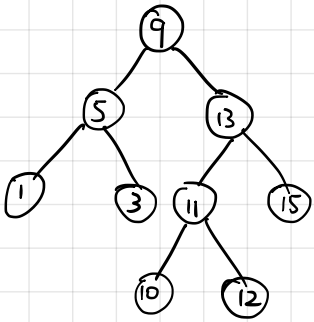
二叉查找树 (BST) 特点

1. All nodes of Left Child < root
2. All nodes of right child > root
3. Left & Right Child tree also a Binary Tree.



O(logn) - Search.

Red Black Tree

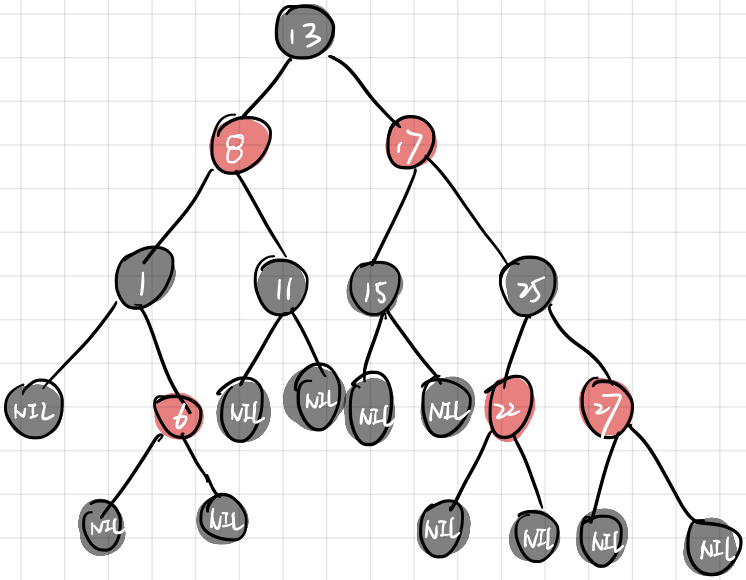
Rule: ① Node is Red / Black.

② Root is Black.

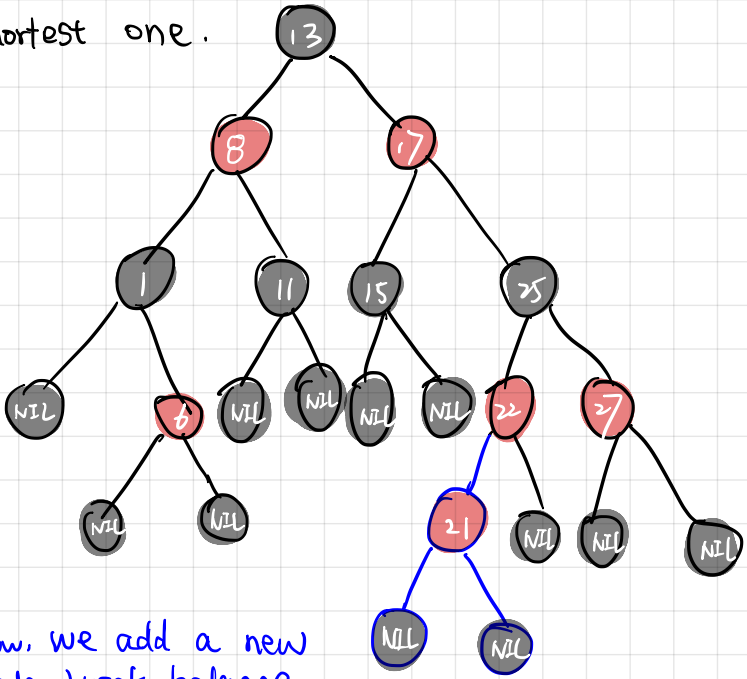
③ Every leaf is a Black NIL NODE.

④ Every Red node's two child MUST be Black.
(No two Consecutive red Node)

⑤ All paths from any nodes to every leaves Contain same number of Black Node



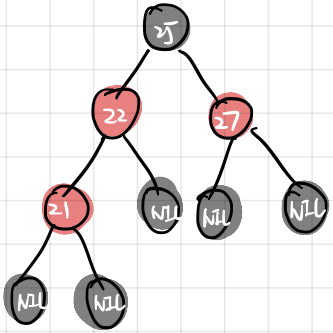
note: the longest path from root to leaves can not exceed the shortest one.



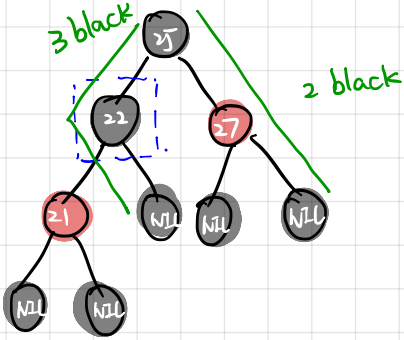
Now, we add a new node, break balance

2 ways to solve : 「Change Color」 & 「Rotation」

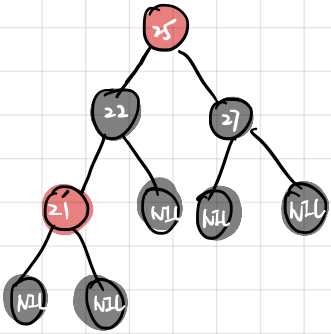
① Change Color



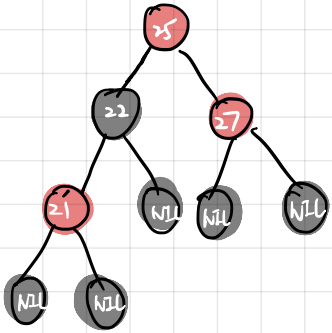
fail # 4



fail # 5

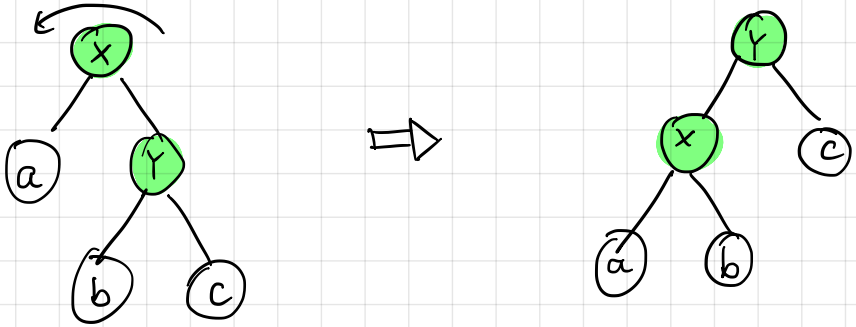


fail #4

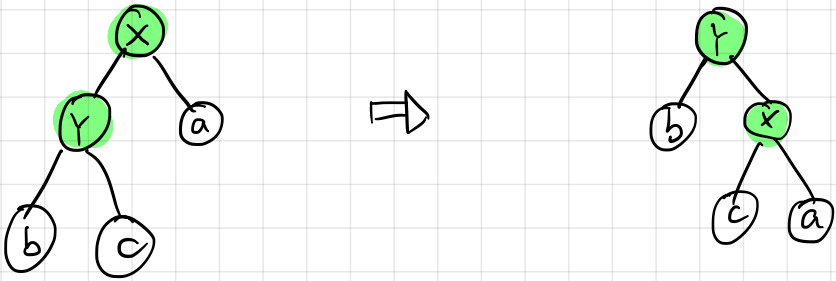


② Rotation.

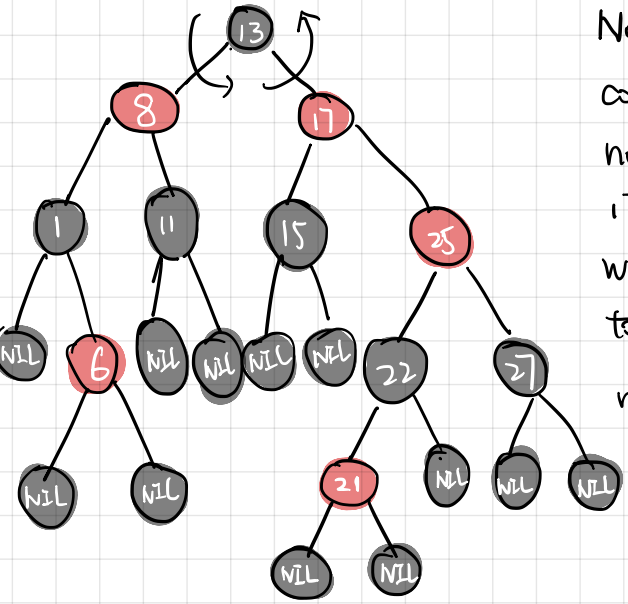
Left Rotation



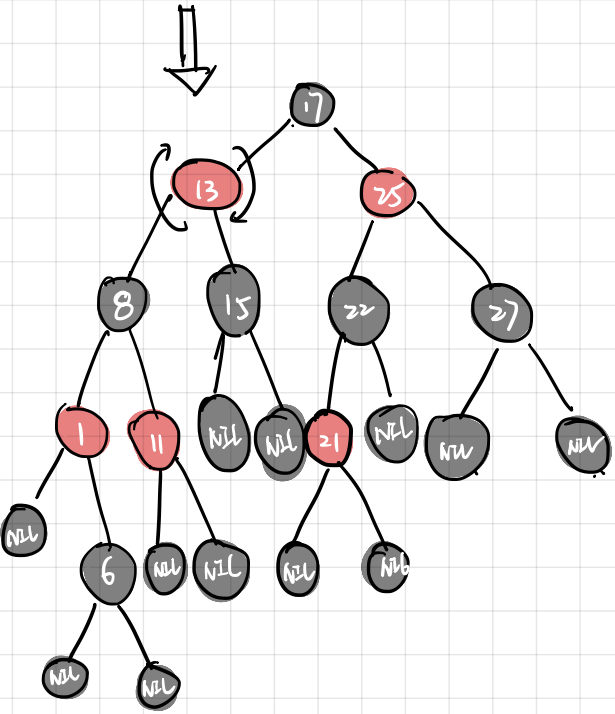
Right Rotation



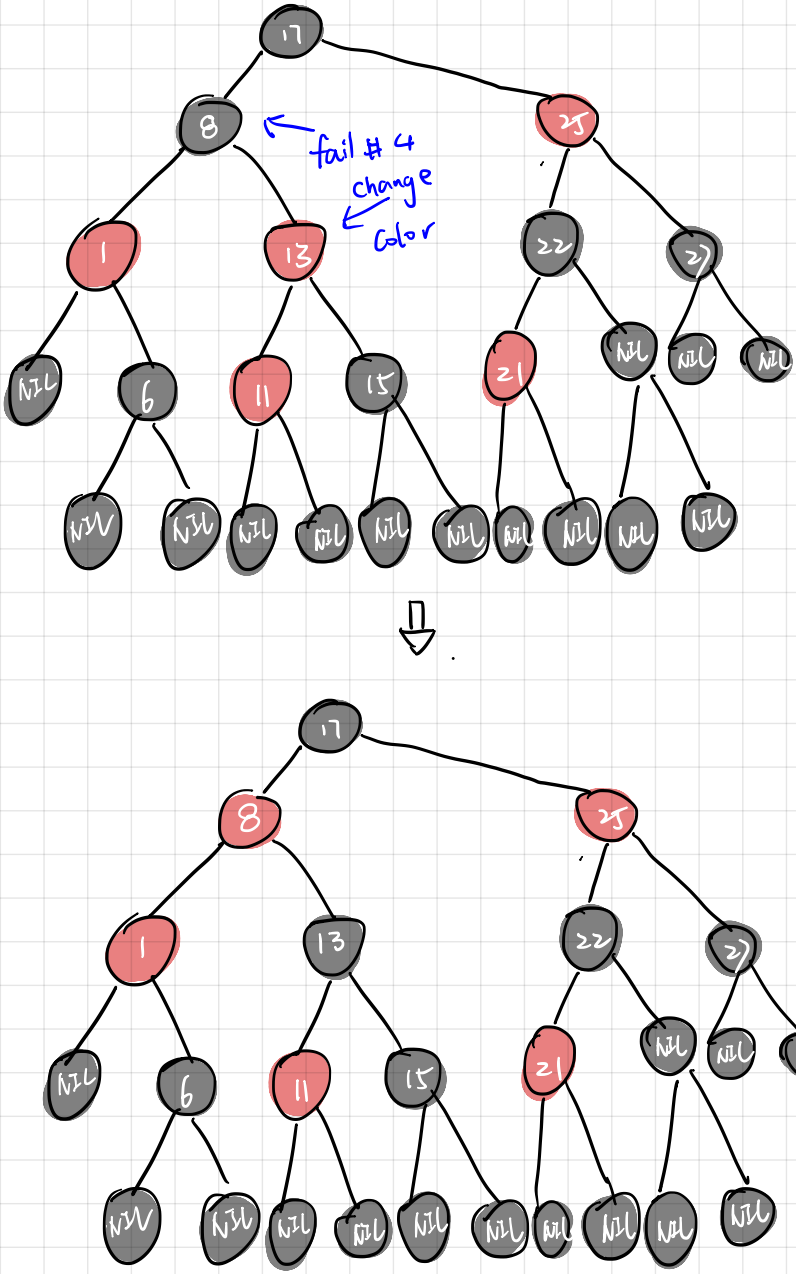
Example.



Now, Node 17 & 25 are two consecutive red nodes, it is not suitable to change node 17 to black. (break #4), nor we can not change node 13 to red (break #2). So we need rotation now.



fail # 5.


17
8
25
fail # 4
change
Color
1
13
22
27
NIL
6
11
15
21
17
8
25
1
13
22
27
6
11
15
21