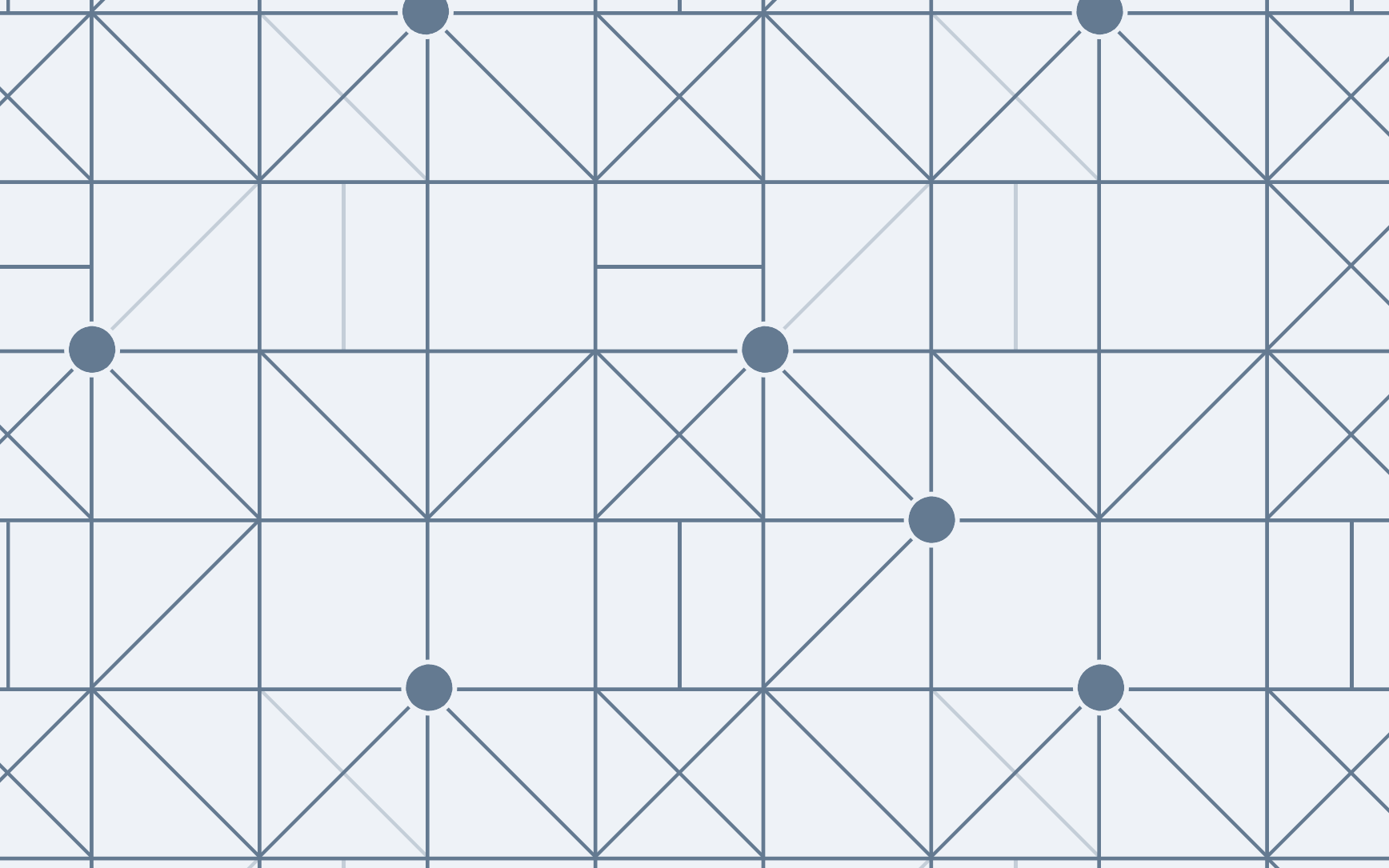
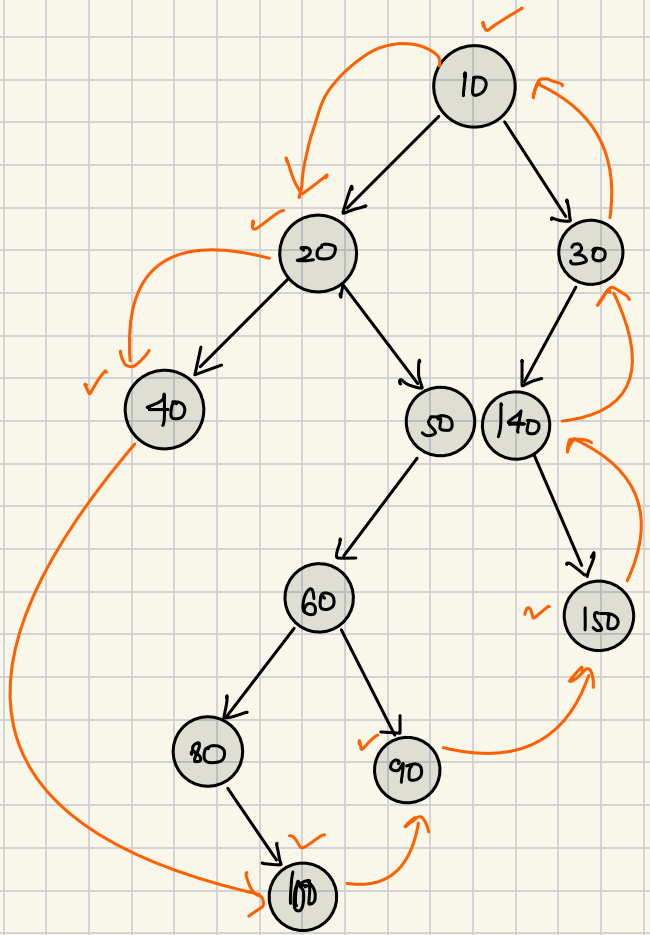


Boundary Traversal



Boundary traversal

root + leftmost Nodes + leaf Nodes + rightmost nodes

{10, 50, 40, 100, 90, 150, 140, 30}

Boundary traversal :

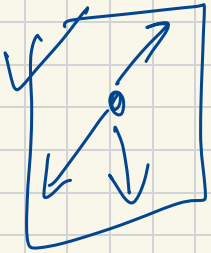
get root Node + Left Boundary + Leaf Nodes + right Boundary.

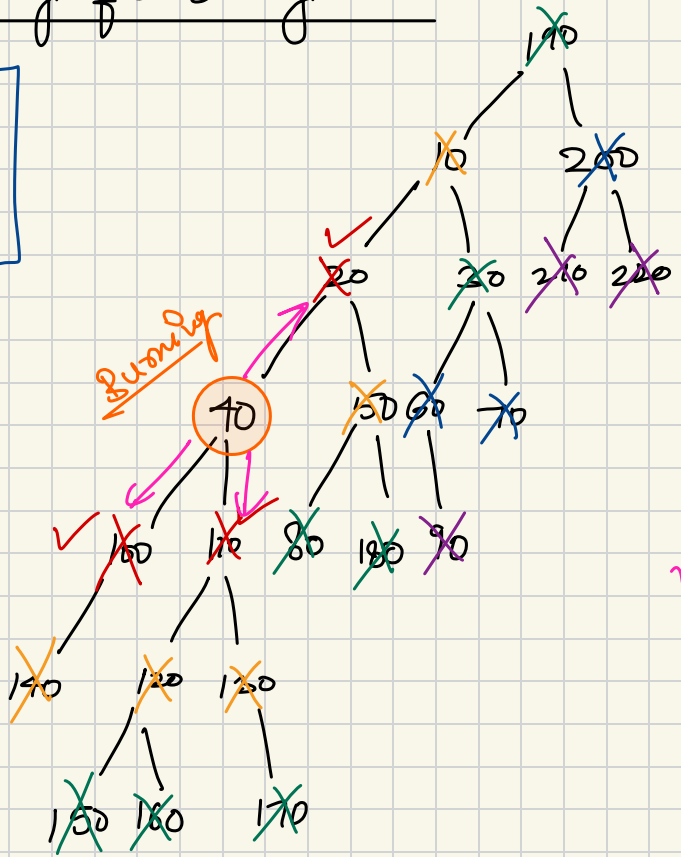
{ root + LB + LN + RB }

top - bottom { pre order }

bottom - top { post order }

Burning of a Binary Tree





min amt of time tree would get burnt ?

Time = 1 2 3 4 5

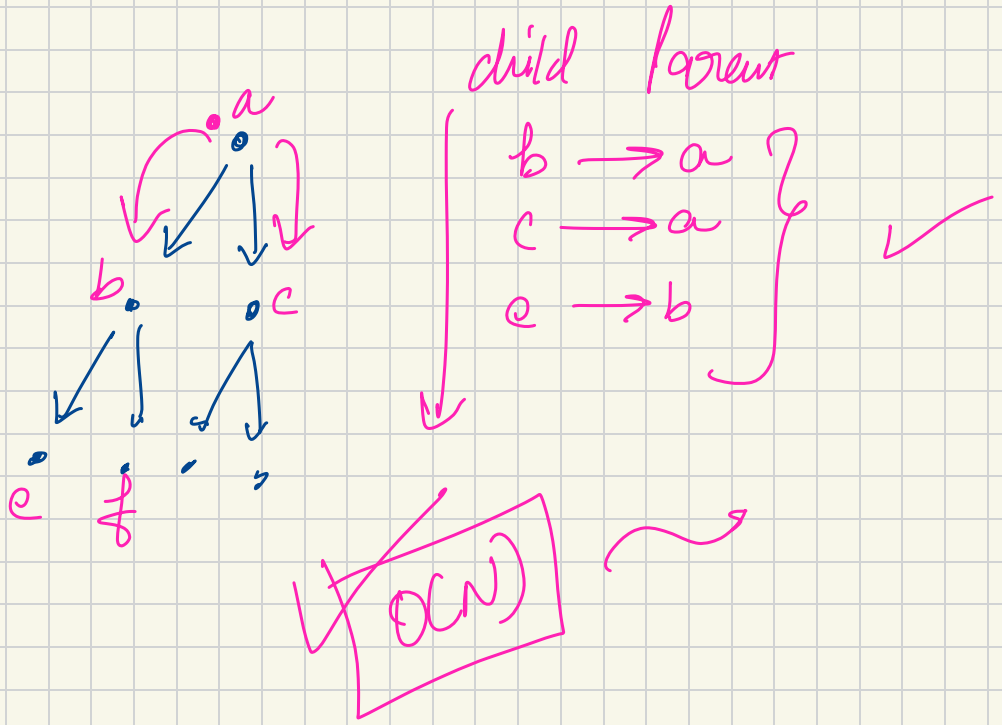
how ?

how to get parent ?

BFS

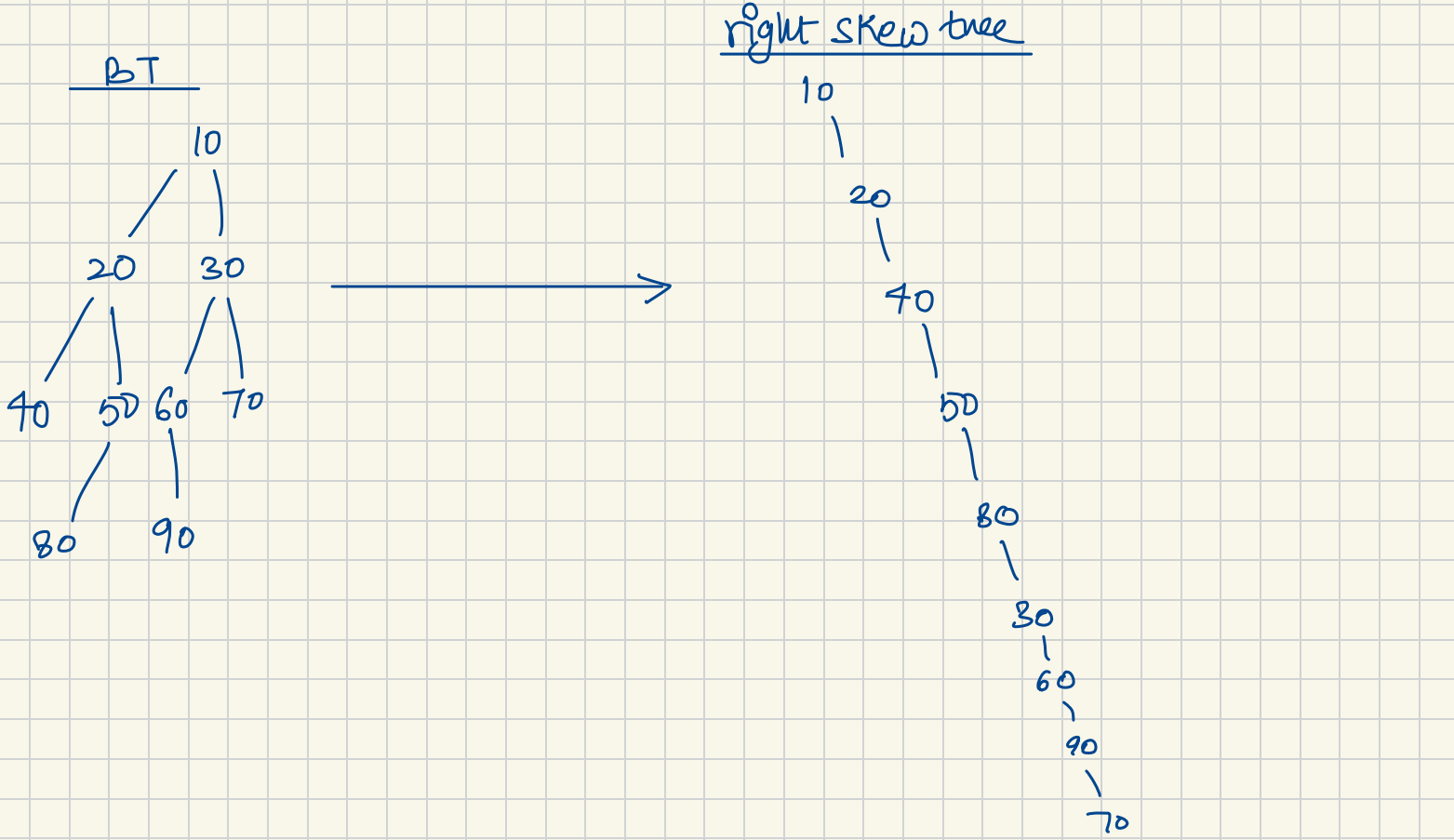
HashMap

child (key) → parent (value)


a
b
c
e
child parent
b → a
c → a
e → b
O(N)

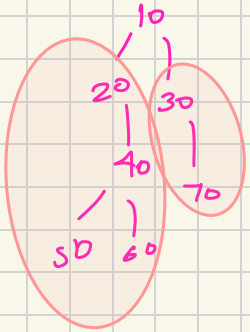
Flatten a Binary Tree to a linked list

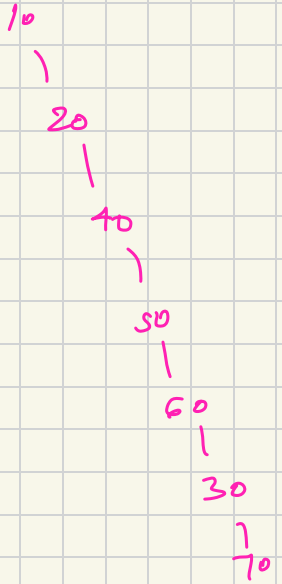
{ in place }

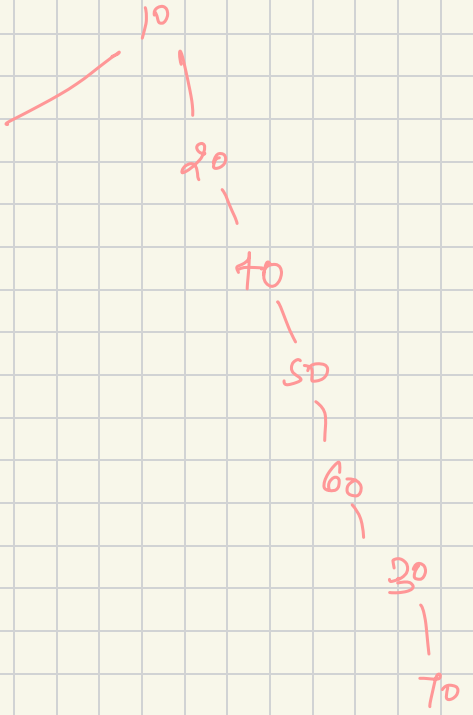


faith: flatten the given Binary Tree

```
void flattenBinaryTree(Node root)
{
}
```







Construct a BT using Pre order and In order traversal

pre order: 10 20 40 50 80 30 60 90 70 { root, lst, rst }

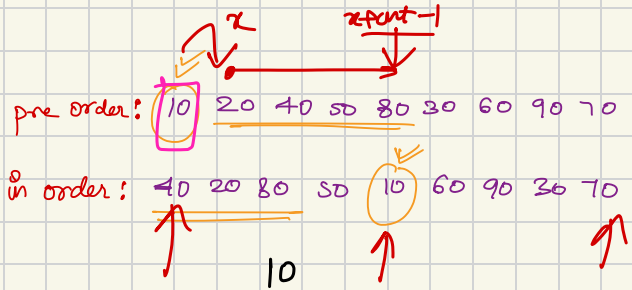
in order: 40 20 80 50 10 60 90 30 70 { lst, root, rst }

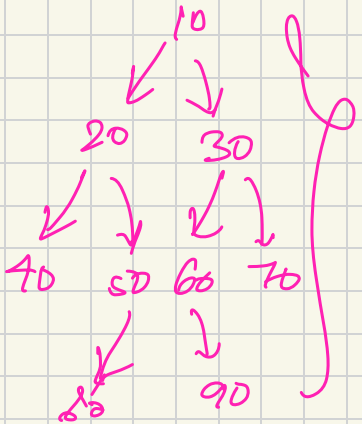
construct BT (pre , in)

{

}

construct the BT and returns from given pre , in





10

pre {20, 40, 50, 80}
in {40, 20, 80, 50}

pre {30, 60, 90, 70}
in {60, 90, 30, 70}

20

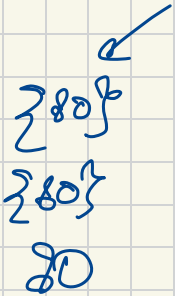
{40}
{40}
40

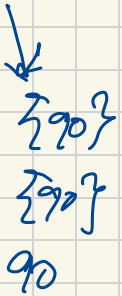
{50, 80}
{80, 50}
50

80

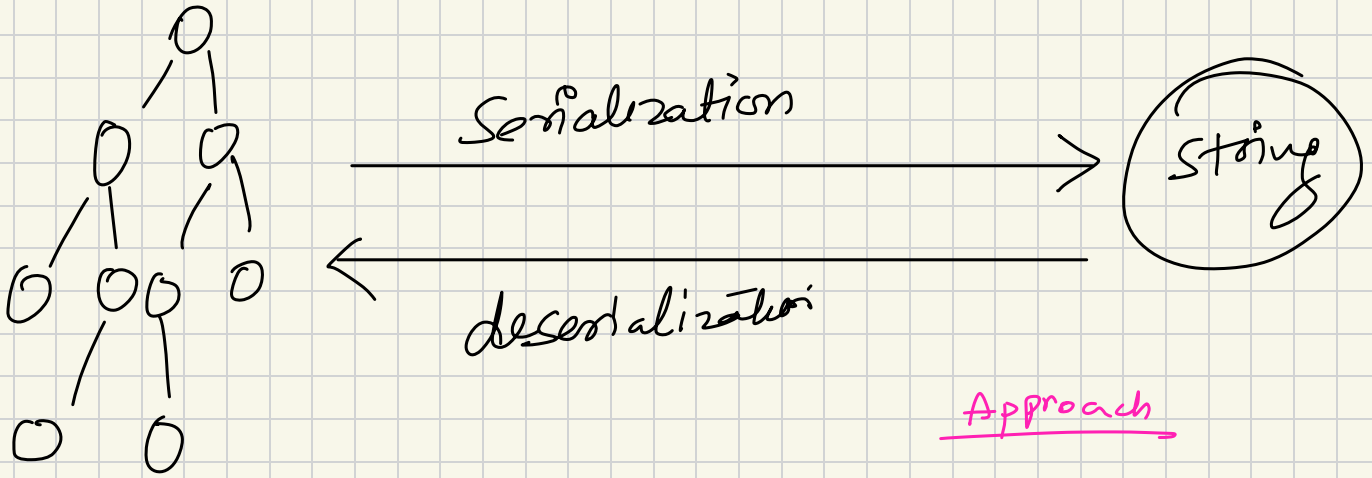
{60, 90}
{60, 90}
60

{70}
{70}
70


{80}
{80}
80

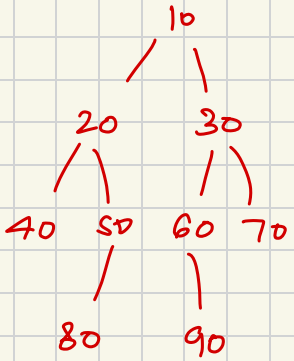

{90}
{90}
90

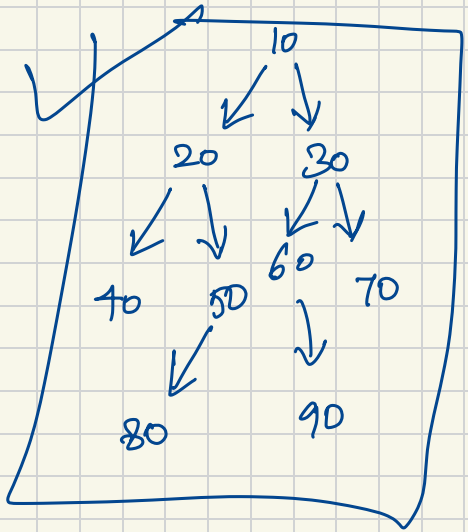
serialize and deserialize



Approach

"preorder # inorder"





preorder

"10, 20, 40, null, null, 50, 80, null, null, null, 30, 60, null, 90, null, null, 70, null, null."

Longest subarray with equal number of 0's, 1's and 2's

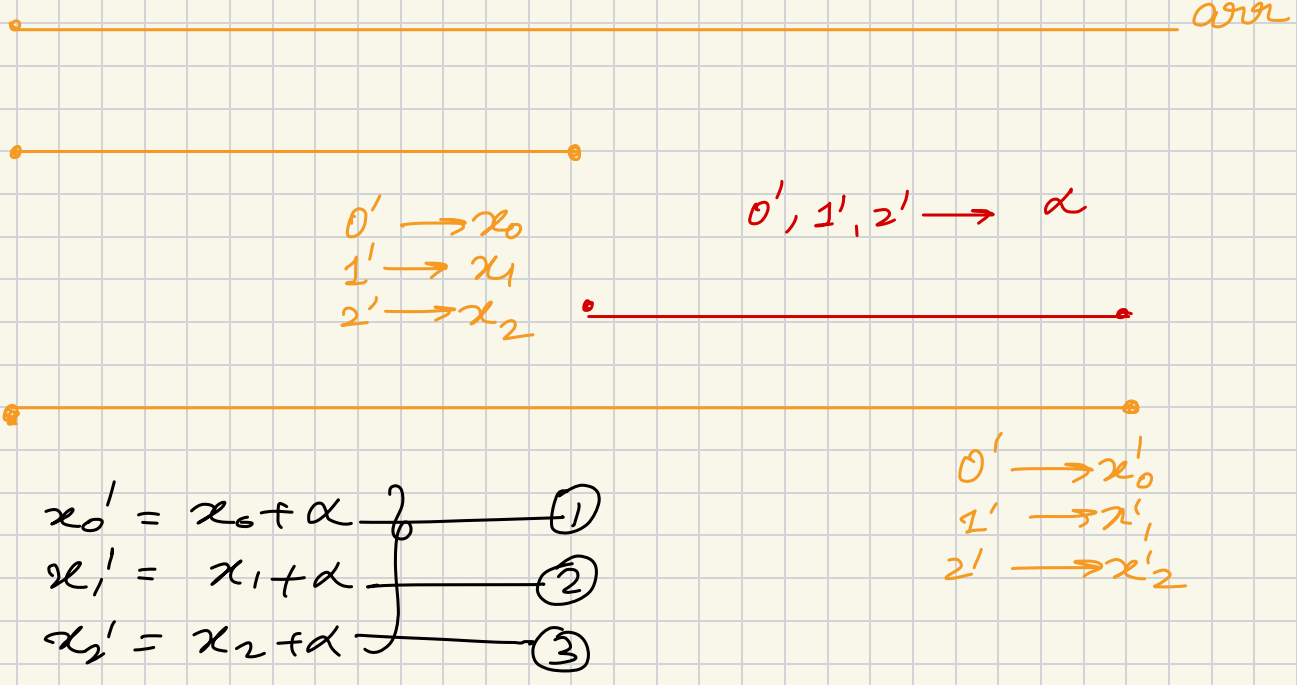
$arr[] = \{1, 1, 2, 0, 1, 0, 1, 2, 1, 2, 2, 0, 1\}$

Brute force

- find all subarrays → $O(N^2)$
- cnt no. of 0's, 1's & 2's → $O(N)$

TC : $O(N^3)$
SC : $O(1)$

$arr[] = \{1, 1, 2, 0, 1, 0, 1, 2, 1, 2, 2, 0, 1\}$



(2) − (1)

$$x_1' - x_0' = x_1 - x_0 \quad \text{——— (4)}$$

(3) − (2)

$$x_2' - x_1' = x_2 - x_1 \quad \text{——— (5)}$$

arr[] = { 1, 1, 2, 0, 1, 0, 1, 2, 1, 2, 2, 0, 1 }

x_0	0	0	0	0	1	1	2	2	2	2	2	2	3	3
x_1	0	1	2	2	2	3	3	4	4	5	5	5	5	6
x_2	0	0	0	1	1	1	1	1	2	2	3	4	4	4
$x_1 - x_0$	0	1	2	2	1	2	1	2	2	3	3	3	2	3
$x_2 - x_1$	0	-1	-2	-1	-1	-2	-2	-3	-2	-3	-2	-1	-1	-2

0 # 0 | 1 # 1 | 2 # -2 | 2 # -1

Key

Key = "$(x_1 - x_0)$ # $(x_2 - x_1)$"