CSE 101 3-7-25

Rotations:

note:

$$Keys(\alpha) \le Key[y] \le Keys(\beta) \le Key[x] \le Keys(\gamma)$$

So Rotations preserve BST properties.

- Summarize RightRotate(T, x):

p $\begin{cases} x.left = y.right \\ y.right.Parent = x \end{cases}$

Parent $\begin{cases} y.Parent = x.Parent \\ x.Parent.(left\ or\ right) = y \end{cases}$

x, y $\begin{cases} y.right = x \\ x.Parent = y \end{cases}$

13.3 Insertion

2 cases

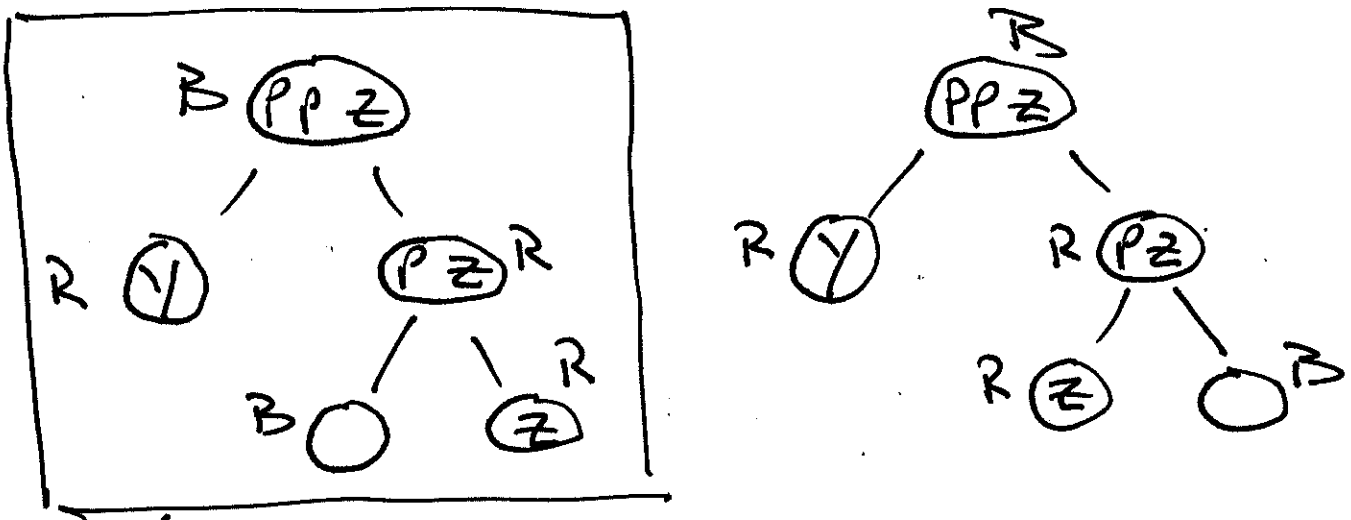
- z.parent is left child of Parent: cases 1, 2, 3

  let y be z.parent right sibling, i.e. z's uncle.

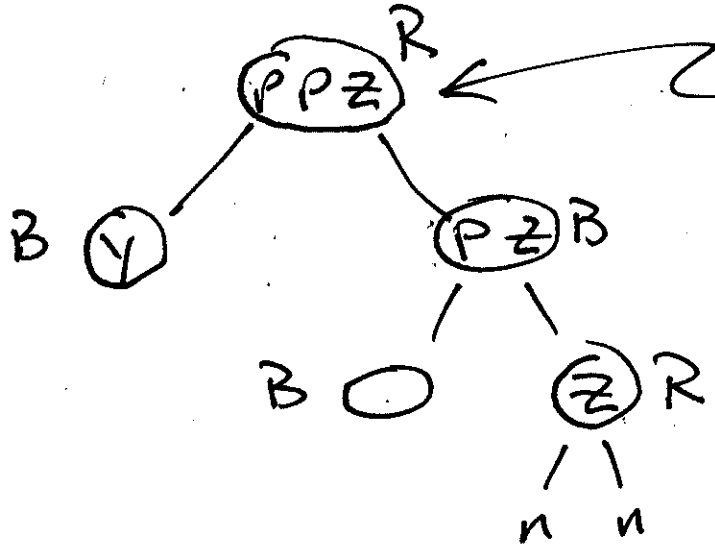
- z.parent is right child of its Parent: cases 4, 5, 6

  let y be z.parent left sibling i.e. z uncle.

Case 4 : y is Red



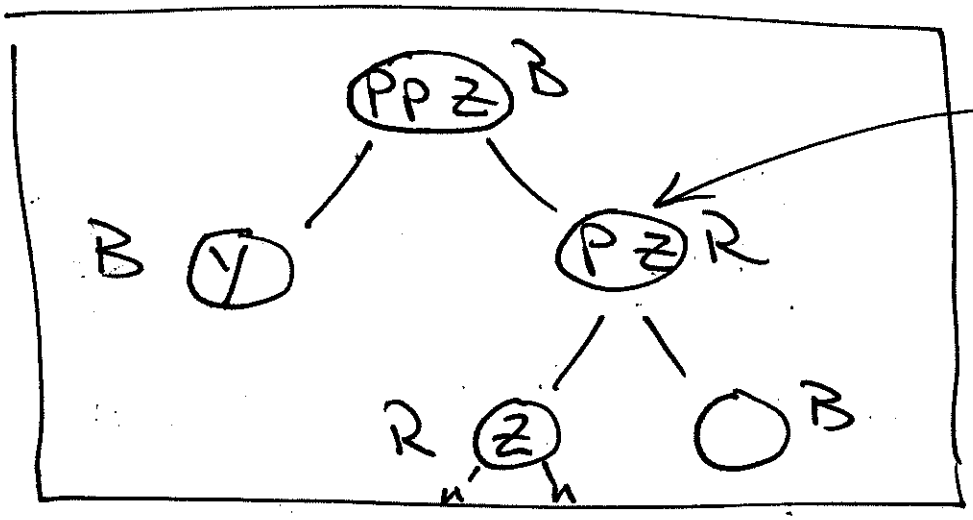
fix colors of y, Pz, PPz. let local variable z 'climb up' to PPz.



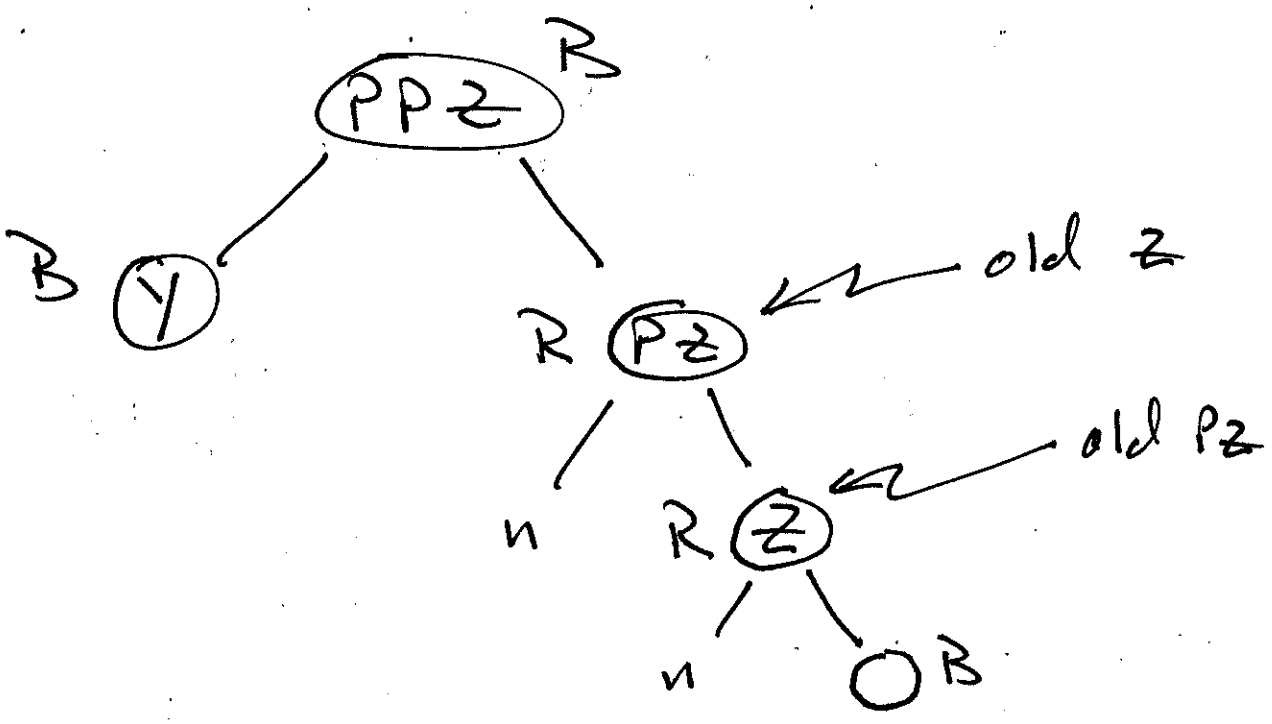
new z

may execute another iteration of while loop.

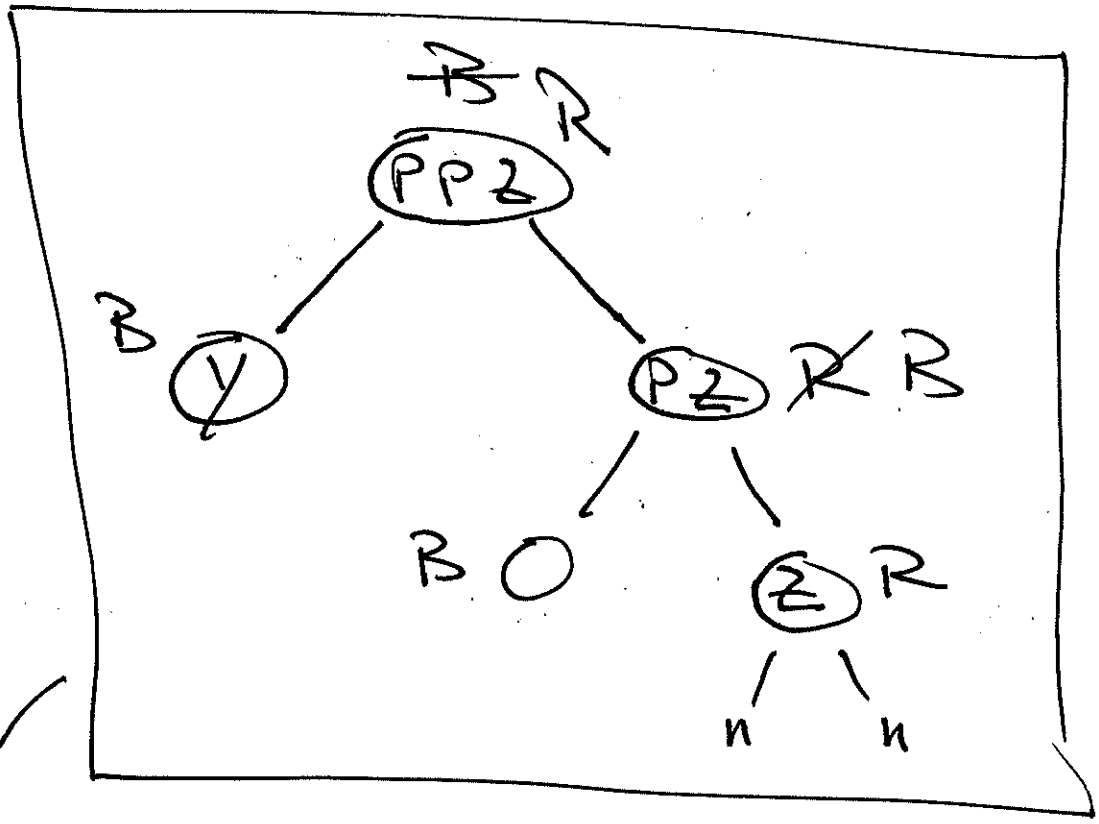
Case 5: y is Black, z == z.Parent.left



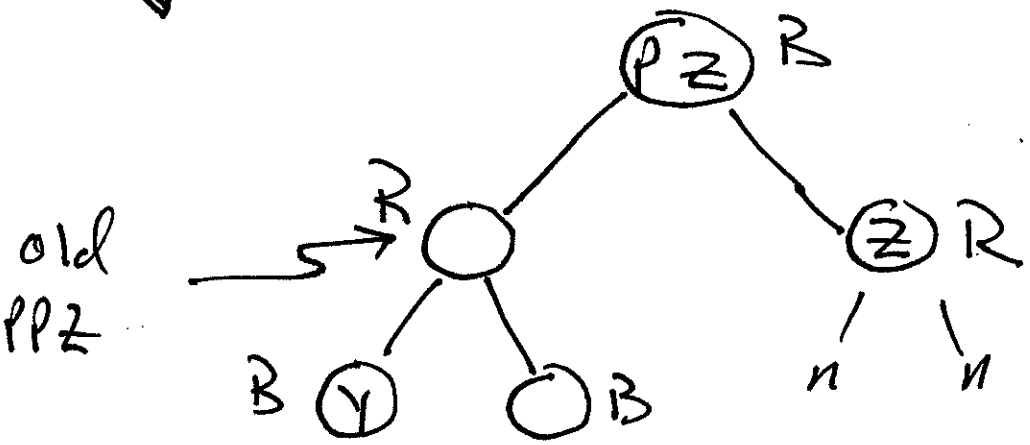
convert to case 6 (z == z.Parent.right)



Case 6: y is Black, z == z.parent.right



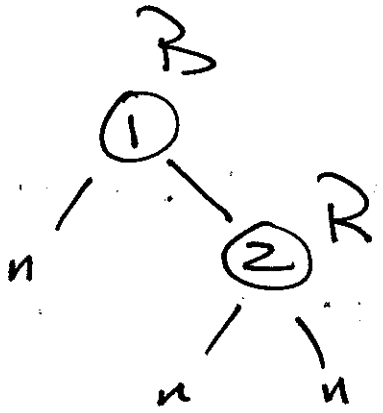
color Pz Black, PPz Red, then left Rotate about PPz.



Ex Insert 1, 2, 3, 4, 5 into an initially empty RBT.

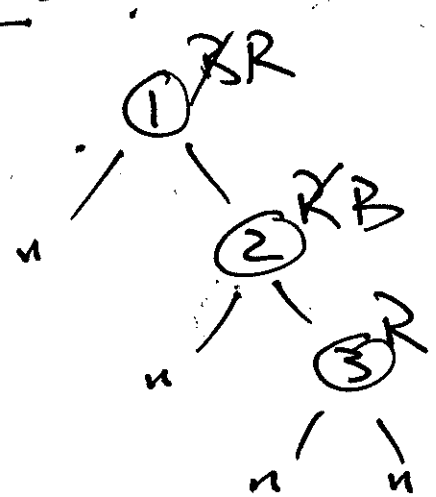
Insert 1:

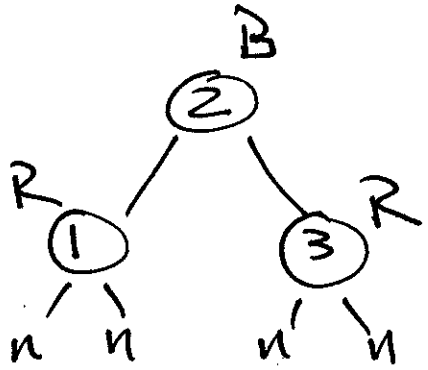
Insert 2:



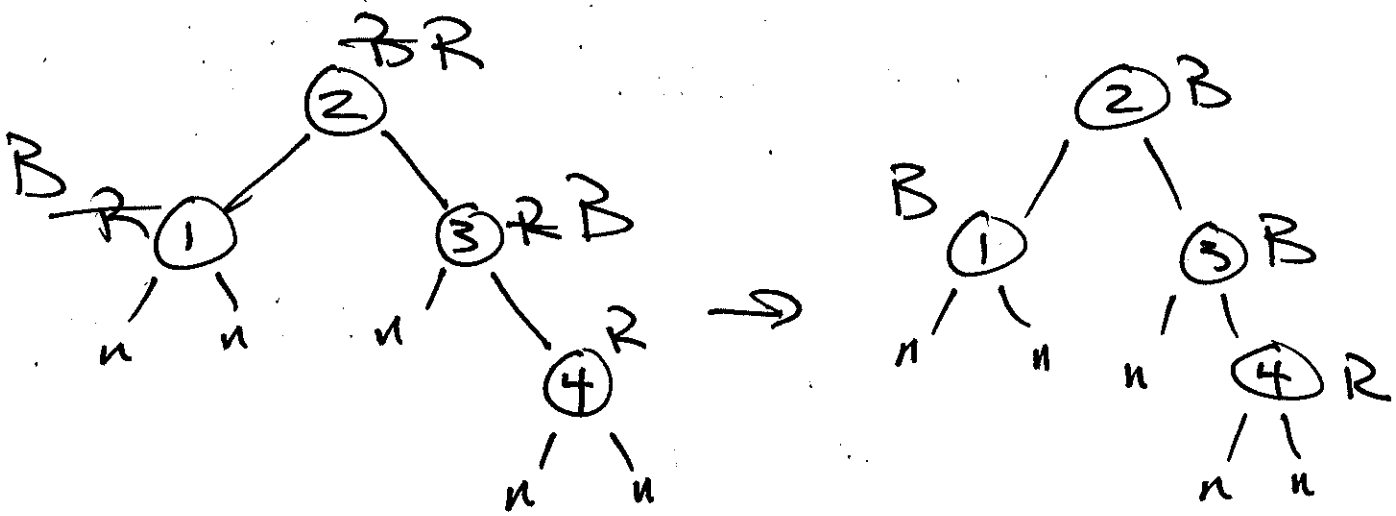
Insert 3:

Case 6

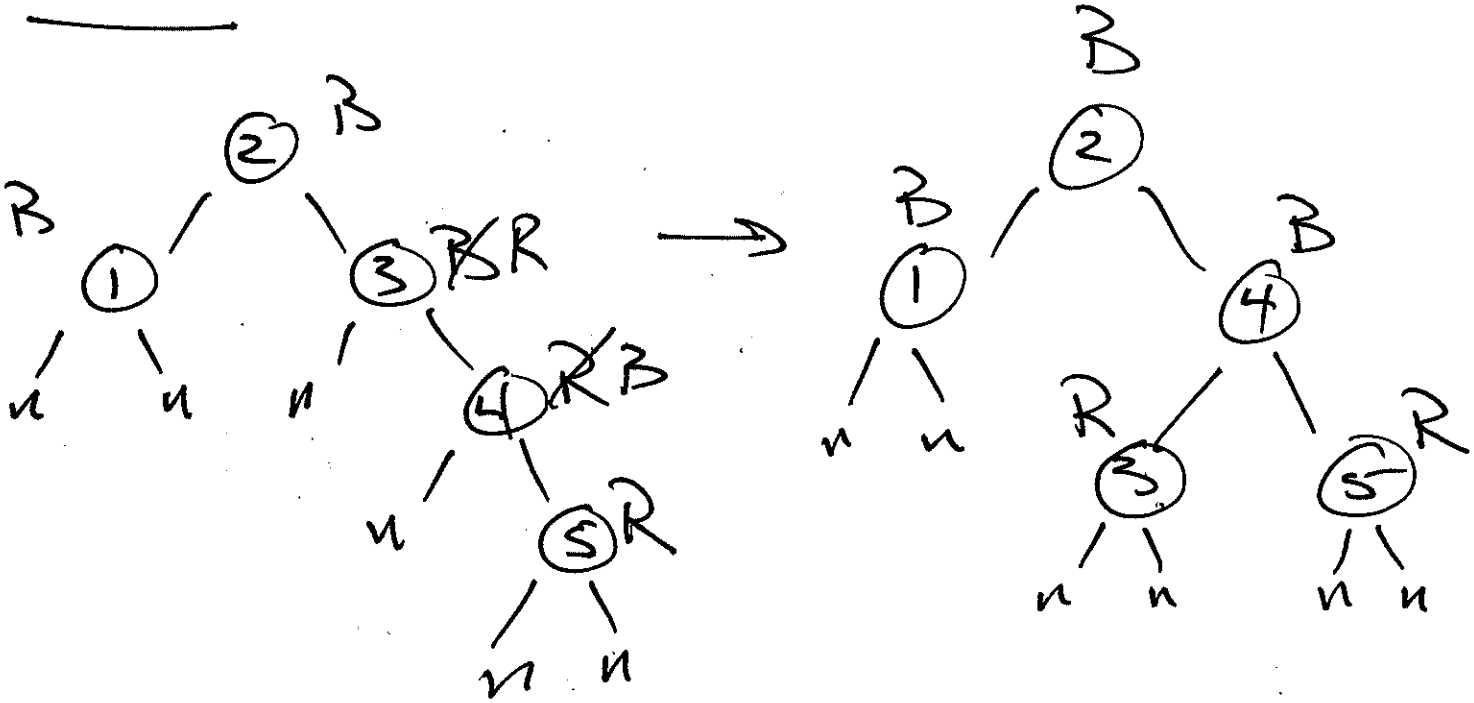




Insert 4 (case 4)



Insert 5



Case 6