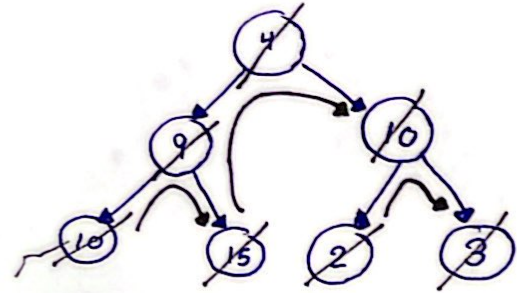


4, 9, 10, 10, 15, 2, 3

0, 0, 0, 0, 0, 0, 0 → output

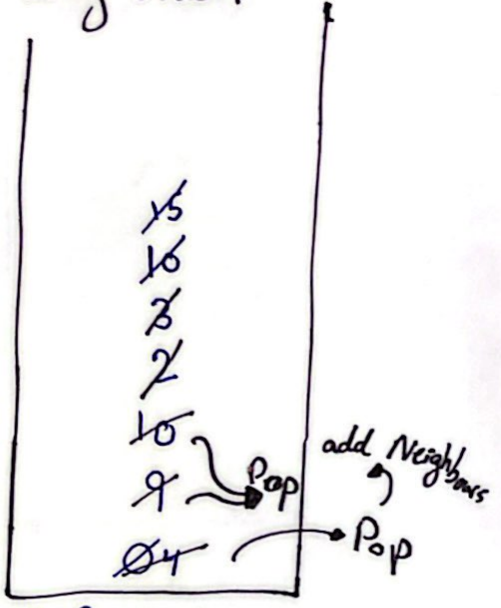
1+2x0
1+2x1
2+2x2
2+2x0
...
arr :- 4, 9, 10, 15, 10, 2, 3

index in Recursive call



Backtracking

using Stack



acted on Each Vertex Individually

Problem

add All Neighbours at Once Stack started From top

Found the Fix

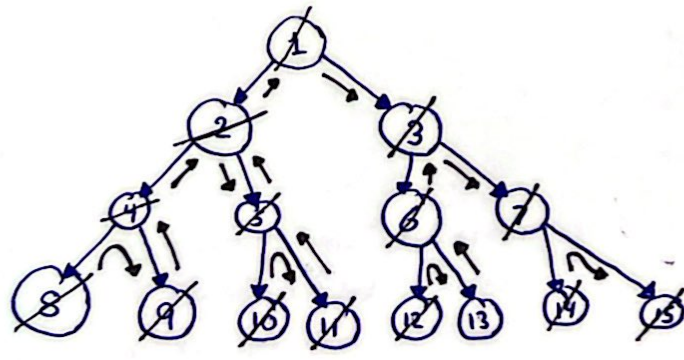
iterative

another Fix

add Neighbours in Reverse

Procedure - 1

bfs → dfs



arr :- 1, 2, 4, 8, 9, 5, 10, 11, 3, 6, 12, 13, 7, 14, 15

(index) i

1 → 0
2 → 1
3 → 2
4 → 3
5 → 4
6 → 5
7 → 6
8 → 7
9 → 8
10 → 9
11 → 10
12 → 11

2 Recursive call

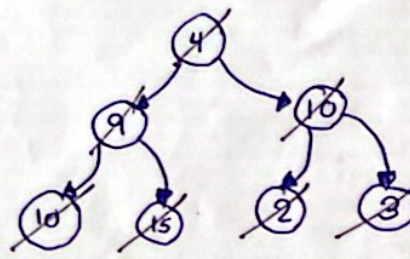
$i * 2 + 1$

$i * 2 + 2$

2^0 1
 2^1 2 4
 2^2 8
 2^3

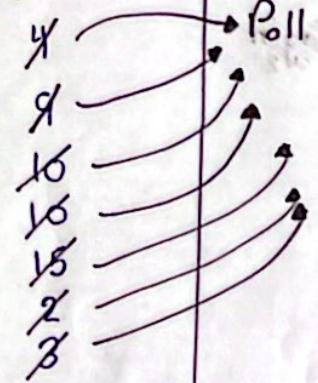
4, 9, 10, 15, 10, 2, 3

arr :- 4, 9, 10, 10, 15, 2, 3



Solution
Make the output
index $\begin{bmatrix} i \times 2 + 1 \\ i \times 2 + 2 \end{bmatrix}$ reverse

using queue



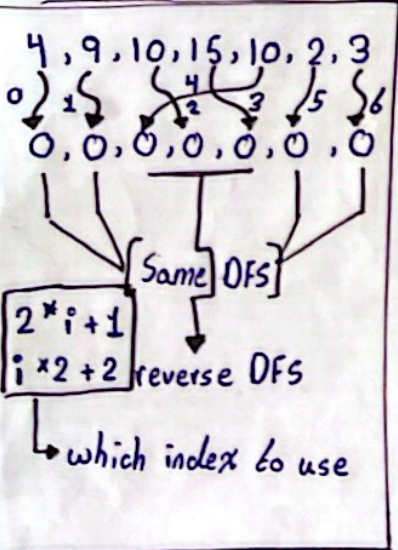
4, 9, 10, 10, 15, 2, 3

Similar =

Perfect

Steps to Help
My Fams

Imagine on queue



1-start by adding the first No. to Queue

2-While queue not Empty, remove and add to other array

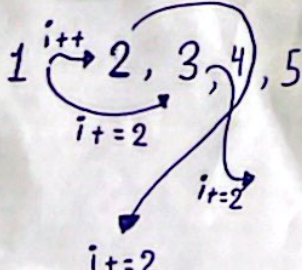
3-Enter to add the neighbours to Queue

4-loop until Finished Sixe

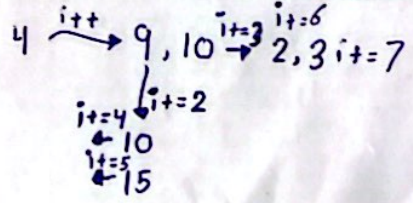


Notice

Every tree node have 2 Children
Maximum



Conclusion :- index varies
if Contains Multiple node
of Children index increase
by one for each children
increase (one children) save
the index but (two children)
increases index to get Node



For Each node consist of 2 child