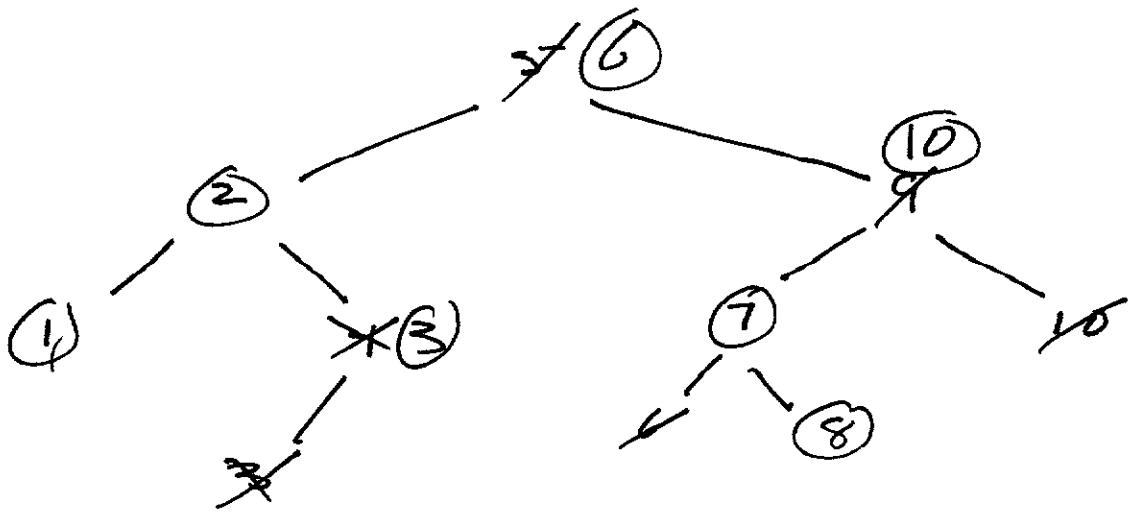
CSR 101 5-20-21

Quiz 4 : solutions Posted (fix error!)

Past : one more day : to Sat. 10 PM

Pa6 : BST Dictionary

Problem 1 on quiz : 5, 9, 7, 2, 6, 4, 8, 3, 1, 10



Delete: 5, 9, 4

c] Pre: 6, 2, 1, 3, 10, 7, 8 ← Correct

d] Post: 1, 5, 2, 8, 7, 10, 6 ← "

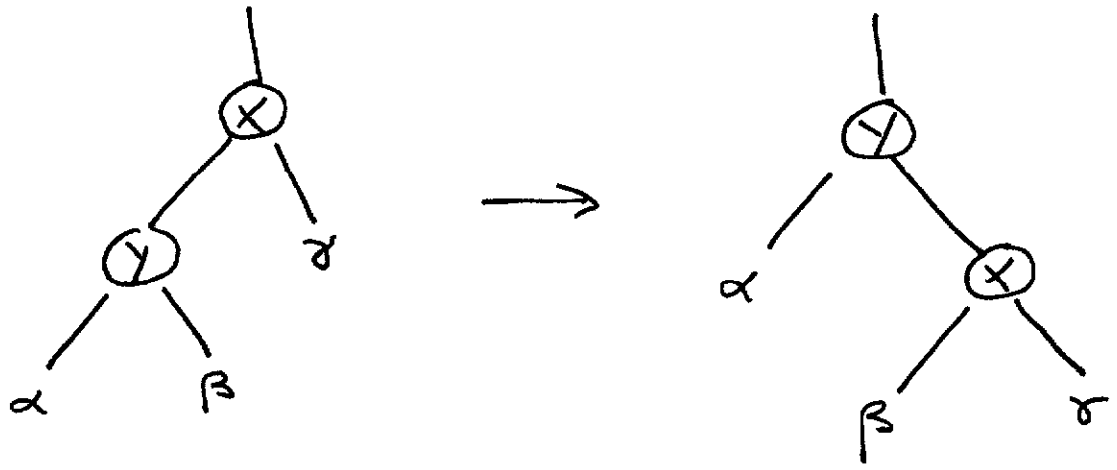
Pa6 comments:

Forward & Backward iterations

Analogy:

List		Dictionary
moveFront()	——	beginForward()
moveBack()	——	beginReverse()
get()	——	currentKey() / currentVal()
next()	——	next()
prev()	——	prev()

RightRotate(T, x)



Summary:

- x.left = y.right
- y.right.parent = x

} switch β

- y.parent = x.parent
- x.parent.{left / right} = y

} reset parent

- y.right = x
- x.parent = y

} switch x, y

Sec. 13.3 in CLRS

Recal from BST: TreeInsert(T, z)

which RBT properties could be violated?

✓ (1) all nodes Red or Black
✓ (2) all nil children (leaves) are Black
✓ (3) Root is Black
(4) Red nodes have only Black children.
✓ (5) # Black nodes on any n^2 descending Paths from a node to a leaf are the same

strategy: color new node z Red

This preserves RBT (5), Then color Root Black, fixing RBT (3).

see

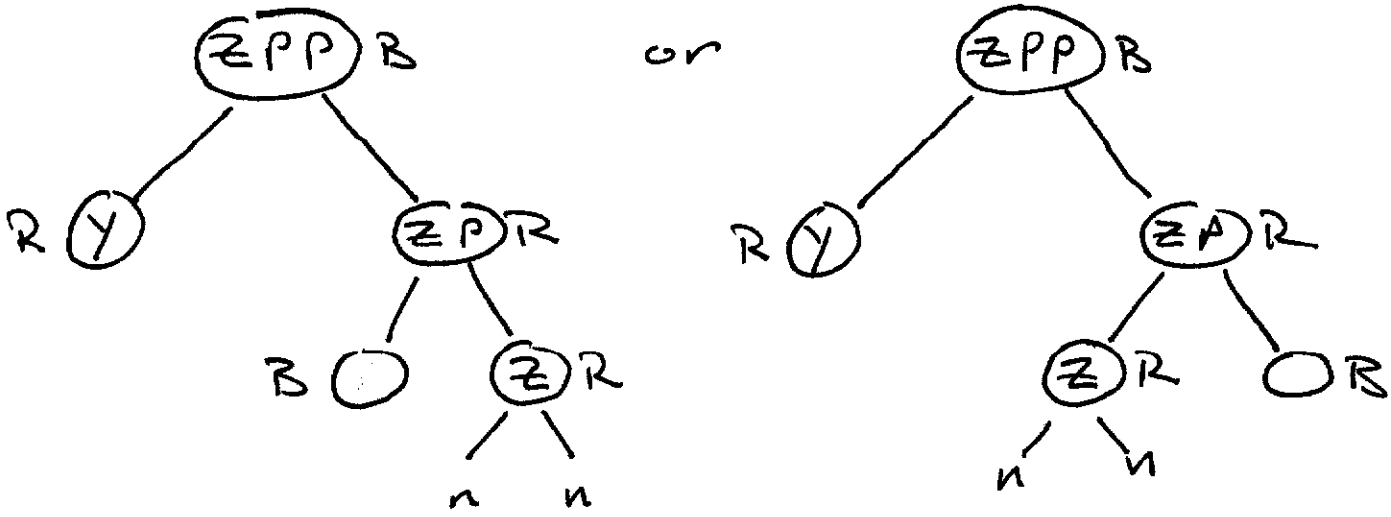
- RB_Insert(T, z)
- RB_InsertFixup(T, z)

Problem: if z.Parent.Color = Red we have violation of RBT (4).

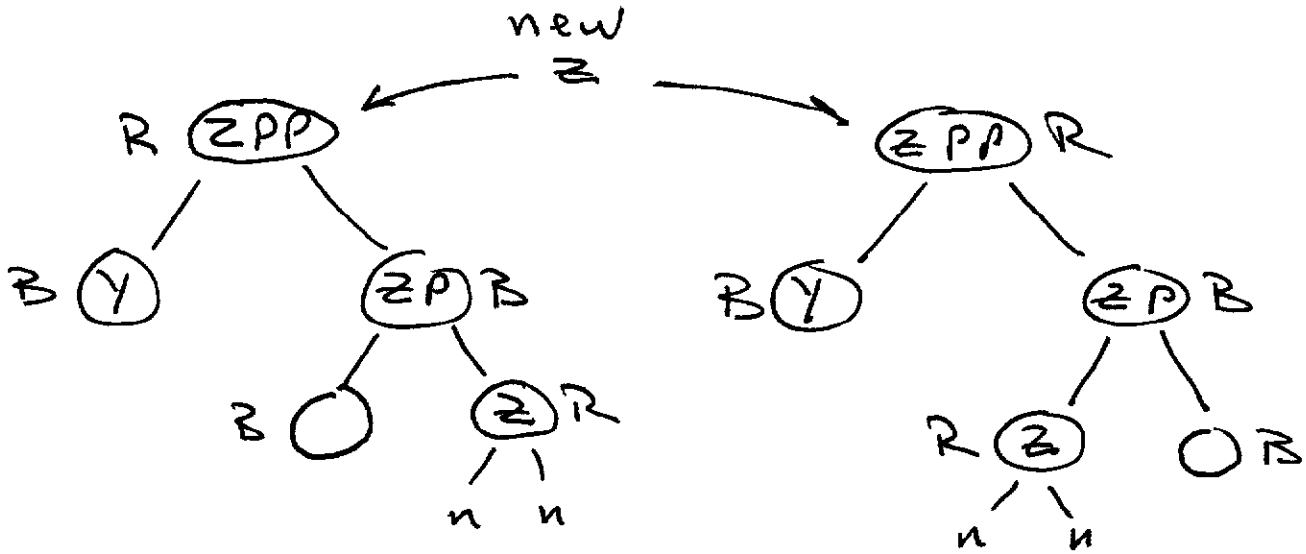
Two cases:

- z.Parent is left child of its Parent
  let y be z.Parent sibling, i.e. right child of z.Parent.Parent, i.e. y is z's uncle.
  ⇒ subcases: {1, 2, 3}
- z.Parent is right child of its Parent
  let y be sibling of z.Parent (uncle)
  ⇒ subcases: {4, 5, 6}

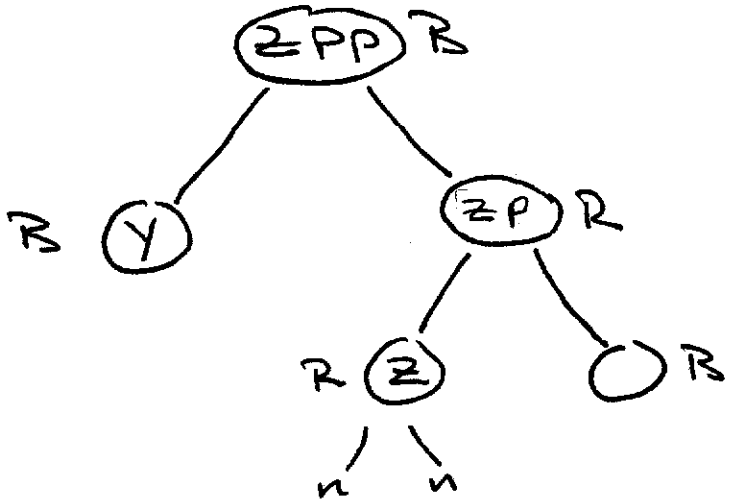
Case 4: y is Red (y == z.Parent.Parent.left)



Fix color relationships among y, zP, zPP, then let z climb-up to zPP.

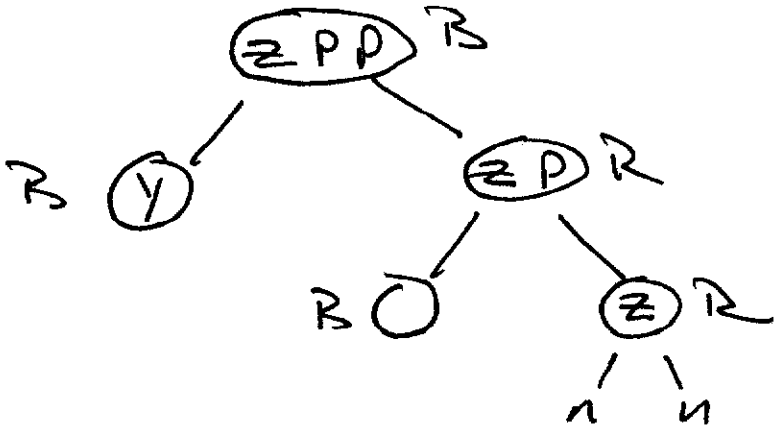


Case 5: y is Black, z == z.Parent.left

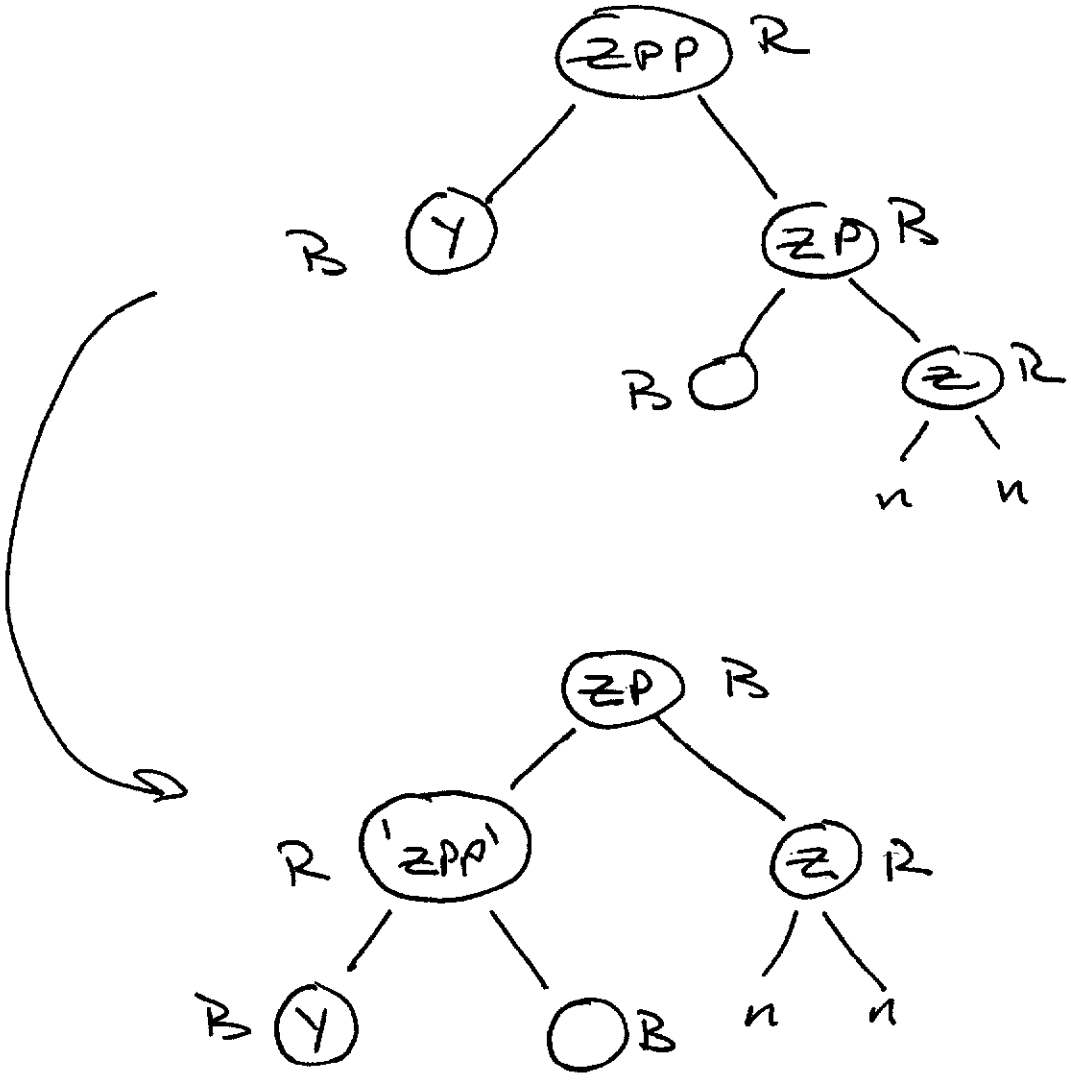


convert to case 6 by letting z climb-up to z.Parent, then RightRotate about z.

Case 6: y black, z == z.Parent.right



Color z.Parent black, color z.Parent.Parent Red, LeftRotate about z.Parent.Parent


zpp
R
B
y
zp
R
B
z
R
n
n
zp
B
R
'zpp'
z
R
B
y
B
n
n