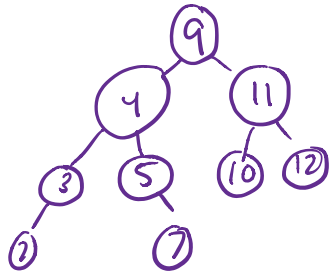
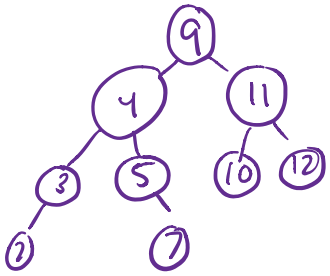
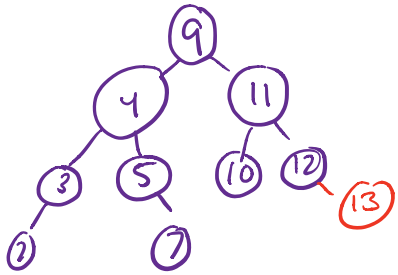


remove(10)
==>

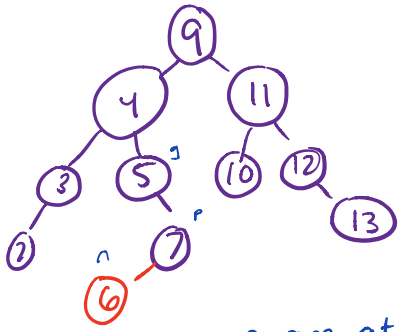


1. Check children (10)
2. leaf node.
3. get parent (10).
4. set parent → left to NULL
5. delete 10

insert(13)



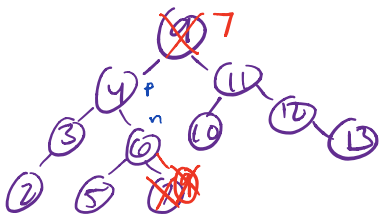
insert(6)
==>



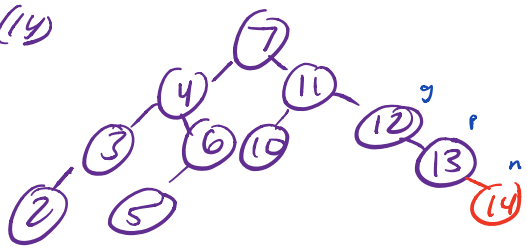
1. Insert 13 @ 12 → right value
2. set all pointers and update Balance of parent and 13.
3. Insert Fix (12,13)
4. Insert Fix (11, 12)

1. go until we are at a leaf, 7's left child
2. update all pointers and balances of 6 and 7.
3. Insert Fix (7, 6)
4. // Right Left case
   rotate Right (7)
   rotate left (5)
5. done, no recursing

remove(9)

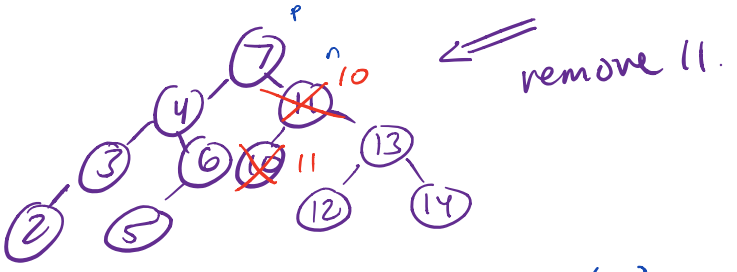


insert(14)
==>



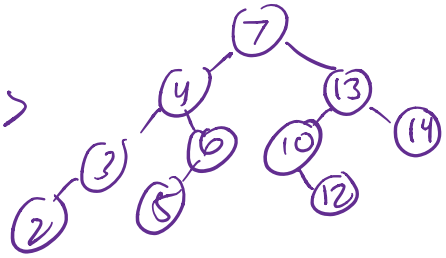
1. check how many children 9 has.
2. 9 has 2 children, so swap w predecessor. (7)
3. set the root to 7.
4. set 6 → right to NULL and delete 9.
5. RemoveFix (6, -1)
6. set Balance (6) = -1 done!

1. Insert 14 @ 13 right
2. Set 13's balance to 1.
3. Insert Fix (13, 14)
4. // Left case
   rotate left (12)
   set Balance of 13 → 0
   set Balance 12 → 0.



remove 11.

=>



Final.

1. swap 11 w/ predecessor (10)
2. set 10 → left to NULL
3. delete 11.
4. RemoveFix (10, 1)
5. ndiff = -1
6. // zig zig case
   rotateLeft (10)
   Set Balance (10) = 1
   Set balance (13) = -1
   done!