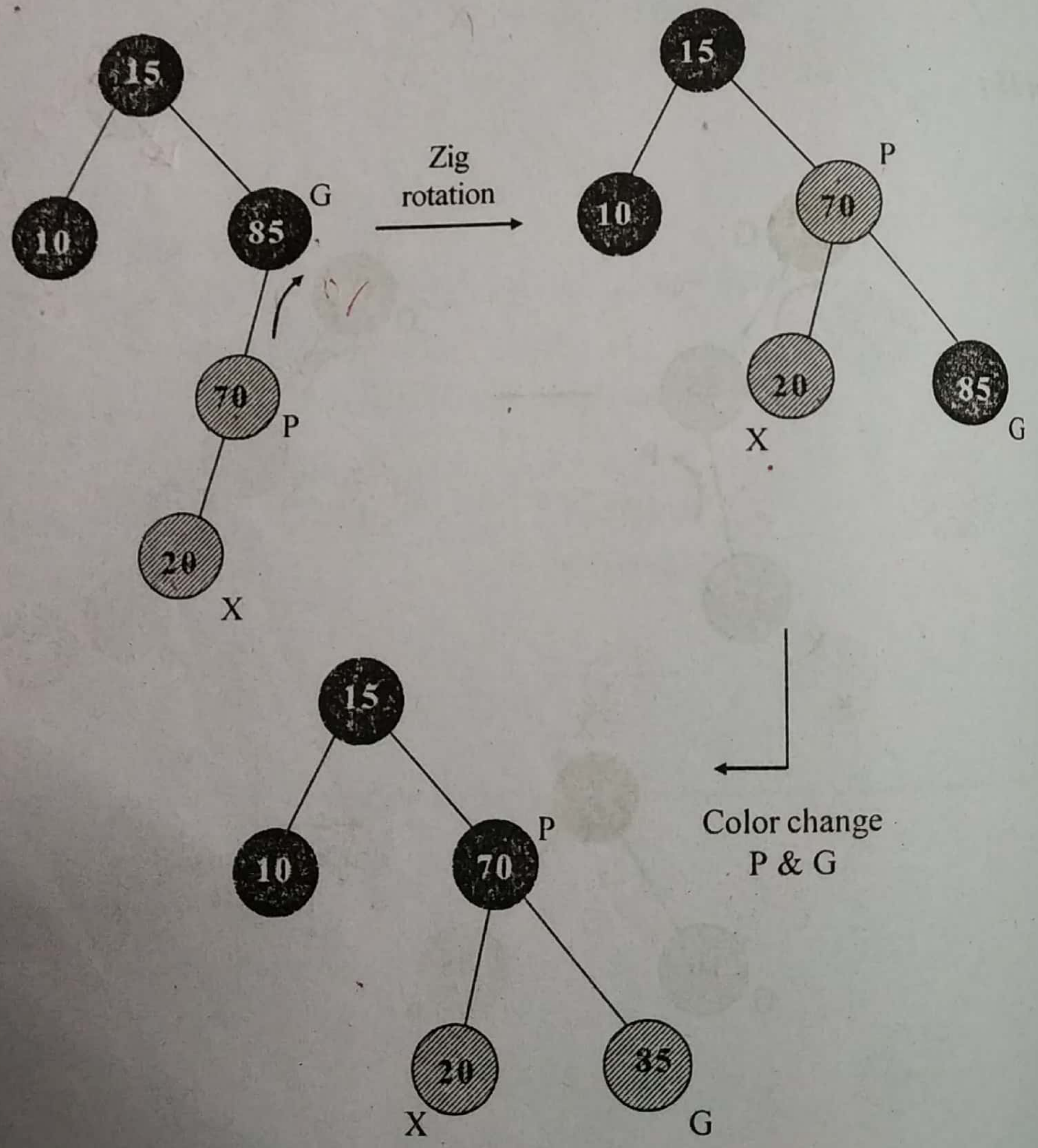
Insert 15 :

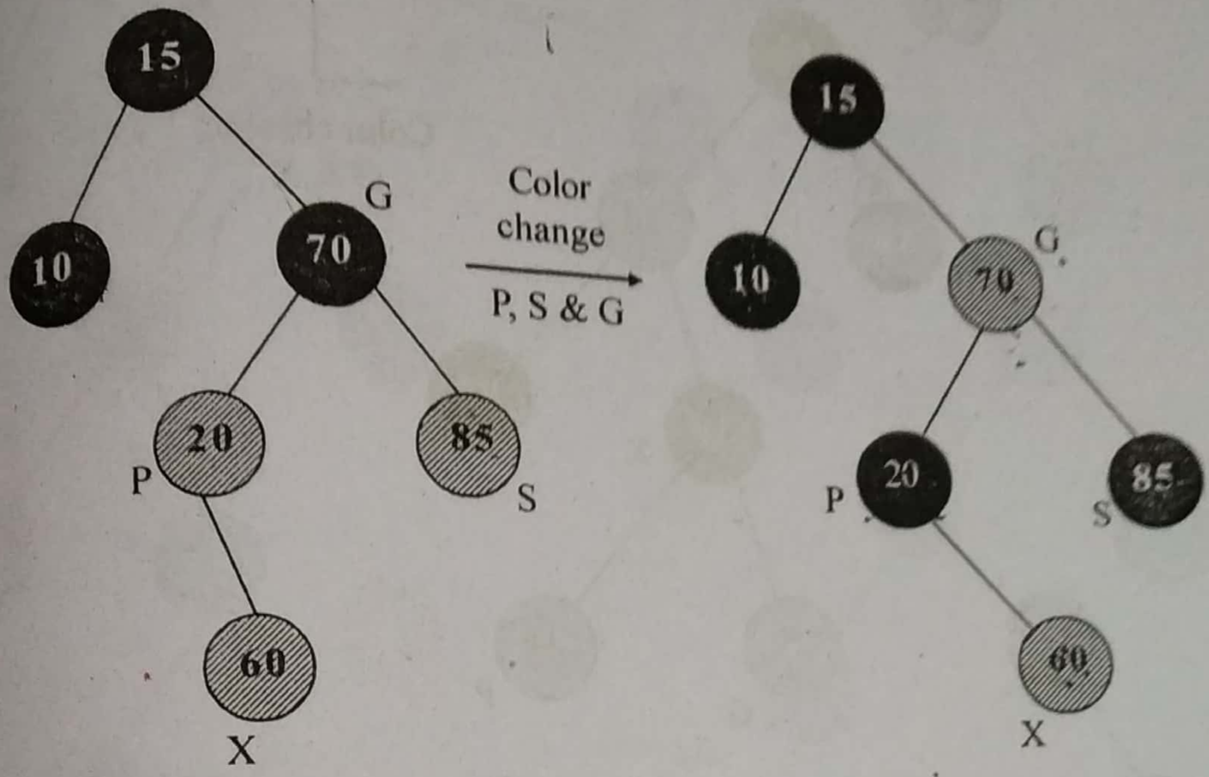


Insert 70 :

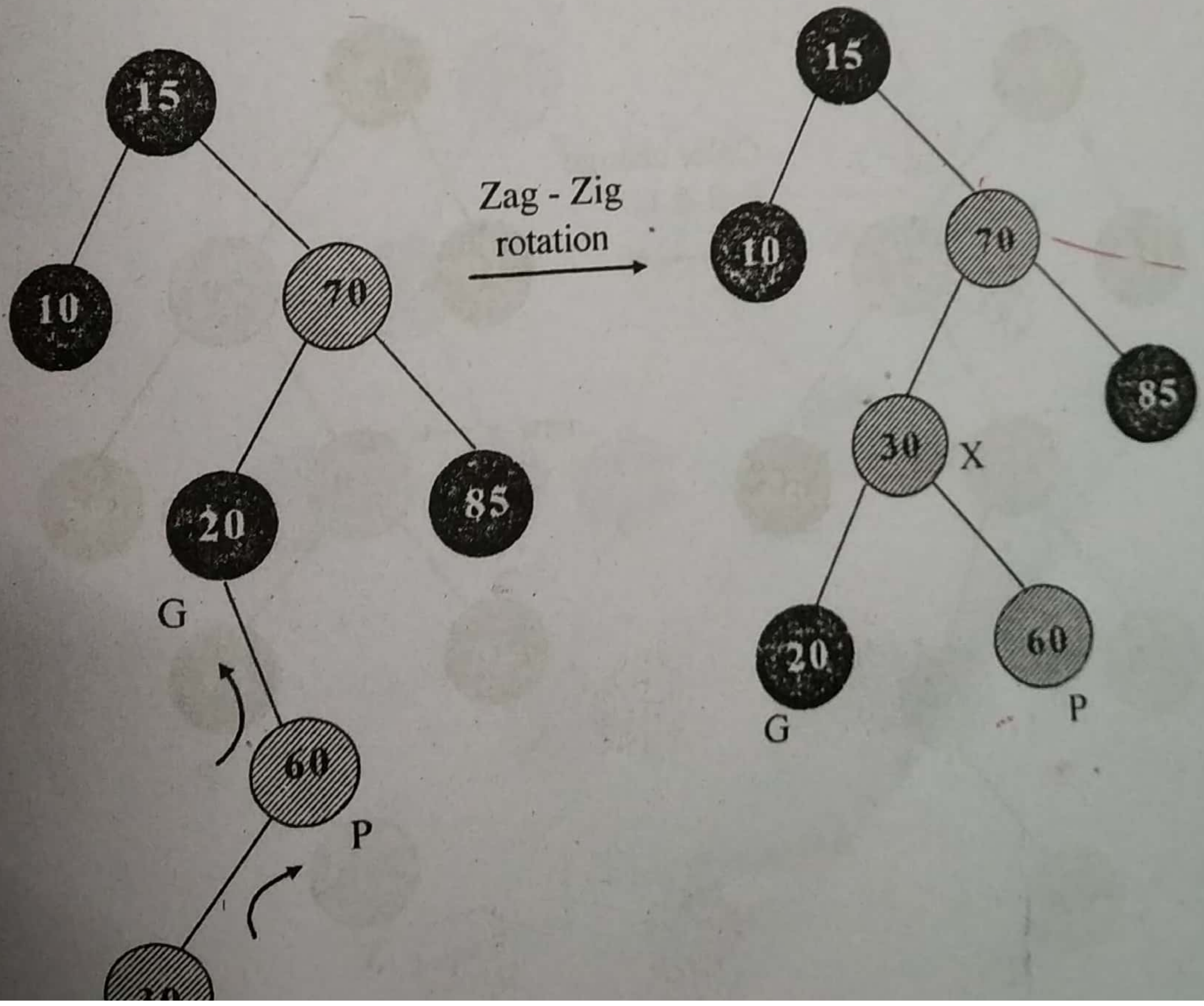


Insert 20 :

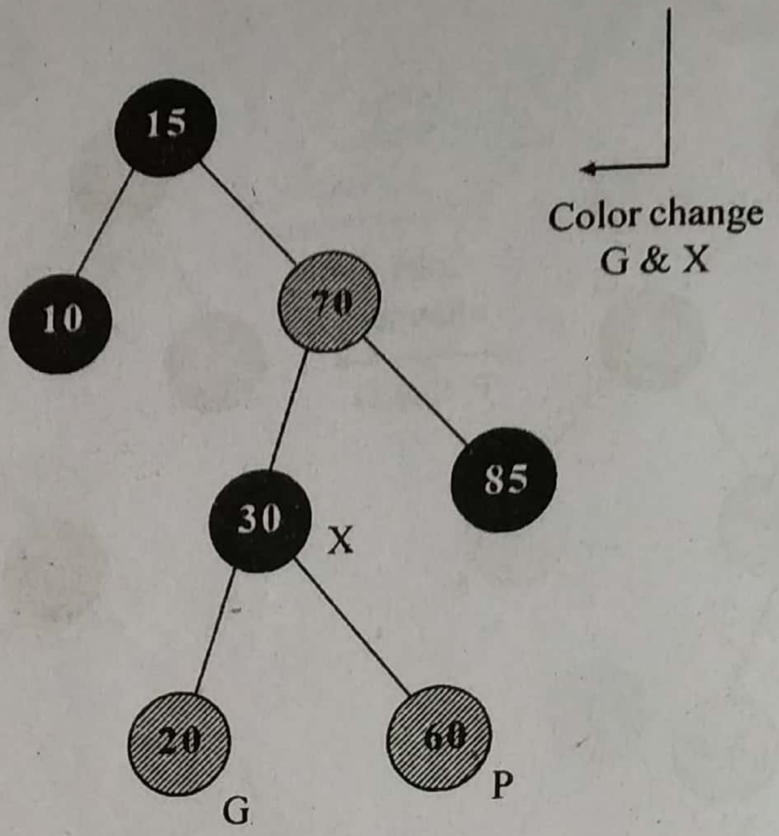




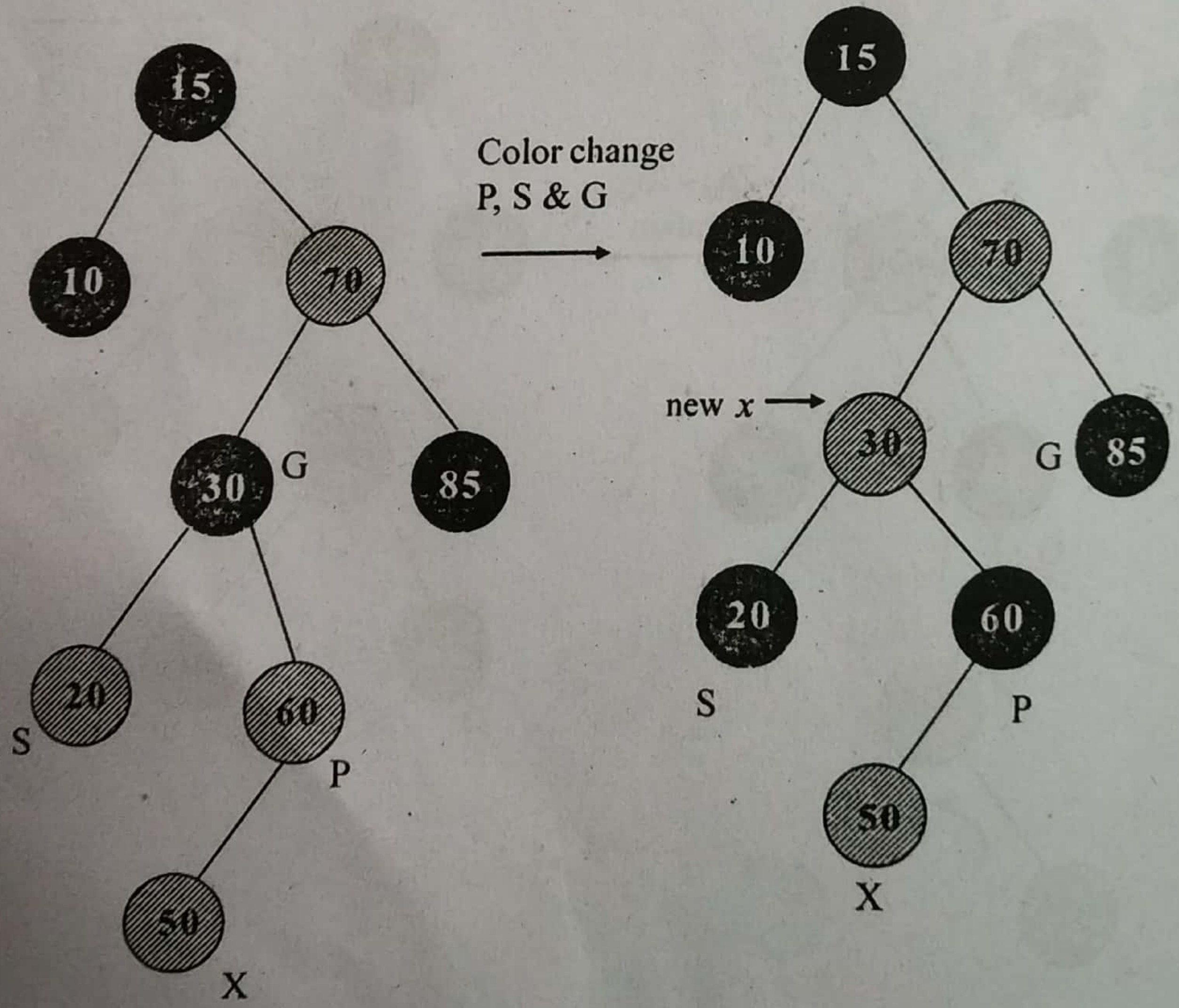
Insert 30 :

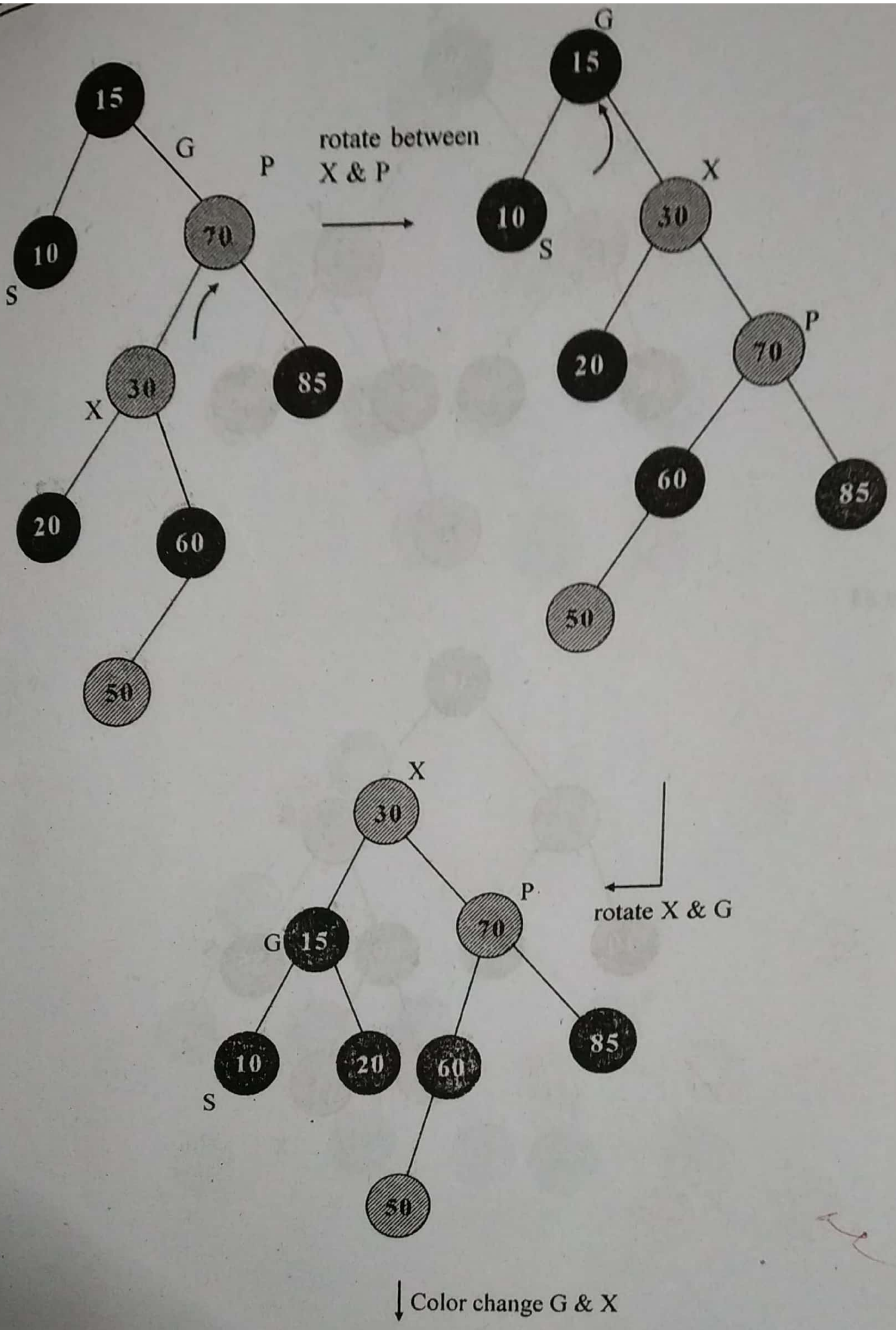


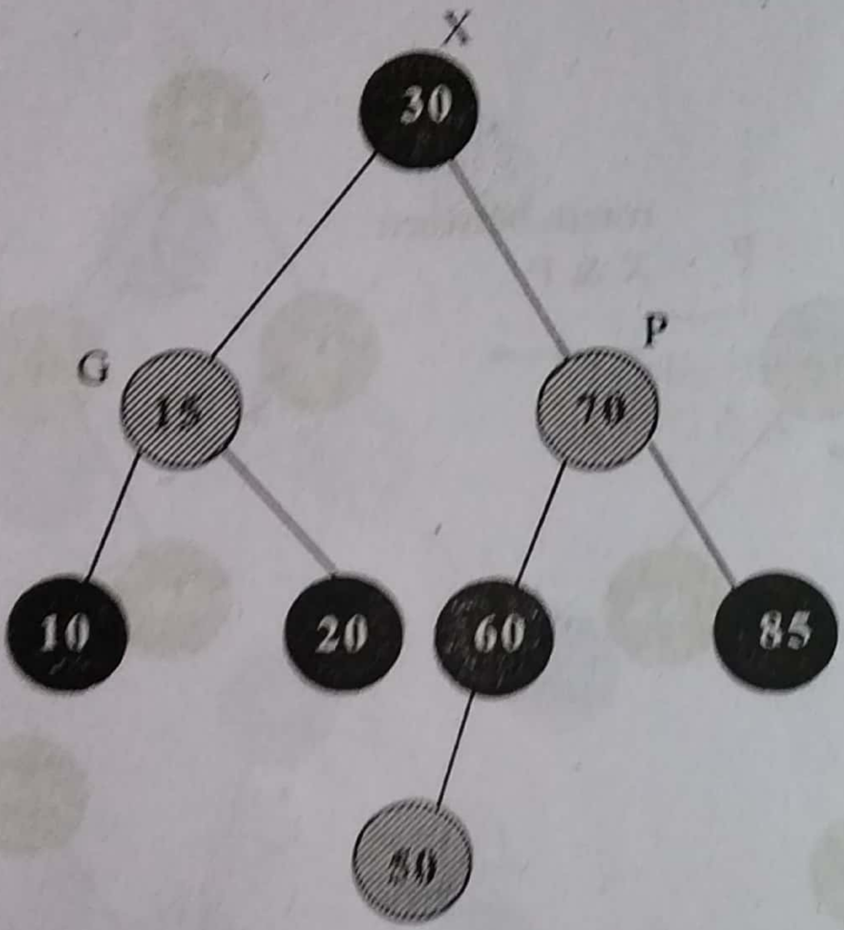
Scanned by CamScanner


15
10
70
30
X
85
20
G
60
P
Color change
G & X

Insert 50 :


15
10
70
30
G
85
20
S
60
P
50
X
Color change
P, S & G
15
10
70
new x
30
G
85
20
S
60
P
50
X


15
G
P
10
S
70
30
X
85
20
60
50
rotate between X & P
G
15
10
S
X
30
20
P
70
60
85
50
X
30
P
G 15
70
10
S
20
60
85
50
rotate X & G
Color change G & X



Scanned by CamScanner