1/29 R-B Trees Insertion

Monday, January 29, 2018 6:36 PM

R-B Trees are BSTs

1. all Nodes are either red or Black
2. The root is black optional
3. All leaves are black (Null leaves)
4. No red node has a red child
5. Every path from Root to leaf has an equal number of black nodes.

public class Node {

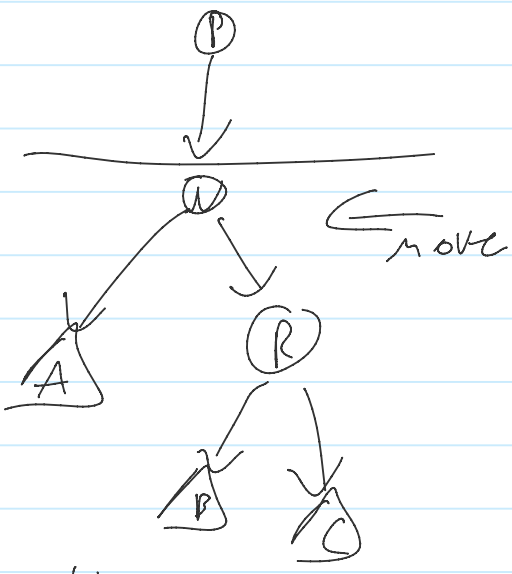
Node Left;
Node Right;
Node Parent;

grandparent()
parent.parent

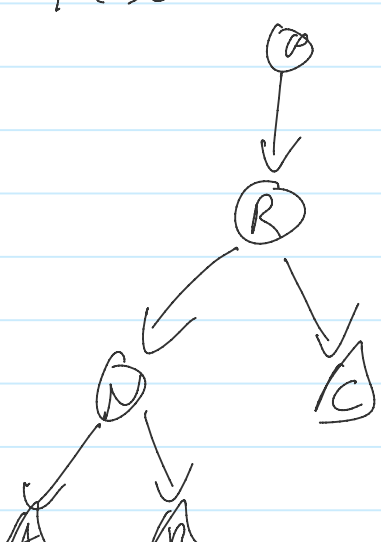
sibling()
opposite child for parent

uncle
parent.siblings

Rotate left



result

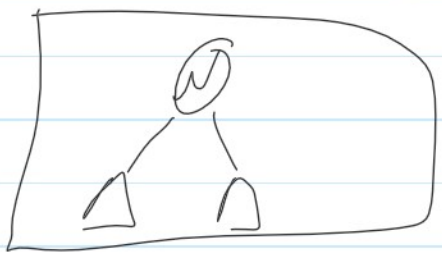


Node insert (root, newNode) //returns new root

```
insert recursive(root, newNode);
repair (newNode);
find new root (newNode); // or findnewRoot(root);
return newRoot;
```

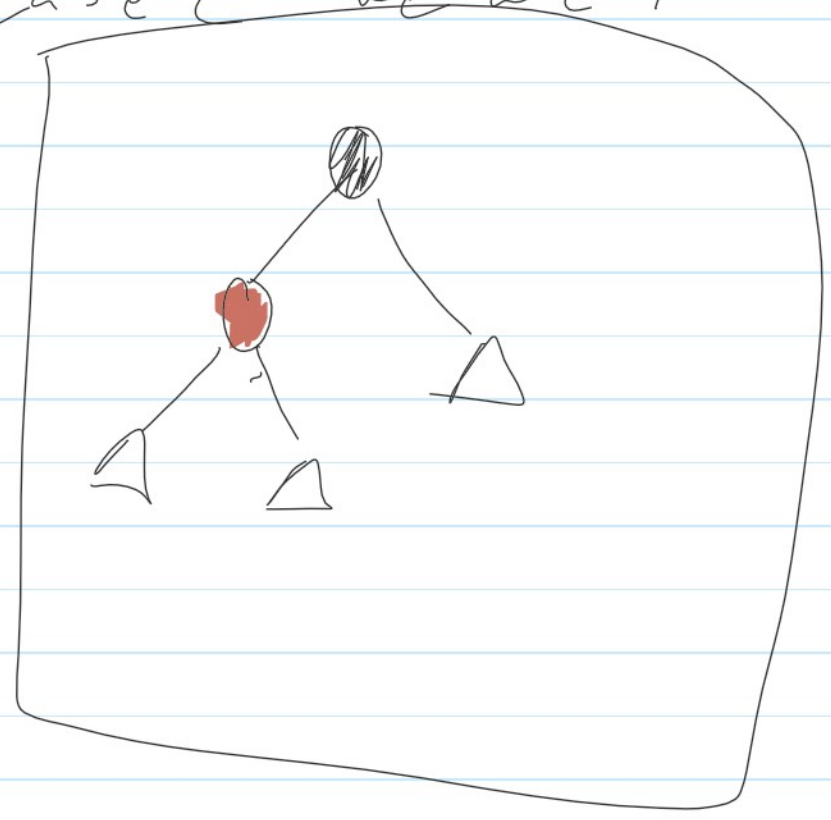
Repair (Node newNode) assume children are correct

Case 1 we have inserted the root



ensure N. Color is black

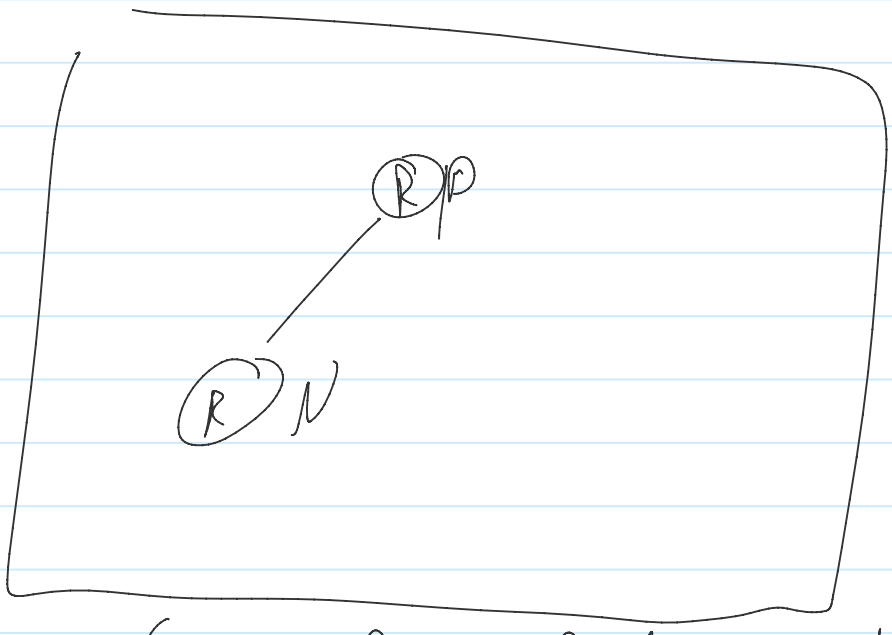
Case 2 we are red and Parent is Black



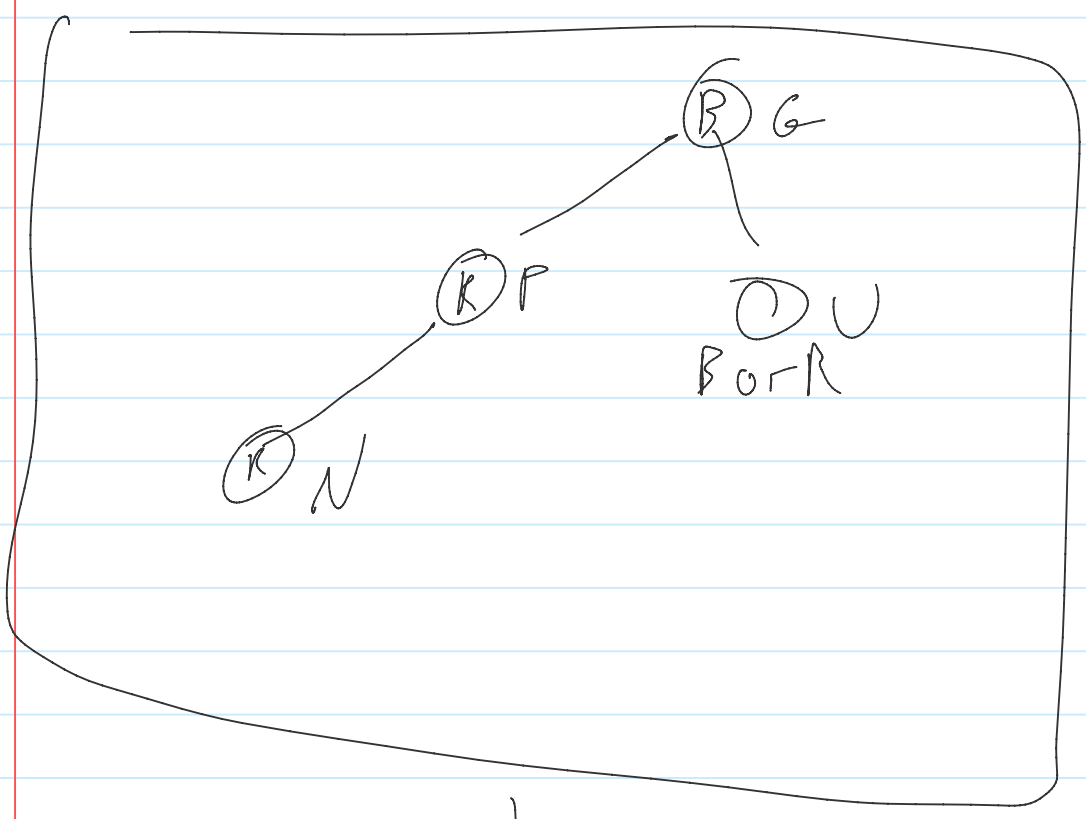
Do nothing
everything looks good

Case 3

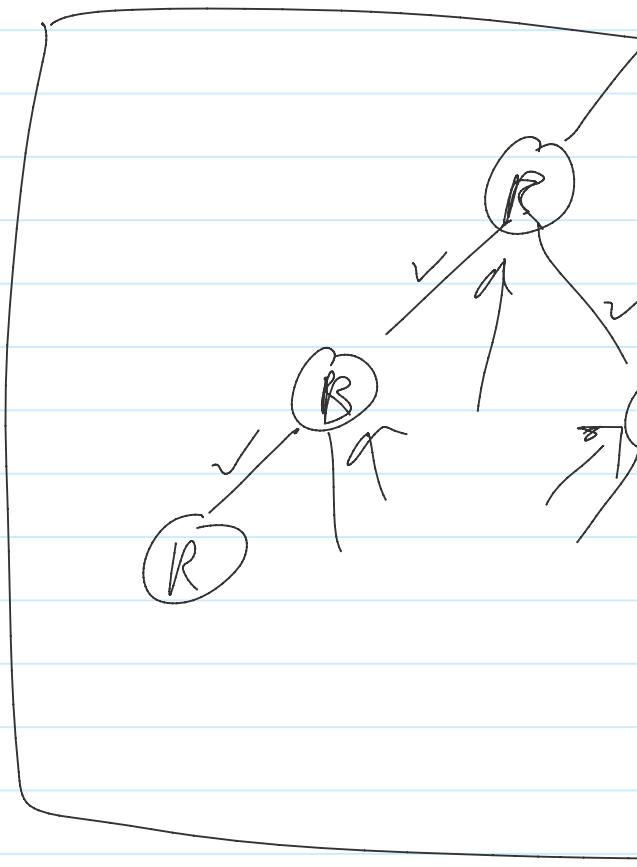
By condition 1 and we did not enter Case 2 or 1
our parent is Red



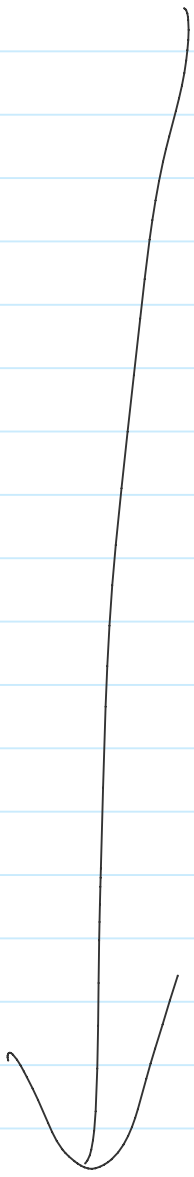
Since P is Red we have to have a grandparent

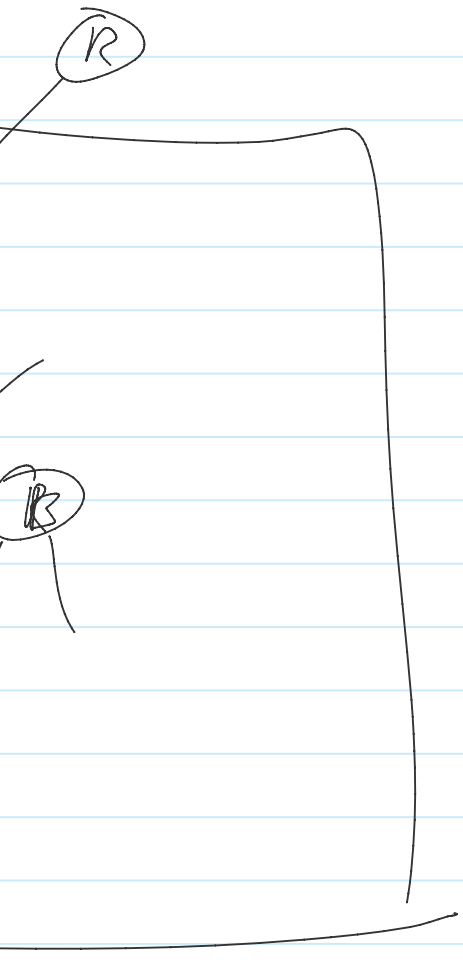


Case 3 or uncle → is Red

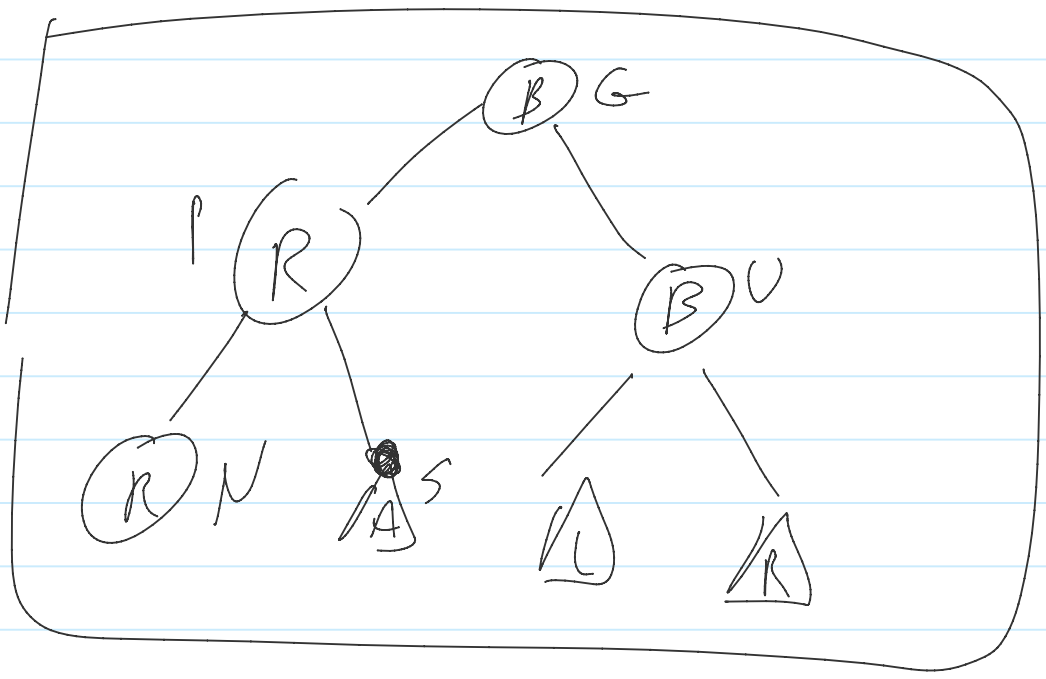


adjust color of pa
uncl
a
grand
repa

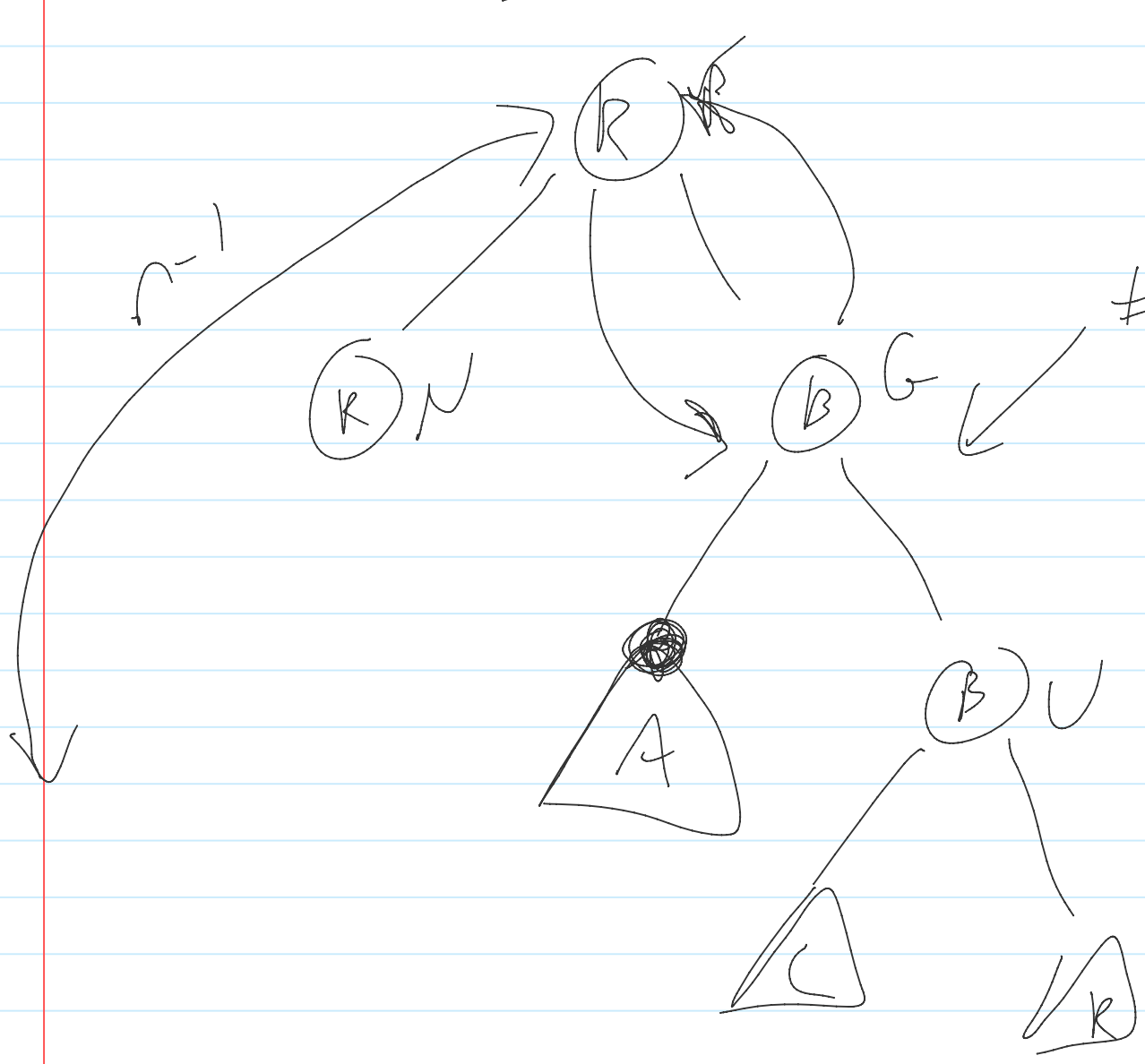




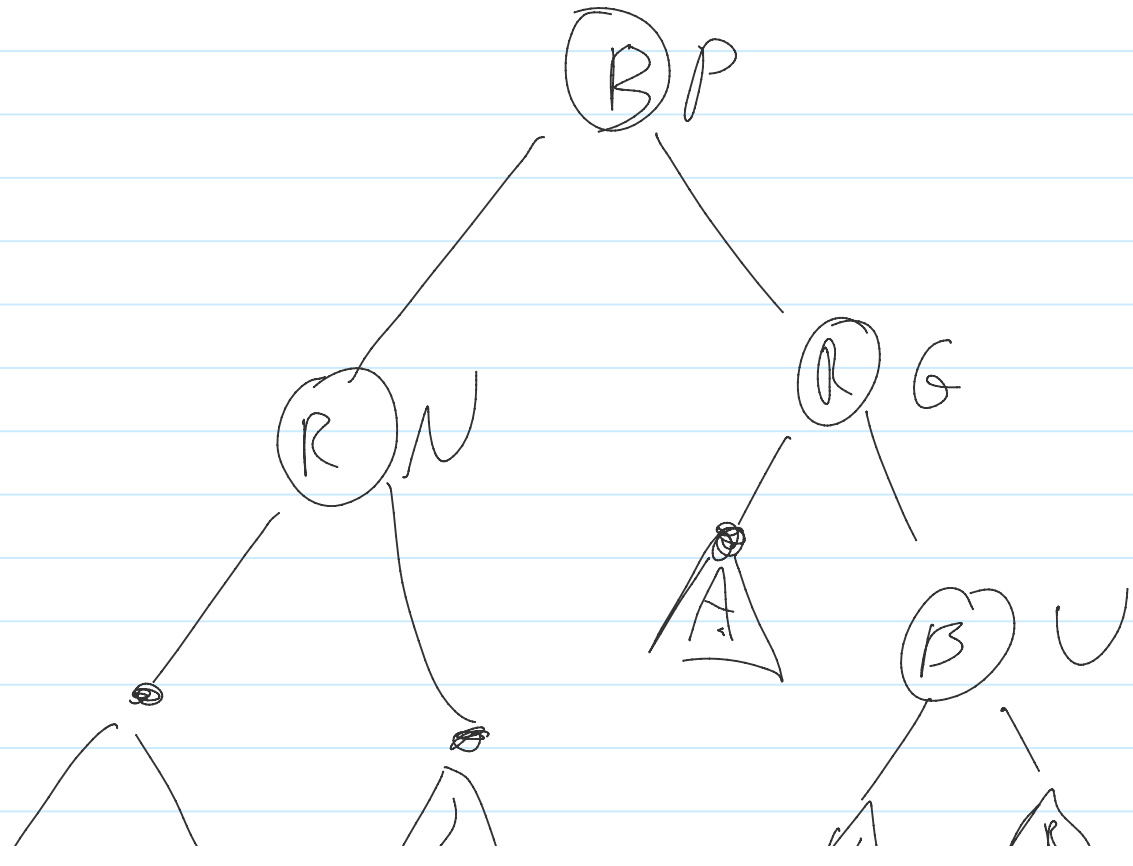
ent
ke
l
dparent
-
ir (grandparent);



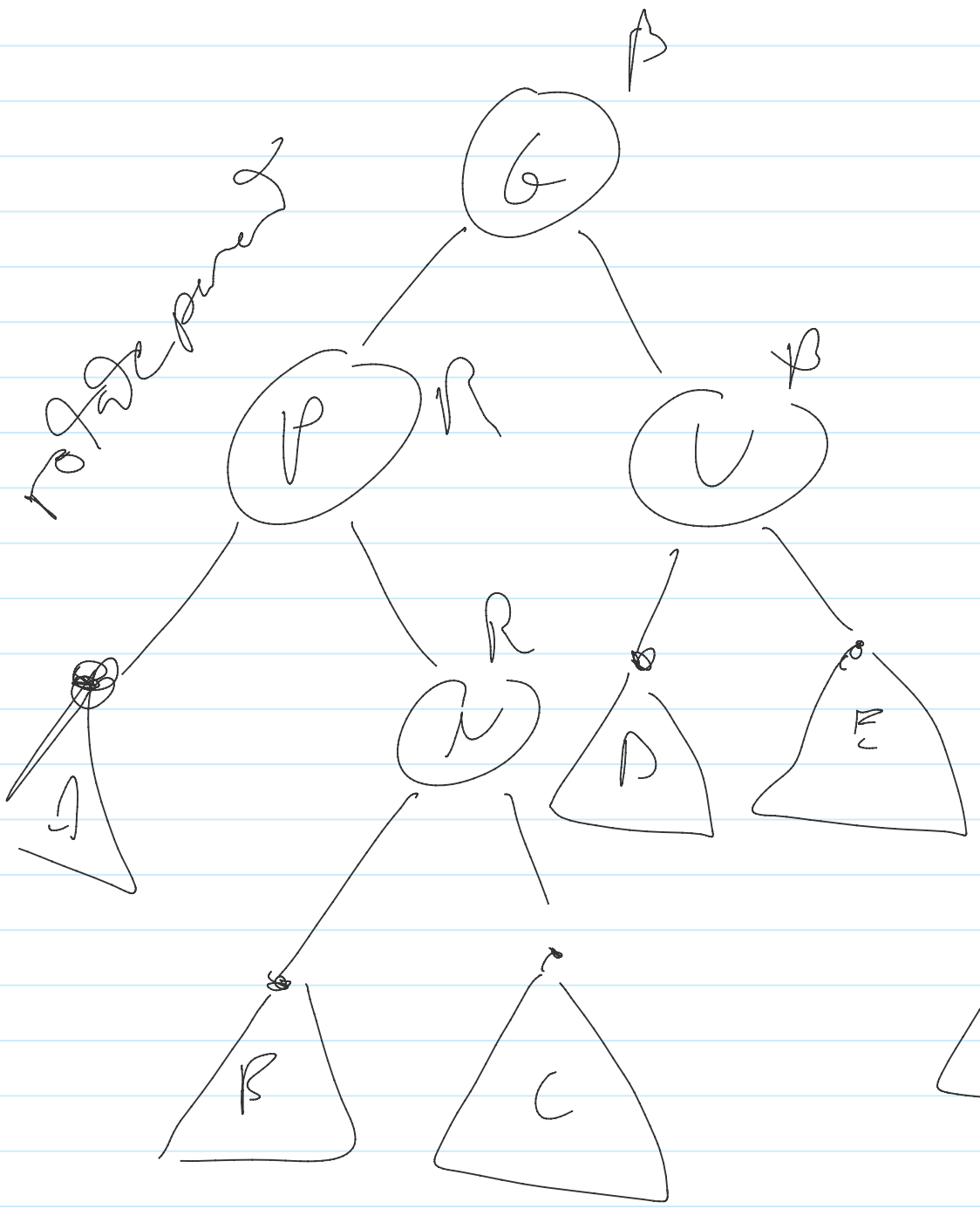
Rotate

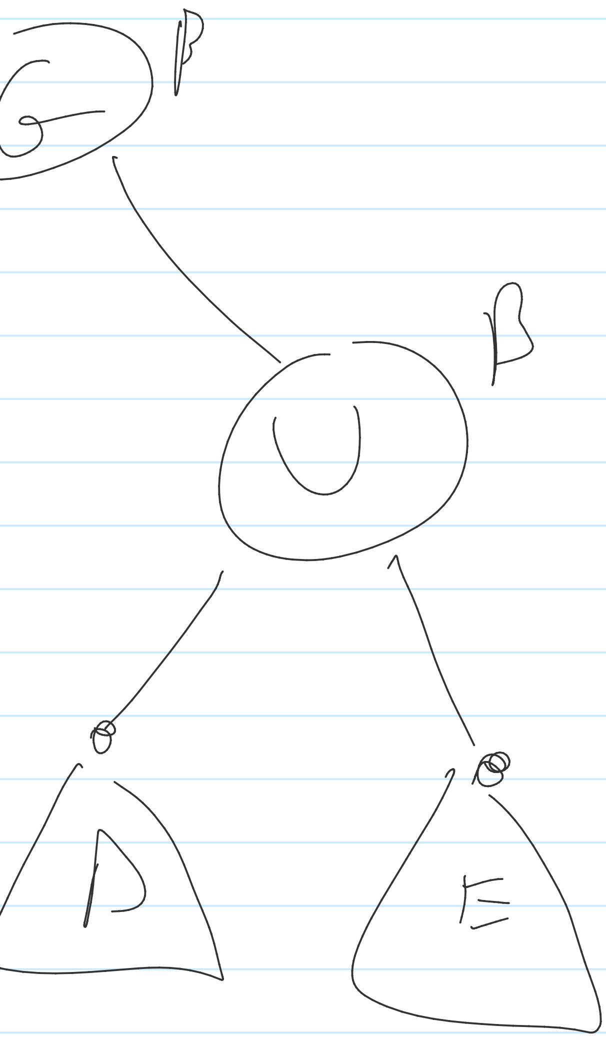


Change
G to red
P to black



1


P
G
P
R
U
P
R
D
E
C
A
B



x our(parent)