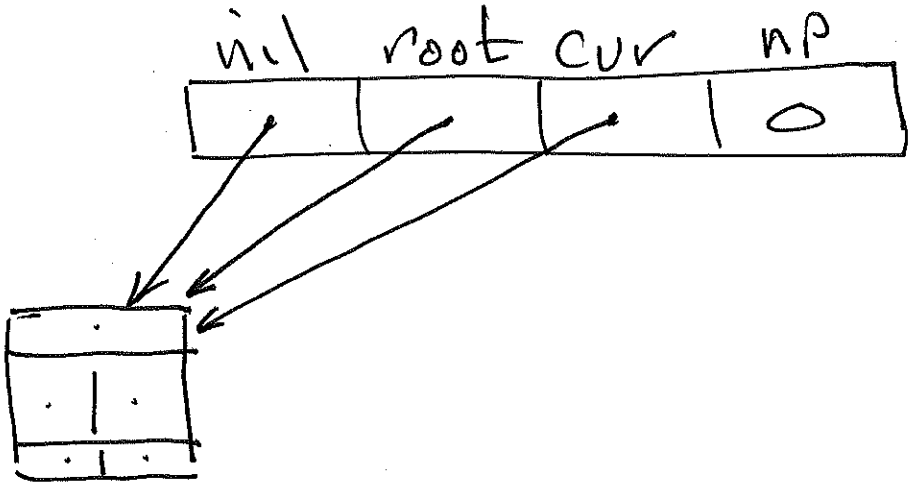
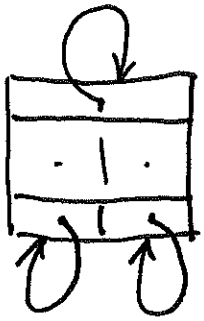
cse 101-02 6-1-23

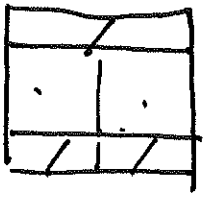
Par: 1 last day → Fri.

Empty state:

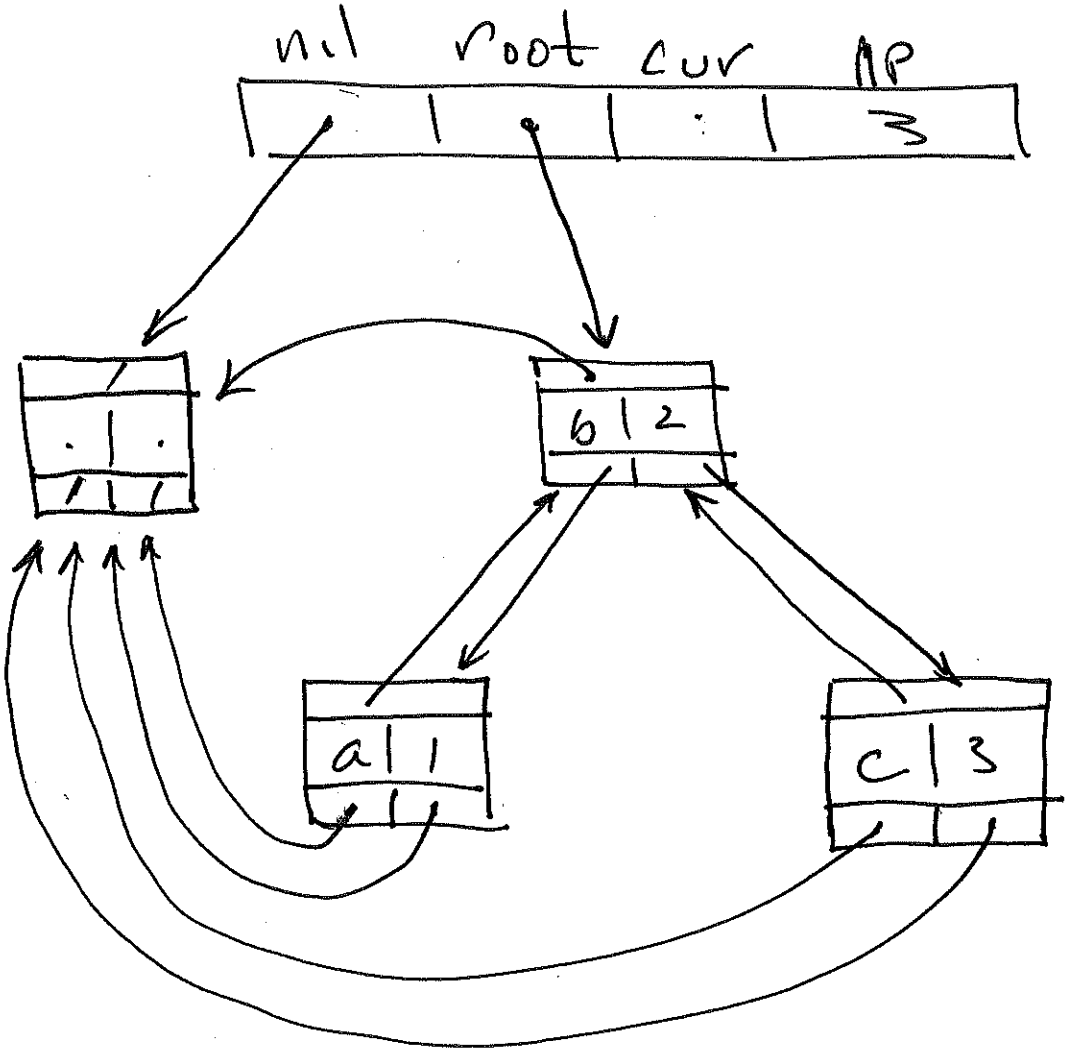




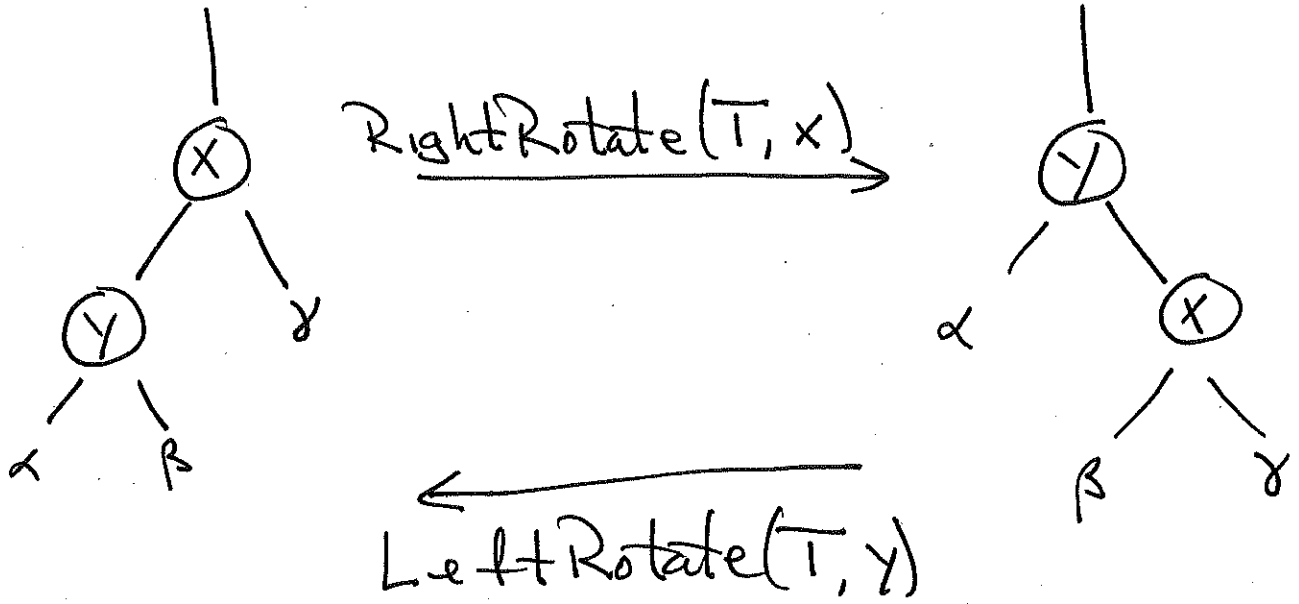
or



non-empty state :



13.2 Rotations



summary of right rotate:

β { • x.left = y.right
  • y.right.Parent = x

Parent { • y.Parent = x.Parent
  • x.Parent.(left or right) = y

x, y { • y.right = x
  • x.Parent = y

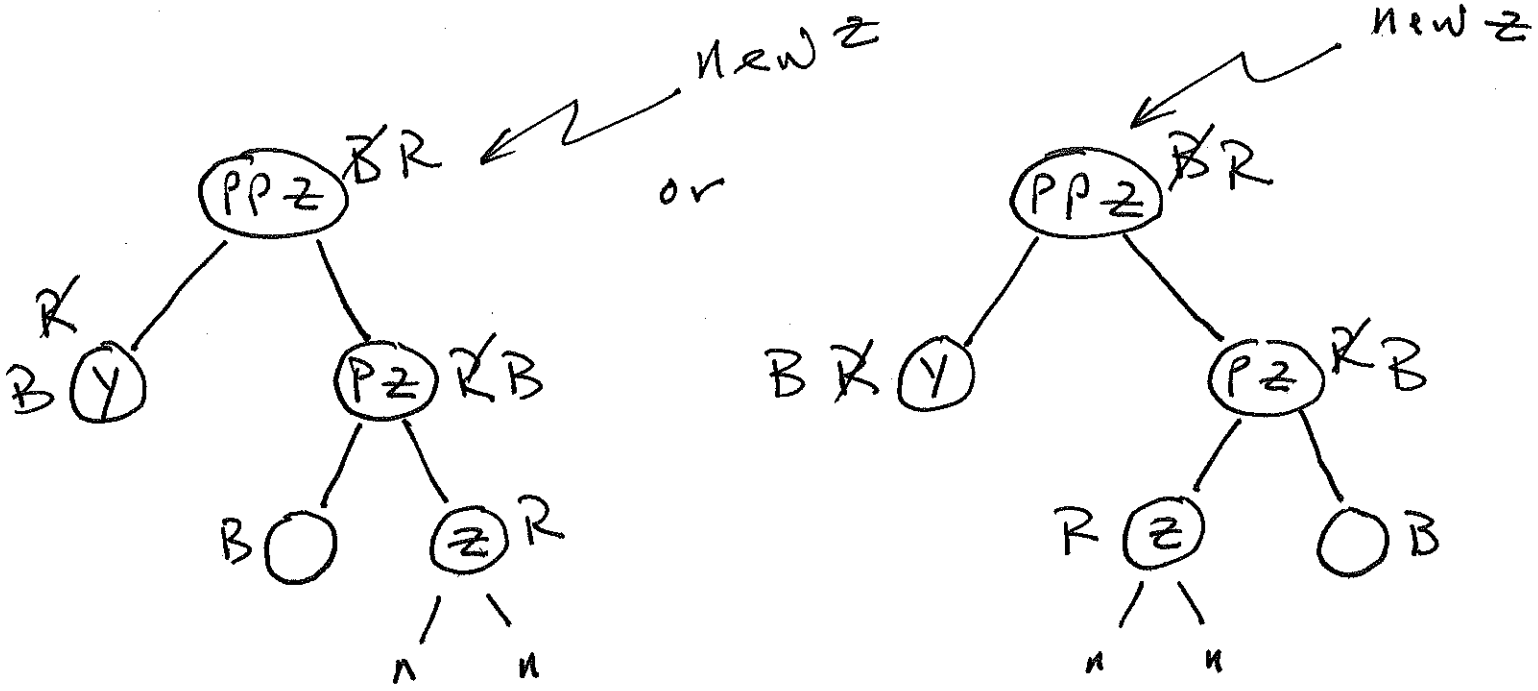
RB_insert_fixup :

cases : 1, 2, 3 : z.parent == z.parent.parent.left

cases : 4, 5, 6 : z.parent == z.parent.parent.right

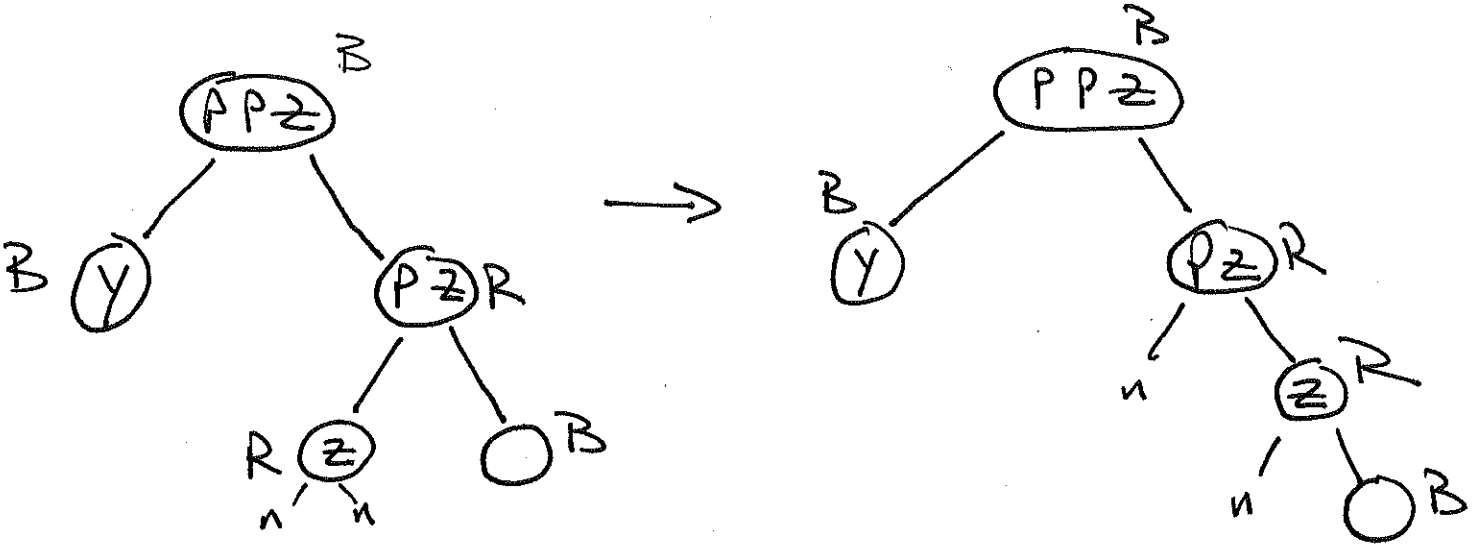
let y = z.parent.parent.left

<u>case 4</u> : y.color == Red



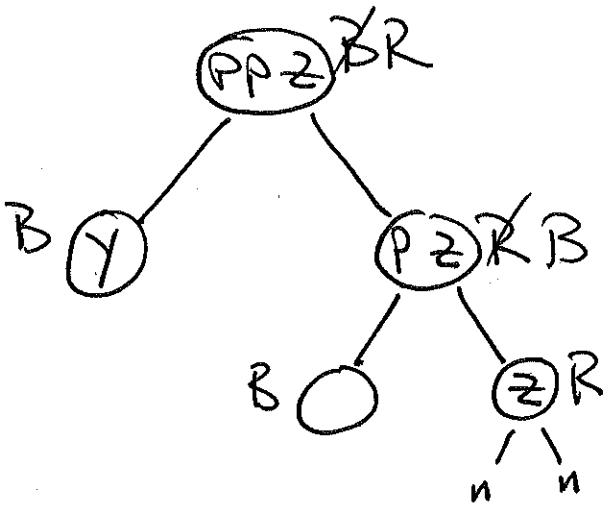
⇒ swap colors {y, pz} ↔ {ppz}. let z = z.parent.parent. Possibly iterate

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

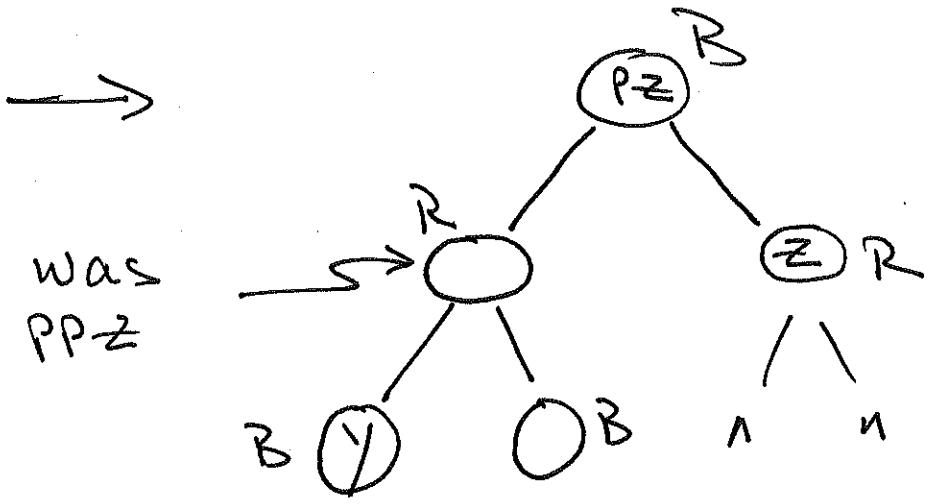


convert to case 6 : let z = z.Parent,
rotate right about (new) z

Case 6 : y.color == Black, z == z.Parent.right



- swap colors {Pz} ↔ {PPz}
- left rotate about PPz

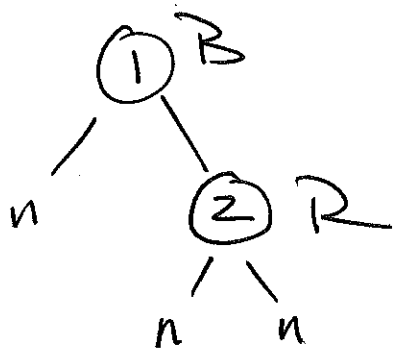


note: loop terminates after cases 5, 6, and it may terminate after case 4.

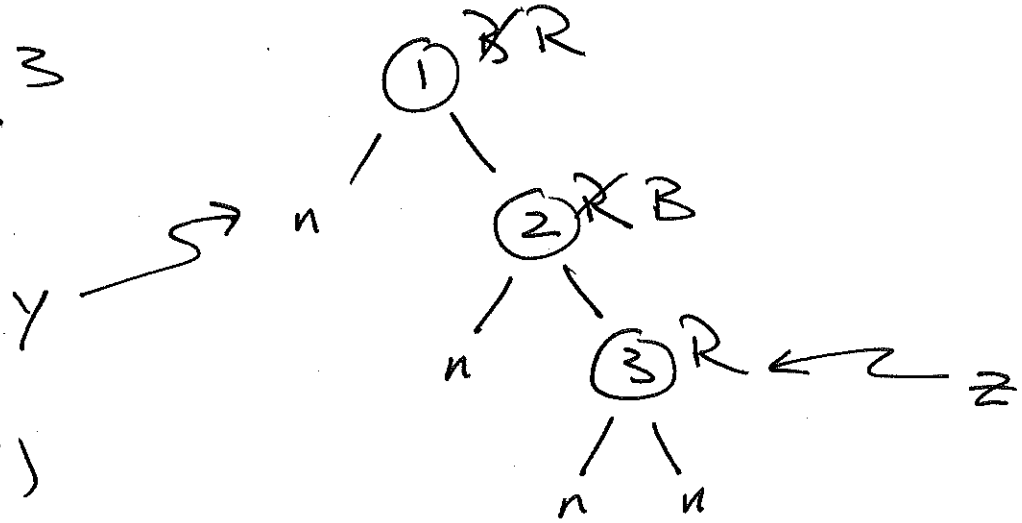
Ex. insert: 1, 2, 3, 4, 5

insert 1

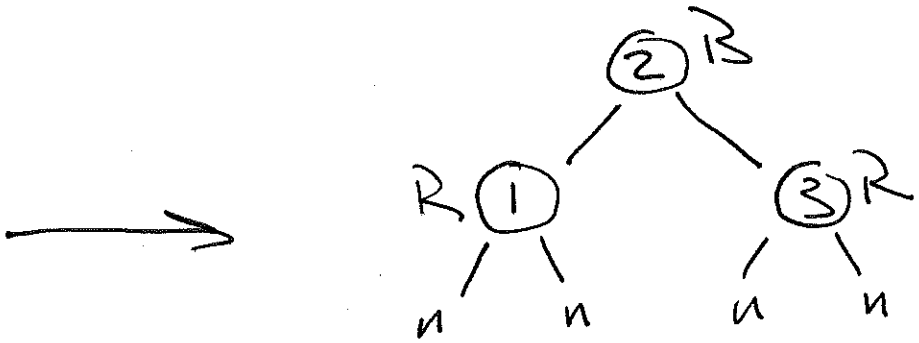
insert 2



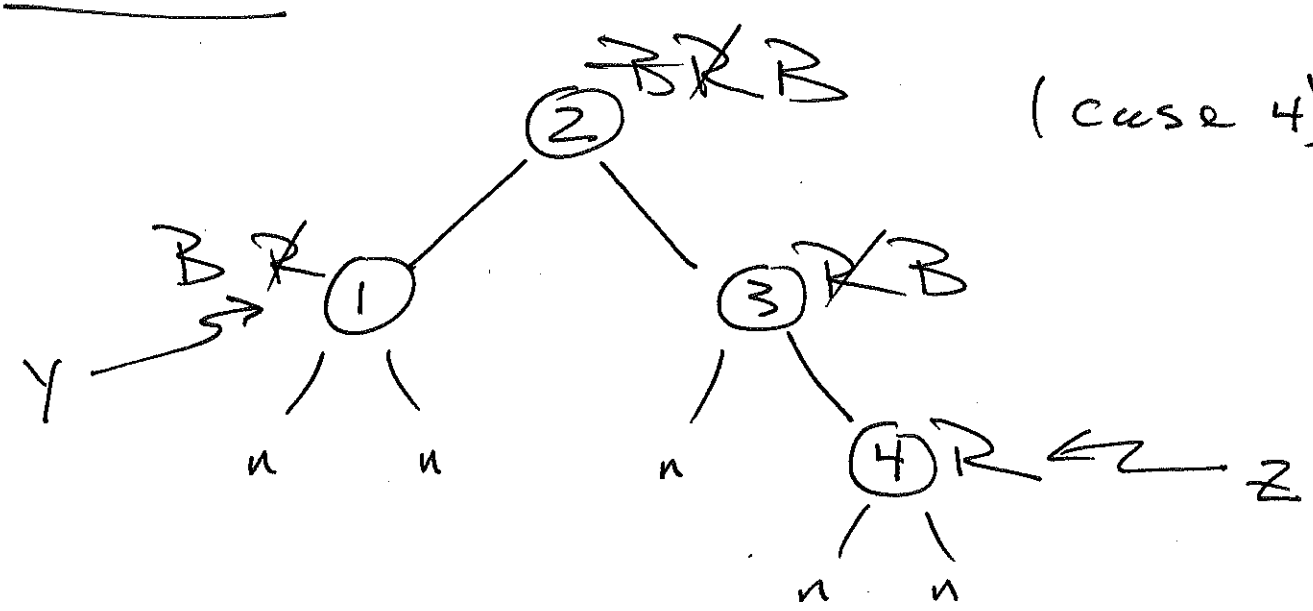
insert 3



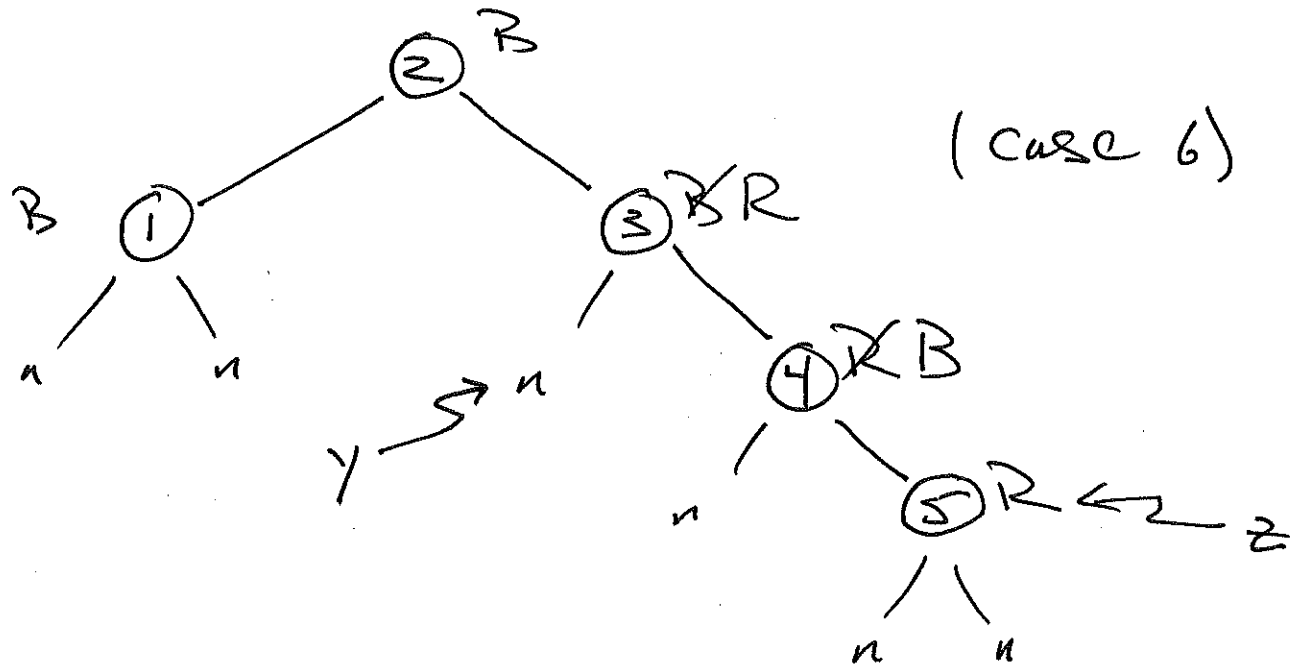
(Case 6)

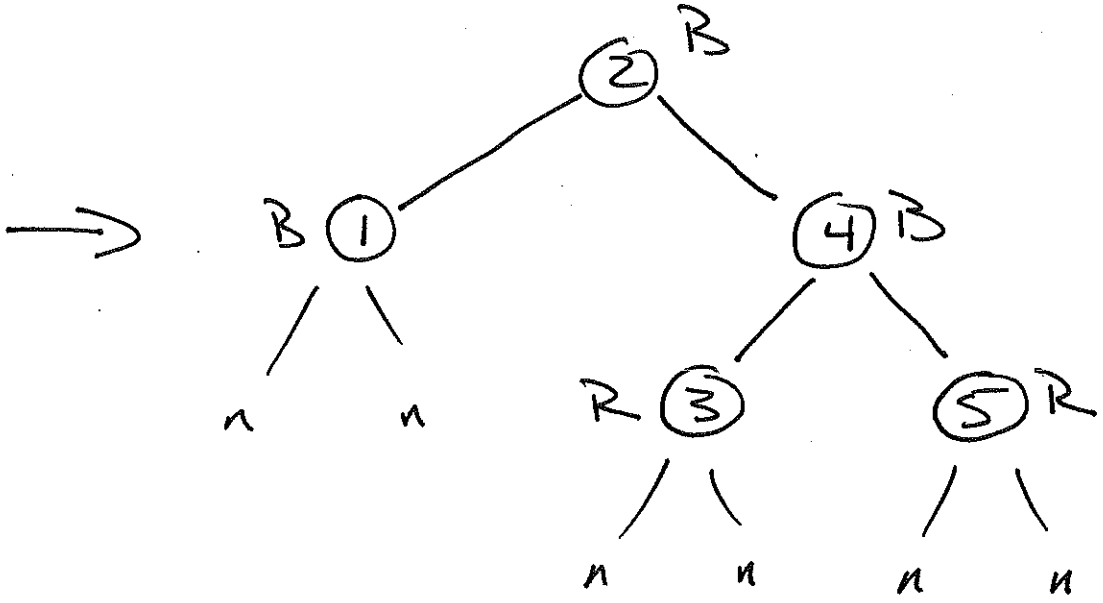


Insert 4:



Insert 5:




2 B
B 1
4 B
n
n
R 3
5 R
n
n
n
n