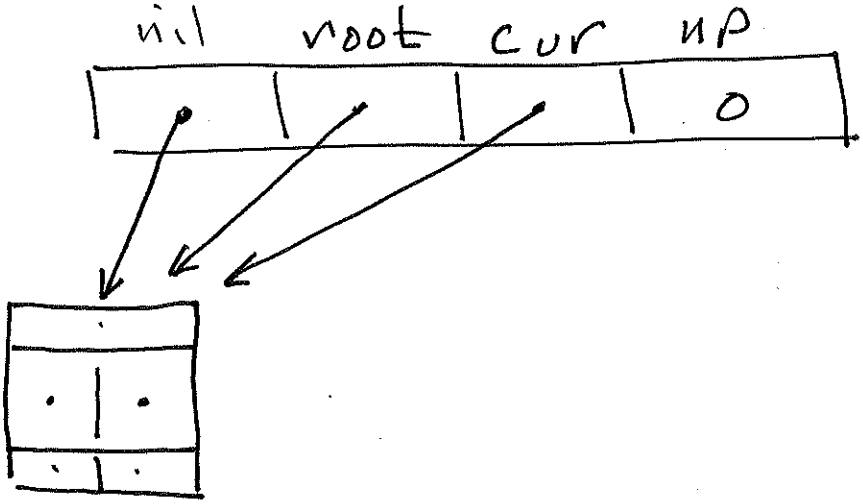
Pa7: one last day: Fri!

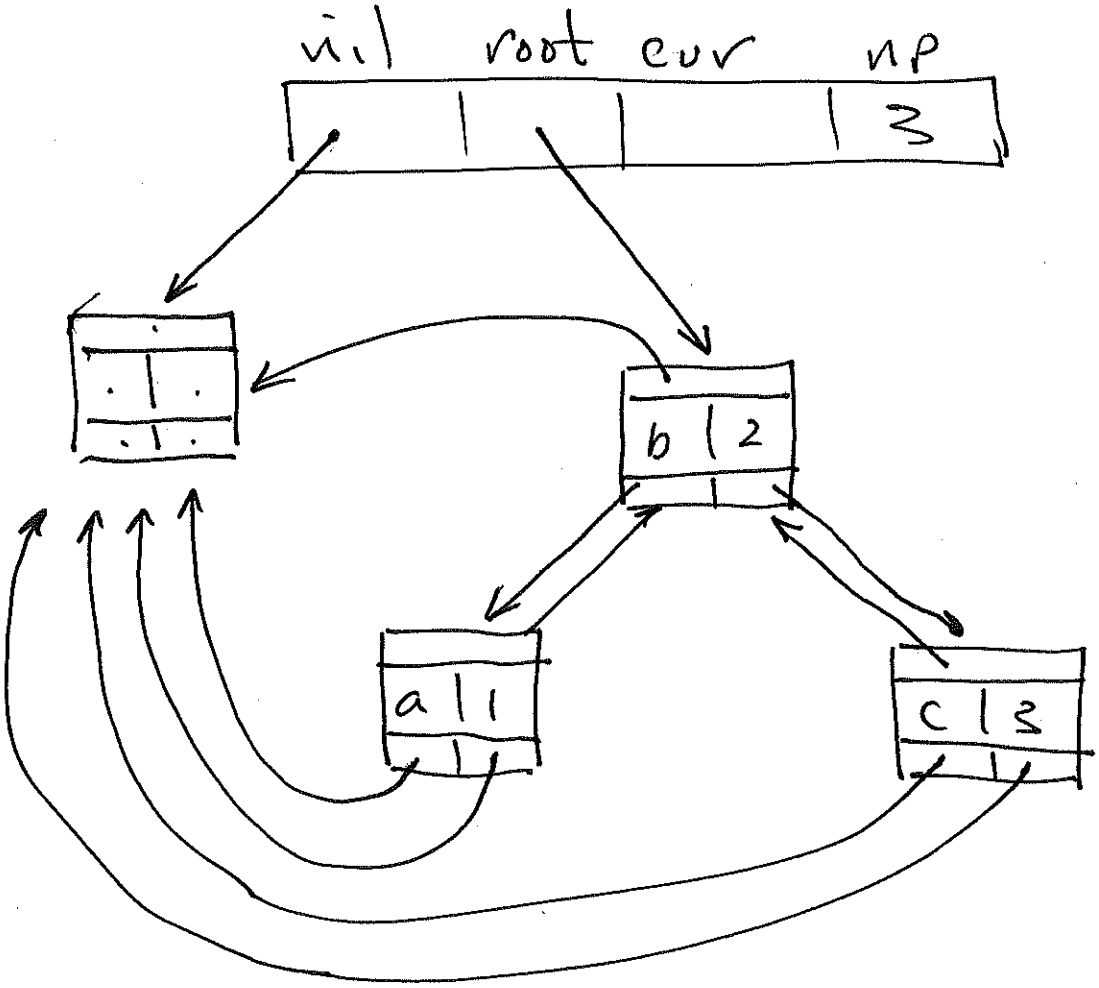
Pa7 questions:

empty state:

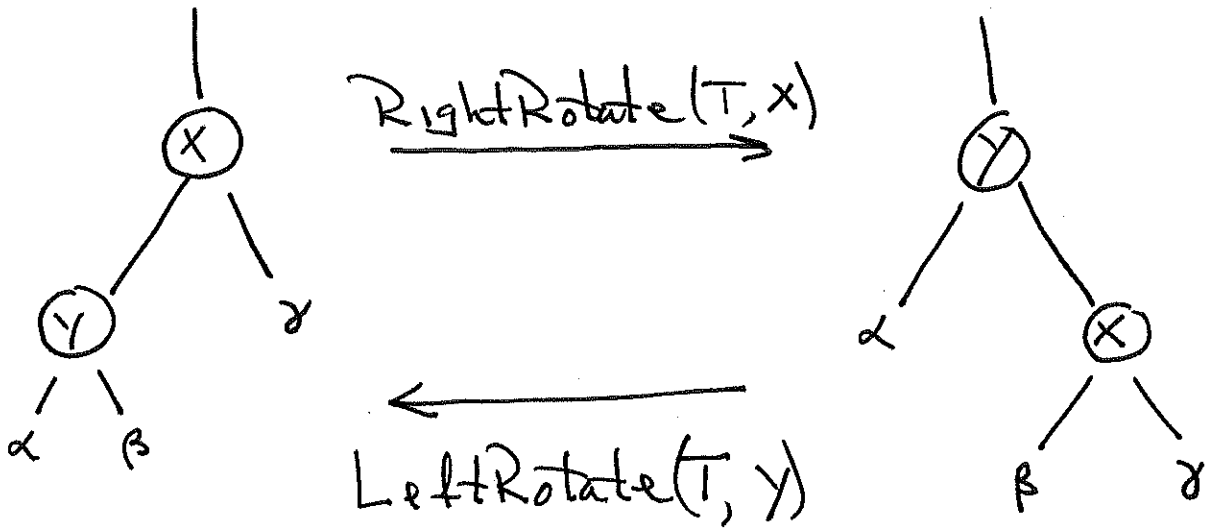


nil node object

non-empty state:



13.2 Rotations in a RBT



Summary: RightRotate(T, x)

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

13.3 Insertion

strategy:

- do BST insert()
- call RB_insert_fix_up()

fix-up cases 4, 5, 6:

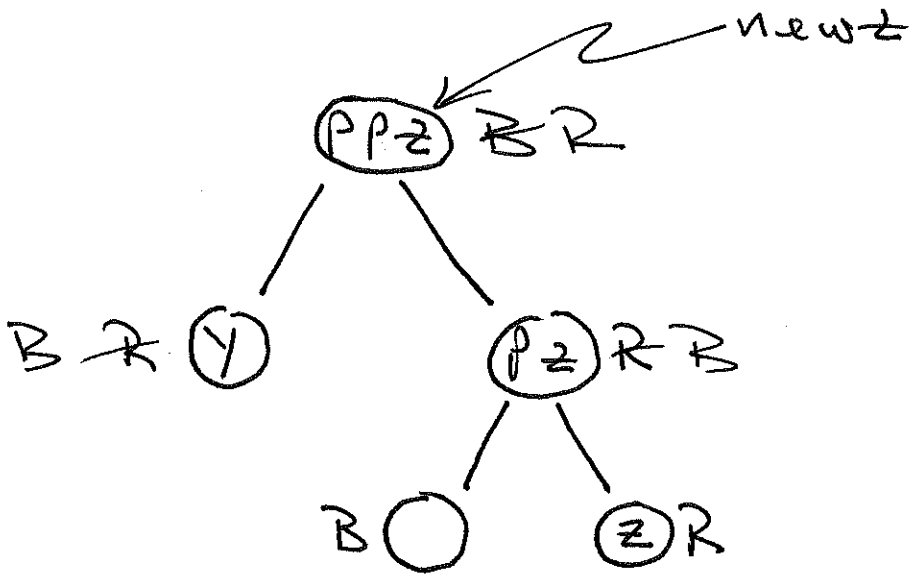
cases 4, 5, 6 defined by

z.Parent == z.Parent.Parent.right

let y = z.Parent.Parent.left

i.e. y is 'uncle' of z

Case 4: y is Red



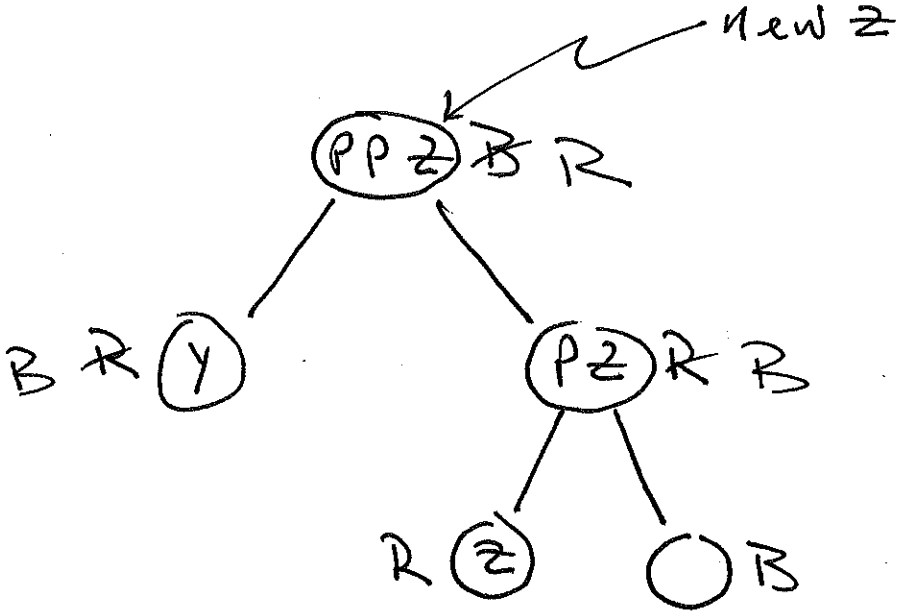
swap colors

y, pz ↔ ppz

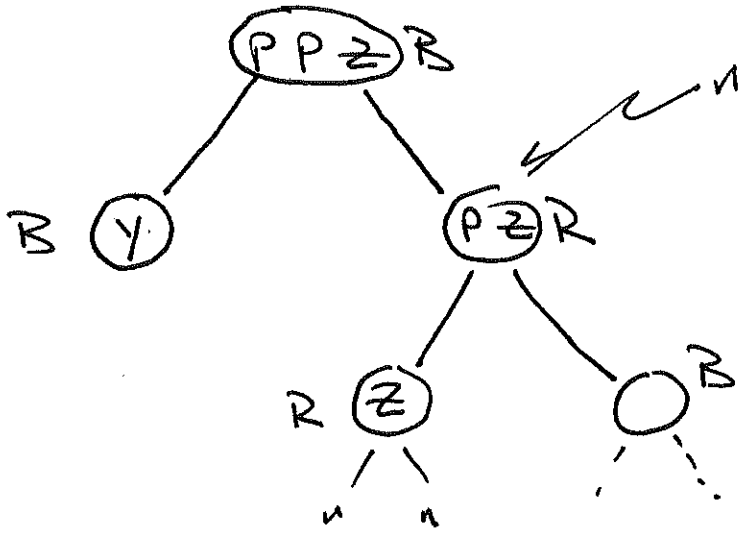
let z climb up to ppz.

Possibly another iteration of while.

or

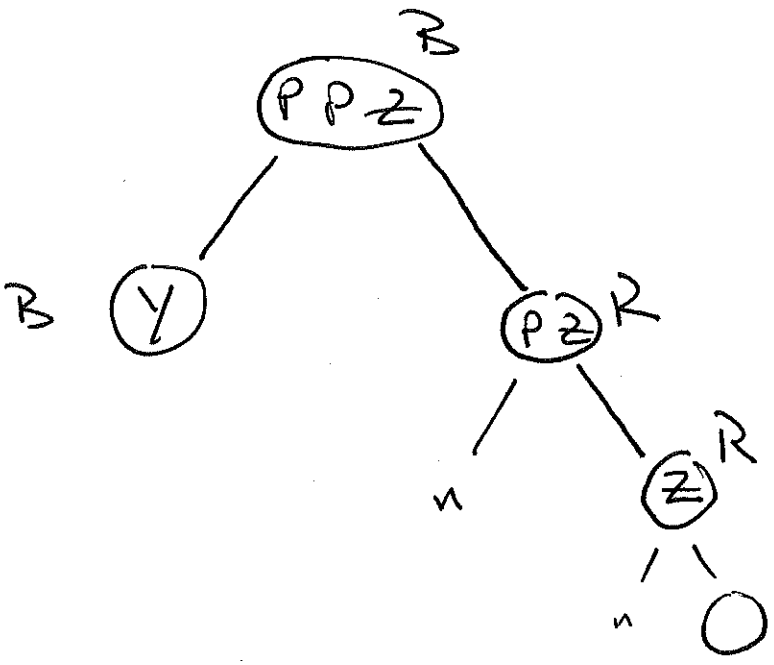


Case 5 : y is Black, z == z.parent.left



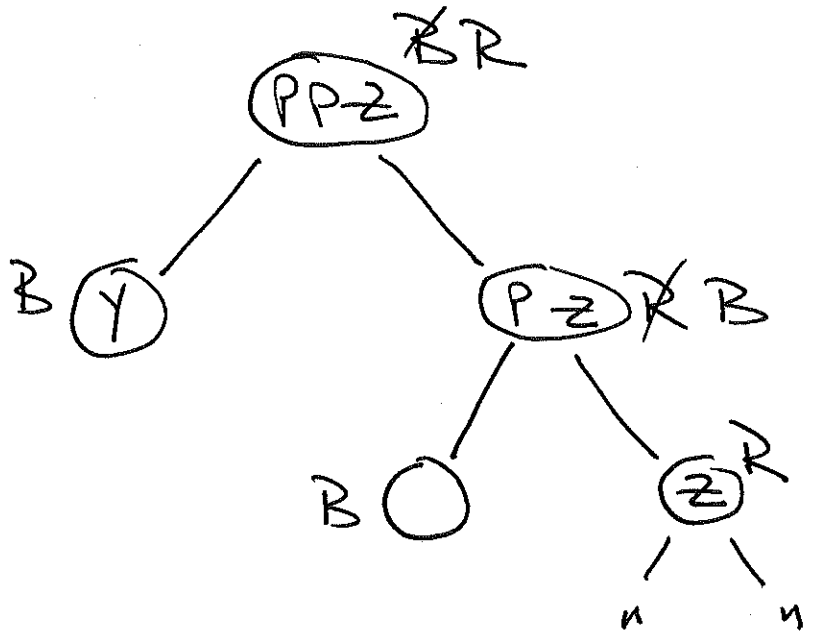
let z = Pz

RightRotate(T, z)

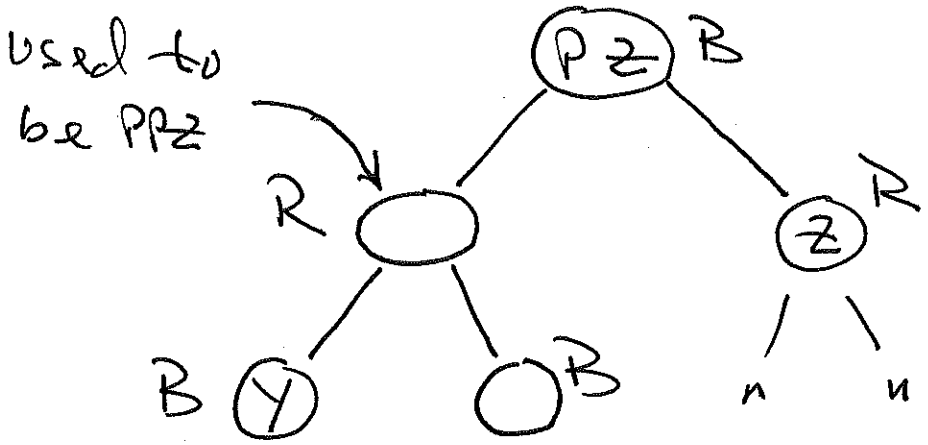


now in Case 6

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



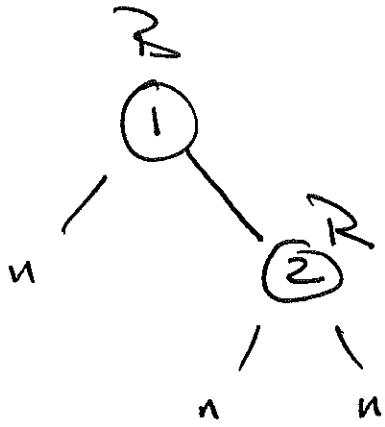
color Pz Black, PPz Red, then left Rotate about PPz



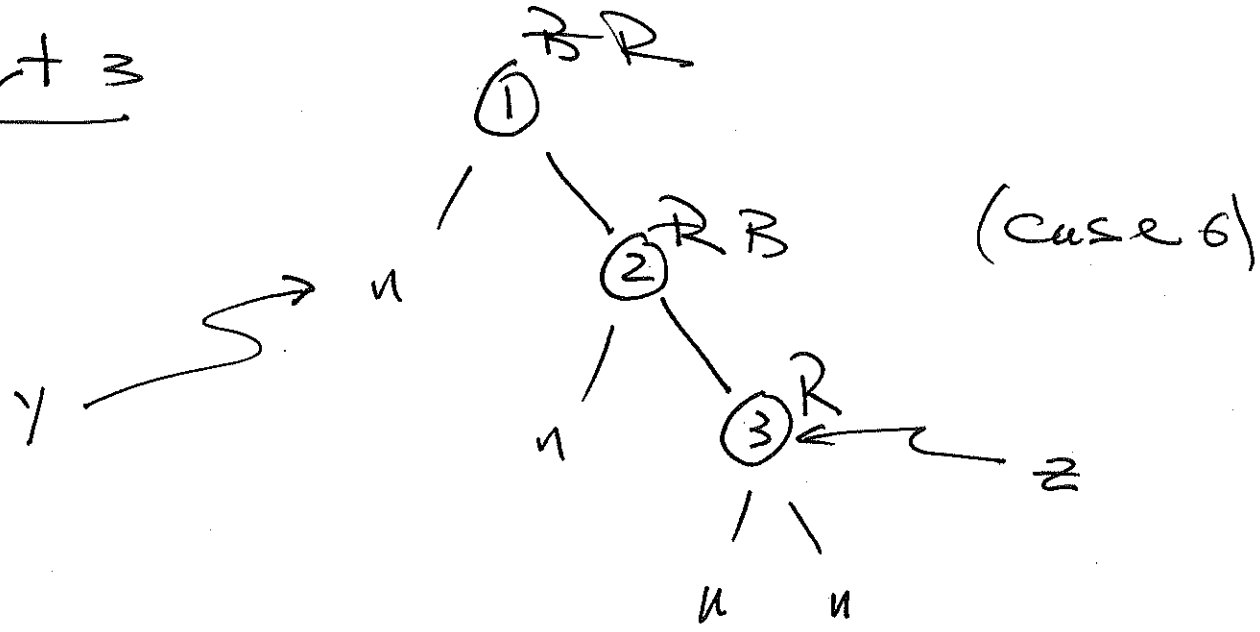
Ex. Insert: 1, 2, 3, 4, 5

insert 1

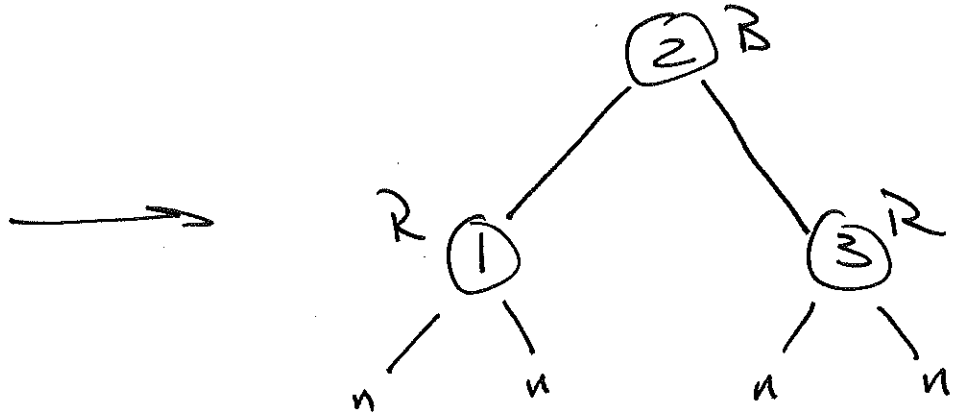
insert 2



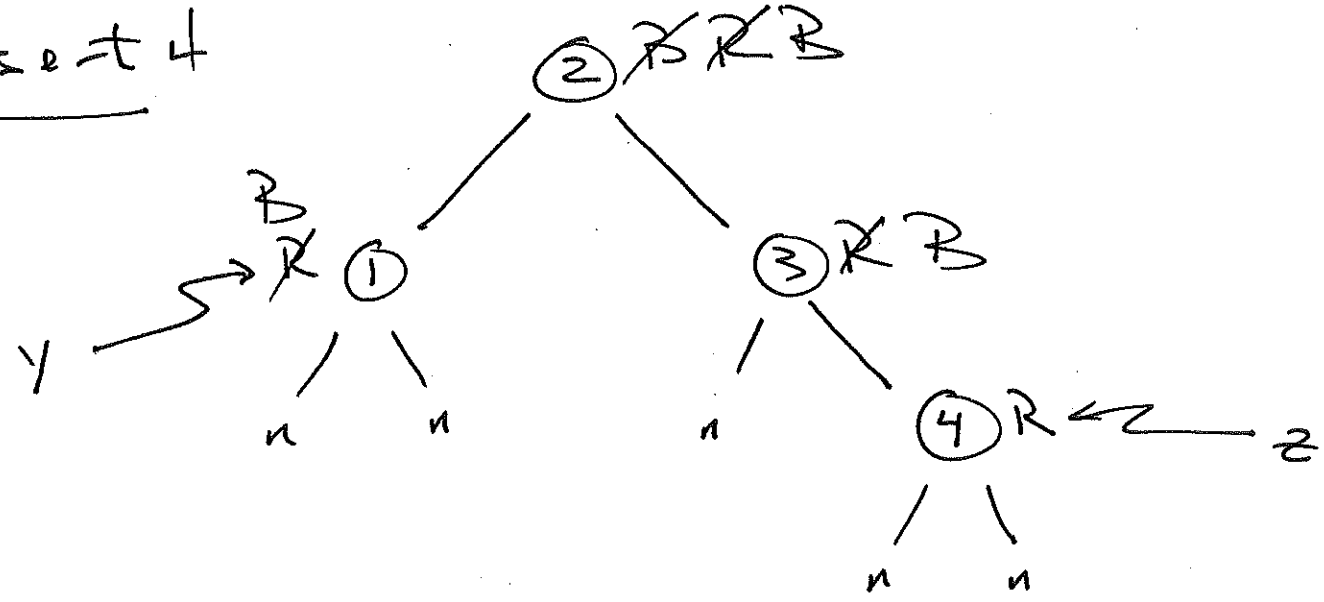
insert 3



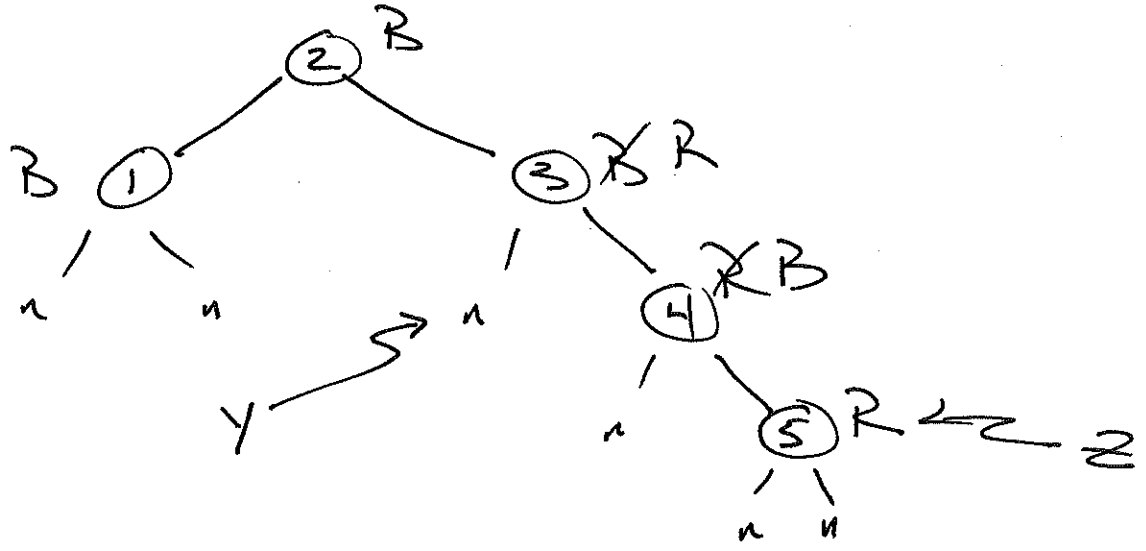
(case 6)

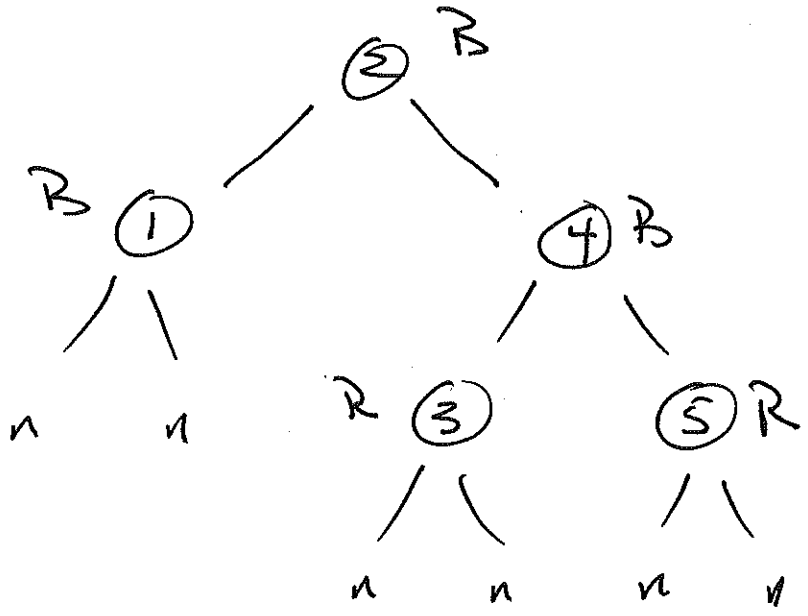


Insert 4



Insert 5




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