FRANKLIN'S LECTURES

S6 First Series MARATHON

ALGORITHM ANALYSIS & DESIGN

Module 1

CST306

1. Define Big Oh, Big Omega and Theta notations and illustrate them graphically. (7)

Asymptotic notations → Oh, Omega, theta, little, omega

↳ time & space.

1. Big Oh → upper bound → worst case.
2. Big Omega → lower bound → Best case
3. Big theta → tight bound → average case

→ Asymptotic notations — mathematical notations
— n grow how it effects on
time / space requirements.

1. Big Oh
   * Worst case
   * upper bound.
   * $f(n) = O(g(n))$ if there exists two const c, no such that $0 \le f(n) \le c \cdot g(n) \cdot \forall n \ge n_0$

2. Big Omega
   * best case
   * lower bound
   * $f(n) = \Omega(g(n))$ — ——

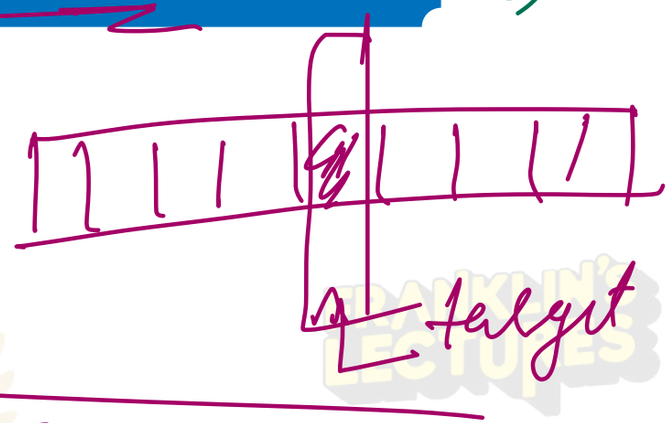
3. Big theta -

$f(n) : \Theta(g(n))$

* Average case
* tight bound.

2. Find the best case, worst case & average case time complexity of binary search.

array → Sorted array.
→ half.



Best case. → target = arr[mid]
→ O(1)
→ Single iteration

Worst case

$\Rightarrow n \to n/2 \to n/4 \to n/8 \cdots$ — best cond^n.

$\Rightarrow \theta(\log_2 n)$

$\hookrightarrow n/2^{k-1}$

$n = 2^k$

$k = \log_2 n$

average case

$\Rightarrow n \to n/2 \to n/4 \cdots$

$\Rightarrow O(\log n)$

Case	Complx.

3. Show that for any real constants a and b, where b > 0, $(n+a)^b = O(n^b)$.

$(n+a)^b = O(n^b)$

$f(n) = O(g(n))$

$f(n) = c \cdot g(n) \rightarrow n^b$

cont

$(n+a)^b \leq c \cdot n^b \quad \forall n \geq n_0 \Leftarrow$ to Show

1. Let $n \geq |a|$

Big-Oh n greater

$n+a$ = Suppose $a=10$

$n=1$ $<$ $n+a=11$

10.

$0 \leq f(n) \leq c \cdot g(n)$

actual value $\rightarrow (n+a)$

(Substitution)

upper limit $\Rightarrow (n+n)$

$\Rightarrow 2n$.

$(n+a) \leq 2n$

power $\Rightarrow (n+a)^b \leq (2n)^b$

$(n+a)^b \leq 2^b \cdot n^b$

$2^b \cdot n^b \rightarrow g(n)$

c.

$f(n) \leq c \cdot g(n)$

$f(n) = O(g(n))$

$(n+a)^b = O(n^b)$

hence proved.

little oh (strictly upper bound).

4. Is $n^2 \in o(n^2)$? Justify your answer.

$$\lim_{n \to \infty} \frac{f(n)}{g(n)} = 0$$

$$= \lim_{n \to \infty} \frac{n^2}{n^2} = 1$$

$f(n) = n^2$

$g(n) = n^2$

← Justification

→ No.

5. Recursion tree method: $T(n)=2T(n/2)+n^2$

$$T(n) = 2T(n/2) + n^2$$

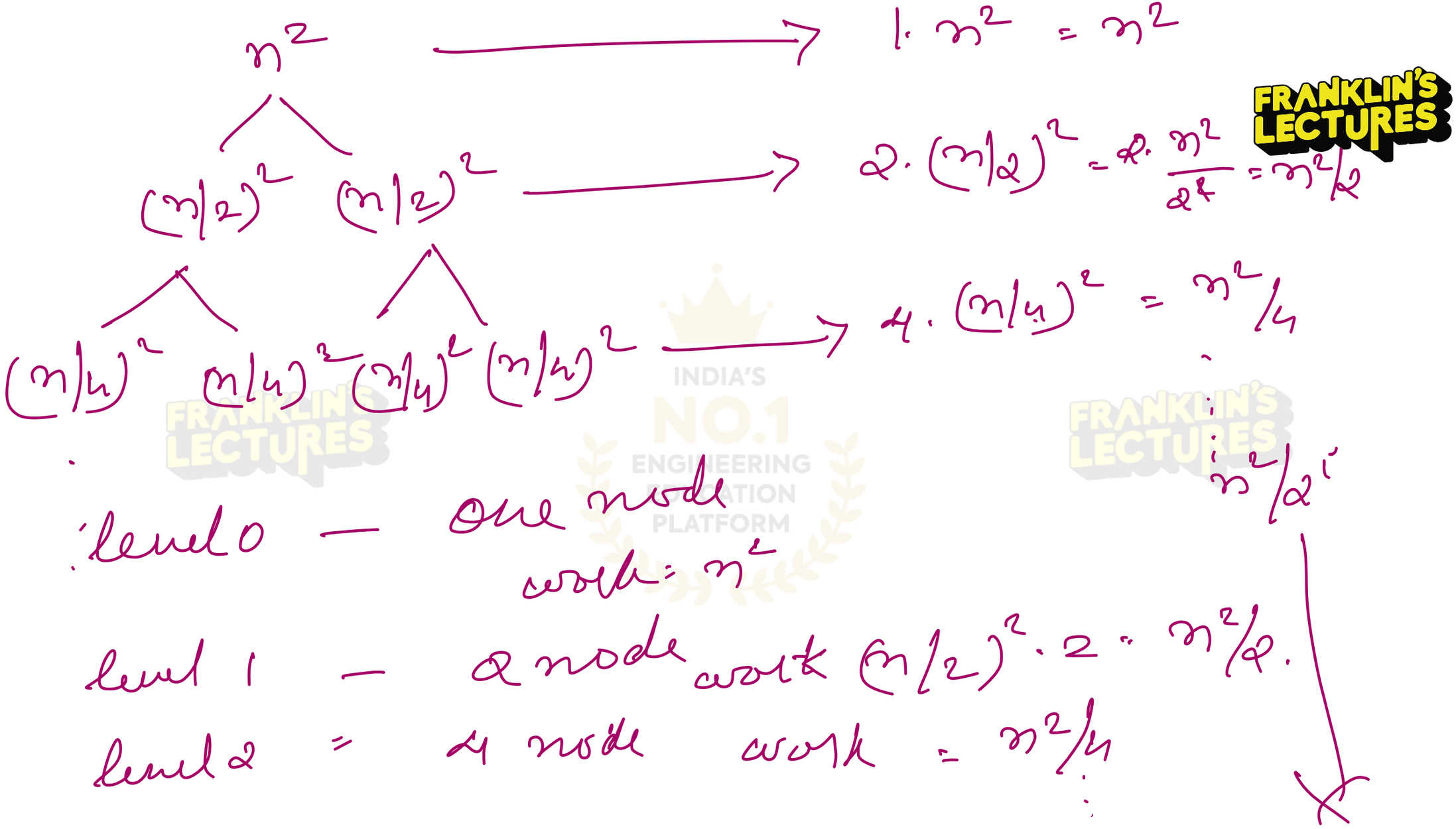
(a) n^2 → amount of work done.

$2T(n/2)$ → This problem is split into 2 sub problems of size $(n/2)$

$$T(n/2) = 2T(n/4) + (n/2)^2$$

$$T(n/4) = 2T(n/8) + (n/4)^2$$

$$\vdots$$


n^2
$1 \cdot n^2 = n^2$
$(n/2)^2$ $(n/2)^2$
$2 \cdot (n/2)^2 = 2 \cdot \frac{n^2}{2^2} = n^2/2$
$(n/4)^2$ $(n/4)^2$ $(n/4)^2$ $(n/4)^2$
$4 \cdot (n/4)^2 = n^2/4$
$n^2/2^i$
level 0 — one node work: n^2
level 1 — 2 node work $(n/2)^2 \cdot 2 = n^2/2$
level 2 = 4 node work = $n^2/4$

Pattern obtain

$n^2 \rightarrow n^2/2 \rightarrow n^2/4 \rightarrow \cdots$

i^{th} level

$\frac{n^2}{2^0} \rightarrow \frac{n^2}{2^1} \rightarrow \frac{n^2}{2^2} \rightarrow \cdots \frac{n^2}{2^i}$

Total cost $\Rightarrow [n^2 + n^2/2 + n^2/4 + \cdots]$

$\Rightarrow n^2 [1 + 1/2 + 1/4 + 1/8 + \cdots]$

GP

$n^2 [2]$

$= 2 \cdot n^2$

GP form = $\frac{a}{1-r}$ → first term, → common ratio

$= \frac{1}{1-1/2} = 1/\frac{2-1}{2} = 1/1/2 = 2.$

$f(n) = 2n^2$ → $g(n)$

↳ c

$T(n) = O(n^2)$

6. Solve the recurrence using Master's theorem:

⇒ $T(n)=3T(n/2)+ n^2$

$T(n) = aT(n/b) + f(n)$

$T(n) = 3T(n/2) + n^2$

$f(n) = O(n^k \log^p n)$

$f(n) = n^2$

$n^2 \times 1$

$a = 3$

$b = 2$

$k = 2$

$p = 0$

$a = 3 \quad b^k = 2^2 = 4$

$3 < 4$ $\quad a < b^k \Rightarrow$ Case 3. ✓

2. 1. $P \geq 0$

2. $P < 0$

case 3(1)

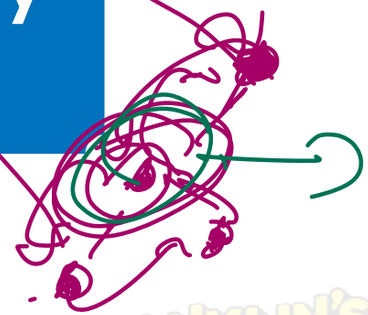
$\Rightarrow \Theta(n^k \log^p n)$.

$\Rightarrow \Theta(n^2 \log^0 n)$

1

$T(n) = \Theta(n^2)$

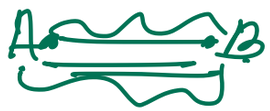
7. Give Breadth First search algorithm for graph traversal. perform its complexity analysis



1. Create an empty queue
2. enqueue the starting vertex & mark it visited.
3. while the queue is not empty
   - 3.1 dequeue vertex v
   - 3.2 foreach unvisited neighbour u of v mark as visited.

1. Time complexity →

$O(V+E)$



2. Space complexity

$O(V)$

BFS / DFS

FRANKLIN'S LECTURES

S6 First Series MARATHON

ALGORITHM ANALYSIS & DESIGN

Module 2

CST306

8. Explain the different operations possible on disjoint sets. Implement UNION using the linked list representation of disjoint sets.

→ Collection of non-overlapping sets.

1. Make_Set(x) — Create a new set with only one element
2. Find(x) — Root of the set where x is an element
3. Union(x,y). — Merge the sets containing x & y into one

Make set (3)
= {3}

linked list representation of UNION

→ list
→ next pointer
→ root pointer

$S_1 = \{A, B\}$ $S_2 = \{C, D\}$

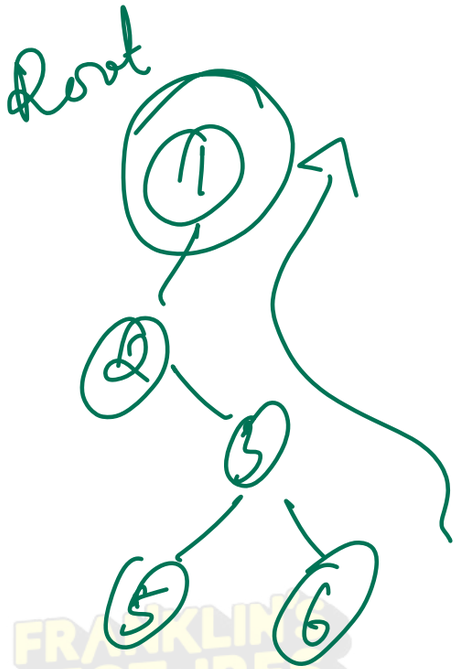
A → C

UNION(S_1, S_2)

= A → B → C → D

↓ A

weighted union — S_1 → 2 elements
S_2 → 100 elements



find(6) => 1

9. Construct an AVL tree by inserting the following elements in the given order: 21, 26, 30, 9, 4, 14, 28, 18, 15.

⇒ 4 rotations

AVL - Binary search tree
→ Balanced.

BF: Height of left subtree − Height of right subtree.

∴ {−1, 0, +1}

21, 26, 30, 9, 4, 14, 28, 18, 15

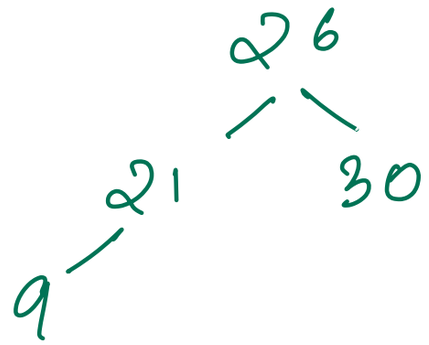
Insert 21 → 21

Insert 26 → 21 — 26

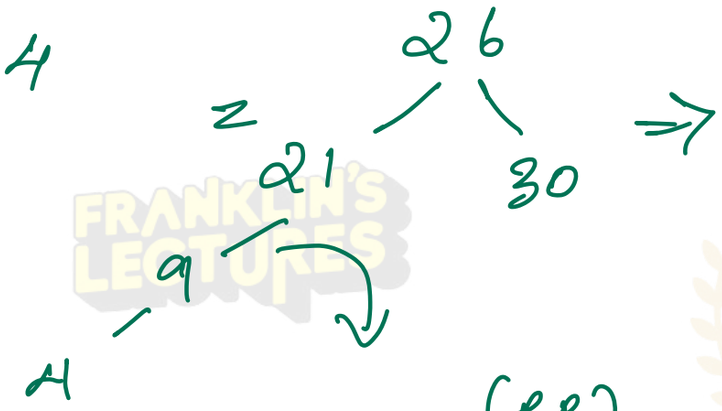
Insert 30 → 

RR imbalance ⇒ RR rotation

insert 9

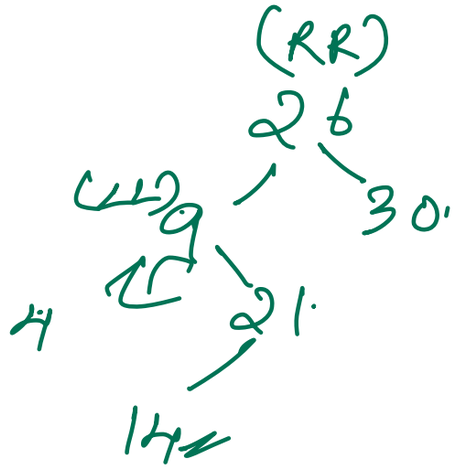


insert 4

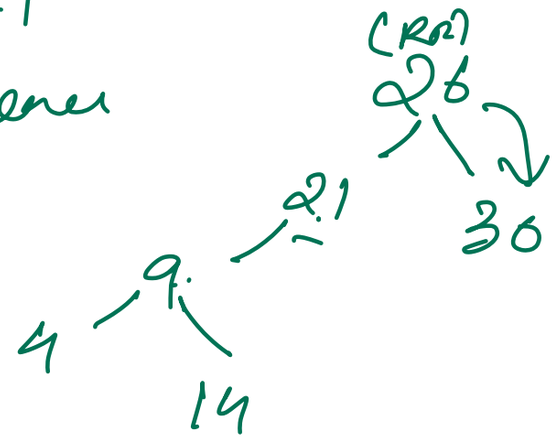


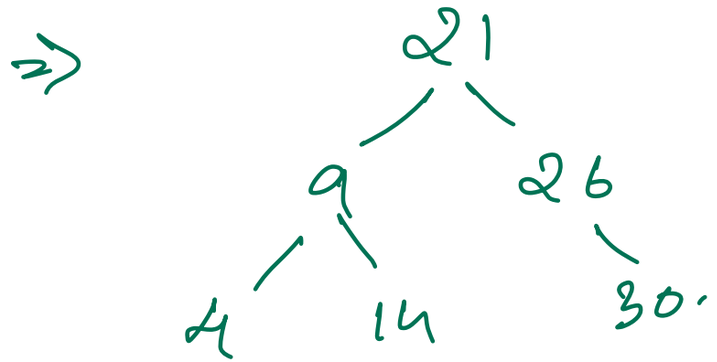
LL imbalance => LL rotation

insert 14

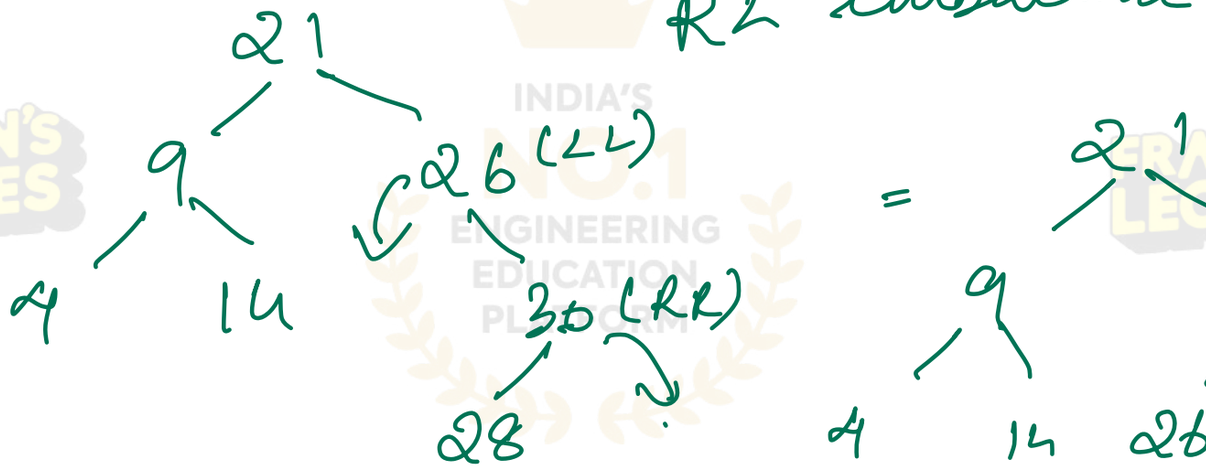


LR imbalance





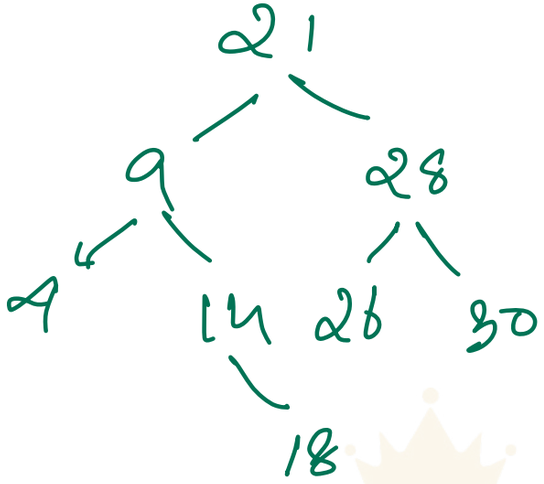
insert 28



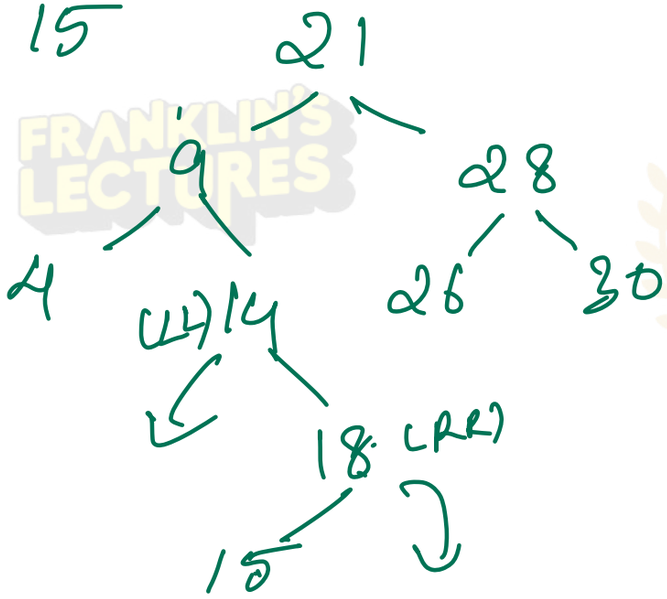
RL imbalance

=

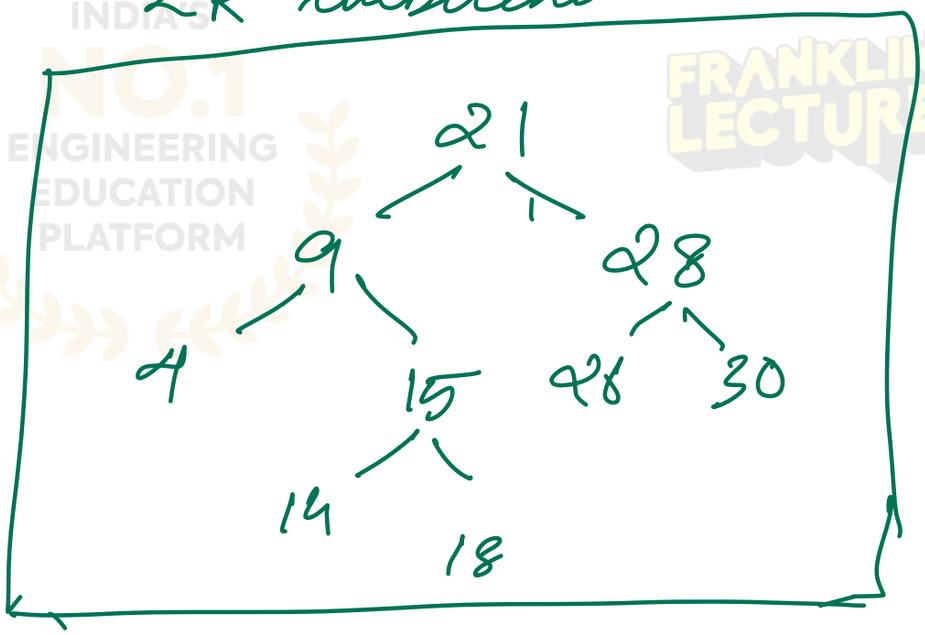
insert 18



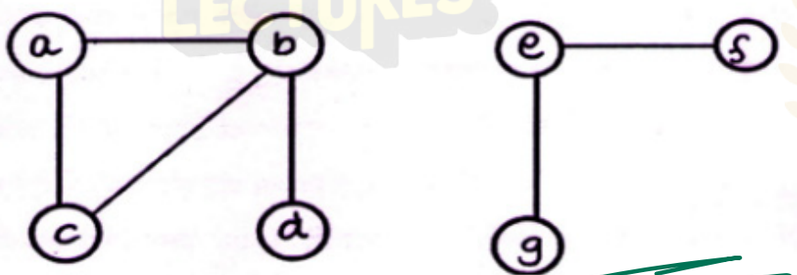
insert 15



LR imbalance



10. Give an algorithm for computing connected components in a graph using Disjoint Set data structure operations. Explain the working of this algorithm on the graph given below



⇒ Strongly connected components

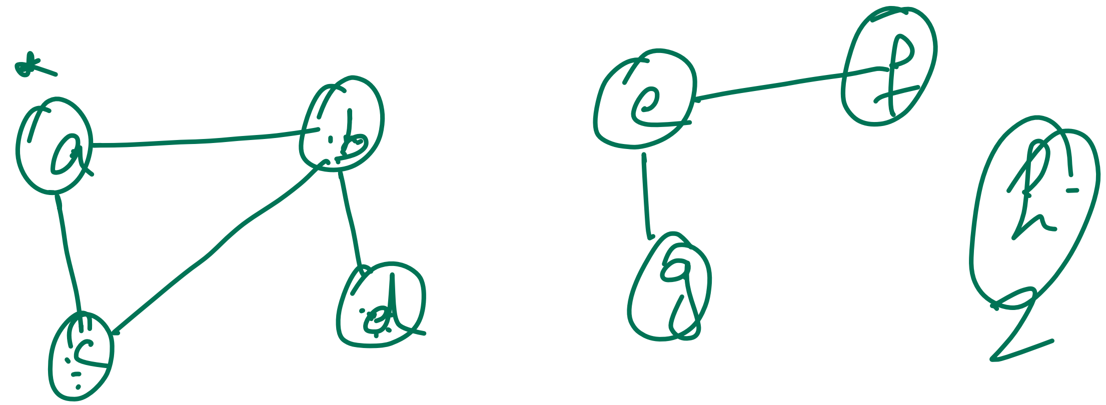
make set(n) :-

1. Elements initiate
2. process edges.
3. Find(x)

Group1: {a, b, c, d} ✓

Group2 = {e, f, g} ✓

Num of unique leaders = Num of groups.



find(a) : a
find(b) : a
find(c) : a
find d : a

} = 1

find e : e
find f : e
find g : e

} = 2.

FRANKLIN'S LECTURES

Step ahead with the right guidance

- Live + recorded classes
- Expert faculty pool
- Digitally illustrated study materials
- Free crash courses
- 24/7 doubt clearance & counselling support
- Self Evaluation Tests

EMI options available

GATE INTEGRATED

S6 SINDOOR BATCH

SCAN QR

9074 745 741
9633 962 277

Watch Now

Click Here

FRANKLIN'S LECTURES

THANK YOU