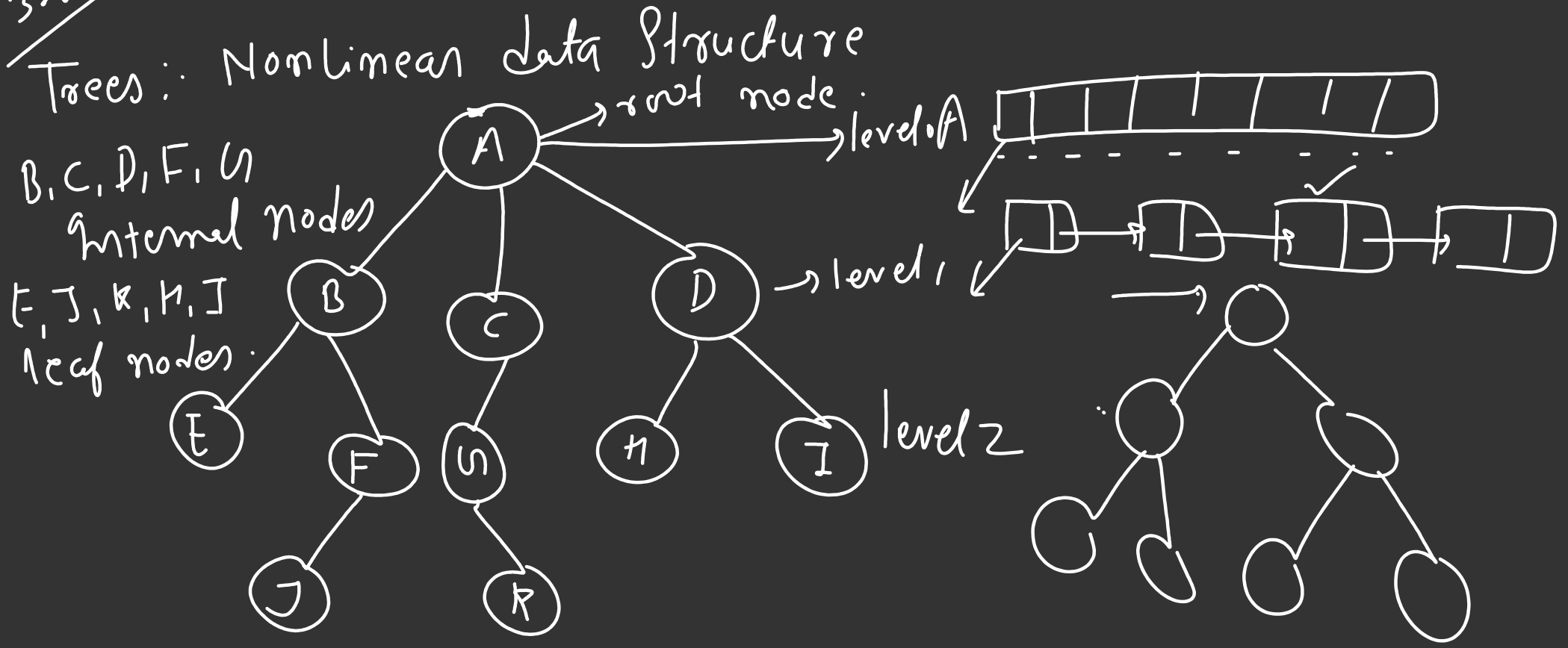
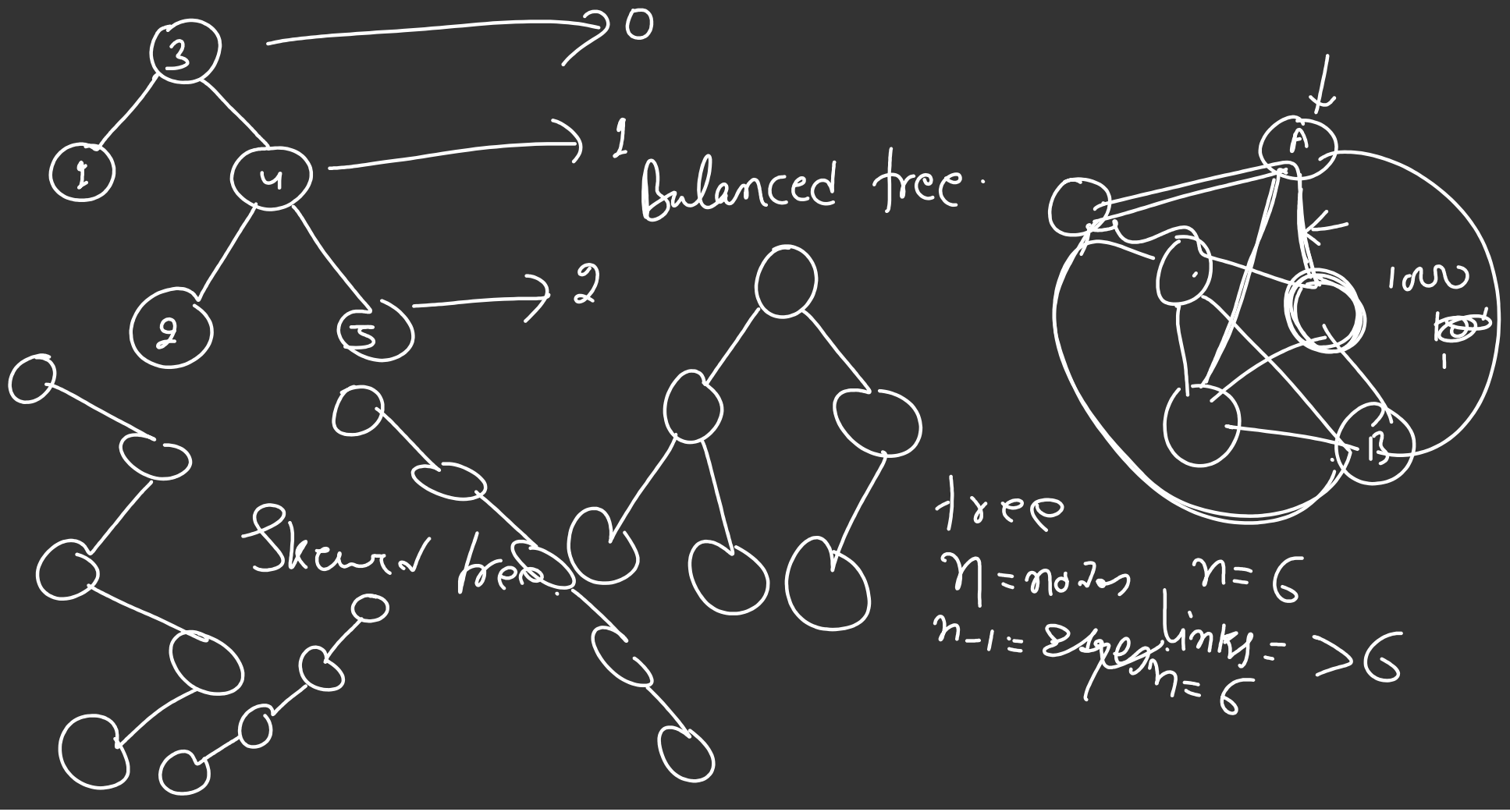
Date
31/01/23

§Data Structures§

N-ary trees.

Trees :: Nonlinear data Structure

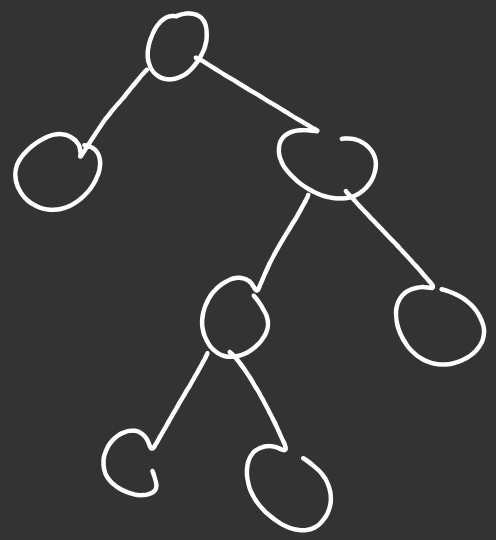



3
1
4
2
5
0
1
2
Balanced tree.
Skewed tree.
tree
n = nodes
n-1 = Edges
n = 6
links = >6
n = 6
A
B
1000

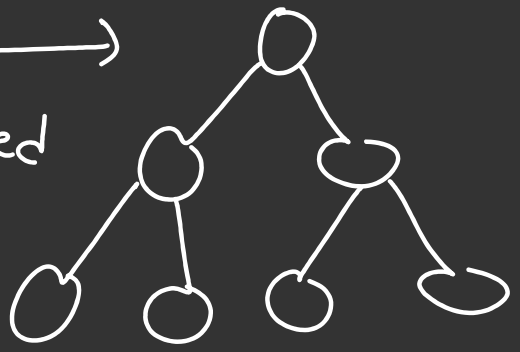
§ Binary tree:: A tree is a binary tree if each node has zero, one or two child.

Types of binary trees::

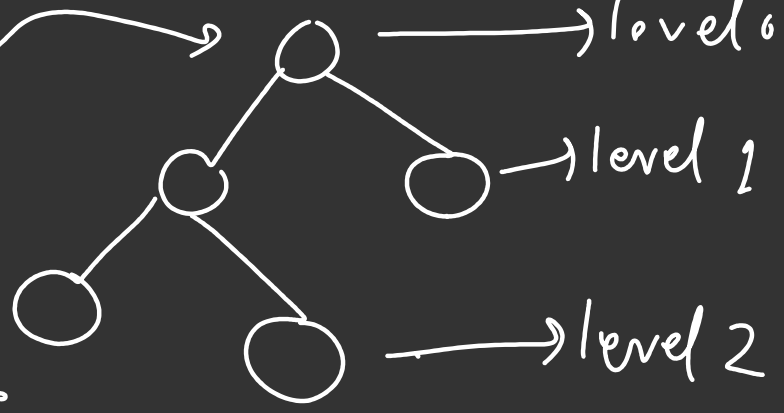
1) Strict binary trees:: A binary tree is called strict binary tree if each node has exactly two children or no children.



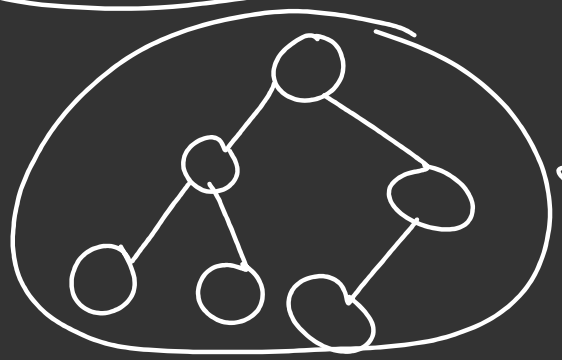
2. Full binary trees :: A binary tree is called full binary tree if each node has exactly two children and all leaf should at the same level



3. Complete binary trees :: A binary tree is called Complete binary tree if Each node has Exactly two children. Right Side can't be fill without filling left



n = 6



← Almost Complete binary tree.

tree

h

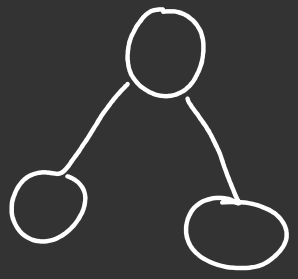
0

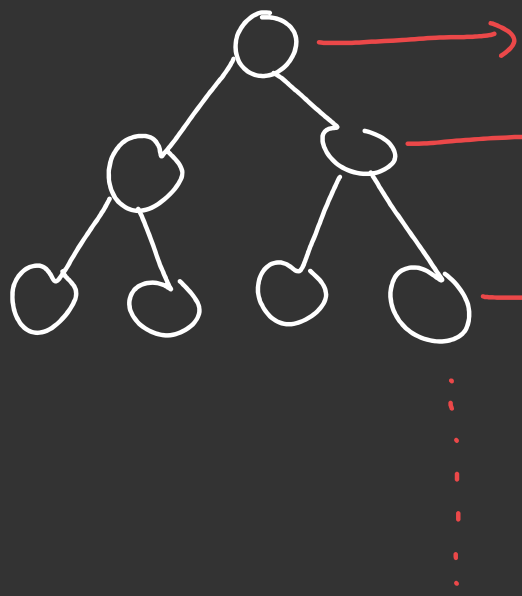
1

n

1

3





$n = 7$

7

$2^0 + 2^1 + 2^2 + \cdots + 2^h = n$

$2^{h+1} - 1 = n$

$2^{h+1} - 1 = 7$

$2^{h+1} = 8$

$h+1 = \log 8$ $\Rightarrow$ $h+1 = 3$

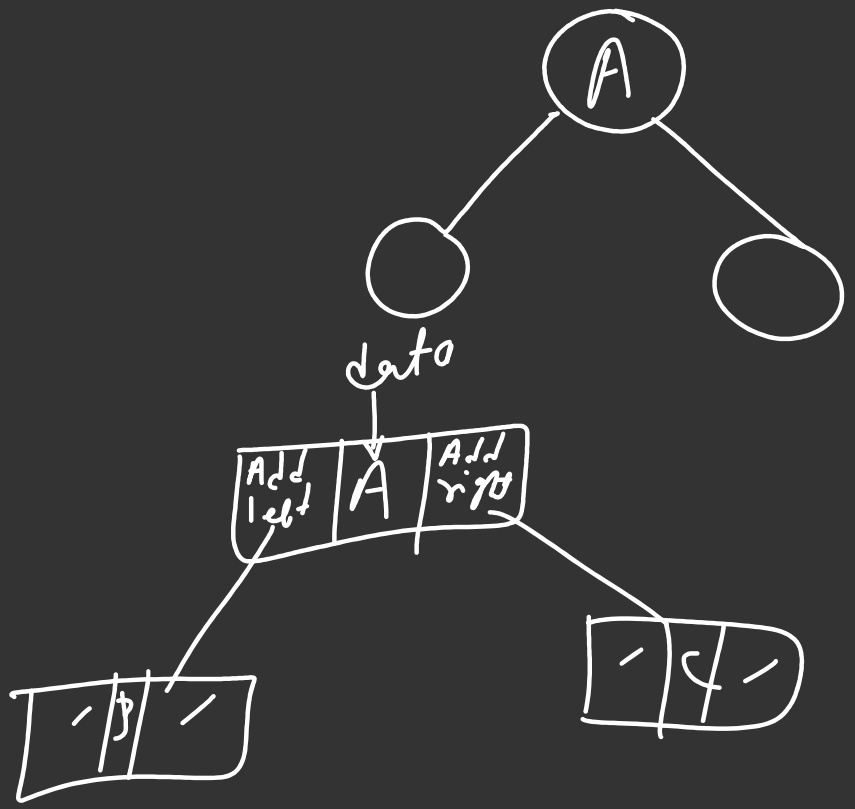
$h = 2$

§ Structure of binary tree::

```
Struct BinaryTreeNode
{
    int data;
    Struct BinaryTreeNode *left;
    Struct BinaryTreeNode *right;
}
```

operations on binary trees::

→ Insertion
→ Deletion
→ Traverse

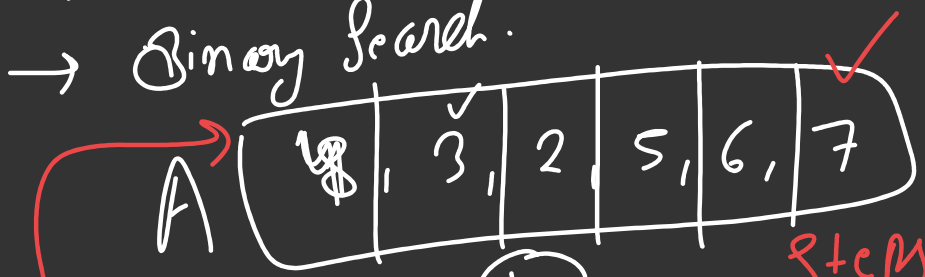


Applications of trees:

→ Used in Compilers

→ Data Compression

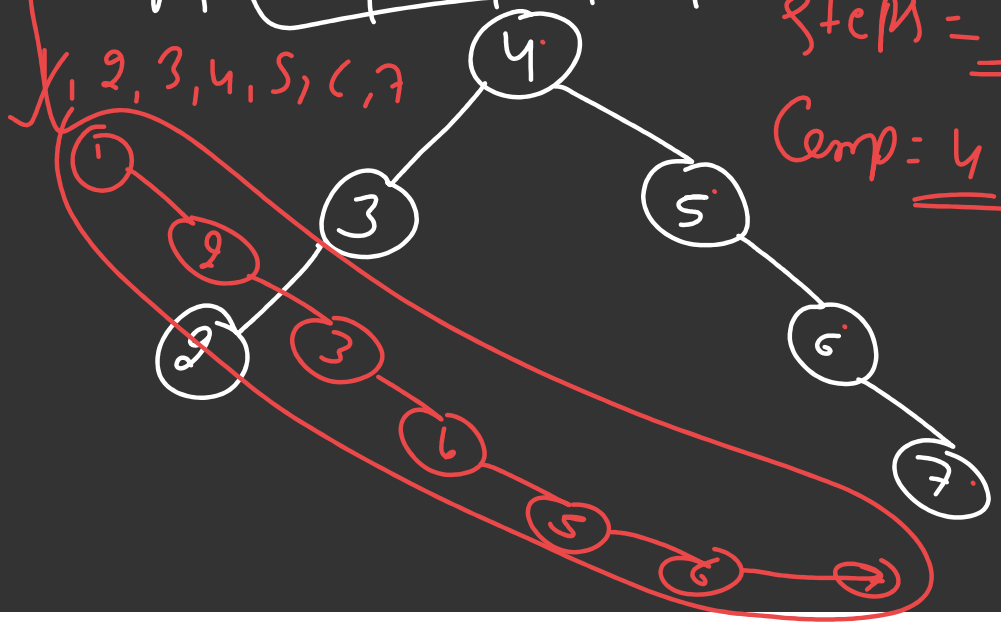
→ Binary Search.

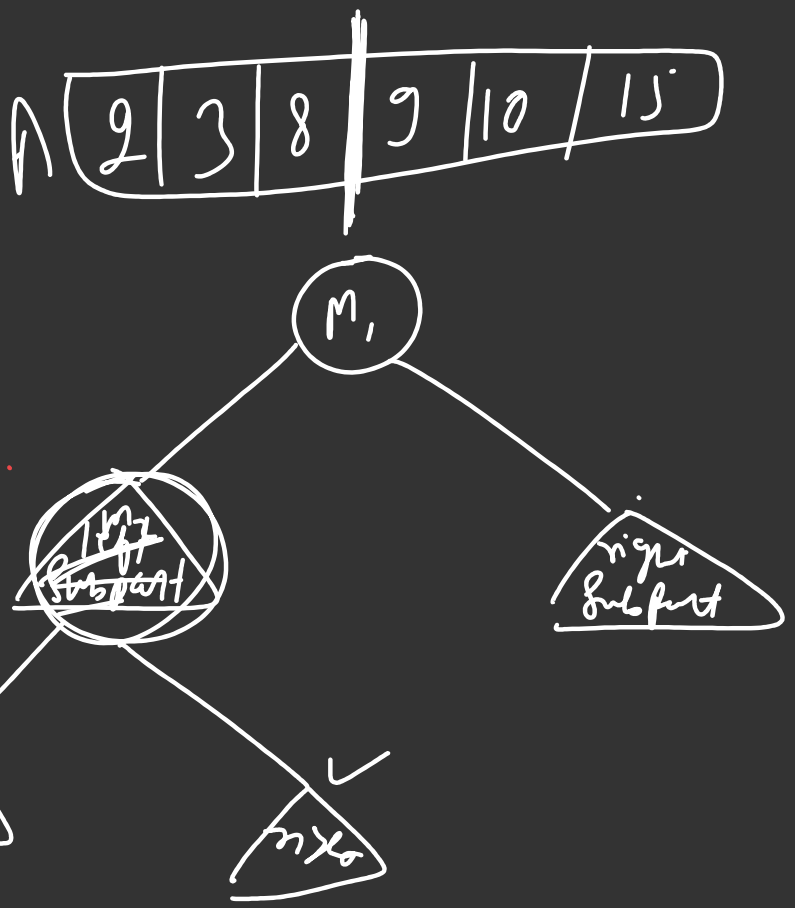


O(n)

Steps = 6 Comp.

Comp: 4

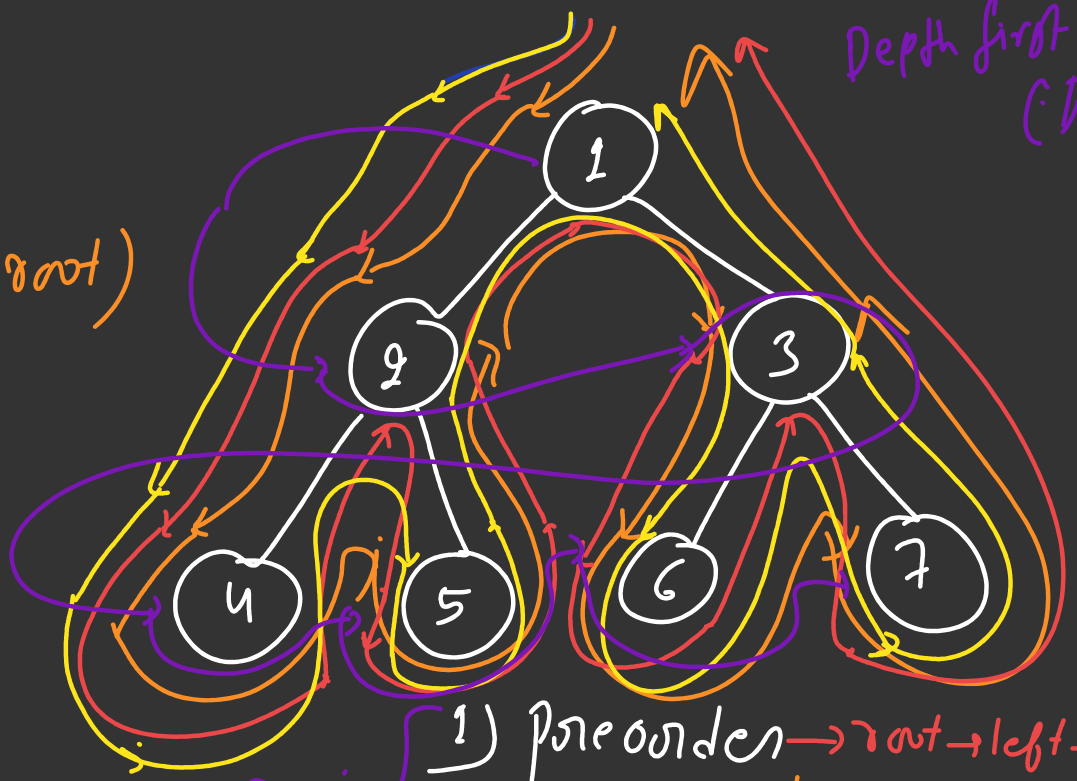




Traverse ::

Preorder :: 1, 2, 4, 5, 3, 6, 7

```
Preorder(Struct BinaryTreeNode *root)
{
    Printf("%d", root->data)
    Preorder(root->left)
    Preorder(root->right)
}
```



Depth first Search (DFS)

DFS

1) Preorder → root → left → right
   1, 2, 4, 5, 3, 6, 7
2) Inorder → left → root → right
   4, 2, 5, 1, 6, 3, 7
3) Postorder → left → right → root
   4, 5, 2, 6, 7, 3, 1

BFS

4) level order → 1, 2, 3, 4, 5, 6, 7

1) 1 hour Exam
✓5 objective / one line. (5 marks) 10 minutes.
✓5 questions. (50 minutes.) 2/3 marks (10 marks.)

2 Searching
3 Sortings (Insertion, Quick, Merge, Bubble, Selection)
Array, LL, Stack, Queue, others

15 marks
40 marks.
15

O, Ω, Θ ....

$