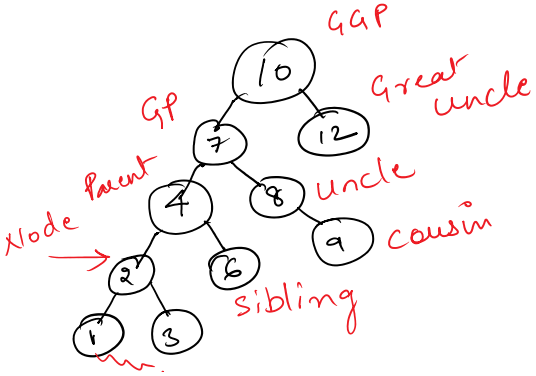
Red - Black Trees

AVL Trees
↳ strictly balanced Trees

Loosely balanced
Self balancing BST

- Every node is Red or Black
- Root is always BLACK
- Every new node when it is inserted will be RED
- If a node is RED, it cannot have a red parent. Each RED node should have a BLACK parent. No two adjacent node can be RED. → RED CONDITION
- Every leaf node which is NULL is BLACK
- Every path from a node to any of its null descendants has same number of BLACK NODES.

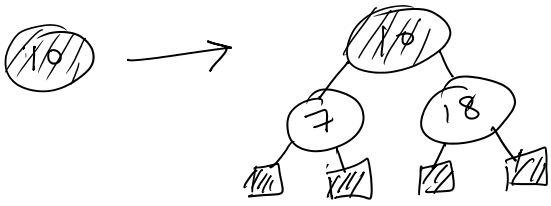


 → BLACK

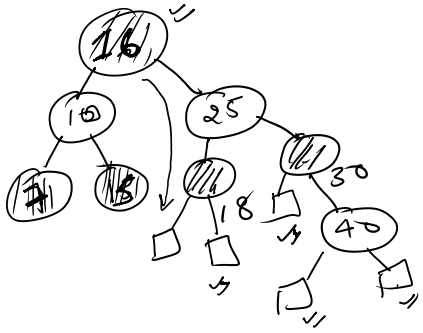
 → RED

Children

#g 10, 18, 7...



#g



valid R-B tree.

Insertion in R-B trees.

Need to recolor/rotate or both whenever the insertion violates the R-B trees properties

2 cases – Node which is being inserted has a RED Parent

Uncle is Red

Parent is Red

GP will be black

RECOLOR

1. Set GP → Red
   - P → Black
   - U → Black
2. After GP → Red
   It might disturb
   Check with color GGP

GP GP

Uncle is Black or NULL

Parent is Red

GP will be Black

2 possible orientations

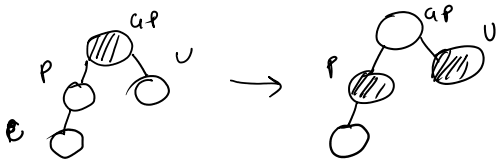
OR1 ZIG-ZIG

1. Promote P
2. Recolor
   GP → R
   P → B

OR2 ZIG-ZAG

1. Promote C
   Promote C
2. Recolor
   GP → R
   C → B

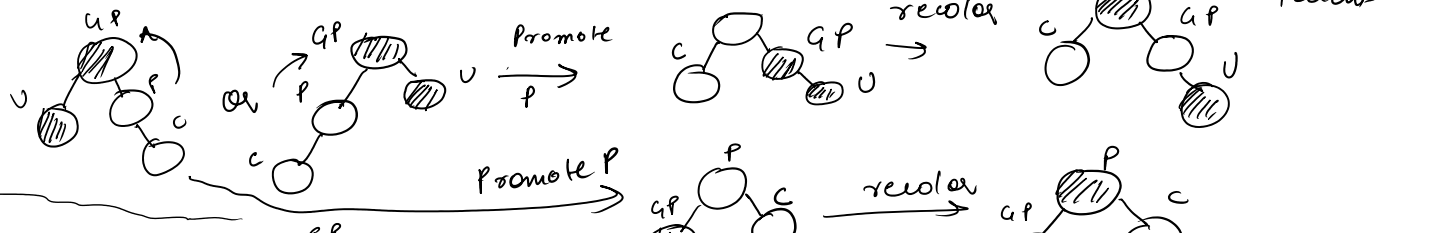
Check with color GGP



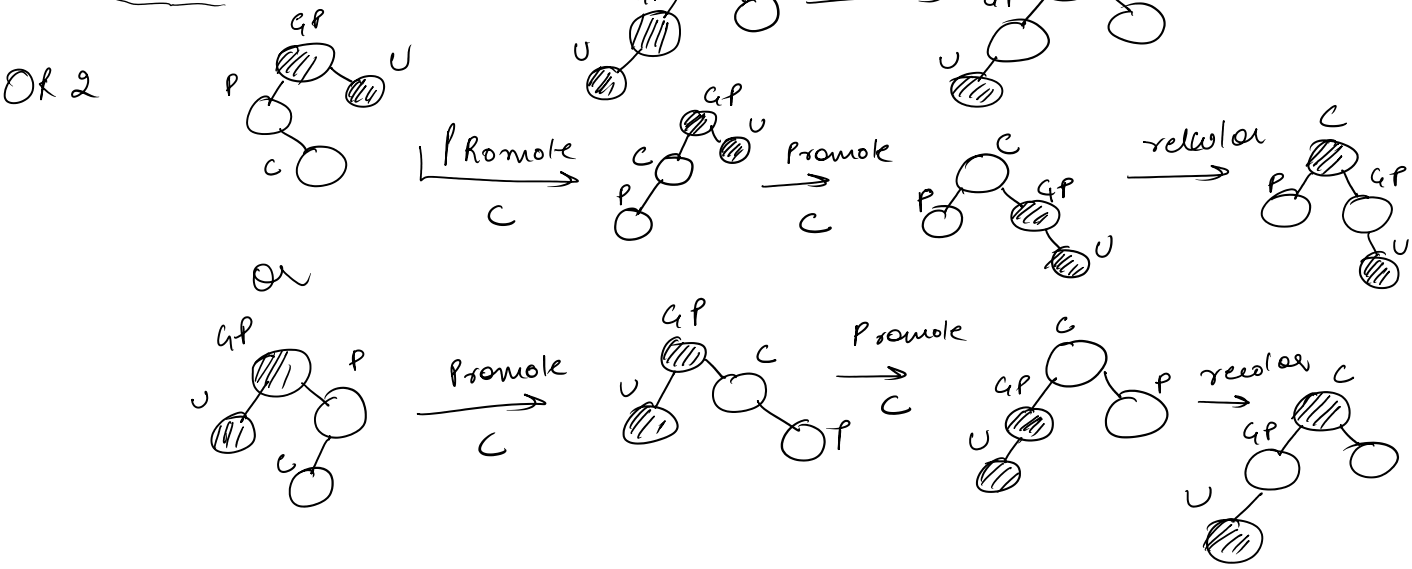
C → B

In these cases, changes are local, so no need for any further changes in the tree

OR 1 ZIG-ZIG – parent & child are both same orientation from their respective parent

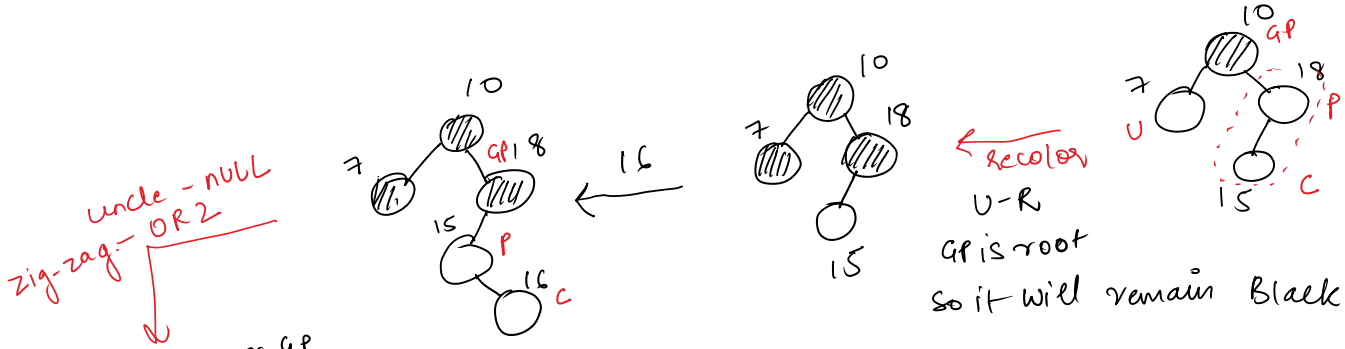


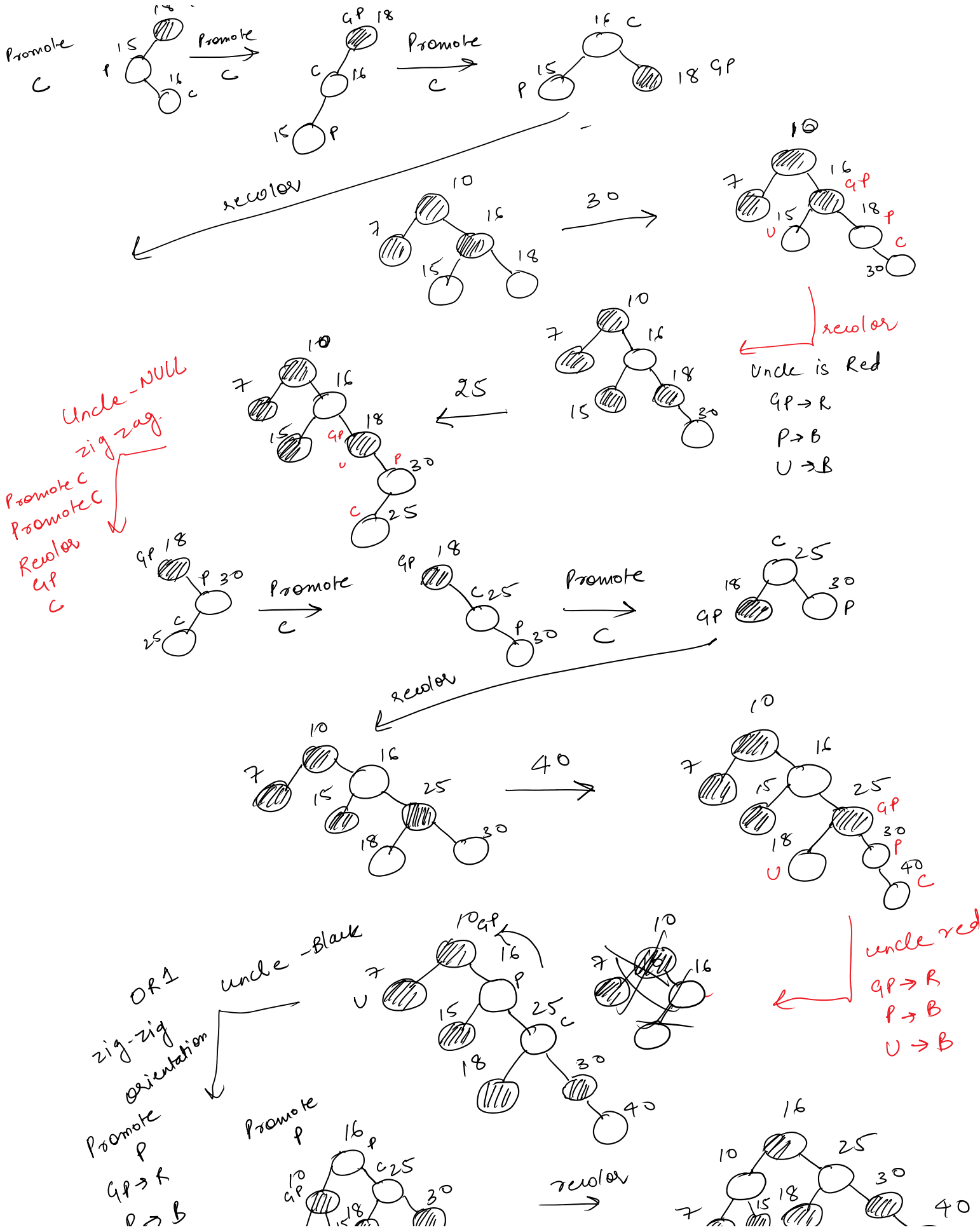
OR 2



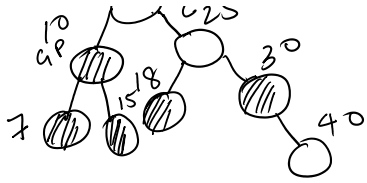
#3 10, 18, 7, 15, 16, 30, 25, 40

B OR




Promote
C
Promote
C
Promote
C
recolor
30
recolor
Uncle is Red
GP → R
P → B
U → B
25
Uncle - NULL
zig-zag
Promote C
Promote C
Recolor
GP
C
Promote
C
Promote
C
40
recolor
uncle red
GP → R
P → B
U → B
OR
zig-zig
orientation
uncle - Black
Promote
P
GP → R
P → B
Promote
P
recolor

gp → r
p → b



recolor →