

Wildcards in Java :-> (Generics)

\* Whenever we don't want to specify the data-type of the object, we use  $\langle ? \rangle$  "Question mark" in Java. This is called "Wildcard".

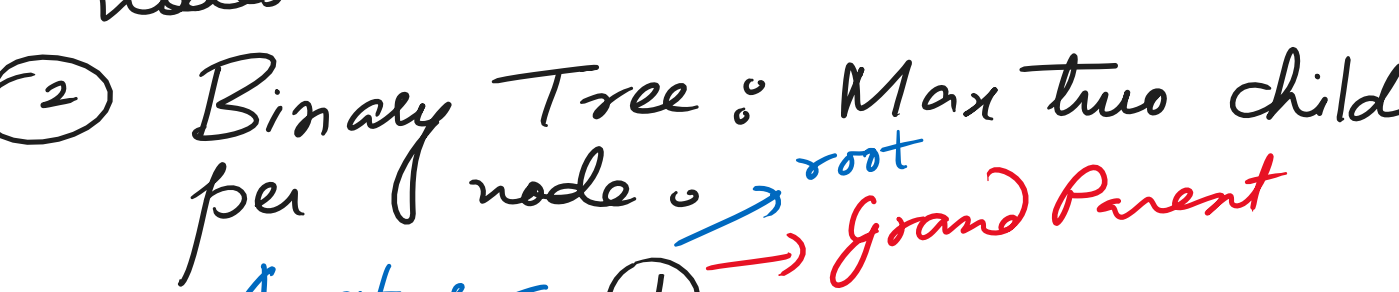
All objects & classes in Java belong to  $\Rightarrow$  Object Class `java.lang.Object`

Introduction to Non-Linear Data Structures

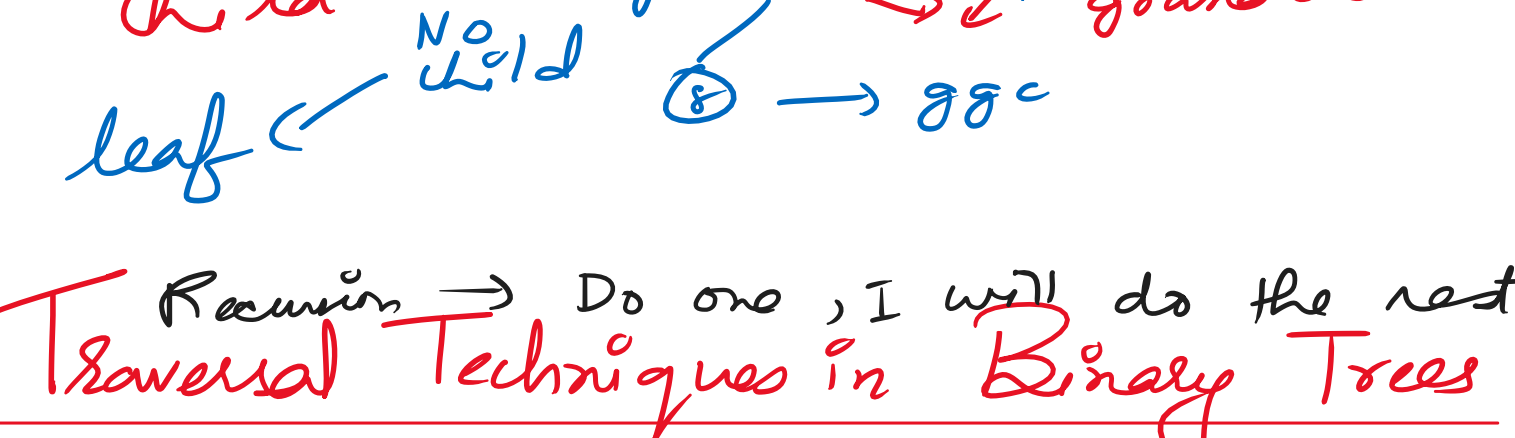
Trees :-> A non-linear data structure containing nodes. (Recursion)

Each node can have children nodes. Depending on the number of nodes, trees are of different types.

① Multiple nodes: Normal Tree

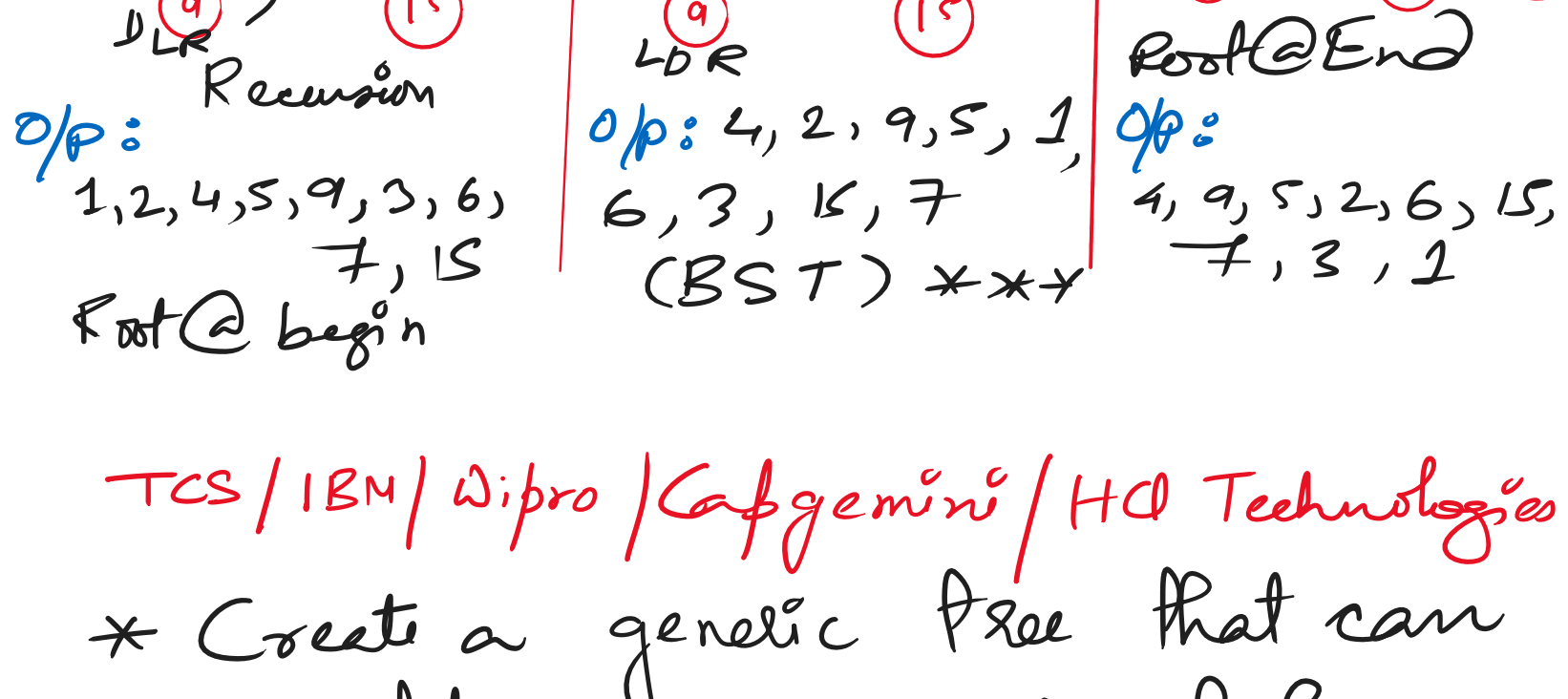


② Binary Tree: Max two children per node.



Recursion  $\Rightarrow$  Do one, I will do the rest.  
Traversal Techniques in Binary Trees

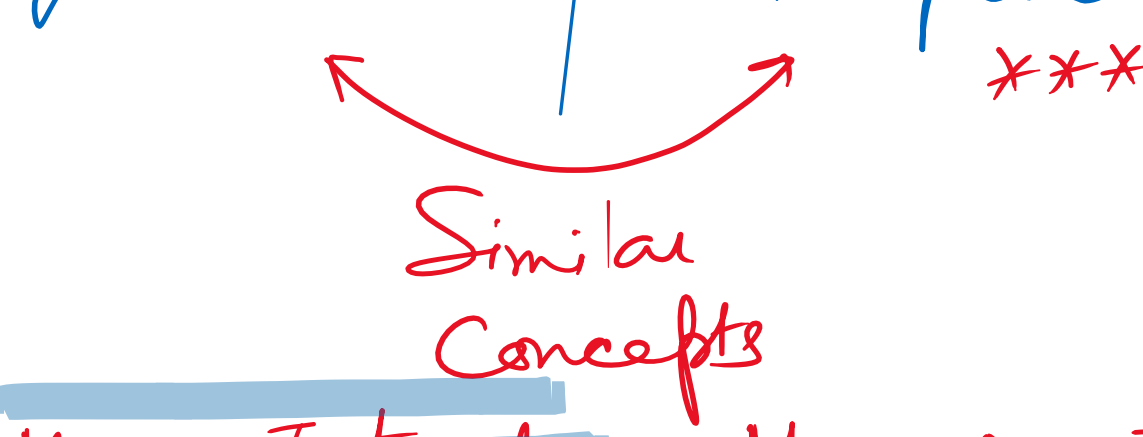
① Depth First Search Traversal



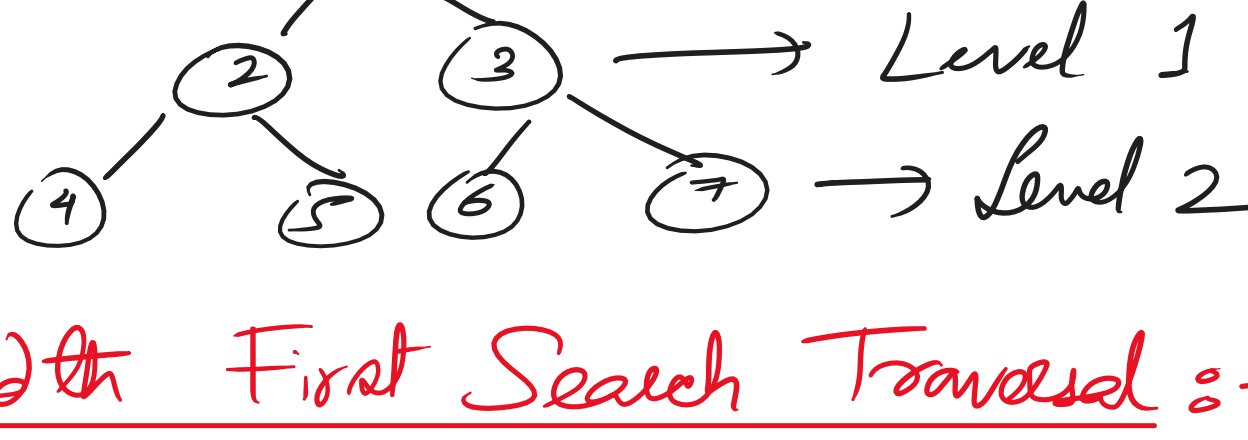
TCS / IBM / Wipro / Capgemini / HCL Technologies

\* Create a generic tree that can accept any data-type as an input & perform the DFS traversals.   
 *TreeNode obj = new TreeNode("");*   
 *TreeNode(" ");*

TreeNode {  
 int data;  
 TreeNode right;  
 TreeNode left;  
}



\* More Interest  $\equiv$  More New Topics  $\equiv$  More Questions



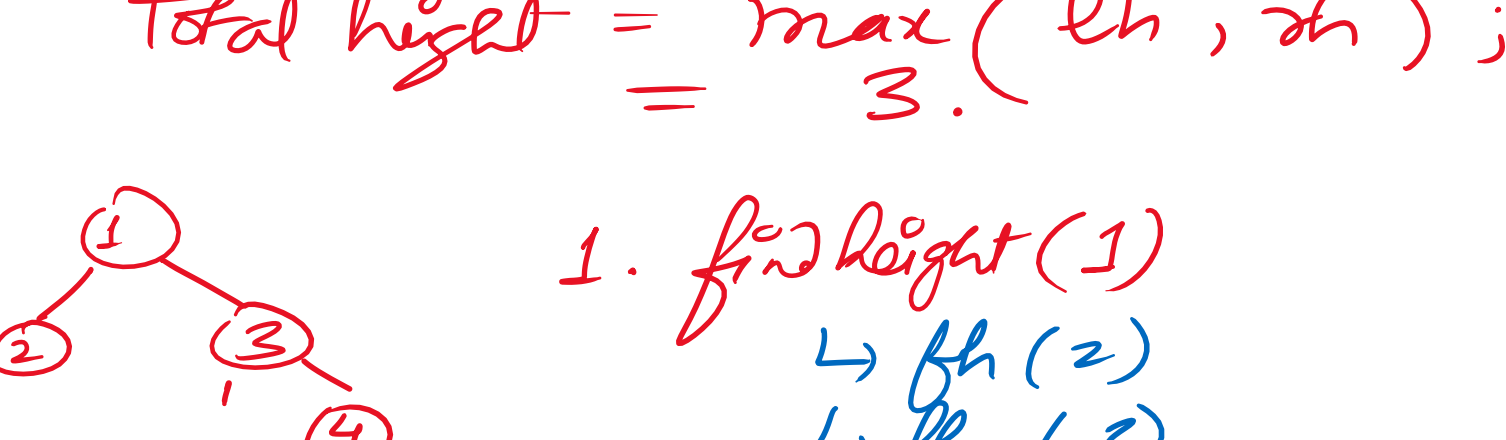
Breadth First Search Traversal :-> "Level Order Traversal"

$\Rightarrow$  Top to Bottom & Left to Right  
\* We use a FIFO (queue) data structure.

O/p :-> 1, 2, 3, 4, 5, 6, 7 and so on.

Binary Trees Important Interview Questions

- \* Height of a Binary Tree
- \* Mirror Image of a Binary Tree
- \* Identical Trees
- \* Level Order Traversal  $\Rightarrow$  Queue



So, Total height =  $\max(lh, rh)$ ;  
= 3.

1. findheight(1)  
  $\hookrightarrow$  fh(2)  
  $\hookrightarrow$  fh(3)  
2. findheight(2)  
  $h = 0 + 0 + 1$   
3. findheight(3)  
  $\hookrightarrow$  null  
  $\hookrightarrow$  fh(4)  
4. findheight(4)  
  $h = 0 + 0 + 1$

Recursion  
  $\hookrightarrow$  DP  
 height of 3 =  $m(0, 1) + 1$   
 =  $1 + 1$   
 = 2

5. Back to findheight(1):  
  $lh = 1$   
  $rh = 2$   
  $\max(1, 2) + 1$   
 = 3

- \* Arrays  $\checkmark$  (Search & Sort)  $\checkmark$
- \* Stacks  $\checkmark$  (Kadane)
- \* Linked Lists  $\checkmark$
- \* Collections \*\*\* Max/Min Heap pending
- \* Degree \*\*\*
- \* \*\*\* Priority Queue

\* Trees  $\rightarrow$  Binary Tree DFS BFS  
 height  $\checkmark$  mirror  $\checkmark$   
 identical  $\checkmark$

- \* Trie / Heaps
- \* Greedy Algo
- \* Back Tracking + Bit Masking
- \* Graphs
- \* DP