Pranitha Rao
2018BIT1123
1803213087
IT-A 23/9/20.

DAA.

Q4.

a) $T(n) = 2T\left(\frac{n}{2}\right) + \frac{n}{\log n}$ → (1)

⇒ $T\left(\frac{n}{2}\right) = 2T\left(\frac{1}{2} \times \frac{n}{2}\right) + \frac{n/2}{\log n/2}$

⇒ $T\left(\frac{n}{2}\right) = 2T\left(\frac{n}{2^2}\right) + \frac{n/2}{\log n/2}$

put in (1)

$T(n) = 2\left[2T\left(\frac{n}{2^2}\right) + \frac{n/2}{\log n/2}\right] + \frac{n}{\log n}$

$T(n) = 2^2 T\left(\frac{n}{2^2}\right) + \frac{n}{\log n/2} + \frac{n}{\log n}$ → (2)

$T\left(\frac{n}{2^2}\right) = 2T\left(\frac{1}{2} \times \frac{n}{2^2}\right) + \frac{n/2^2}{\log n/2^2}$ from (1)

$T\left(\frac{n}{2^2}\right) = 2T\left(\frac{n}{2^3}\right) + \frac{n/2^2}{\log n/2^2}$

put in (2)

⇒ $T(n) = 2^2\left[2T\left(\frac{n}{2^3}\right) + \frac{n/2^2}{\log n/2^2}\right] + \frac{n}{\log n/2} + \frac{n}{\log n}$

Malina Rai
2018BIT1123
18032138087
IT-A

23/9/20.

DAA.

Q4.

a) $T(n) = 2T\left(\frac{n}{2}\right) + \frac{n}{\log n}$ → (1)

$T\left(\frac{n}{2}\right) = 2T\left(\frac{1}{2} \times \frac{n}{2}\right) + \frac{n/2}{\log n/2}$

$T\left(\frac{n}{2}\right) = 2T\left(\frac{n}{2^2}\right) + \frac{n/2}{\log n/2}$

put in (1)

$T(n) = 2\left[2T\left(\frac{n}{2^2}\right) + \frac{n/2}{\log n/2}\right] + \frac{n}{\log n}$

$T(n) = 2^2 T\left(\frac{n}{2^2}\right) + \frac{n}{\log n/2} + \frac{n}{\log n}$ → (2)

$T\left(\frac{n}{2^2}\right) = 2T\left(\frac{1}{2} \times \frac{n}{2^2}\right) + \frac{n/2^2}{\log n/2^2}$ from (1)

$T\left(\frac{n}{2^2}\right) = 2T\left(\frac{n}{2^3}\right) + \frac{n/2^2}{\log n/2^2}$

put in (2).

$T(n) = 2^2\left[2T\left(\frac{n}{2^3}\right) + \frac{n/2^2}{\log n/2^2}\right] + \frac{n}{\log n/2} + \frac{n}{\log n}$

$T(n) = 2^3 T\left[\frac{n}{2^3}\right] + \frac{n}{\log n/2^2} + \frac{n}{\log n/2} + \frac{n}{\log n}$

$\log 2^{k-(k-1)}$

$\log_2 2^1$.

$\log \frac{2^k}{2^{k-1}}$ $\log 2^{k-(k-1)}$

$\log 2^1$

$T(n) = 2^3 T\left[\frac{n}{23}\right] + n\left[\frac{1}{\log \frac{n}{2^2}} + \frac{1}{\log \frac{n}{2}} + \frac{1}{\log n}\right]$

1 1 }

k times.

$T(n) = 2^k T\left[\frac{n}{2^k}\right] + n\left[\frac{1}{\log \frac{n}{2^{k-1}}} + \frac{1}{\log \frac{n}{2^{k-2}}} + \cdots + \frac{1}{\log \frac{n}{2}} + \frac{1}{\log n}\right]$ → (3)

Now $T(1) = 1$

$\frac{n}{2^k} = 1$ or $n = 2^k$ or $k = \log n$

put in (3)

$T(n) = n T[1] + n\left[\frac{1}{\log \frac{2^k}{2^{k-1}}} + \frac{1}{\log \frac{2^k}{2^{k-2}}} + \cdots + \frac{1}{\log 2^k}\right]$

$\Rightarrow n + n\left[\frac{1}{\log_2 2^1} + \frac{1}{\log_2 2^2} + \cdots + \frac{1}{\log_2 2^k}\right]$

$k-(k-1)$

$\log^{R-(R-1)}$

$2 \log 1$

$n + n\left[\frac{1}{1} + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{R}\right]$

$n + n \log k$.

$\therefore$ T.C. = $n \log k$.

$\Rightarrow$ $n \log \log_2 n$ Ans.

Q4. b). $T(n) = T(\sqrt{n}) + \theta(\log n)$

$\Rightarrow T(n) = T(\sqrt{n}) + \log n$.

$\Rightarrow$ let $n = 2^k$. $k = \log_2 n$

$T(2^k) = T(2^{k/2}) + \log_2 2^k$ k^{0+1}

$\Rightarrow T(2^k) = T(2^{k/2}) + k$.

$\Rightarrow$ $\hookrightarrow$ let this be equal to S.

$S(k) = S\left(\frac{k}{2}\right) + k$.

Applying masters

$a = 1$ $b = 2$ $f(k) = k$.

$\Rightarrow k^{\log_b a} = k^{\log_2 1} = k^0 = k^0 = 1$

k^0

$\therefore f(k) = k$

& $k \log_b a = k^0$

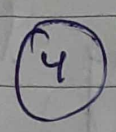
$\therefore f(k) > k \log_b a$

$\therefore$ T.C. $= \theta(f(k)) = \theta(\log_2 n)$.

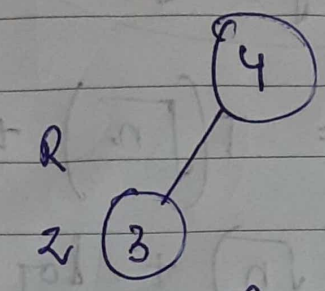
Q8 2:

4, 3, 6, 7, 11, 5, 8, 9.

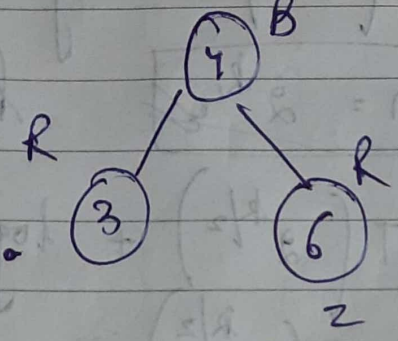
Step 1: 4 →



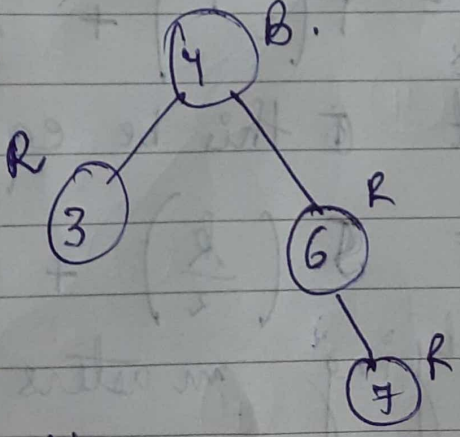
(2) 3 →



(3) 6 →

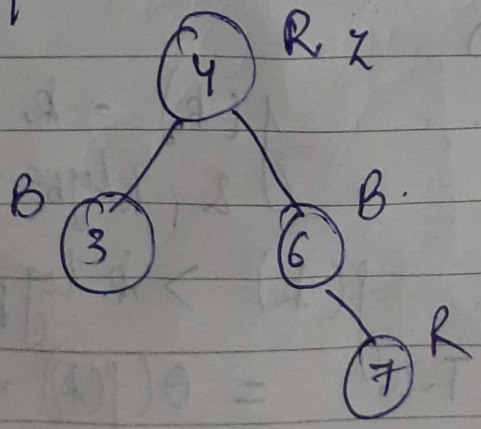


(4) 7 →



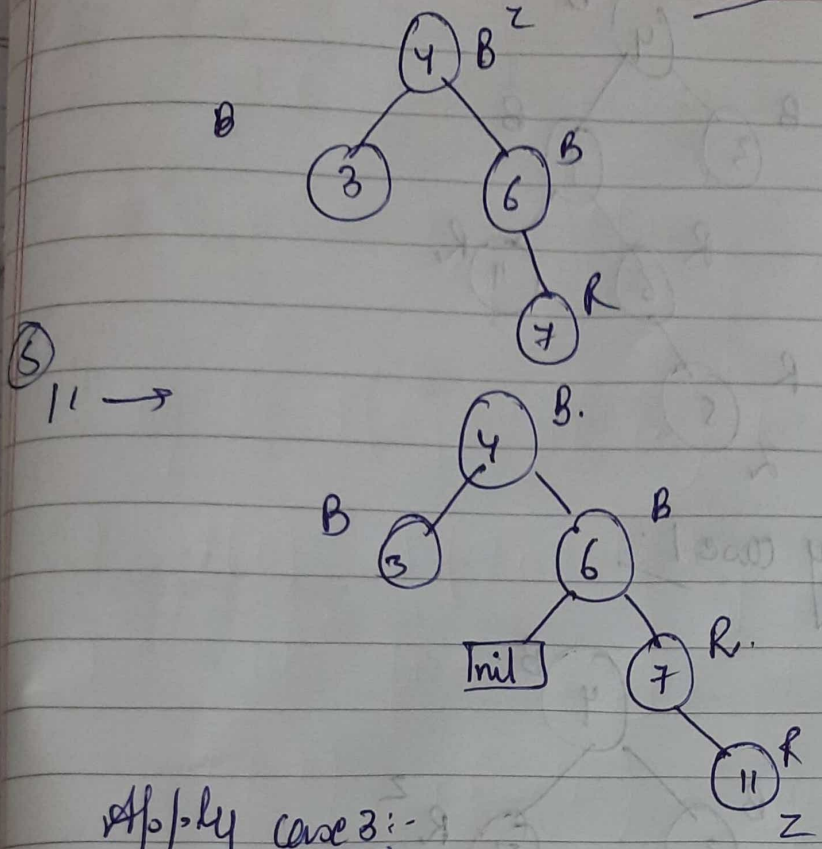
Apply case 1:-

make 3 & 6 Black & grandparent red & make it new z

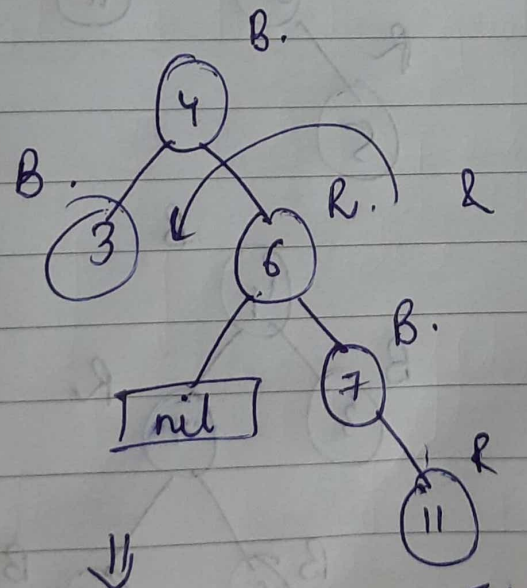


classmate
Date
Page

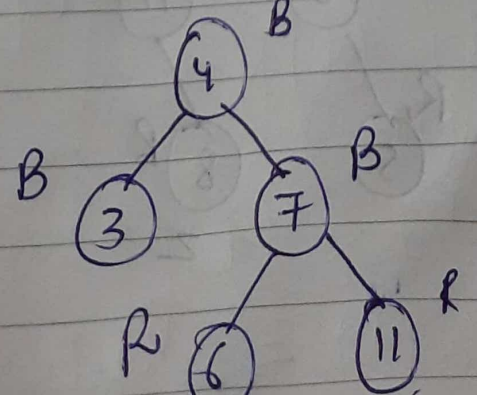
As root node can't be red.



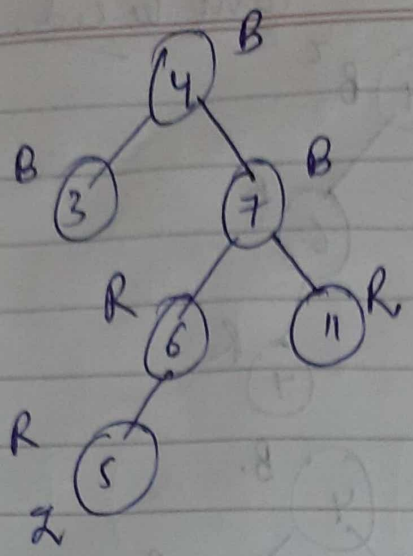
Apply case 3:-
z's uncle is black so make 7 black & 6 red.



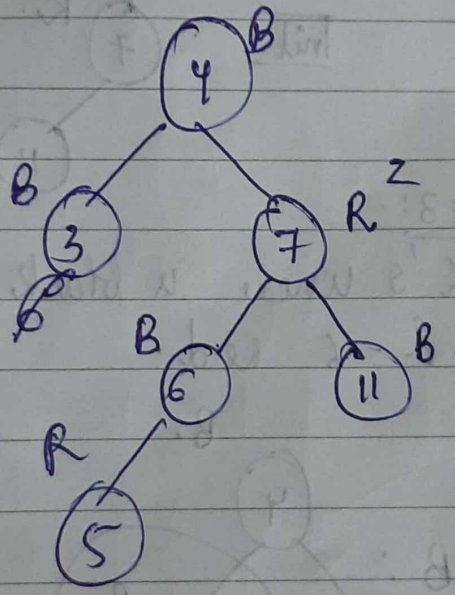
& left rotate around 6.



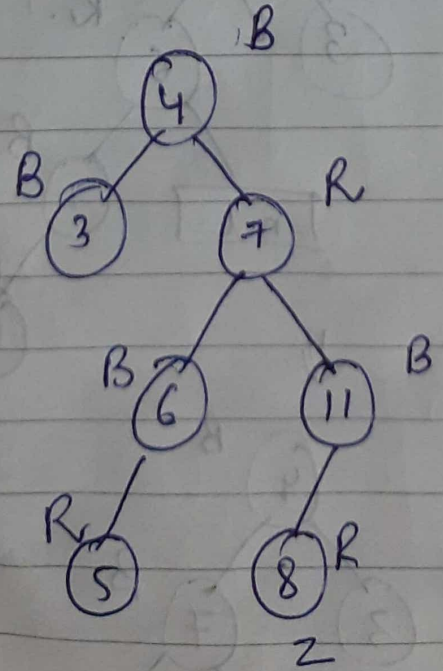
(6) 5→



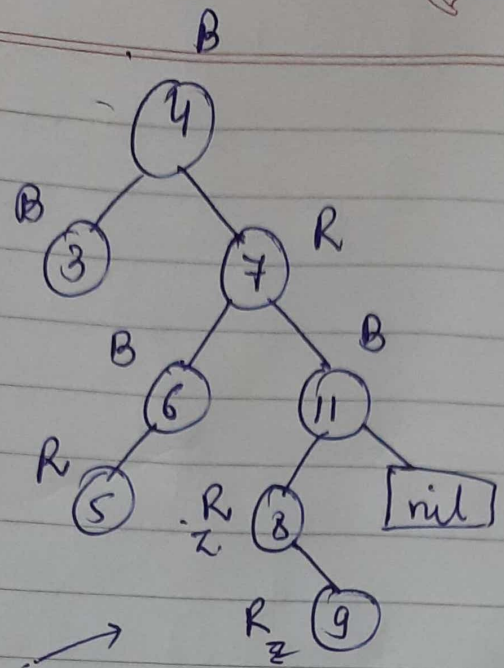
Apply case 1:



(7) 8→

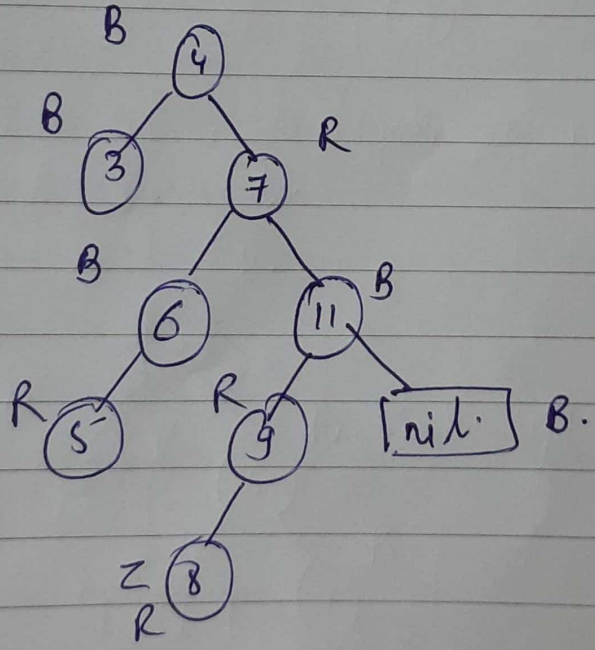


8) g →



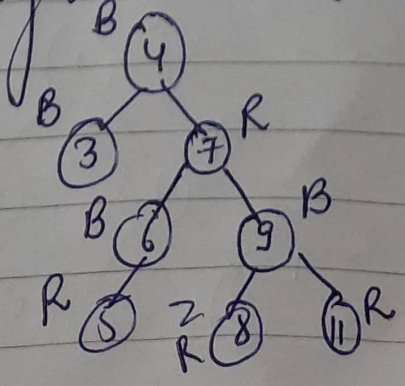
Apply case 2:

Set 8 as new z & left rotate.



Case 3:-

Color node 9 black & 11 red & right rotate around 11.



Malvina Rai
180321308 7.

Q3.

Insert fixup (T, z)

case 1
1. while z ≠ root [T] & color P[z] = red
2. do if P[z] = left [P[P[z]]]
3. then y ← right [P[P[z]]]
4. if color [y] ← red
5. then color [P[z]] ← black
6. color [y] ← ~~red~~ Black
7. color [P[P[z]]] ← red
8. z ← P[P[z]]

case 2
9. else if z = right[z]
10. then z ← P[z]
11. left rotate (T, z)
12. ~~color [z] ← black~~

case 3
13. color P[z] ← black
14. color P[P[z]] ← red
15. right rotate (T, P[P[z]])
16. else (same as then clause)
with left & right exchange
17. color (root [T]) ← black

Q2. R-B tree. (Binary Search Tree)

for every node, left child should be < key value (root) & right child should be > root value.

→ Every node in R-B tree is either red or black.

→ Root node is always black.

→ Leaf nodes are always black.

→ If any node is red, both of its children must be black.

→ **AXX**

For each node, every path from node to leaf contains same no. of black nodes. (all path from node to leaf has same black height).

bh x.

of any node x.

Black height of node:- It is the no. of black nodes including leaf nodes on the path x to leaf but not counting x.

The height should be same for all paths from x to leaf.

Black height of tree:- Black height of root

Use of B-B tree

1. CPU scheduling (fair policy)
2. map fun()
3. set fun()