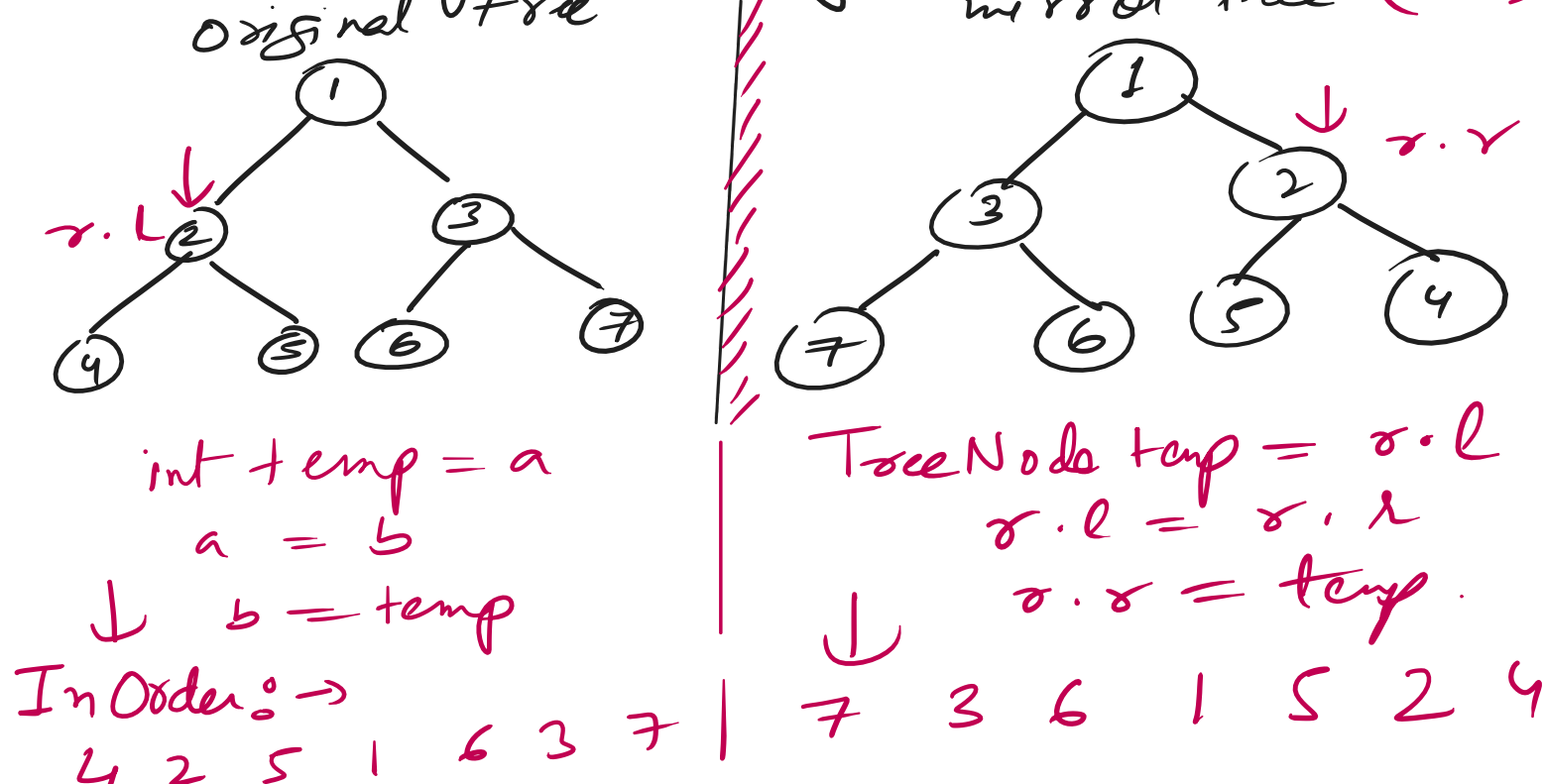


Binary Trees : Important Questions

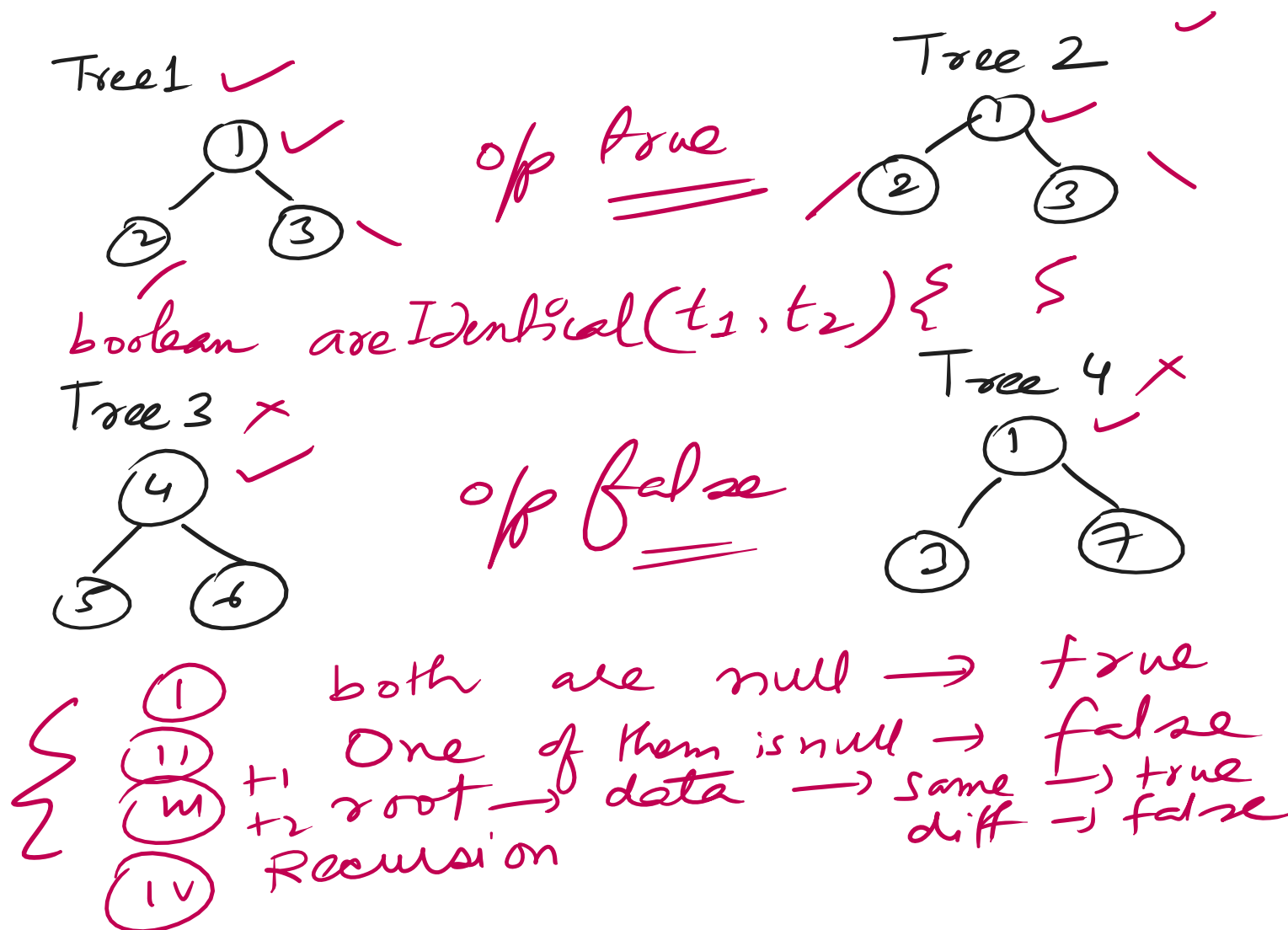
- Level Order Traversal (BFS) ✓
- Generic Tree (Any Data Type) ✓
- Mirror of a Binary Tree ✓
- Identical Trees ✓

Accenture | TCS | DXC | Infosys

Mirror of a Binary Tree (Recursion)



Identical Trees



- both are null → true
- One of them is null → false
- t1 root → data → same → true
- t2 root → data → diff → false
- Recursion

Introduction to Search Trees

- Binary Search Tree
- AVL Trees
- Red Black Trees

Each node in the BST has a unique property.

$$L < \text{Node} < R$$

int[] arr = 50, 30, 40, 20, 35, 10

TC $\log(N)$

Insert ()

Search ()

Delete ()

In Order: Ascending Order

BST → Not height balanced.

1, 2, 3, 4, 5, 6

Insert → BST

Right Heavy Skewed Tree

Left Heavy

search (6) $O(n)$

$O(n)$

$O(n)$

$O(n)$

$O(n)$

$O(n)$

$O(n)$

$O(n)$

$O(n)$

$O(n)$

$O(n)$

$O(n)$

$O(n)$

$O(n)$

$O(n)$

$O(n)$

$O(n)$

$O(n)$

$O(n)$

$O(n)$

$O(n)$

$O(n)$

$O(n)$

$O(n)$

$O(n)$

$O(n)$

$O(n)$

$O(n)$

$O(n)$

$O(n)$

Important checkpoints for the delete function in a BST

* Node with only 1 child

if (r.r == null)

temp = r.l

r = null

return temp

if (r.l == null)

temp = r.r

r = null

return temp

Node With 2 Children

Inorder Successor

Less no of Steps

can

Delete root

root.right

find min (root.right)

temp.data

Inorder: 10, 20, 30, 40, 41, 42, 45, 50, 55, 57, 60, 70, 80

root

root.right

temp

Inorder: 1, 0, 1, 3, 4, 5

temp = findMin (root.right)

root.data = temp.data

delete (root.right, temp.data);