DAA Assignment

AGRANSH KASTOGI
(2015500010)

1- Solve the following recurrence rel^n:
$T(n) = 4T(n/2) + n^2$

Ans: Recurrence rel^n : $T(n) = 4T(n/2) + n^2$
Comparing with $T(n) = aT(n/b) + f(n)$

$a = 4$ & $b = 2$

$\therefore a \geq 1$ & $b > 1$

$n^{\log_b a} = n^{\log_2 4} = n^2$

$f(n) = \Theta(n^{\log_b a})$

By using Master Theorem:

$T(n) = \Theta(n^{\log_b a} \log(n)) = \Theta(n^2 \log n)$

2- Let $\{a_i\}$ be the sequence given by:
$a_k = a_{k-1} + k$ with $a_0 = 0$. Solve this recurrence rel^n & find a_{100}.

Ans. $a_k = a_{k-1} + k$

We can substitute a_{k-1} in terms of a_{k-2}:

$a_k = a_{k-2} + k-1 + k$

$= a_{k-2} + 2k-1$

We can continue this process, substituting a_{k-2} in terms of a_{k-3}, and so on, until we reach a_0:

$a_k = a_0 + 1 + 2 + \cdots + k$

$= a_0 + \frac{k(k+1)}{2}$

We are given that $a_0 = 0$, so:

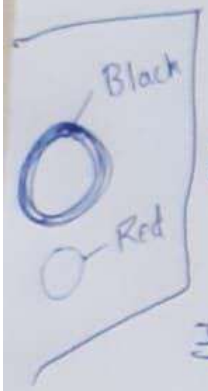
$$\boxed{a_k = \frac{k(k+1)}{2}}$$

To find a_{100} $\Rightarrow$ we can substitute $k = 100$

$$a_{100} = \frac{100(100+1)}{2} = 50(101) = \underline{5050}.$$

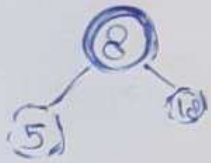
3. Create a Red-Black Tree by inserting following sequence of number:-
8, 18, 5, 15, 17, 25, 40 & 80?

Ans. Inserting 8: Tree is empty. So inserting new node as root node with Black Colour.

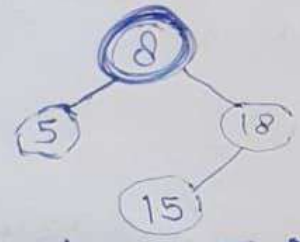


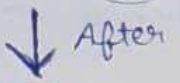
Inserting 18: Tree is not empty. So insert new node with Red Colour.

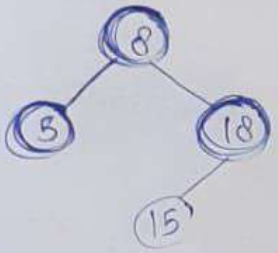
Insert 5: Tree is not empty. So insert new node with Red Colour



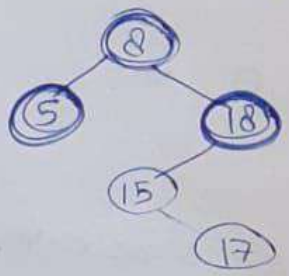
Inserting 15: Tree is not empty. So insert new node with Red Color



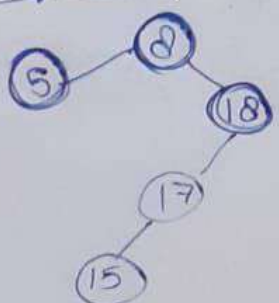




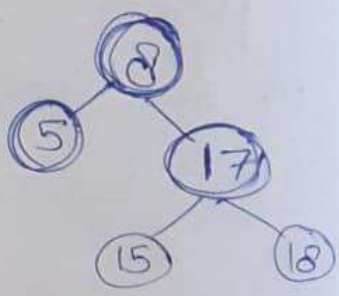
Inserting 17: The Tree is not empty. So insert new node with Red Colour



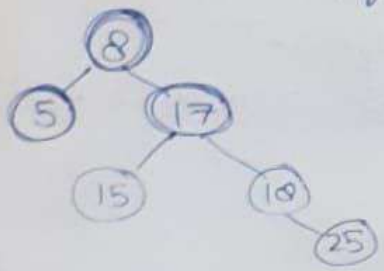
After Left Rotation



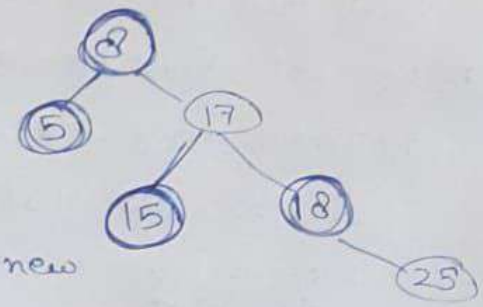
After Right Rotation



Inserting 25: Tree is not empty. So insert new node with Red Color.

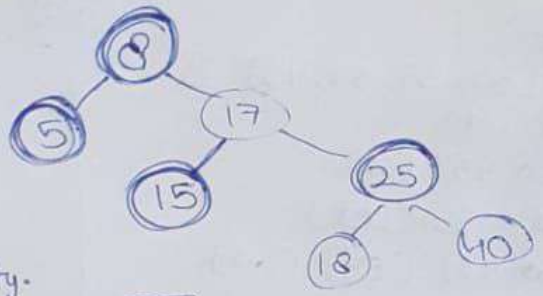


After Recolor

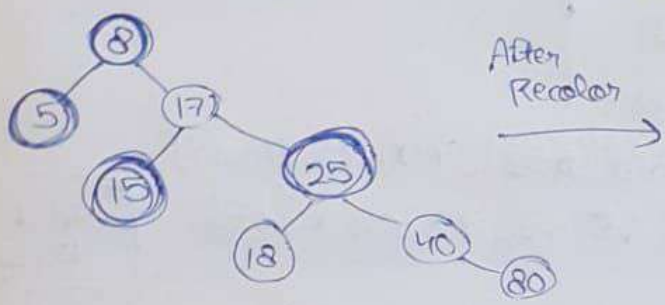


Inserting 40: Tree is not empty. So insert new node with Red Color.

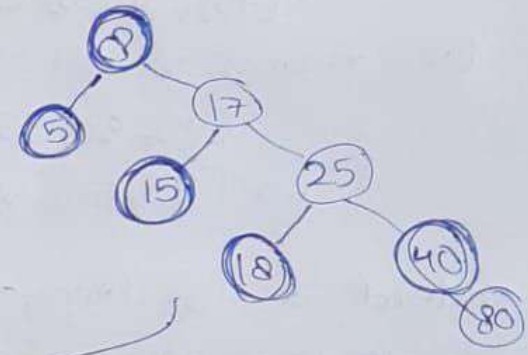
After LL Rotation & Recolor



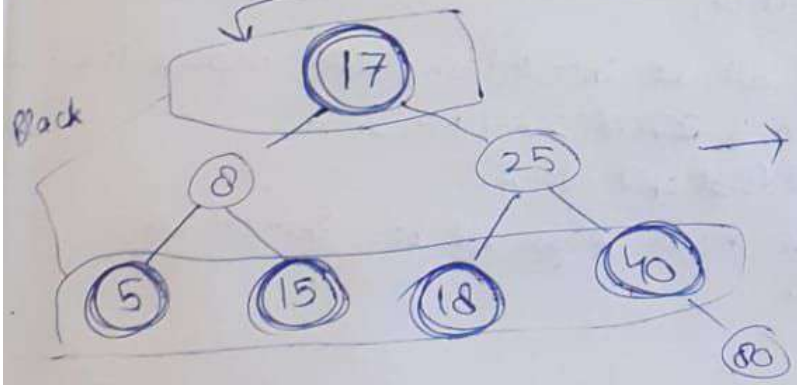
Inserting 80: Tree is not empty.
So insert new node with Red Color.



After Recolor



After Left Rotation & Recolor



Finally, this tree is satisfying all the properties of Red Black Tree & it is a perfect Red Black Tree.

4. Input : 65 70 75 80 85 60 55 50 45 is given. Sort these elements using Quick Sort Algorithm?

Ans: (i) Let Pivot Element = 65 [65, 70, 75, 80, 85, 60, 55, 50, 45]

(ii) Dividing array into 2 sub arrays

[60, 55, 50, 45] [70, 75, 80, 85]

Elements less than Pivot

Elements greater than Pivot.

(iii) Recursively sorting each sub-array. We can apply same steps to each sub-array.

- SubArray [60, 55, 50, 45]
  - Pivot = 60 (Let)
  - [55, 50, 45] [60]
    - Pivot = 55 (Let)
      - [50, 45] [55] [60]
        - Pivot = 50 (Let)
          - [45] [50] [55] [60]

[65]

- Subarray [70, 75, 80, 85]
  - Pivot = 70 (Let)
  - [75, 80, 85] [70]
    - Pivot = 75
      - [80, 85] [75]
        - Pivot = 80
          - [ ] [70] [75] [80] [85]

Now combining all sub-arrays

[45, 50, 55, 60] [65] [70, 75, 80, 85]

⇒ [45, 50, 55, 60, 65, 70, 75, 80, 85] Sorted Array

5. Construct a B-Tree of minimum degree 't' as 3 & a sequence of integers 10, 20, 30, 40, 50, 60, 70, 80 & 90 in an initially empty B-Tree.

Ans.

(i) Initially Root is NULL. Let us first insert 10.

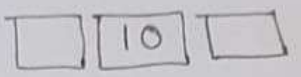
(ii) Now insert 20, 30, 40 & 50. They all will be inserted in root because the maximum no. of keys a node can accomodate is $2n-1$. which is 5.

(iii) Now insert 60. As root node is full, it will first split into two, from 60 will be inserted into the appropriate child.

(iv) Let us now insert 70 & 80. These new keys will be inserted into the appropriate leaf without any split.

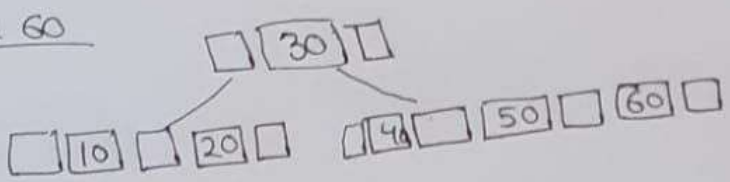
(v) Let us now insert 90. This insertion will cause a split. The middle key will go up to the parent.

(i) Inserting 10

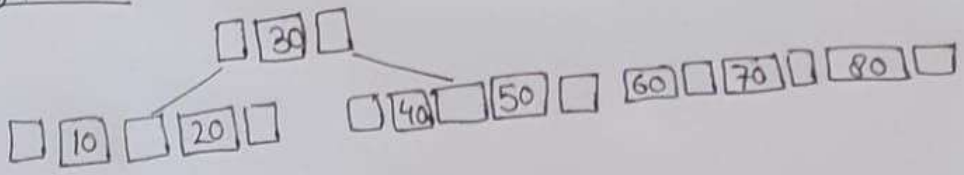


(ii) Inserting {20, 30, 40, 50}

(iii) Inserting 60



(iv) Inserting {70, 80}



(v) Inserting 90