Queue :

↳ CPU scheduling, Ready Queue, Device Resource Sharing.

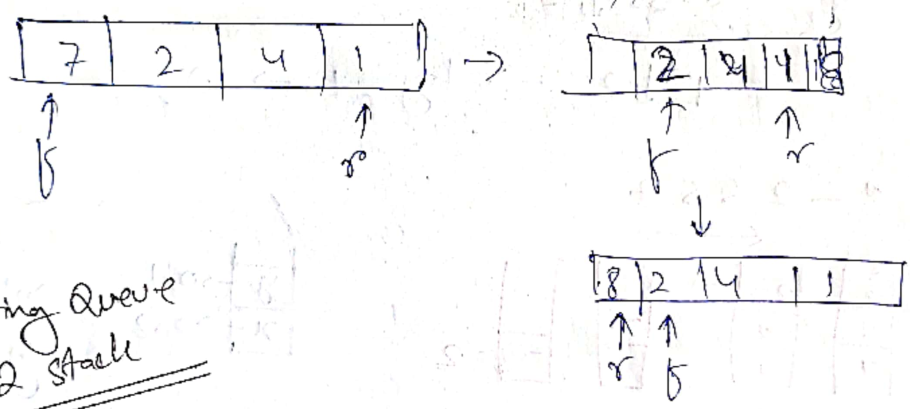
Empty
↳ front = -1 & rear = -1

full
↳ rear = Max - 1

In Circular Queue

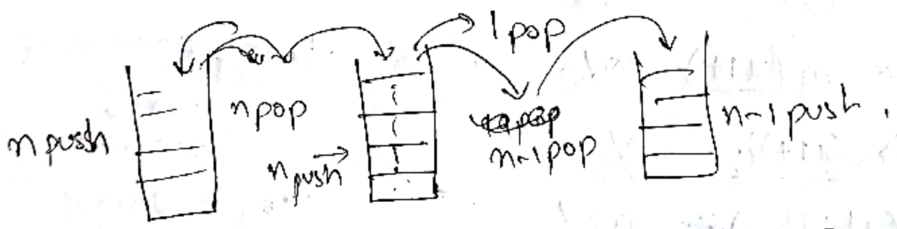
☆ rear = (rear + 1) % Max

(front == (rear + 1) % Max) || (front == 0 && rear == Max - 1)



Implementing Queue using 2 stack

n – enqueue & 1 Dequeue



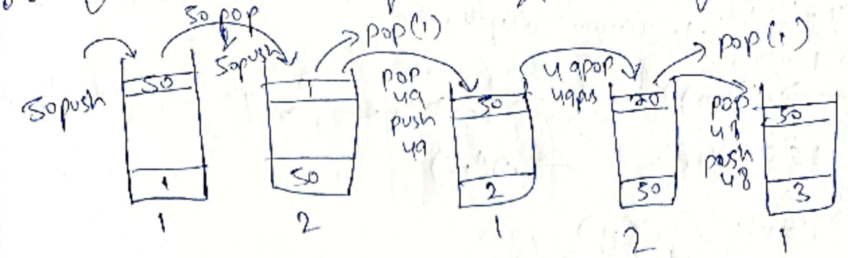
∴ Enquee = n + n + n-1 = 3n-1 push

Dequeue = n + 1 + n-1 = 2n pop.

Rule

1. push → S1
2. pop & push to S2
3. only 1. pop for Deque
☆ 4. Pop all from S2 & push to S1 → For stability

∴ for 50 enqueue & 2 Dequeue ∴



∴ Enqueue = 50 + 50 + 49 + 49 + 48 = 246

Dequeue = 50 + 1 + 49 + 49 + 1 + 48 = 198

444.

Deque.

Double ended Queue.

Insert rear | Delete front
Delete rear | Insert front

Priority Queue :

Inserted in any order but deletion will be based on priority.

→ Heap Max
→ Heap Min

Linked List

Array is fixed length.

For insertion. → ordered Array → shift + insert
Unordered " → insert at end.

For deletion → Ordered Array → Delete & shift
Unordered Array → Delete & shift last element to that position

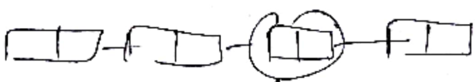
Null pointer dereferencing

The head pointer is null.
and we are try to access the data & next results error.

return middle element

for odd, as usual

for even,



middle element 2

Suppose 2 set

① Union → 1 → 2 → 4 + remove duplicate $\to O(n)$ $\to O(n^2)$ $= O(n^2)$
3 → 1 → 5

② Intersection → $O(n)^2$ 1 → 2 → 4, 3 → 1 → 5 → Finding duplicate

③ membership → YES? YES? → $O(n)$

④ Cardinality → Count $O(n)$

⑤ ca

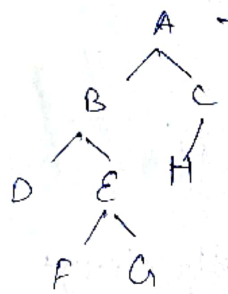
Tree

- Tree is set of nodes & edges
- Acyclic graph
- Tree having n node always has n-1 edges.
- Tree is a connected graph.

• Recursion Linear (factorial)
• Fibonacci series (Non-Linear)

Rooted Tree: one node is selected as root

Except leaf node, All are Internal node including Root node.



hight → 0
1
2
3

hight of tree : longest root to leaf path (no. of edges) → 3

E is at Depth 2
G is at Depth 3

* Ancestor : E's ancestor is B A (parent & Above in path.)

* Descendant : B's descendant is E F G (child & blow in path.)

General Tree → In which each node is having 0 or more children.

Forest
↳ It is collection of 1 or more general Tree.
& each genral tree called components

☆ No. of edges in foreset having P components & n nodes
Total = (n - P edges)

K-ary tree → 0 or K children

★ If L is no. of Leaves & I is no. of Internal nodes

$$Leaf = I(k-1) + 1$$

Degree of Node: no. of edges its connected to.

Handshaking algo → Sum of degree = 2 × no. of edges (simple graph)

I is internal node, degree of Internal = k+1 (with exception (root)).

Total nodes = L + I

Total no. of edges = L + I − 1

Handshaking Theorem.

$$\underbrace{L \times 1}_{\text{leaf degree}} + \underbrace{I(k+1) - 1}_{\text{Internal node degree}} = \underbrace{2(L + I - 1)}_{2\,|Edges|}$$

← for root.

Complete n-ary tree. each node, n children..

L = 41, I = 10, n = ?.

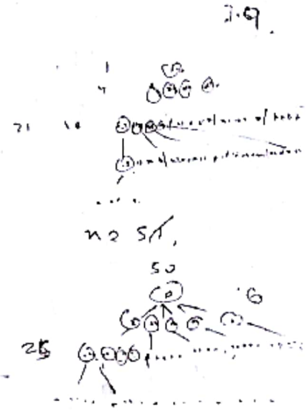
A. 3   C. 5

B. 4   D. 6

$$L = I(n-1) + 1$$

$$41 = 10(n-1) + 1$$

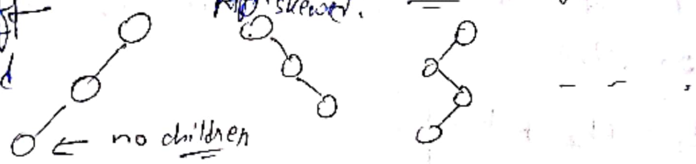
$$4+1 = n \Rightarrow n = 5$$



Binary Tree: 0, 1 or 2 children.

Skewed Binary Tree: In which each node is having 1 children except 1.(Leaf node).

~~Right~~ Left Skewed   ~~Right~~ Skewed.

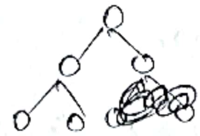


← no children

How many distinct Skewed Binary Tree possible

★ $\frac{1}{n+1} \, {}^{2n}C_n$ n = no. of node.

Full Binary tree. (0 or 2)

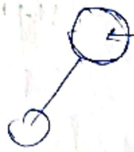


~~All leafs are at same height~~

Minimum Height of Binary Tree with n node.

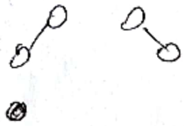
$$\lfloor \log_2 n \rfloor$$

Counting Unlabelled Tree



no data. unlabelled tree.

no. of unlabelled trees possible? 1 2 5 14 42 →

1. $n=1$ $T(1) = 1$
2. $n = 2$ $T(2) = 2$

3. $n = 3$ $T(3) = 5$

L	R	
2	0	→ 2
0	2	→ 2
1	1	→ 1

→ 5.

4. $n = 4$. $T(4) = 14$

L	R	
3	1	→ 5
1	3	→ 5
2	2	→ 2×2

→ 14

5. $n = 5$ $T(5) = 42$

~~L~~	~~R~~
~~4~~	~~0~~
~~0~~	~~4~~
~~3~~	~~1~~
~~21~~	~~3~~

$\therefore \frac{1}{n+1} \, {}^{2n}C_n$ Catalan No.

Binary Tree.

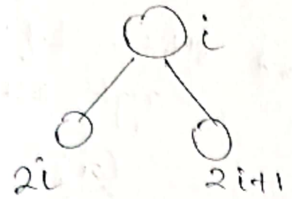
In any Bintree if N_0 is no. of leaf node, N_1 is of no. of node with 1 children & N_2 is No. of node with 2 children.

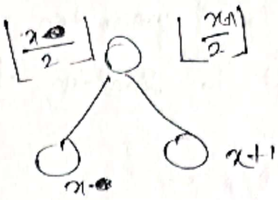
$$\boxed{N_0 = N_2 + 1}$$

Binary Tree Representation

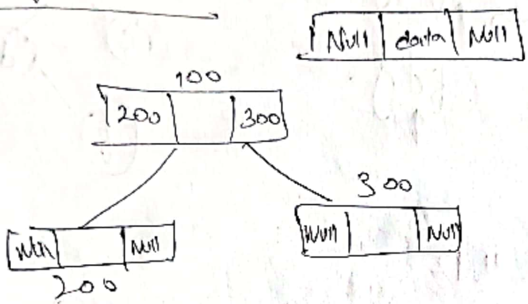
Heap : Sequential Representation

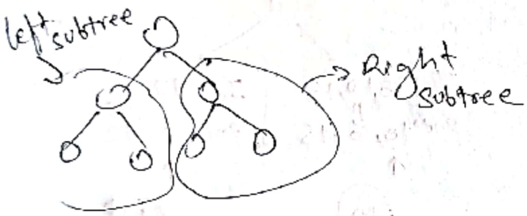
↳ An array is used to store Binary Tree





node Representation

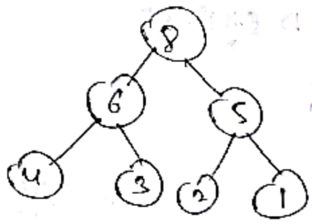




Tree structure is Recursive

Binary Tree Traversal

1. Inorder "
2. Preorder "
3. Postorder "



1. Inorder : Left, Root, Right

4 6 3 8 2 5 1

Root Root

Root

2. Preorder : Root Left Right

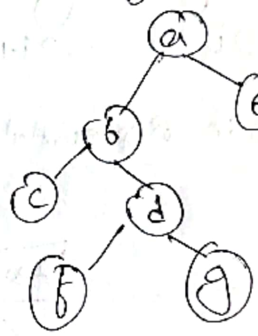
8 6 4 3 5 2 1

Root Left Right

a b c d f g e ← Preorder.

a (c b f d g) a (e) ← Inorder.

1. Find root in preorder.
2. Take the root in Inorder traversal and split in Left and right subtree

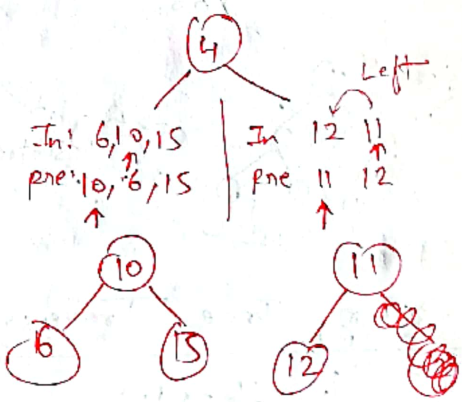


Preorder
b c d f g → d f g

Inorder. (c) b (f d g) → (f) d (g)

Preorder : 4, 10, 6, 15, 11, 12

Inorder : (6, 10, 15) 4 (12, 11)



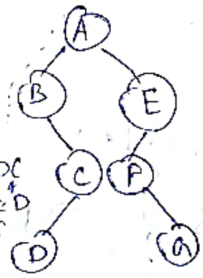
Pre : A B C D E F G

In : B D C A F G E

Post : ?

In: B D C
Pre: B C D

In: D C
Pre: C D

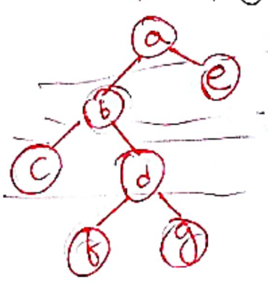


In : F G E
Pre : E F G

∴ Post : D C B G F E A

Post : c f g d b e a

Inorder : c b f d g a e



0
1
2
3

~~X~~ hight = ? 3

Y no. of ~~leaf~~ node in Left subtree 5

~~X~~ Z, no. of node in Right sub 1

$10x + 2^y + 3z$?

$10 \times 3 + 2^5 + 3 \times 1$

$30 + 32 + 3$

65 Ans

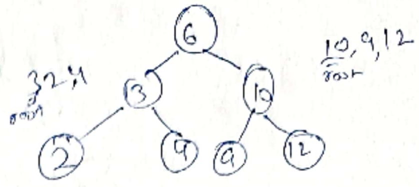
BST

↳ Inorder is Increasing Order Sequence (Unique)

Preorder.

6 (3 2 4) 10 9 12

root ↓ less ∴ left ↓ more ∴ right →



Post order

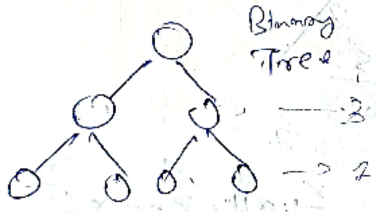
(2, 4, 3) (9, 12, 10), 6

3 root, 10 root

less ∴ left, more Right, root

adbefgh is post order traversal of full binary Tree

Can we constructed the tree ?



∴ adbefgh — Left, right, root.

∴ can be made//

3
7
15
31
63
127

$\boxed{2^{h+1} - 1}$

★ BST : Inorder : Cant be Constructed.

2-ary Tree : Preorder & post order traversal we can Construct Tree Uniquely.

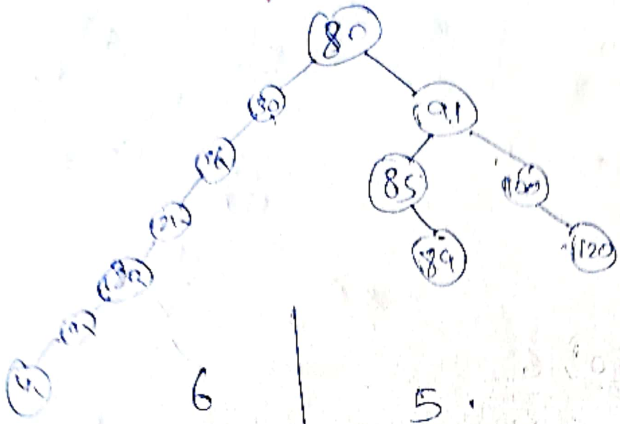
Time Complexity of Search operation in BST

$\to O(h)$ — skewed $O(n)$; balanced $O(\log_2 n)$

O(h) { Left most Node: Min element (BST) ; Right " " : Max " (BST) } → Not always Leaf.

Q. BST constructed with key value: 4, 9, 20, 21, 26, 30, 91, 89, 85, 89, 100, 120.

with 80 as Root of BST. The no. of BST possible is ______.



6 | 5.

$\frac{1}{6+1} {}^{6\times2}C_6$ | $\frac{1}{5+1} {}^{5\times2}C_5$

$\frac{1}{7} {}^{12}C_6$ | $\frac{1}{6} {}^{10}C_5$

$\cancel{\frac{1}{7} \times \frac{12\times11}{2} = \frac{66}{7}}$ | $\cancel{\frac{1}{6} \times \frac{10\times9}{2} = \frac{45}{6}}$

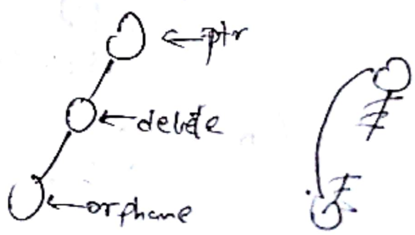
$\frac{1}{7} \times \frac{12\times11\times10\times9\times8\times7}{6\times5\times4\times3\times2}$ | $\frac{1}{6} \times \frac{10\times9\times8\times7\times6}{5\times4\times3\times2} = 42$

132

$\underbrace{132 \times 42}_{\text{Combination}} = 5544$

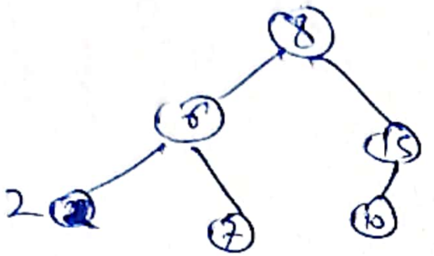
Deletion in BST

1. Deleting Leaf node: Direct Delete.
2. Deleting node with 1 child: hold the parent of the node to be deleted and connect the orphane child of the deleted node to the parent of deleted node



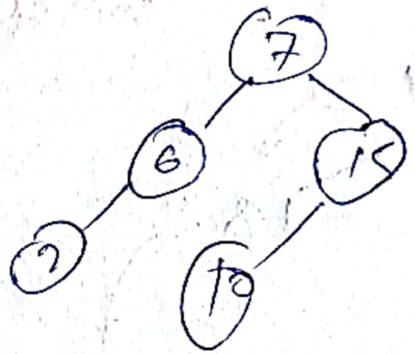
(3) Deleting node with 2 children:

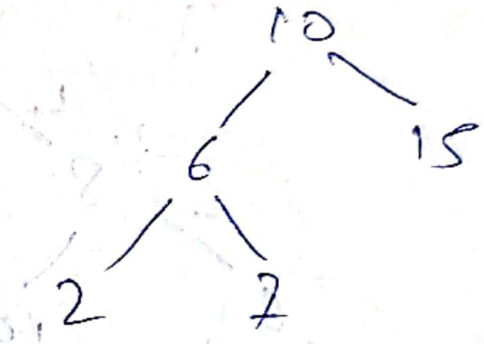
The node is replaced with, Inorder predecessor or Inorder successor.



Inorder = 2 6 7 8 10 15

↑ predec... ↑ successor.





Inorder min don't have predecessor

max " " successor.