Segment Trees

- Priyansh Agarwal

Problem 1

- Given an array of ***N*** elements and ***Q*** queries, In each query you will be given two indices ***L*** and ***R***, you need to output the sum of values of the array from index ***L*** to ***R***.

$(1 <= N, Q <= 10^5)$

- **Example:**

  A = [3, 5, 2, 8]

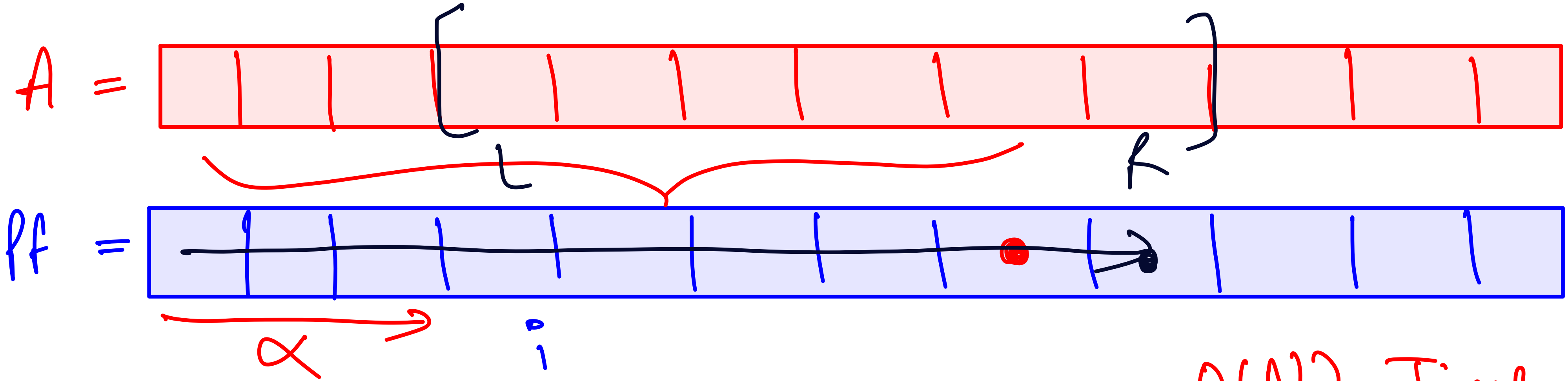
  Query: 2 3, Output: 7

  Query: 1 4, Output: 18

Time & Space with

Prefix sum = O(N+Q) time

O(N) space



$$pf[i] = \sum_{j=0}^{i} A[i]$$

O(N) Time & Space for precomputa-

$$A[L] + A(L+1) + A(L+2) \; --- \; A[R]$$

$$pf[R] = \sum_{i=0}^{R} A[i]$$

$$pf[L-1] = \sum_{i=0}^{L-1} A[i]$$

$$\underbrace{pf[R] - pf[L-1]} = \sum_{i=L}^{R} A[i]$$

Problem 2

Difference Array

- Given an array of ***N*** elements filled with ***0s*** and ***Q*** queries, In each query you will be given two indices ***L*** and ***R*** and an integer ***X***, you need to add ***X*** to all the indices of the array from ***L*** to ***R***. Find out the resulting array after all the queries.

Brute force = $O(Q \cdot N)$ ***($1 <= N, Q <= 10^5$)***

$O(N+Q)$ time

$N \to 10^5$

$Q \to 10^5$

$O(N)$ space

- **Example:**

  A = [0, 0, 0, 0]

  Query: 2 3 6, A = [0, 6, 6, 0]

  Query: 1 4 7, A = [7, 13, 13, 7]

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

3, 6, 8

+8

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

−8

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

A =

1	2	3	4	5	6	7	8	9	10
0	0	0	0	0	0	0	0	0	0

2, 7, 3 =

1	2	3	4	5	6	7	8	9	10
0	3	0	0	0	0	0	-3	0	0

O(1)

3, 6, 5 =

1	2	3	4	5	6	7	8	9	10
0	3	5	0	0	0	-5	-3	0	0

O(1)

8, 10, 2 =

1	2	3	4	5	6	7	8	9	10
0	3	5	0	0	0	-5	-1	0	0

O(1)

Resultant =

1	2	3	4	5	6	7	8	9	10
0	3	8	8	8	8	3	2	2	2

O(N)

Difference Arrays

for every query

→ Add X at index L

→ subtract X at index R+1

for resultant array

→ Take the prefix sum

Given an array A of N elements filled with 0s initially and Q queries.

$Q \rightarrow 10^5$

Query $\rightarrow$ L, R, X

$N \rightarrow 10^5$

$\hookrightarrow$ make every value in array (a[i]) from L to R as

a[i] = a[i] xor X

find out the resultant array.

A =

0	0	0	0	0	0	0	0	0	0
1	2	3	4	5	6	7	8	9	10

2, 5, 4 =

0	4	0	0	0	4	0	0	0	0
1	2	3	4	5	6	7	8	9	10

3, 8, 2 =

0	4	2	0	0	4	0	0	2	0
1	2	3	4	5	6	7	8	9	10

Resultant =

0	4	6	6	6	2	2	2	0	0
1	2	3	4	5	6	7	8	9	10

Problem 3 (Segment Tree)

- Given an array of ***N*** elements and ***Q*** queries. Each query is one of the ***2*** types.

1: Change the value of A[index] to X → log n

2: Find the sum of all values from L to R → logn

$N \cdot Q$

(1 <= N, Q <= 10⁵)

The queries can be interleaved so we need to evaluate both of them quickly.

✗ 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2

✓ 2 1 1 2 1 2 2 1 1 1 2 2 2 2 1 1 2 1 2 2

Problem 3

- **Example:**

  A = [4, 5, 1, 2]

  Query: **2** 3 4 - Sum from 3 to 4 = 1 + 2 = 3

  Query: **1** 3 7 - Change 3rd element to 7, A = [4, 5, 7, 2]

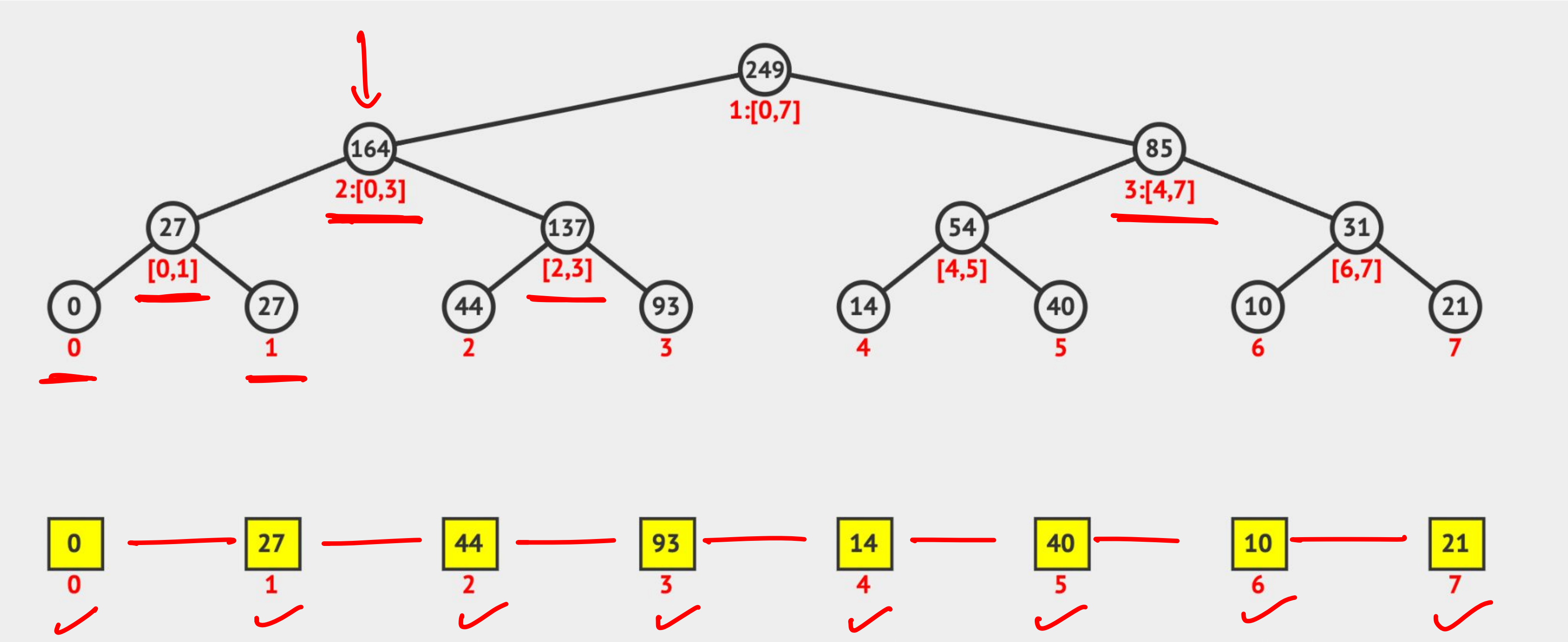
  Query: **2** 2 4 - Sum from 2 to 4 = 5 + 7 + 2 = 14

Segment Trees to the Rescue!

Let’s Learn Segment Trees!

- Let’s divide the array into multiple segments
- How quickly can we reach a particular node?
- How much information is relevant at each node?

Let’s Learn Segment Trees!



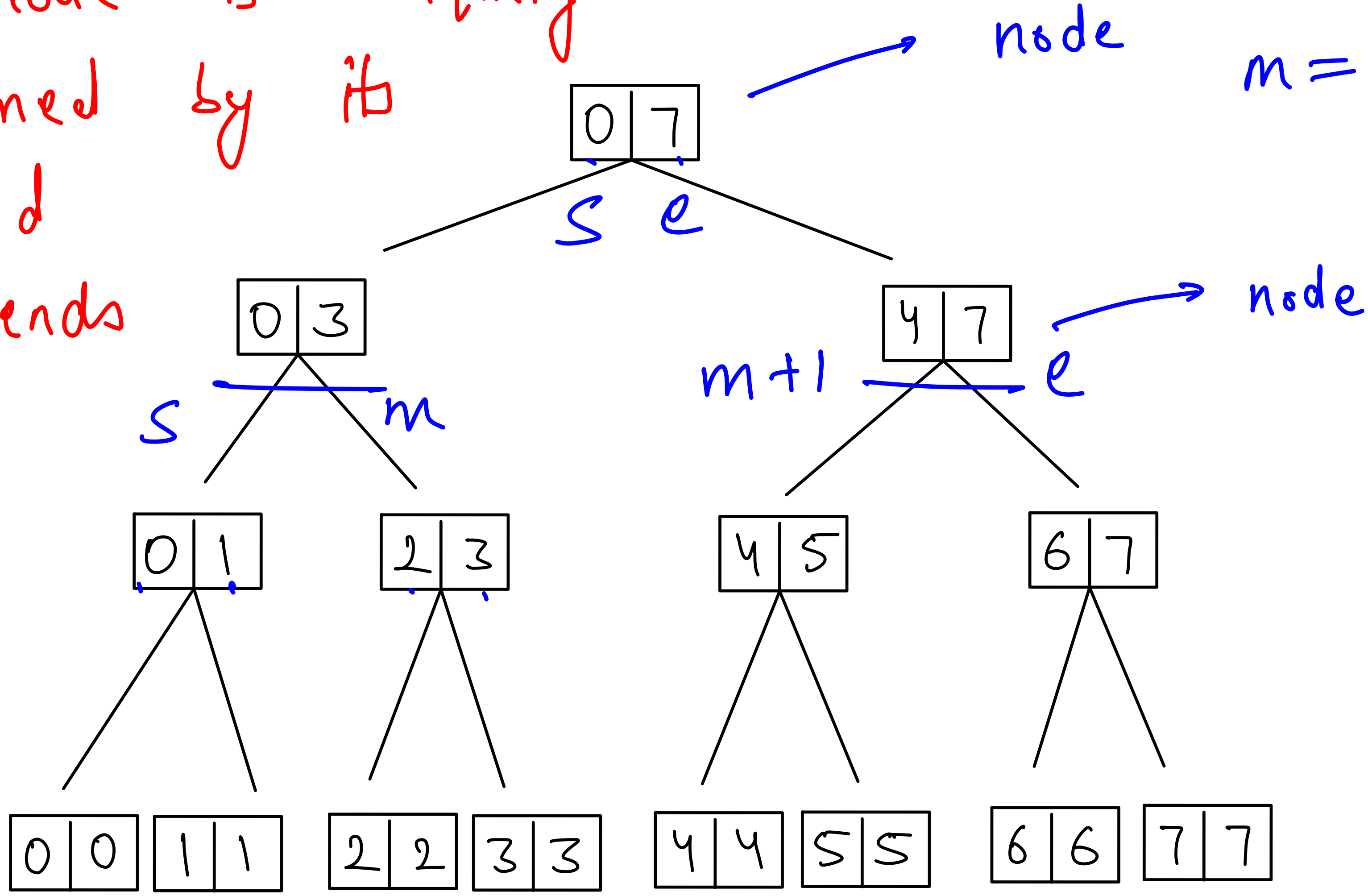
every node is uniquely determined by its left and right ends

$$m = \frac{s+e}{2}$$

$s = 0$

$e = 3$

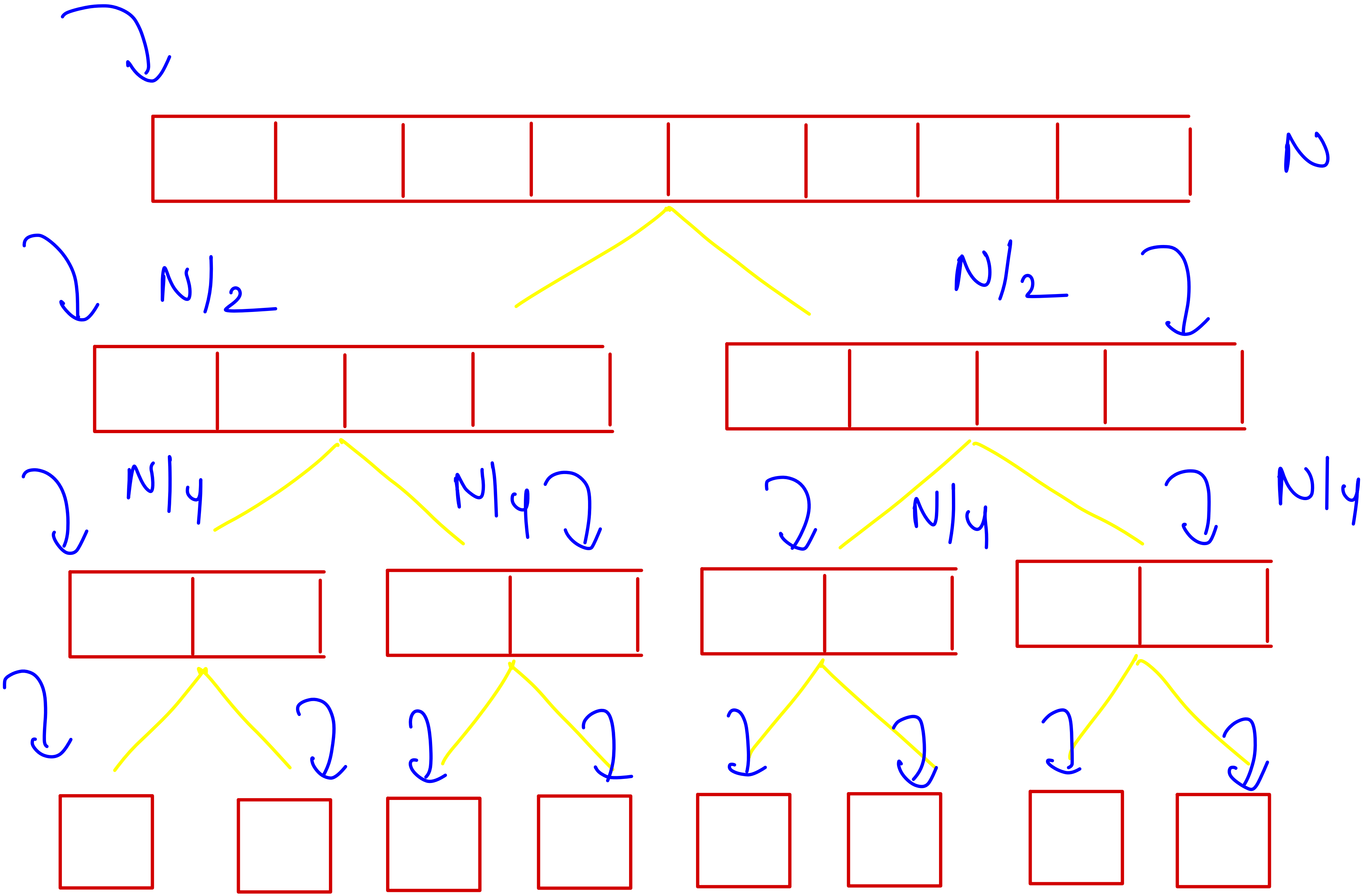
$m = 1$



nodes closer to the root are wider → they have more elements from the array

leaf node

corresponds to an actual element in the array


N
N/2
N/2
N/4
N/4
N/4
N/4

Space $\rightarrow 1 \cdot N + 2 \cdot N/2 + 4 \cdot N/4 \; - \ldots\ldots$

$$\rightarrow \sum_{l=1} 2^{l-1} \cdot \boxed{\frac{N}{2^{l-1}}}$$

$\rightarrow$ no. of elements from the array in a node at level l

leaf node $\rightarrow \frac{N}{2^{l-1}} = 1 \Rightarrow N = 2^{l-1}$

$$l-1 = \log_2 N$$

$$l = \log_2 N + 1$$

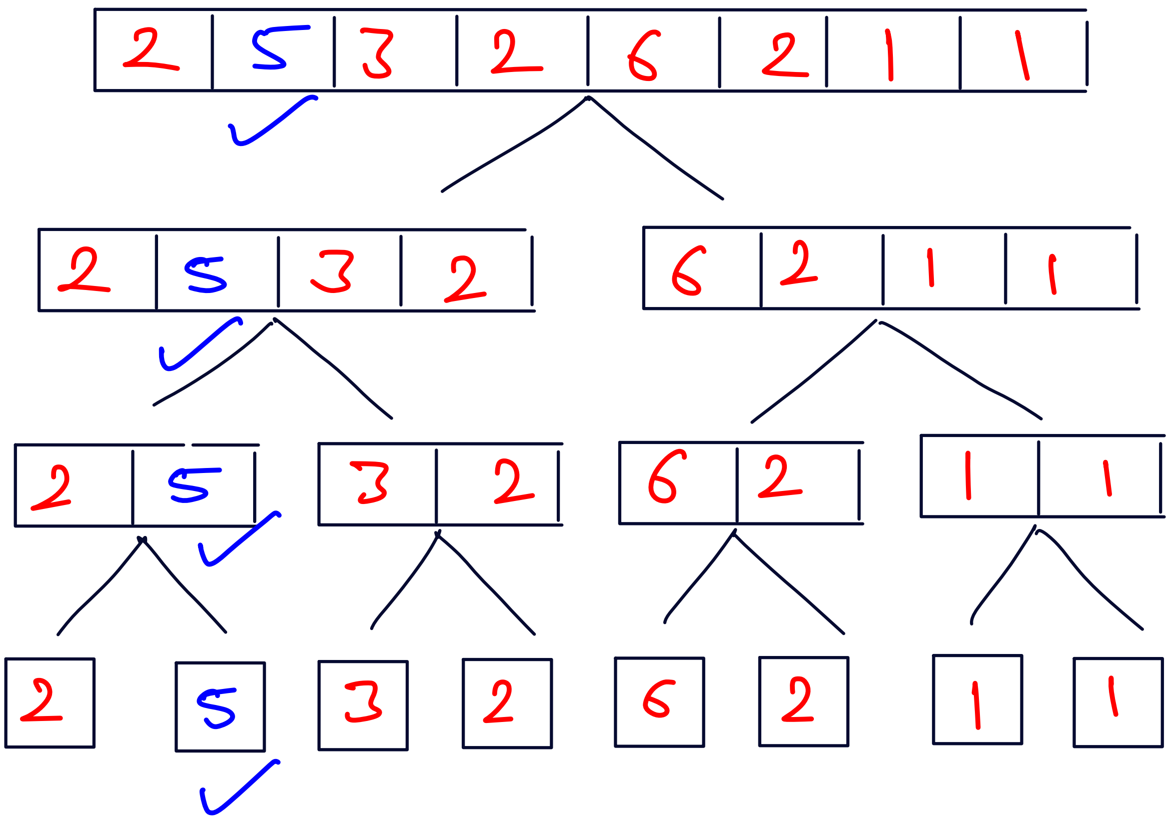
$$\rightarrow \sum_{l=1}^{\log_2 N + 1} \cancel{2^{l-1}} \cdot \frac{N}{\cancel{2^{l-1}}}$$

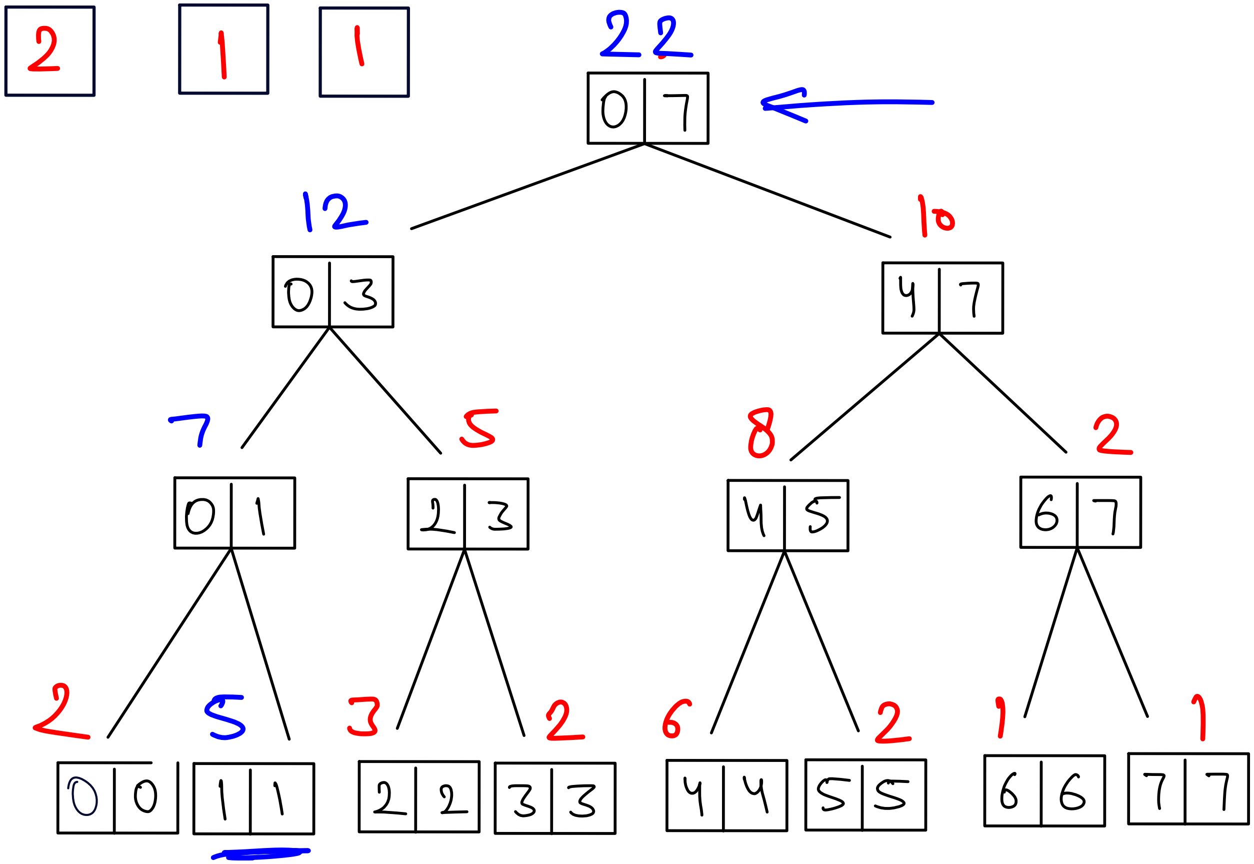
$$\rightarrow N(\log_2 N + 1) = \underline{\underline{N \log N}}$$

when we store the entire subarray in every node

2	1	3	2	6	2	1	1

$1 \rightarrow 5$





Space complexity $\longrightarrow$ Assume N is a power of 2

$$1 \cdot (1) + 2 \cdot (1) + 4 \cdot (1) \ldots \ldots N(1)$$

$$S.C = N + \frac{N}{2} + \frac{N}{4} + \frac{N}{8} \ldots \ldots)$$

$$= \underline{\underline{2N-1}}$$

$$N = 2^k \implies 1\underbrace{0000000000}_{k \text{ zeros}}$$

$N/2 \longrightarrow 1000000000$

$N/4 \longrightarrow 100000000$

$N/8 \longrightarrow 10000000$

$N + N/2 + N/4 + N/8 \cdots 1$

$$N = 1\underbrace{000000000000000}_{k \text{ zeros}}$$

$N = 2^k$

$$\underbrace{111111111111111}_{k+1 \text{ ones}}$$

$k = n$

$$\begin{array}{r} 10000000000000000 \\ - \quad 1 \\ \hline 0000000000000001 \end{array}$$

$= 2^{k+1} - 1 \Rightarrow 2N - 1$

$N = 2^4 \Rightarrow 16$

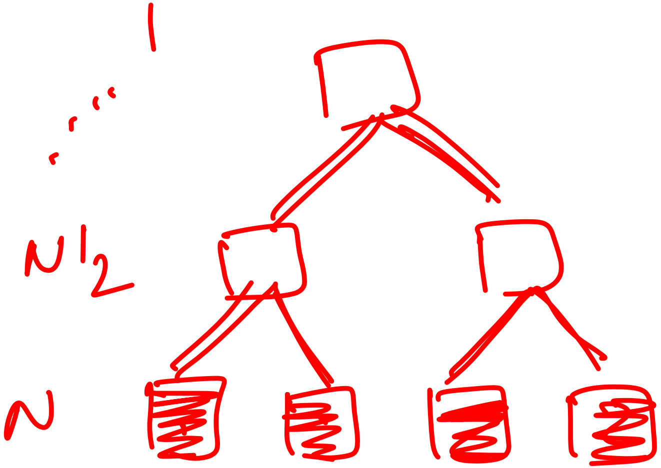
$N = 10000$

$N + N/2 + N/4 \; \cdots\cdots \; 1$

$$\left.\begin{aligned} N &= 10000 \\ N/2 &= 01000 \\ N/4 &= 00100 \\ N/8 &= 00010 \\ N/16 &= 00001 \end{aligned}\right\} \quad 11111 \Downarrow 31 \Downarrow 32-1 = 2N-1$$

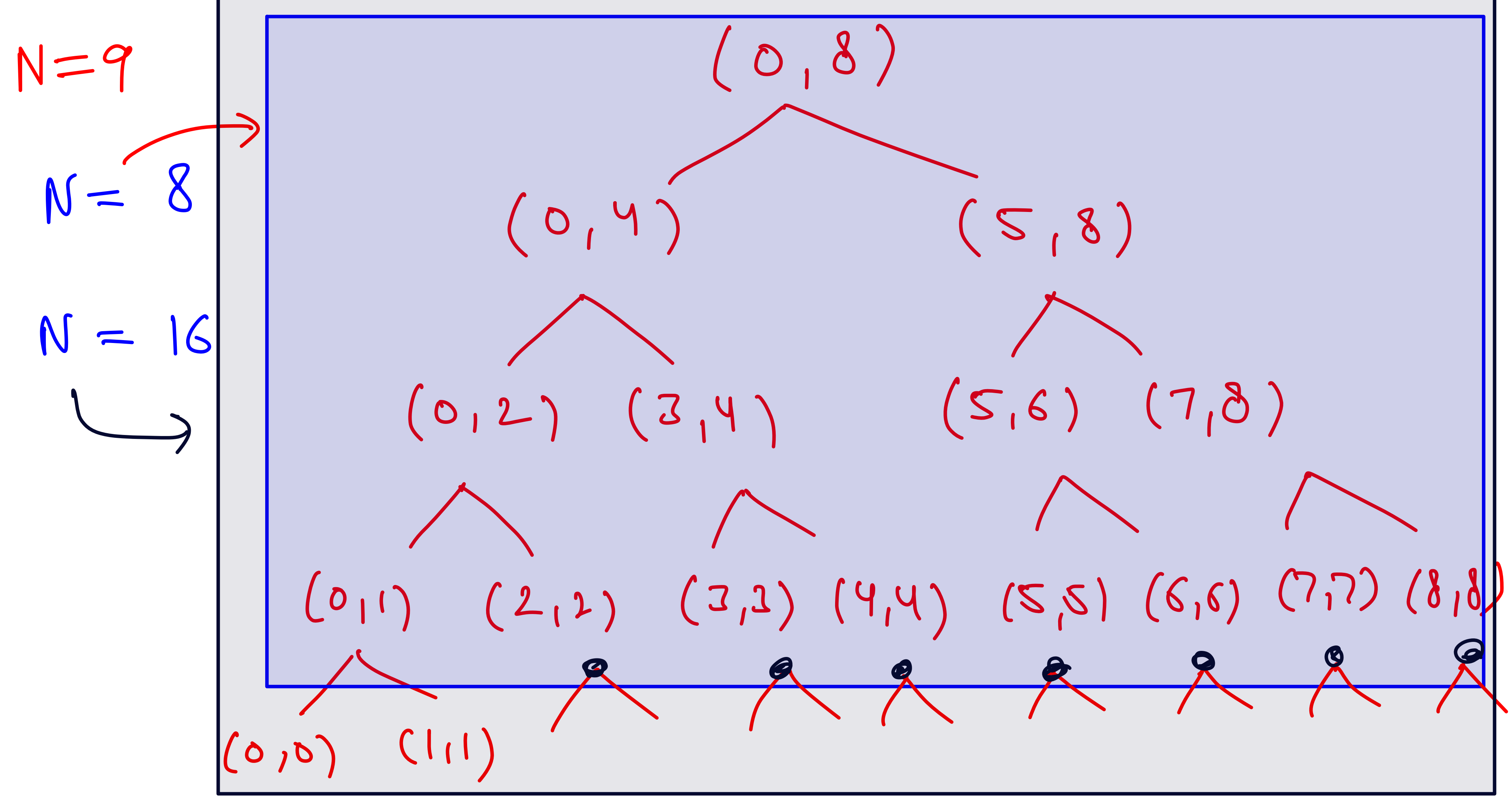
Number of Nodes in a Segment Tree

- N is a power of 2
    - N Leaves
    - N - 1 internal nodes
    - 2*N - 1



Number of Nodes in a Segment Tree

- N is a not a power of 2
  - N is <= 2^K
  - Number of nodes for N <= Number of nodes for 2^K
  - 2^K <= 2N
  - Number of nodes for N <= Number of nodes for 2N
  - Number of nodes for N <= 2(2N) - 1
  - Number of nodes for N < 4N



$N = 9$

$\underline{\text{nodes}(8)} < \text{nodes}(9) < \text{nodes}(16)$

$2 \cdot 8 - 1 \qquad\qquad 2 \cdot 16 - 1$

if N is not a power of 2

nodes (N) $\leq$ nodes $(\underline{2^k})$

such that 2^k is the first power of $2 \geq N$

$\underline{\text{nodes}(N)} \leq \underline{2 \cdot (2^k)} \; \underline{-1}$

2^k = 1st power of $2 \geq N$

$$\underline{\underline{2^k \leq 2N}}$$

$N = 9 \longrightarrow 16 \leq 18$

$N = 5 \longrightarrow 8 \leq 10$

$N = 19 \longrightarrow 32 \leq 38$

$$\begin{tikzcd} 2^{k-1} \arrow[rr, bend left, "\times 2"] \arrow[r, dash] & N \arrow[rr, bend left, "\times 2"] \arrow[r, dash] & 2^k & {} \end{tikzcd}$$

① $\text{nodes}(N) \leq 2 \cdot (2^k) - 1$

② $2^k \leq 2 \cdot N$

$\Rightarrow$ $\text{nodes}(N) \leq 2 \cdot (2N) - 1$

$\text{nodes}(N) < 4N$

In general for any N

$$\underline{\underline{\text{nodes}(N) < 4N}}$$

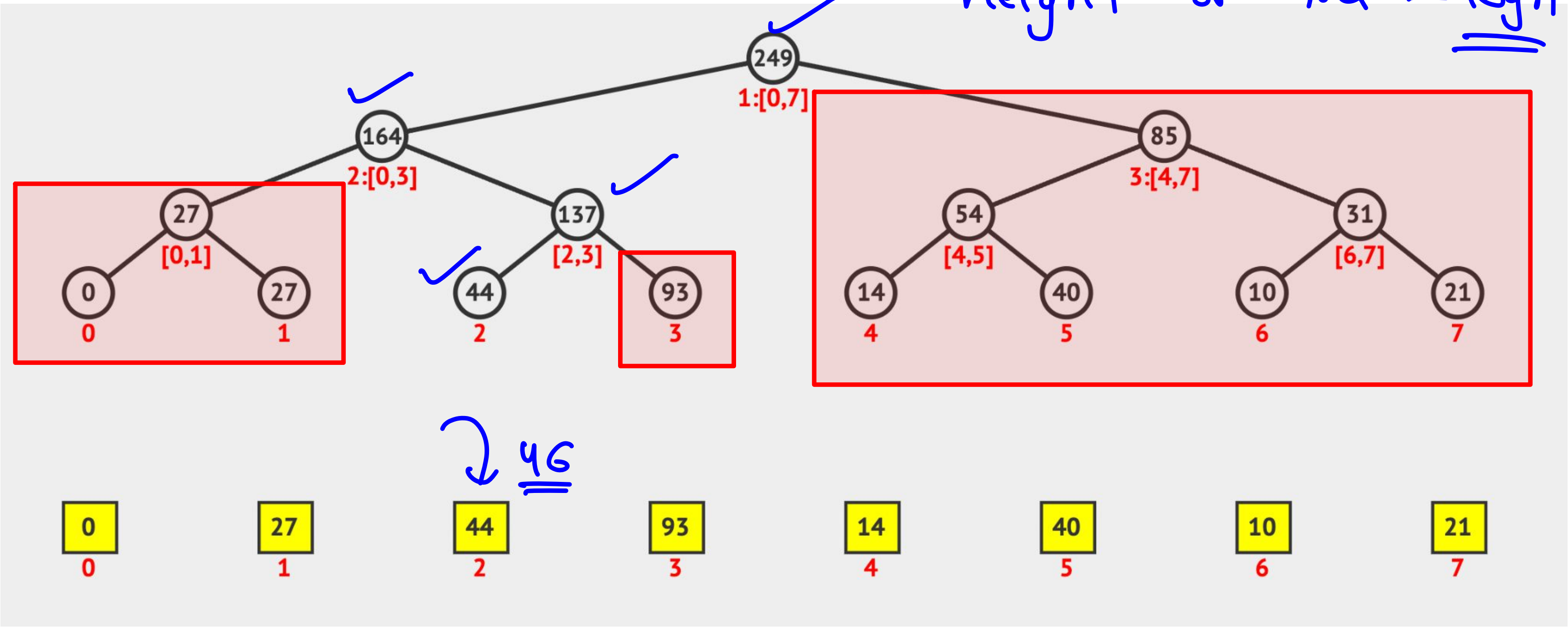
Point Update

- Reach the leaf node corresponding to the index
- Update all the nodes from leaf to the root involved in this
- How much time does this take?
- Is our entire tree updated?

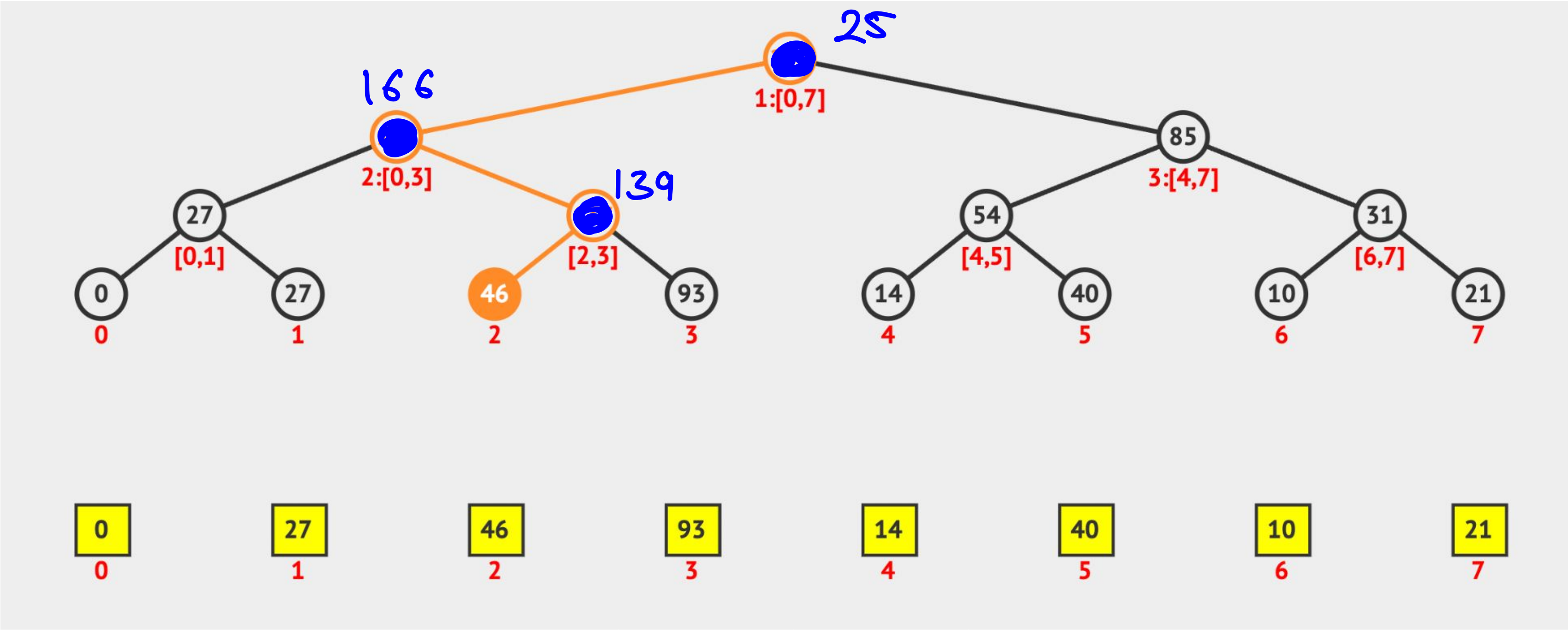
nodes to update = nodes from root to exact leaf

Point Update - Index 2 (Value - 46)

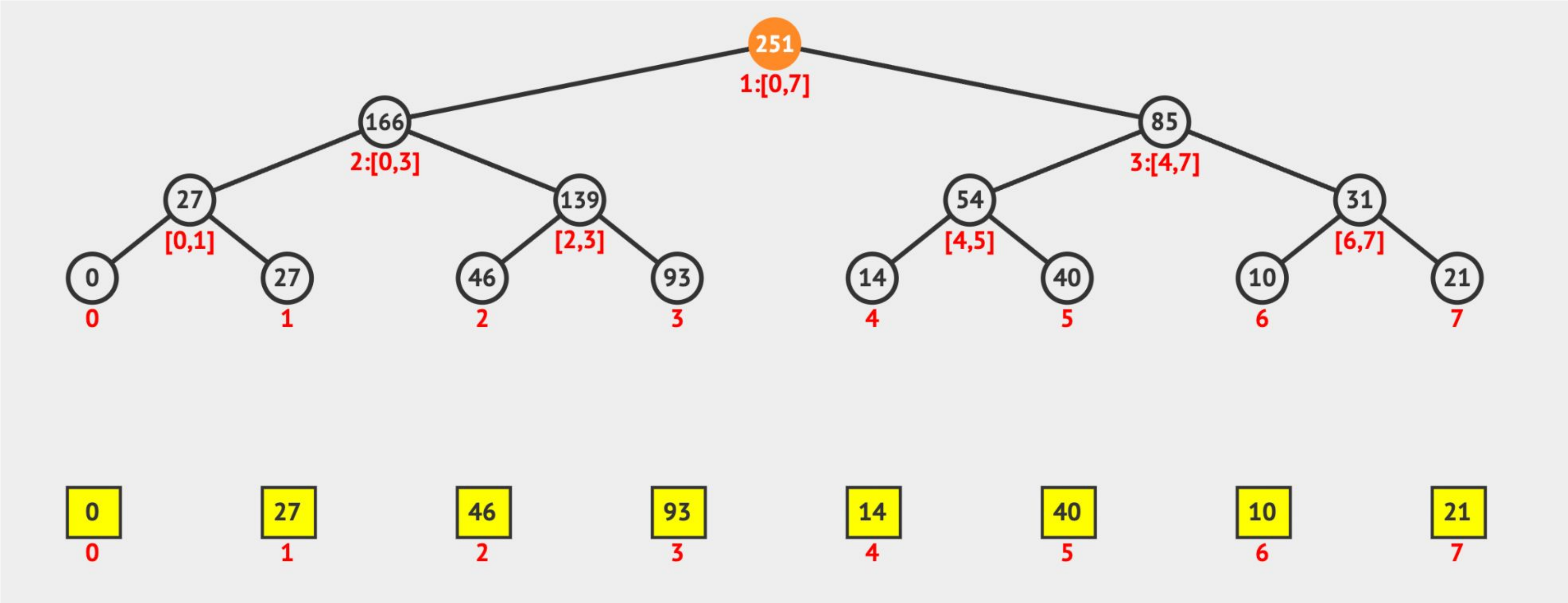
height of tree = log n



Point Update - Index 2 (Value - 46)



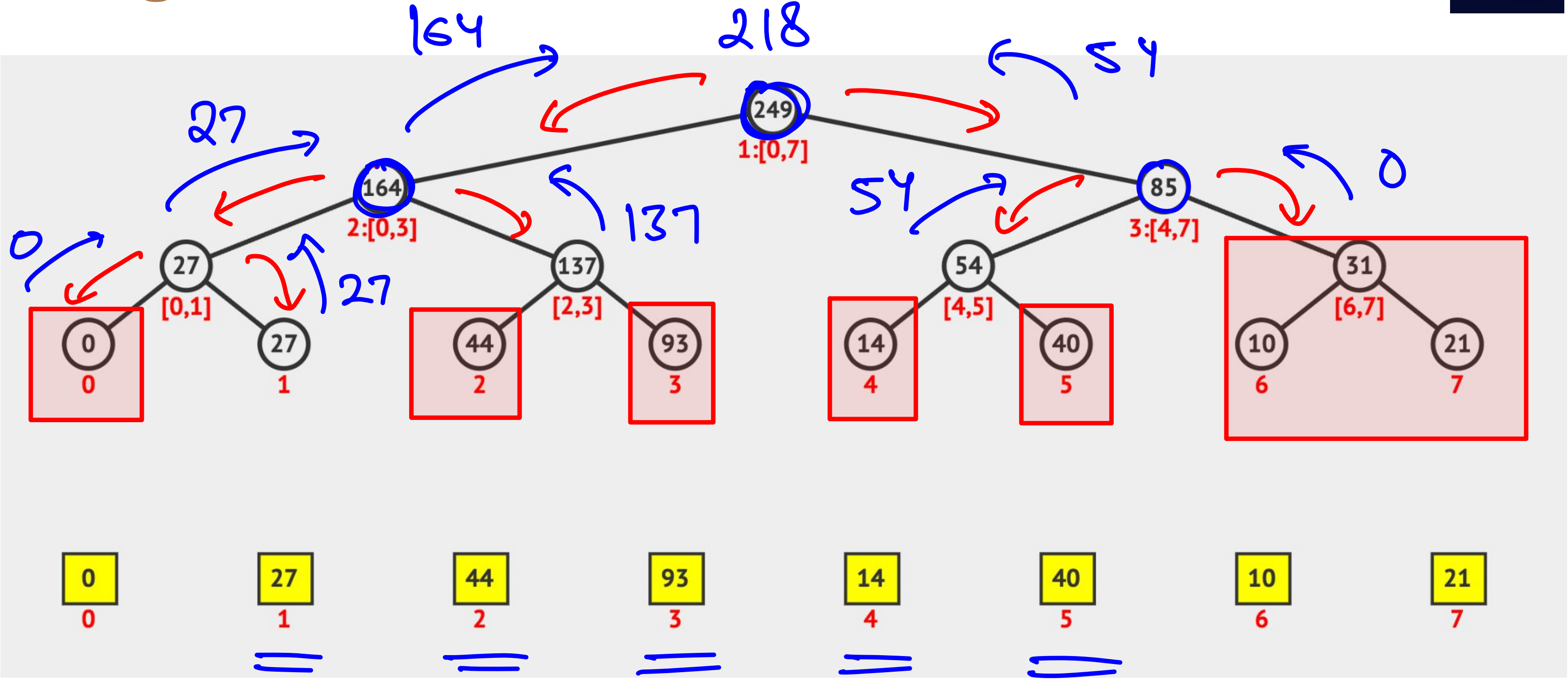
Point Update - Index 2 (Value - 46)



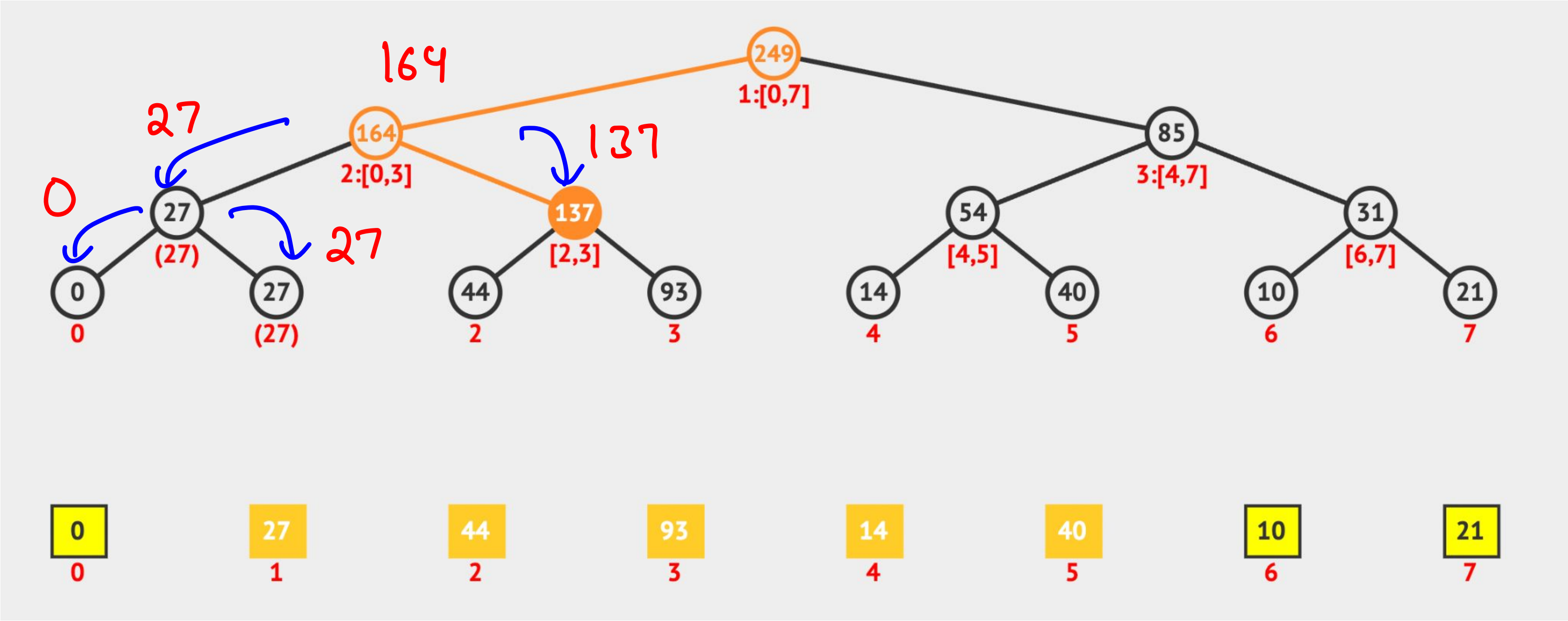
Range Queries

- How to reach the relevant range?
- What happens when current range is completely useful? → return
- What happens when current range is completely useless? → return
- What happens when current range is partially useful? → go down
- How many nodes can we explore at one level? → ≤ 4
- What is the time complexity of this? → $\leq 4 \log n$

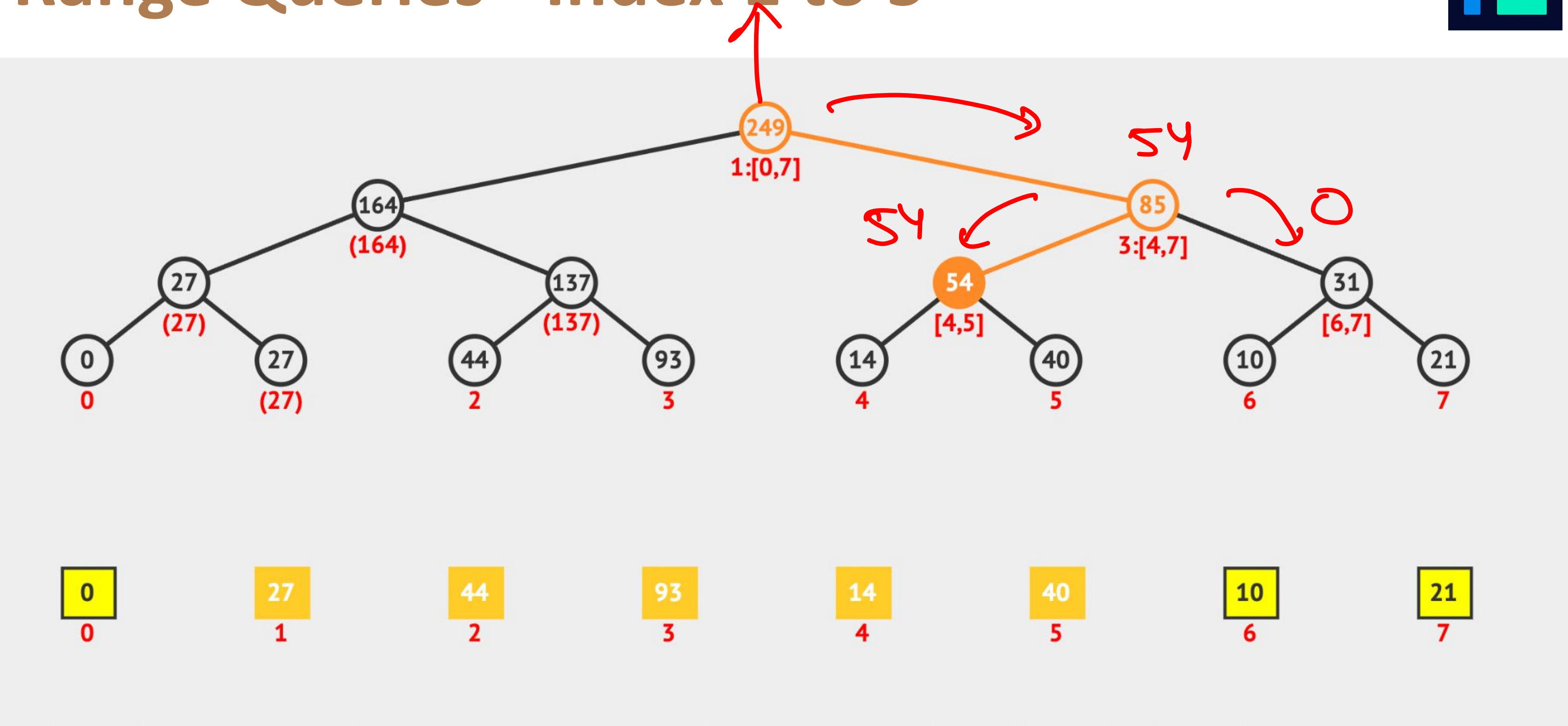
Range Queries - Index 1 to 5



Range Queries - Index 1 to 5



Range Queries - Index 1 to 5



Range query

- segment tree range
- query range

① Return 0

no contribution

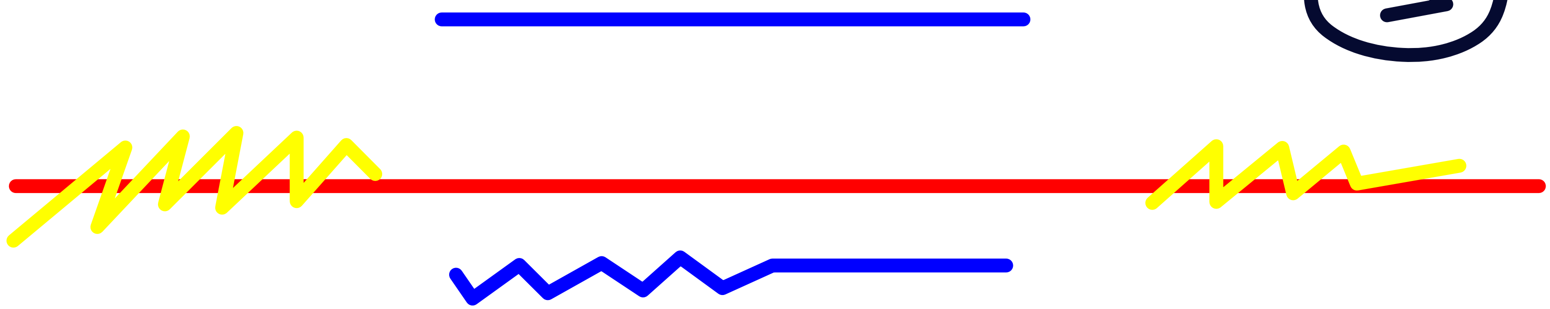
② Return node value

full contribution

③ Go down → Return L+R contrib.

partial contribution

3



$s, e \to$ segment tree

$l, r \to$ query

1. $s > r$ or $e < l$ if

   no contribution

2. $s \geq l$ and $e \leq r$ else if

   complete contribution

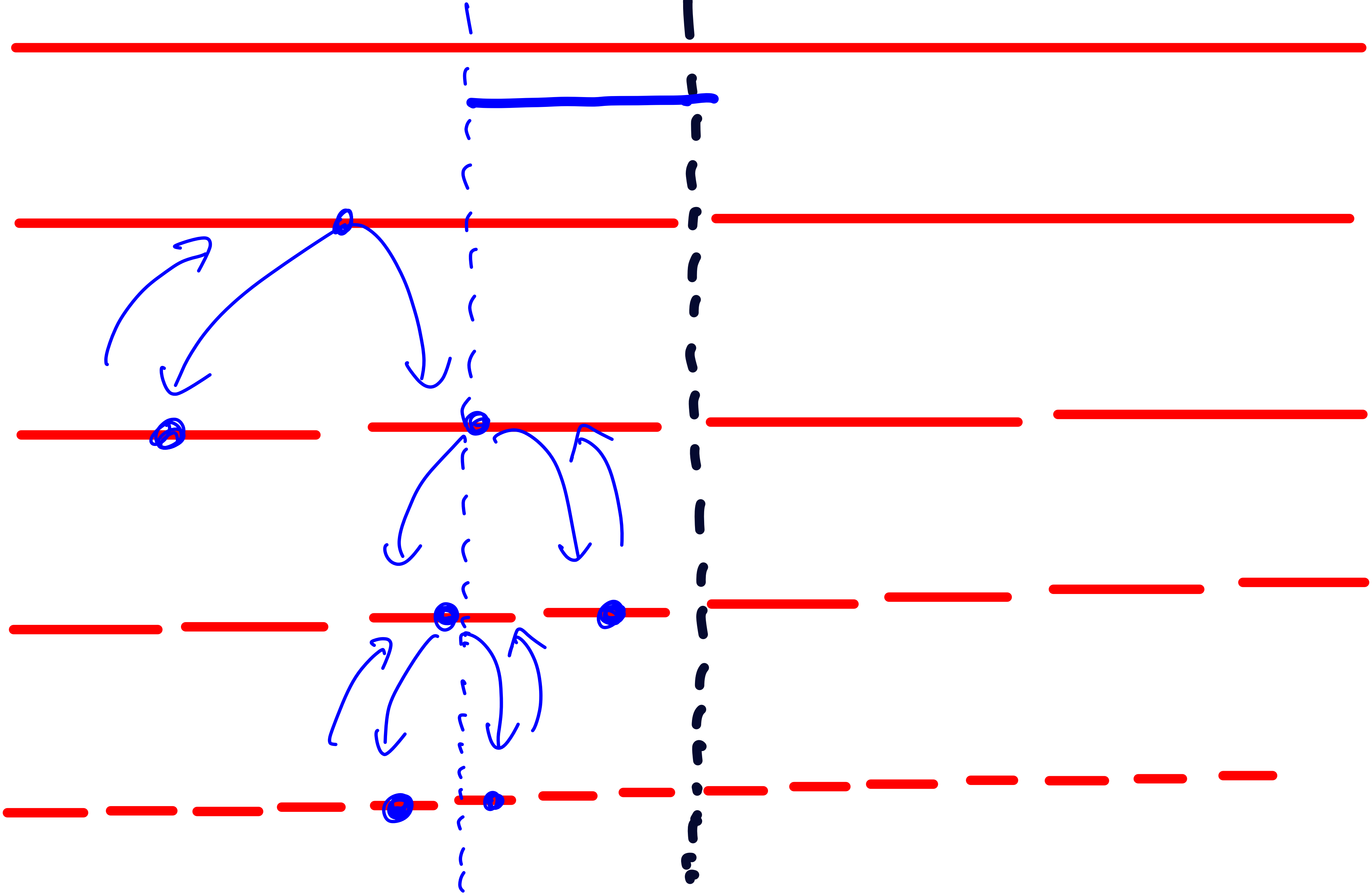
3. partial contribution else

we only go down a node to its children when it is a partial overlap

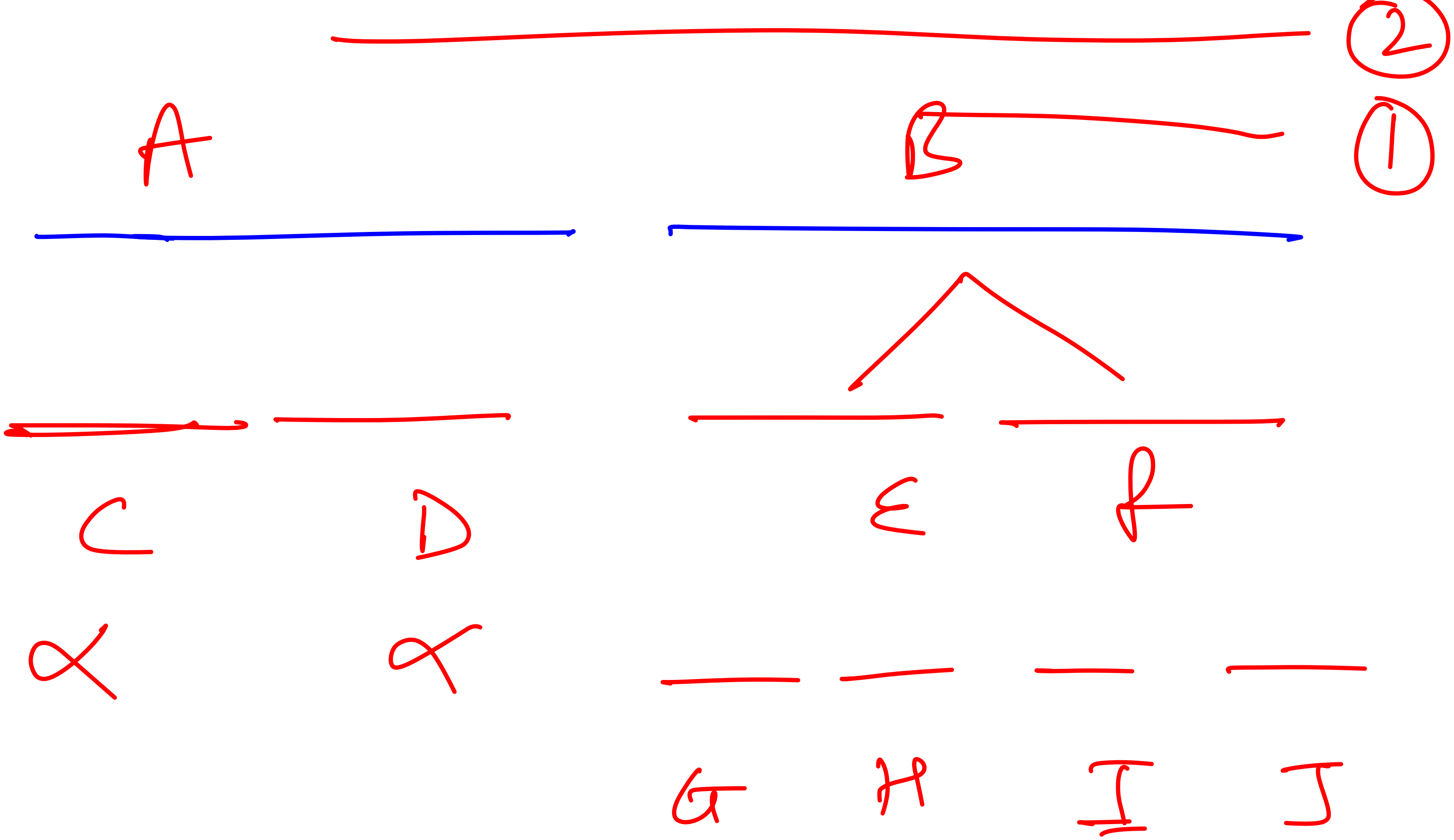
T.C $\propto$ partial overlap

for every level we will not visit more than 4 nodes

T.C $\rightarrow$ $4\log n$



A query range is contiguous



2
A
B
1
C
D
E
F
α
α
G
H
I
J

Building The Segment Tree

- What defines a node uniquely? (node_no. , s and e point)
- Dynamic Programming? No Divide and Conquer Yes
- Can we define each node with a unique integer?
- How to find child and parent nodes of a particular nodes?
- How to store the information? array
- What is the time complexity to build the tree? ?

$x \to$ left $2x$

$x \to$ right $2x+1$

$x \xrightarrow{\text{parent}} x/2$

Array	10	2	3	4	2	1	1	7
	0	1	2	3	4	5	6	7

first node in tree

→ root = node 1

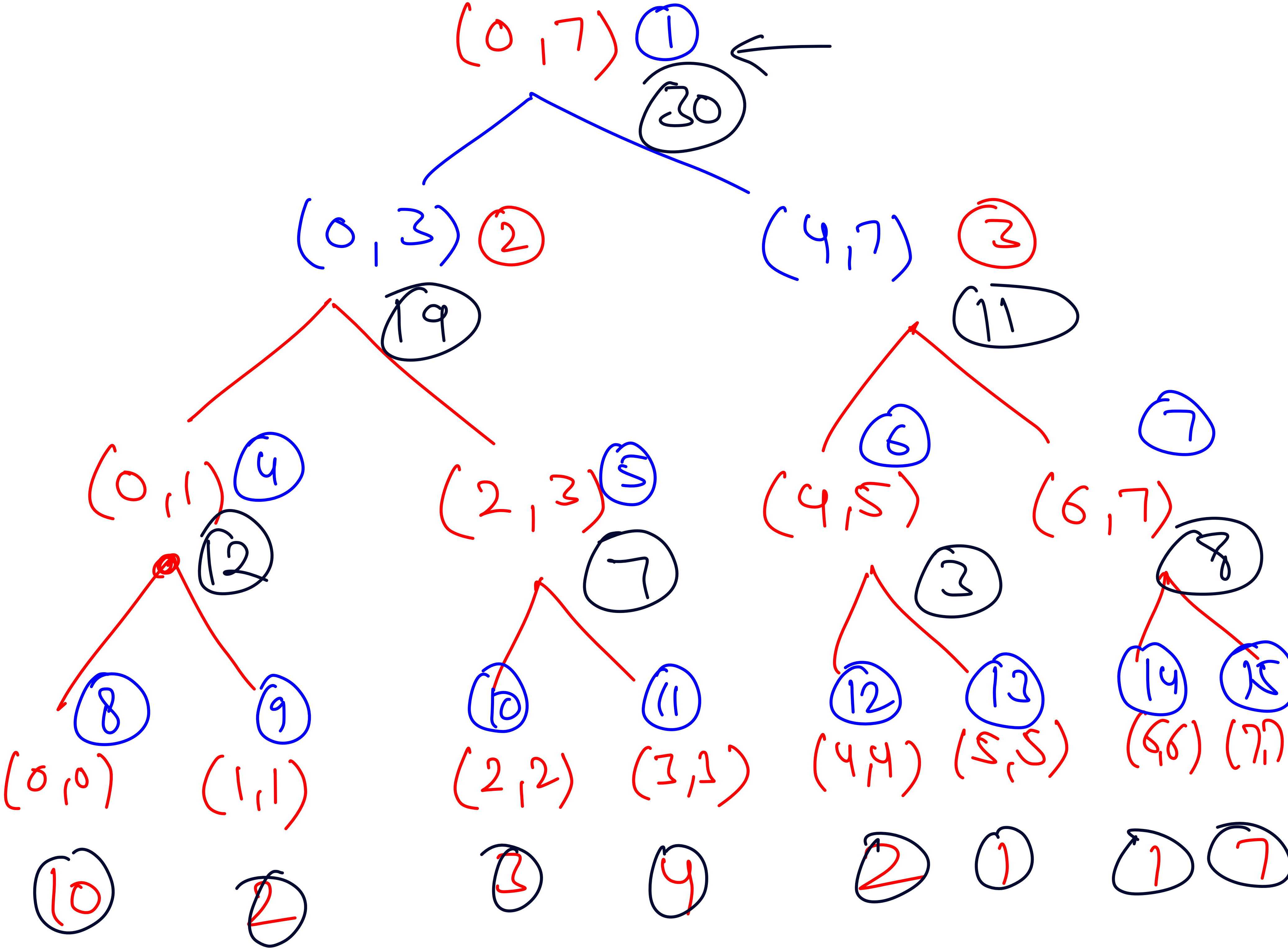
→ range = $(0, n-1)$

left and right child of every node

left = $\underbrace{2 \cdot \text{node\_val}}$ $[s, m]$

right = $2 \cdot \text{node\_val} + 1$ $[m+1, e]$

if node has $[s, e]$ as the range



(0,7) 1 30
(0,3) 2 19
(4,7) 3 11
(0,1) 4 12
(2,3) 5 7
(4,5) 6 3
(6,7) 7 8
(0,0) 8 10
(1,1) 9 2
(2,2) 10 3
(3,3) 11 4
(4,4) 12 2
(5,5) 13 1
(6,6) 14 1
(7,7) 15 7

Problems

- Range Min
- Range Xor
- Range GCD