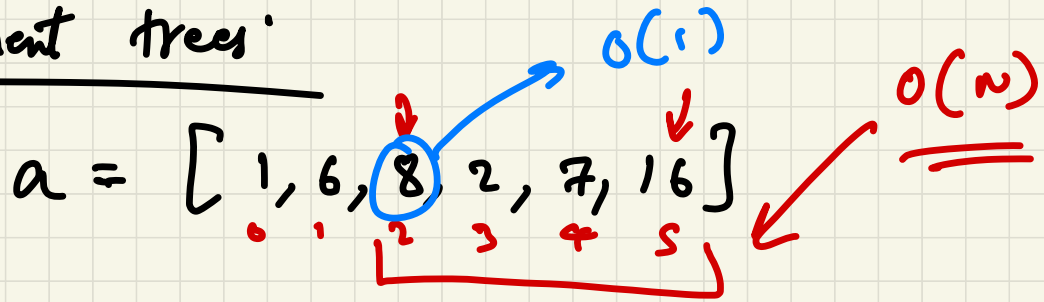
Segment trees:



O(N) not good enough, find O(log N) sol.

↳ Answer → Segment tree.

Segment → Perform query on a range

(sum, max, avg, min, prod)

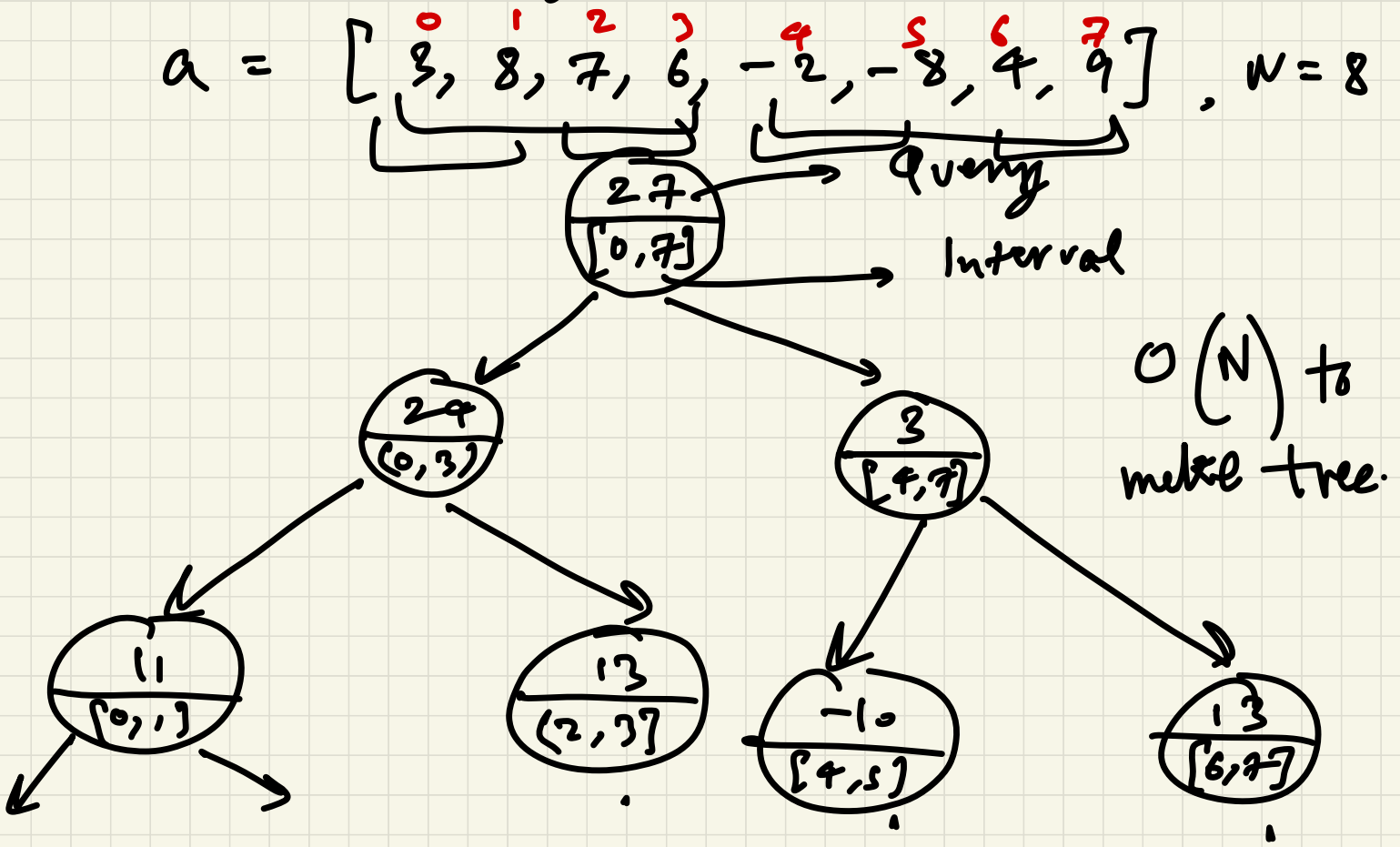
in O(log N)

update → O(log N)

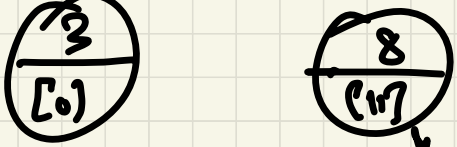
ST is a binary tree which has interval info & operation.

Ex: Sum btw any 2 integer.

$a = [3, 8, 7, 6, -2, -8, 4, 9]$, $N = 8$



$O(N)$ to make tree.



leaf node → only 1 element in array.

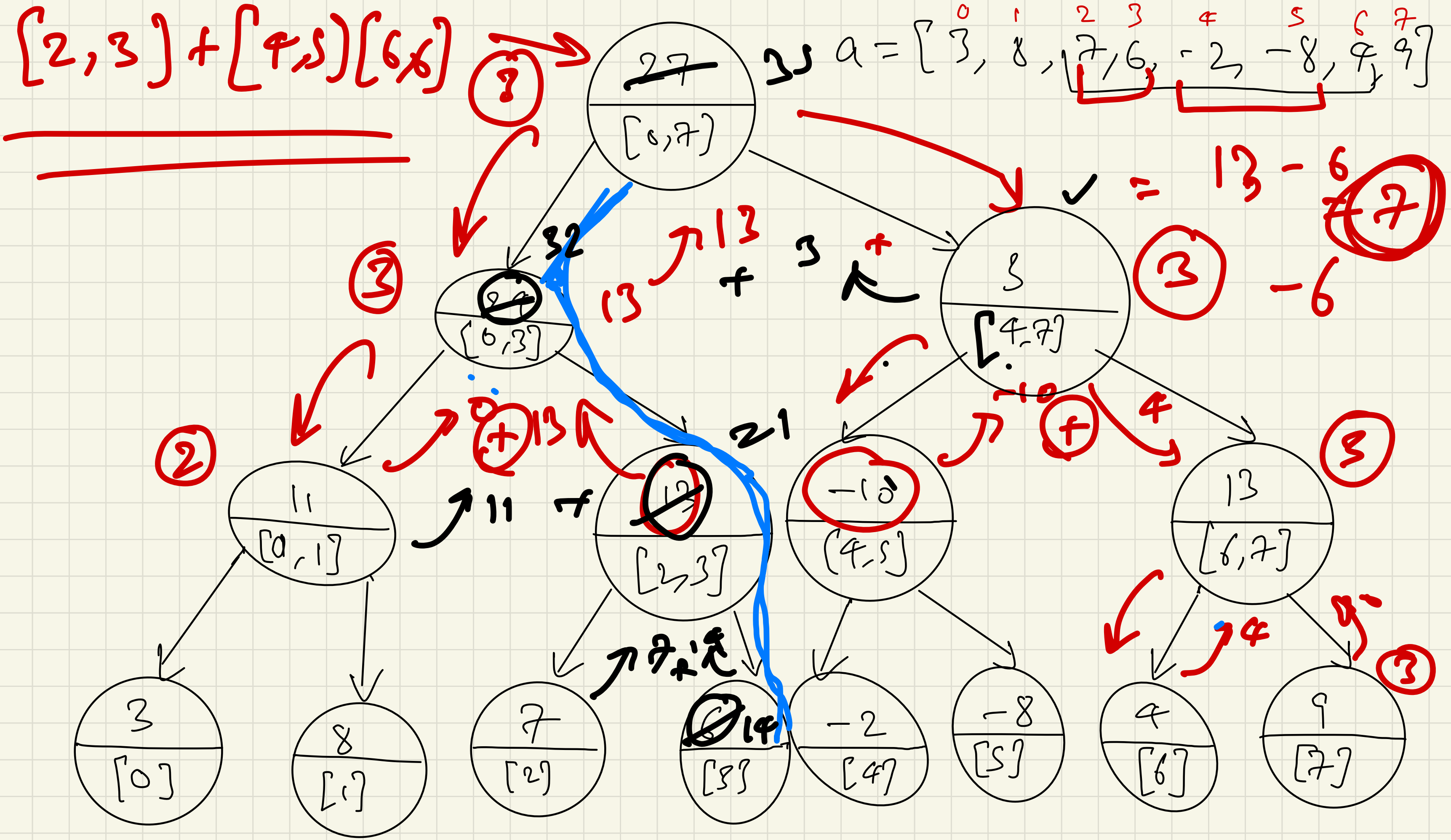
∴ ST is Full binary tree.

we did this in lecture 1:

Leaf Nodes = N

Internal nodes = N - 1

Total Nodes = 2N - 1



[2,3] + [4,5] [6,8]
a = [3, 8, 7, 6, -2, -8, 4, 9]
0 1 2 3 4 5 6 7
27
[0,7]
35
24
[0,3]
32
8
[4,7]
= 13 - 6
= 7
-6
11
[0,1]
12
[2,3]
21
-10
[4,5]
13
[6,7]
3
[0]
8
[1]
7
[2]
6
14
[3]
-2
[4]
-8
[5]
4
[6]
9
[7]

Q: Sum btw $[2, 6]$ = $[2, 3] + [4, 5] + [6, 6]$ etc.

example only.

Cases

① Node interval is inside query interval

ex: $[4, 5]$ → return the value.

② Node interval is completely outside query interval. (node start index) > (query end index) or

$[2, 6]$ → $[7, 7]$ (Node end < query start)

↳ return the default value of the query. In this case = 0.

(3) Overlapping: [2, 6]

Ex: [0, 3] or [0, 7]

How to update in O(logn) time:

Change index 3 with 14 ⇒ (3, 14)

① Check whether index lies in interval.
[0, 7]

② If yes, then check child nodes
if child range is out, no change in val, just return.

③ In the end, you will reach leaf.
update leaf as version will
update the tree.

a= [3, 4, 9, 2, 7]

0 1 2 3 4

m = x1