Bonus Lecture - 5
Segment Tree

RECAP

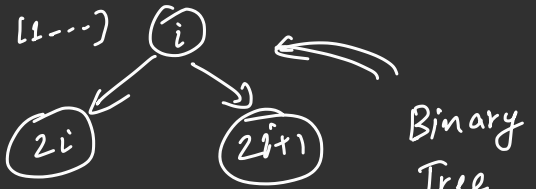
Introduction

Given an Array. $(1 \le N \le 10^5,\ 1 \le a[i] \le 10^9)$

Process queries of 2 types: $(1 \le q \le 10^5)$

⇒ 1 l r ⇒ print $a[l] + a[l+1] \dots a[r]$

⇒ 2 i x ⇒ Increment the value of $a[i]$ by x. (i.e. $a[i] + = x$).



Intuition behind Segment Tree

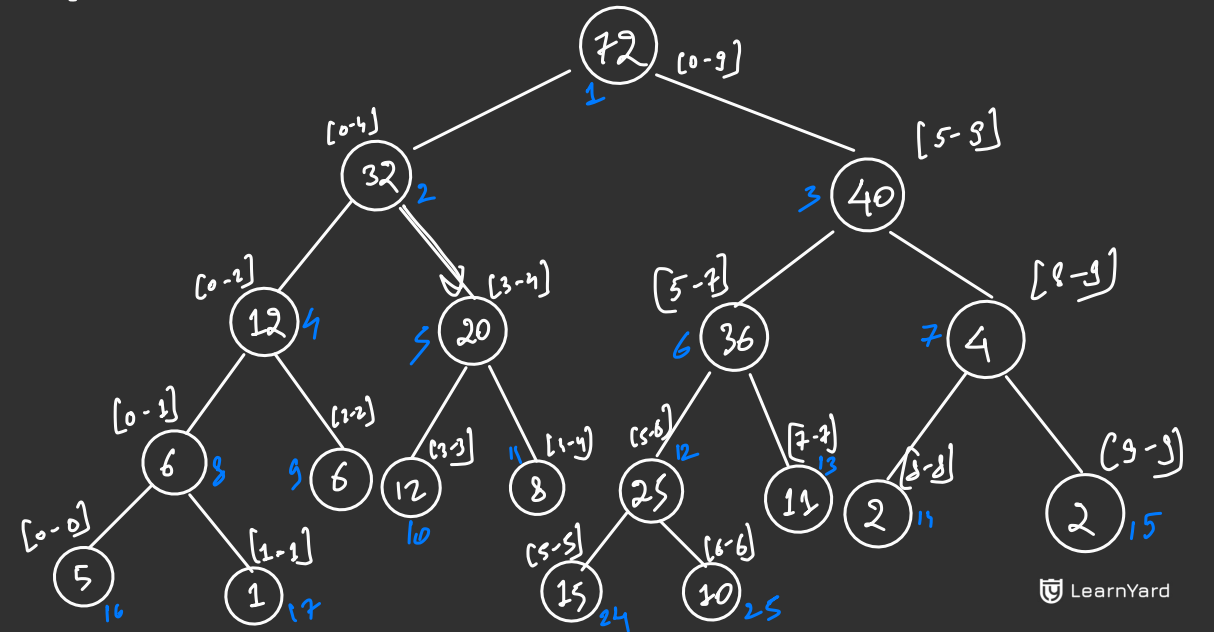
Each node will store some info about a segment of the array. (subarray)

More detailed Explanation

Pre-processing (build)

arr = [5, 1, 6, 12, 8, 15, 10, 11, 2, 2]

0 1 2 3 4 5 6 7 8 9



ceil $(\log_2 N)$ $\Rightarrow$ height

$1 + 2 + 4 + 8 \cdots\cdots \longrightarrow 2^{\log_2 N + 1}$

$\Downarrow$

$\log_2 N + 2$

$$\Rightarrow \frac{1 * \left(2^{\log_2 N + 2} - 1\right)}{2 - 1} \Rightarrow 2^2 * 2^{\log_2 N} - 1$$

$$\Rightarrow 4 * N - 1$$

Updating ⇒ i × (a[i] += x)

index i ∉ [tl, tr] ⇒ return;

else

↳ if (qi ≤ mid)
recur(left_child)

else
recur(right_child)

O(log N)

seg[ti] = merge(lc, rc)

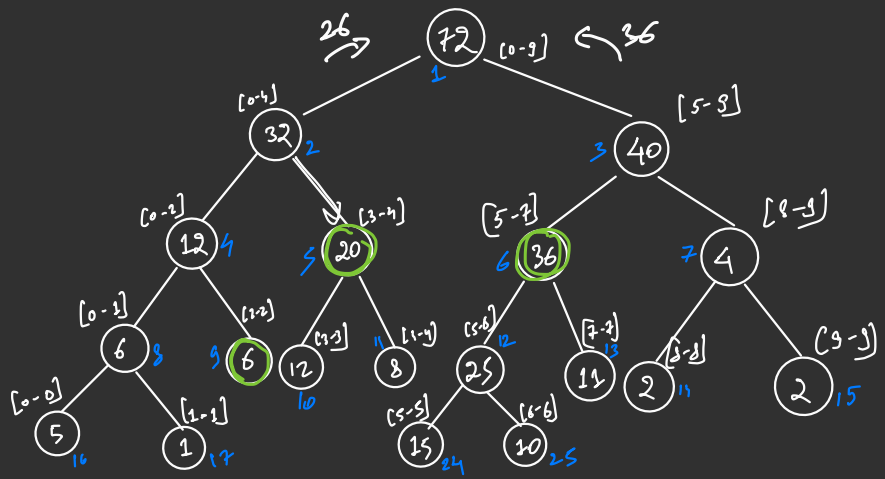
Querying ⇒ [l, r]

query(i , tl, tr, ql, qr)

0	1	2	3	4	5	6	7	8	9
5	1	6	12	8	15	10	11	2	2

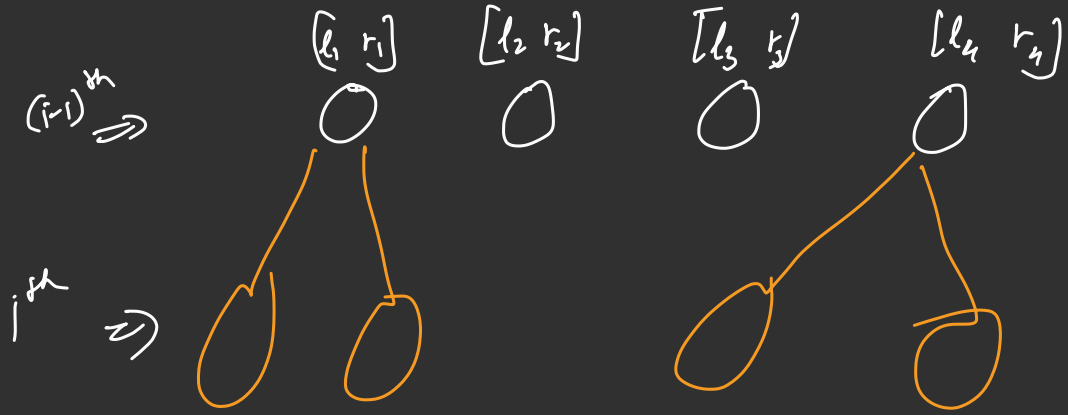
arr = [5, 1, 6, 12, 8, 15, 10, 11, 2, 2]

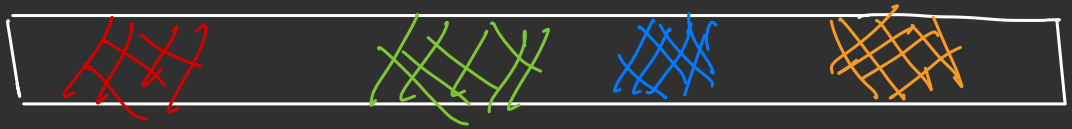
ql, qr ⇒ [2, 7]



Time Complexity for Querying :

For level 0, 1, 2 ⟹ ≤ 4 nodes explored.





Then, T.C. = $O(\log N)$

Let’s do a problem

1. Range Sum Queries

Let’s implement

Thank You!

Reminder: Going to the gym & observing the trainer work out can help you know the right technique, but you'll muscle up only if you lift some weights yourself.

So, PRACTICE, PRACTICE, PRACTICE!