4/5/2023 16:40

Wednesday

Trees (data structure)

→ non-Linear In Nature

Ex: array, stack, queue & L.L are Linear

→

Root, parent Node, child Node, Sibling, Edge, Leaf, subtree Depth, Height, Height of tree, Level, Degree of Node, Null

Binary Tree Traversals:

Tree Traversal algorithms can be classified broadly into two category

Depth-First Search (DFS) Algo

Breadth-First Search (BFS) Algo

1) DFS Algo classied into 3

1) Preorder Traversal (current - left - right)

2) Inorder Traversal

(left - current - right)

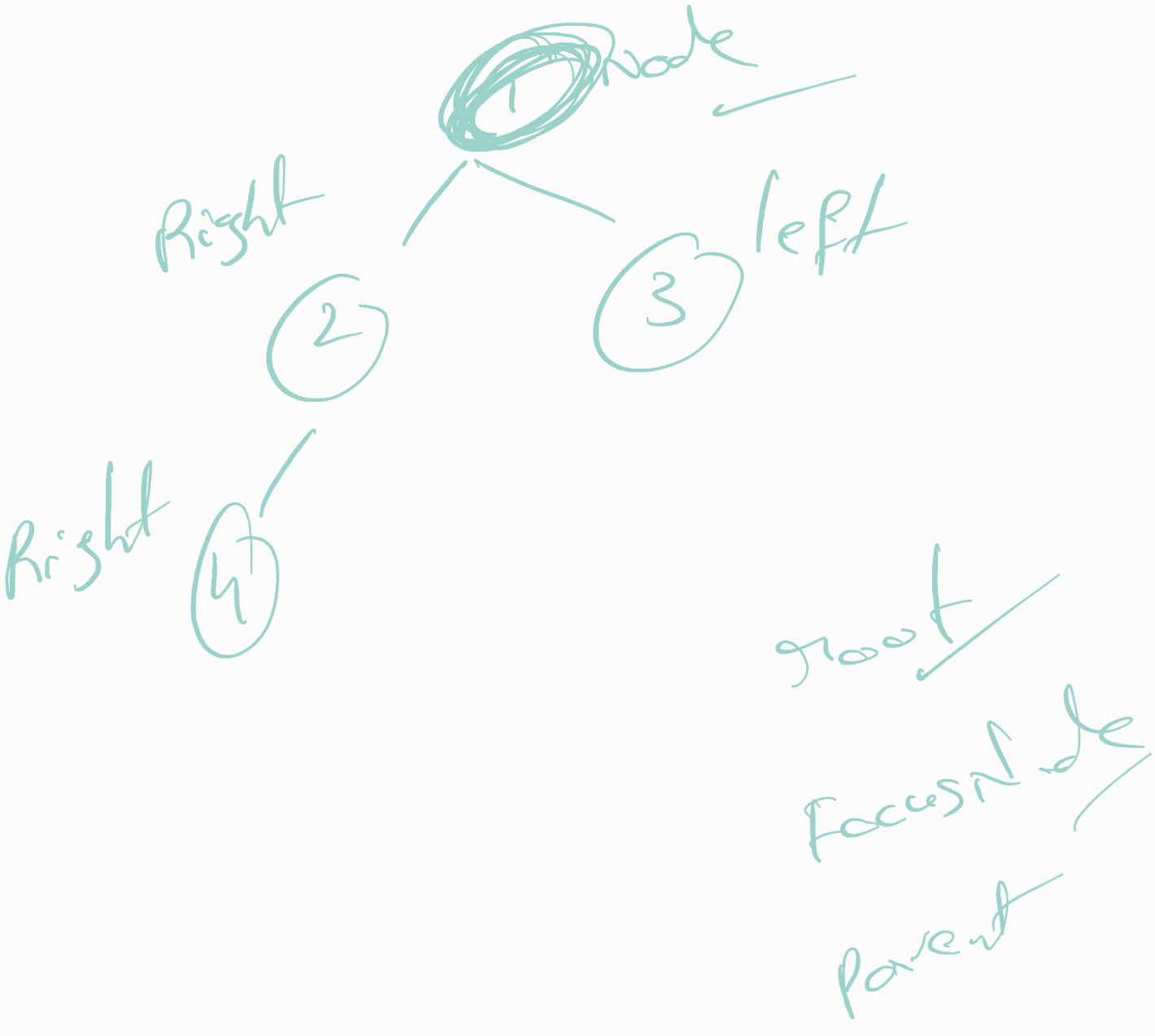
3) postorder Traversal

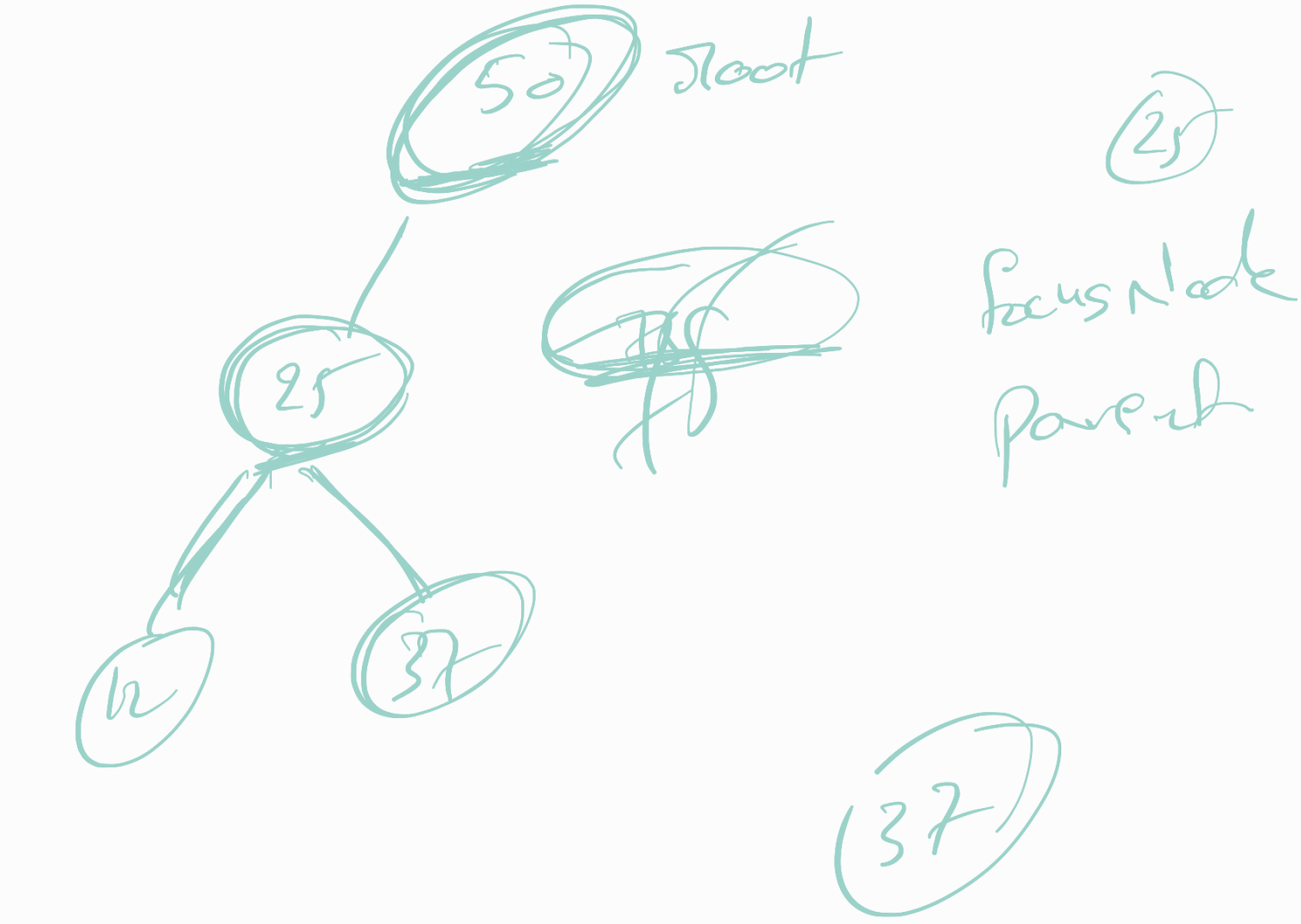
(left - right - current)

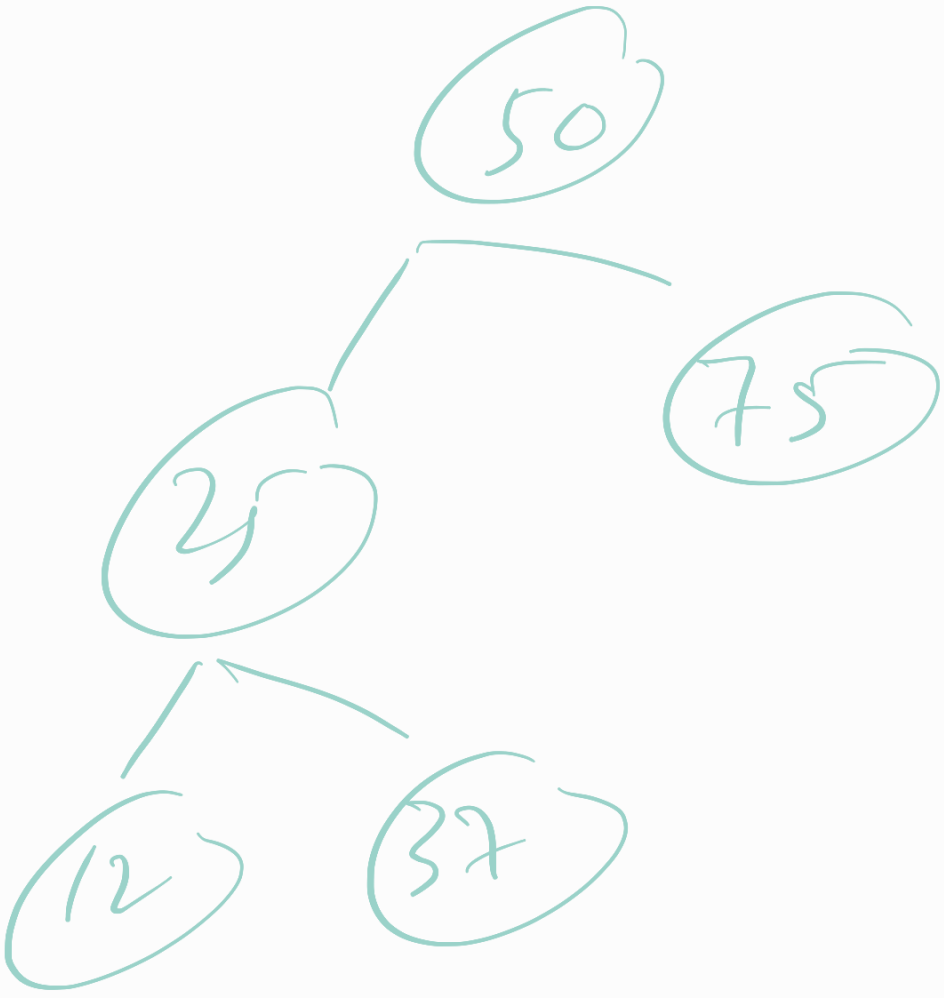
Breadth - first - Search (BFS)

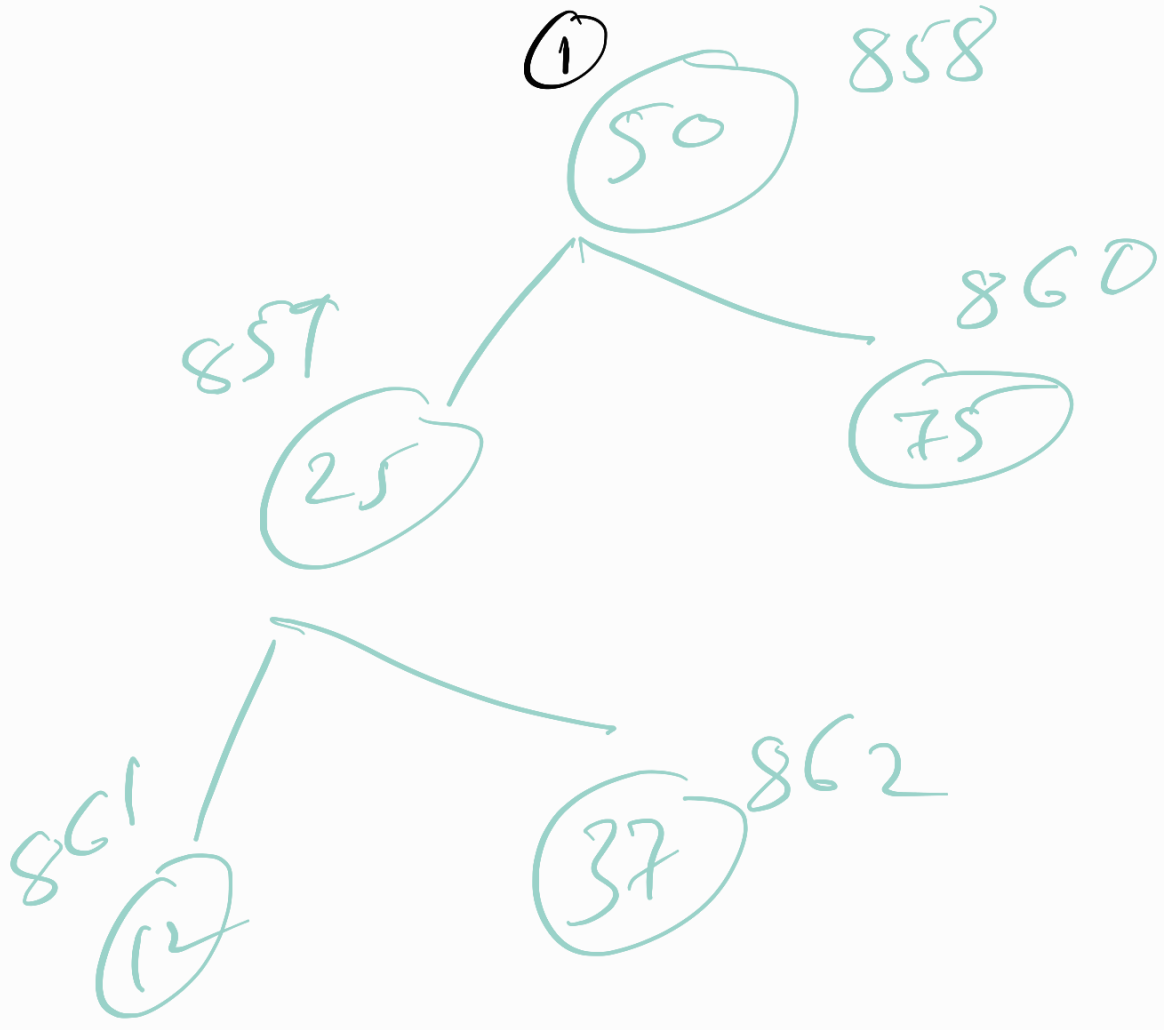
1) Level order Traversal.

Implementation of Binary Tree



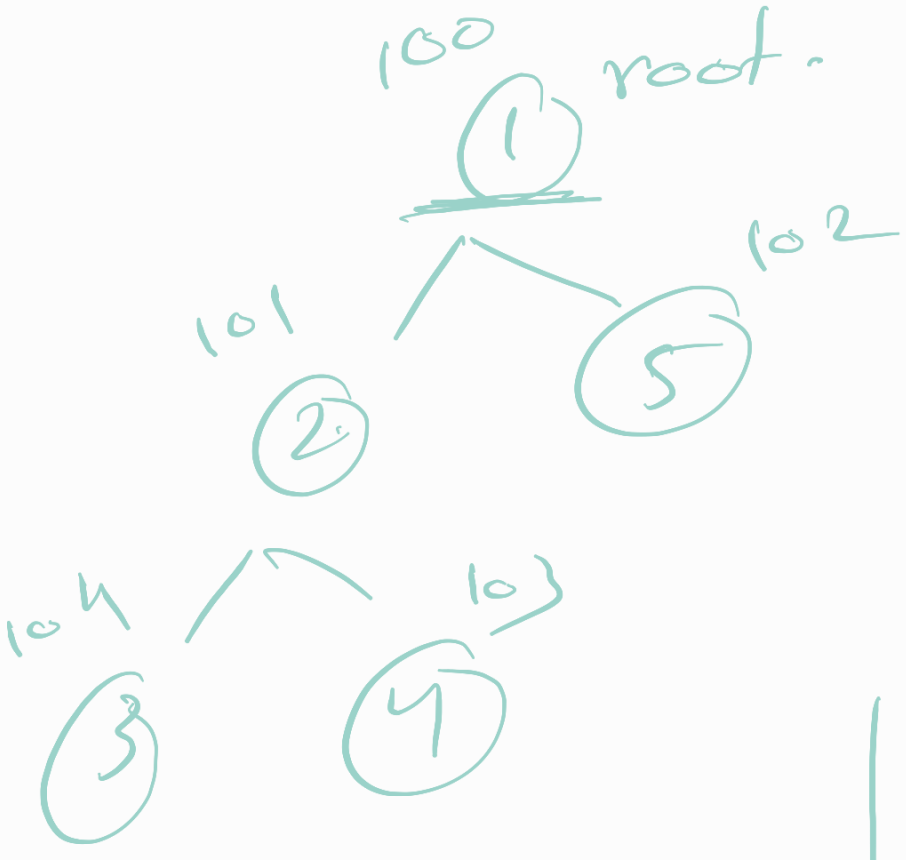

50
root
25
focusNode
parent
25
12
37
37


50
25
75
12
37

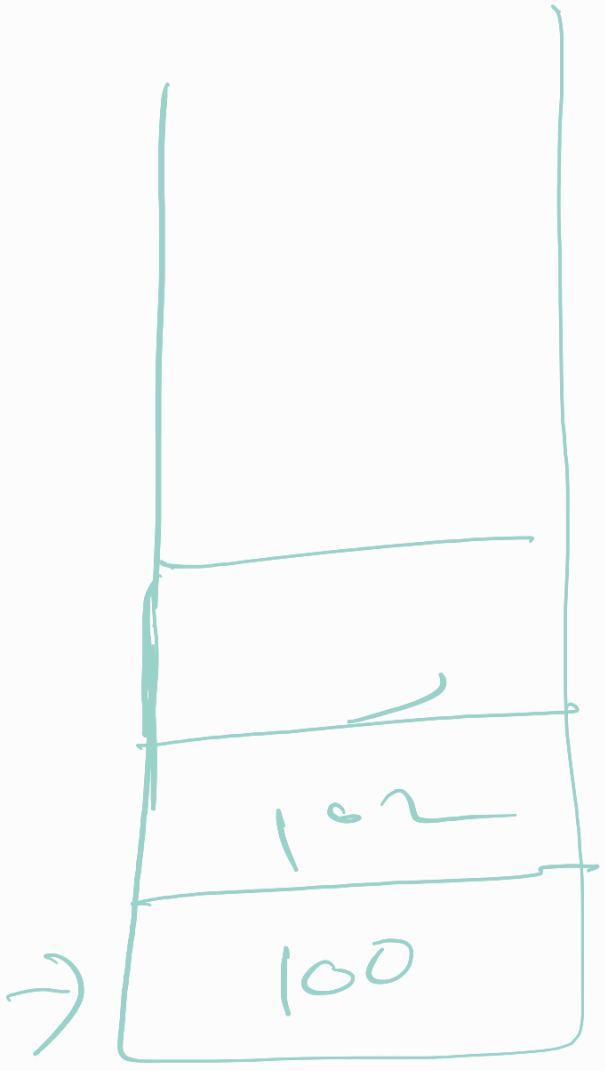

1
858
50
85C
75
857
25
861
12
862
37

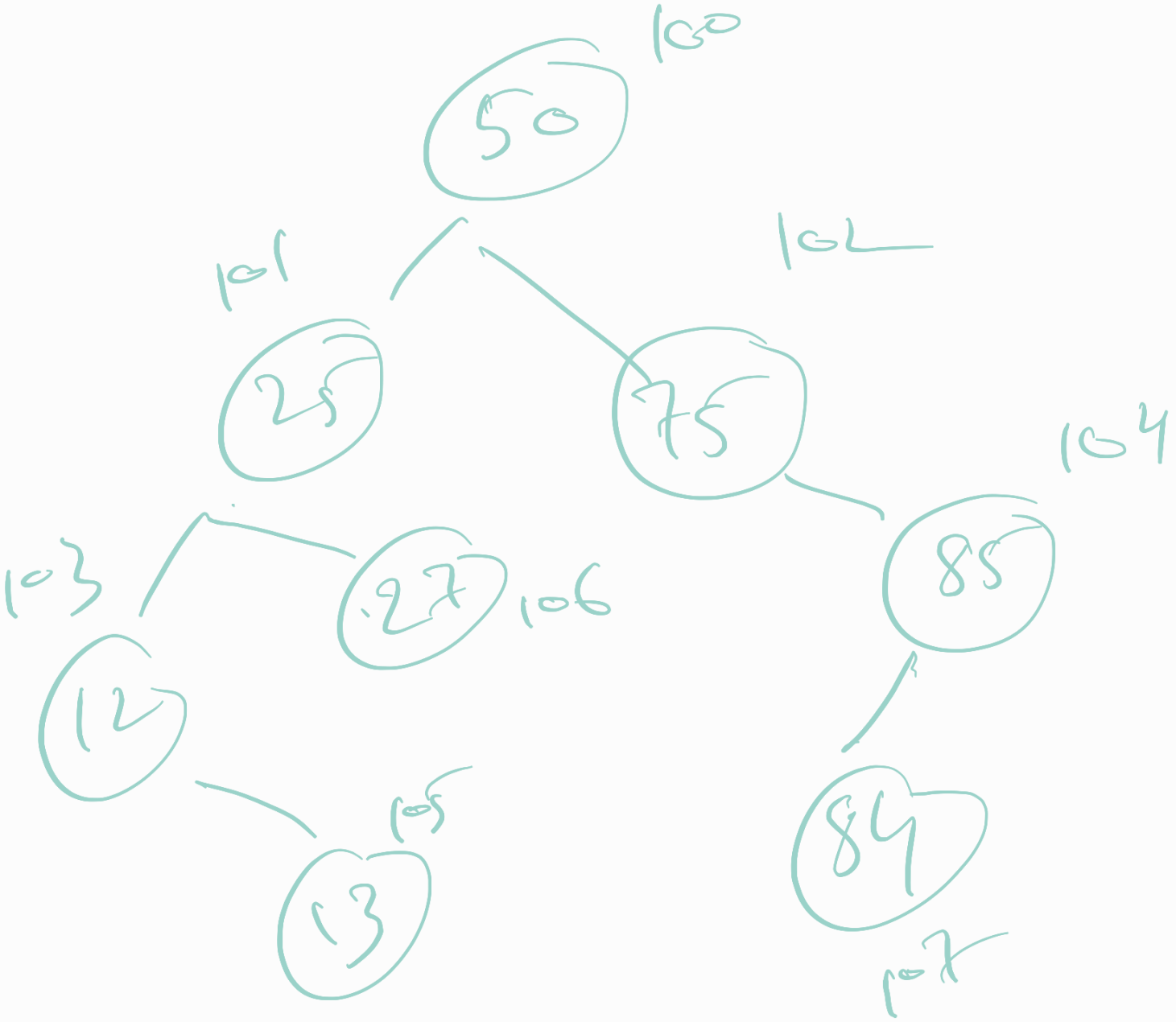

862
3
859
2
858
1
50

Preorder Traversal (N-L-R)

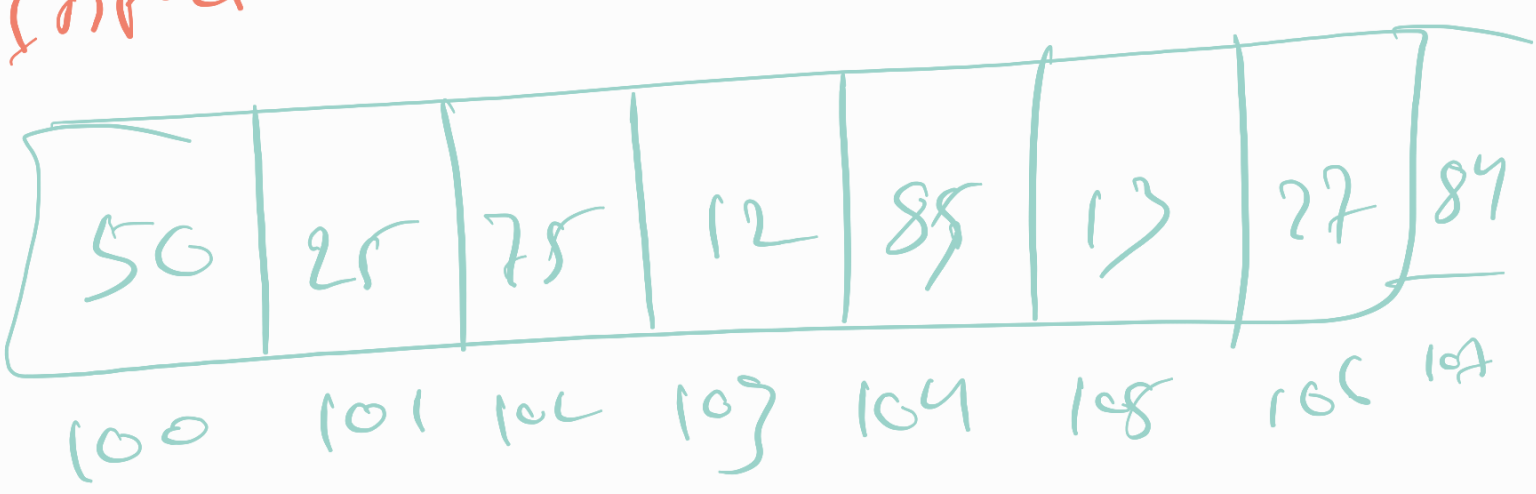


1, 2, 5, 4, 3



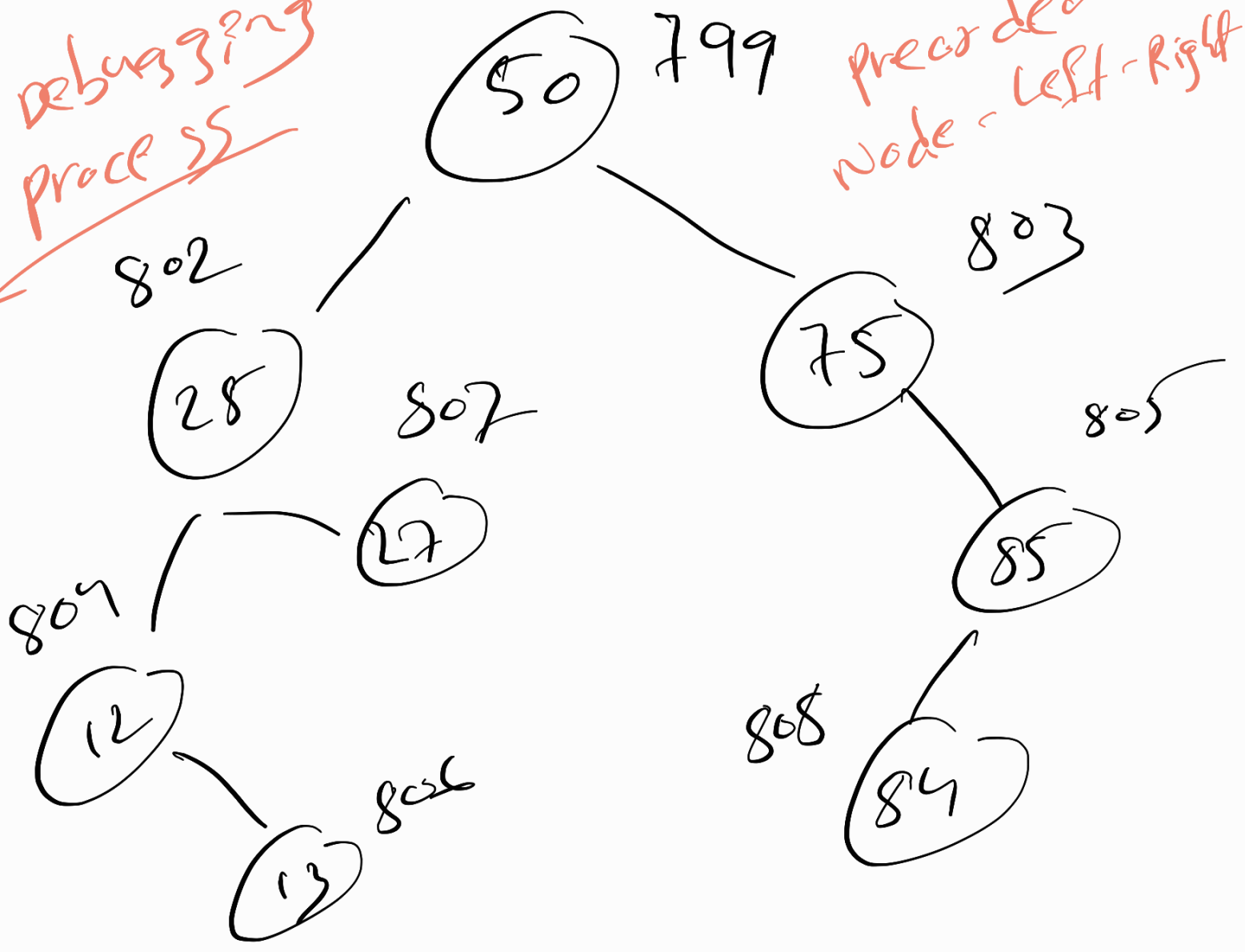

100
50
101
25
102
75
103
12
27
106
104
85
105
13
84
107

Input


50
25
75
12
85
13
27
84
100
101
102
103
104
105
106
107

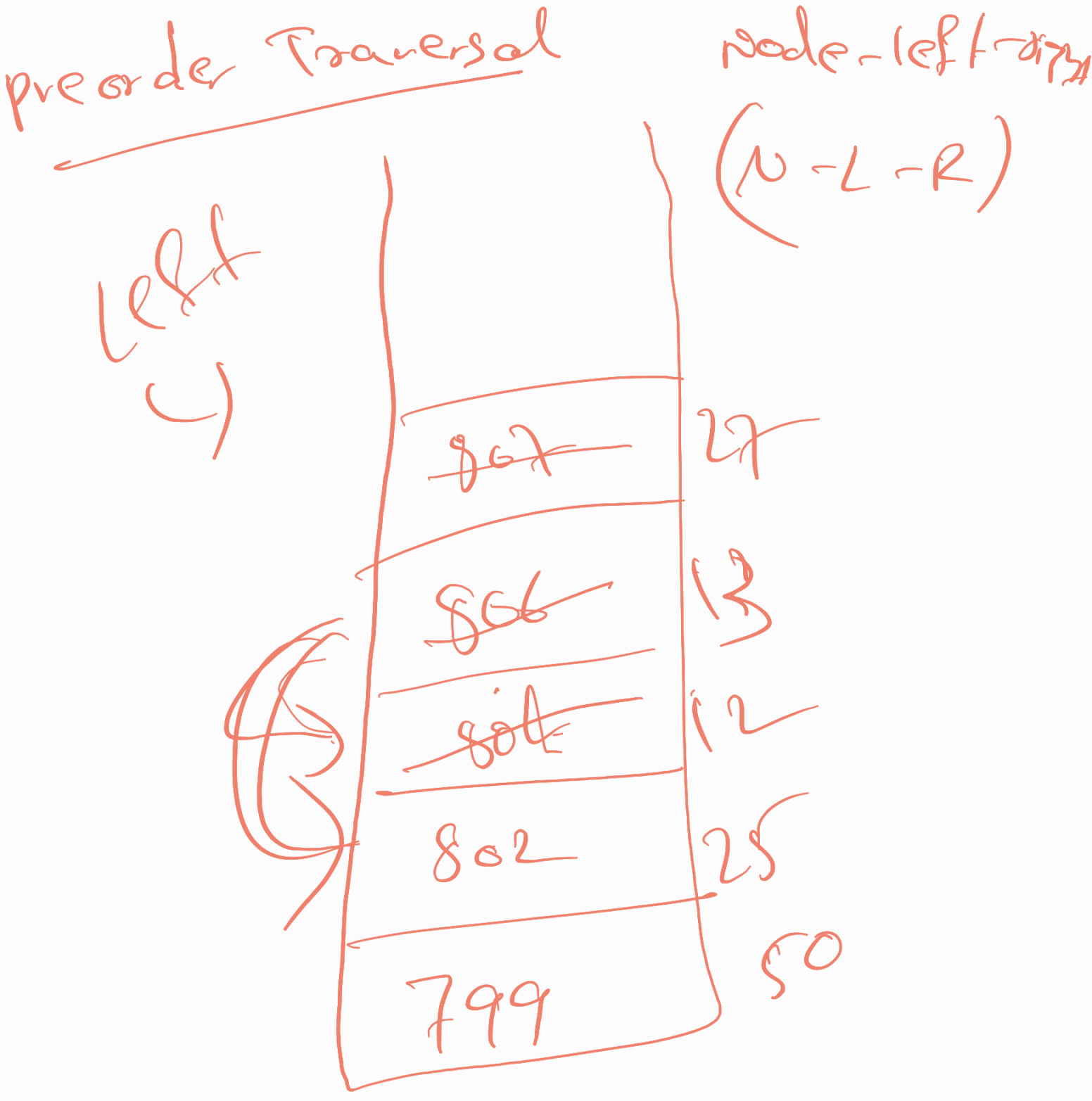
Debugging process

preorder
node - left - right



50	25	75	12	85	13	27	84

799 802 803 804 805 806 807 808


preorder Traversal
node-left-right
(N-L-R)
left
)
807
27
806
13
801
12
802
25
799
50

right

So traversal dp is

50, 25, 12, 13, 27, 75, 85, 84

Preorder → Node - left - right
Postorder → left - right - Node
Inorder → left - Node - right
} DFS

Node - left - right } BFS (level order traversal)