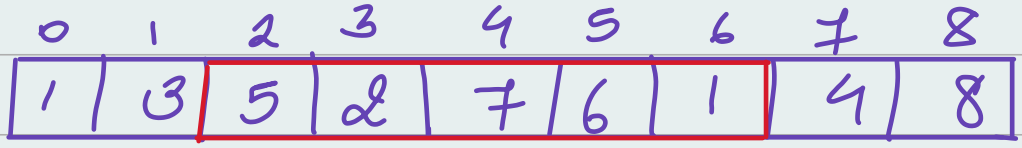
Range Query Interview Questions

Q)



Sum b/w index (2, 6)

Ans : Run a for loop from index 2 to 6.

Worst case : Time Complexity = O(N)

→ Do better.

Algorithm : Square Root Decomposition.

* Used to optimize queries

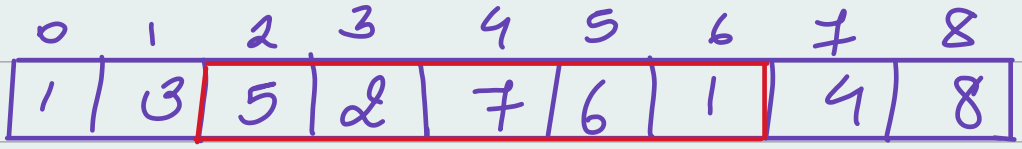
* Reduces time complexities to $O(\sqrt{N})$

steps :-

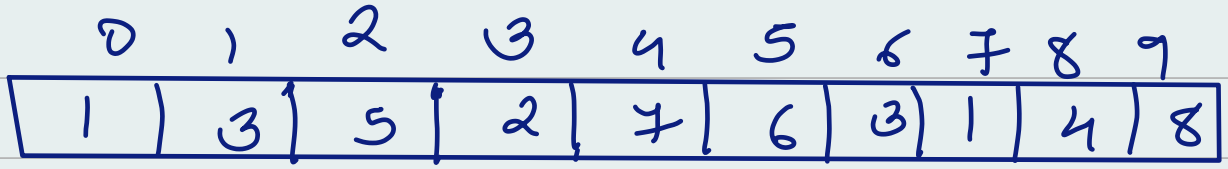
① Divide the array into blocks of size $\sqrt{N}$.

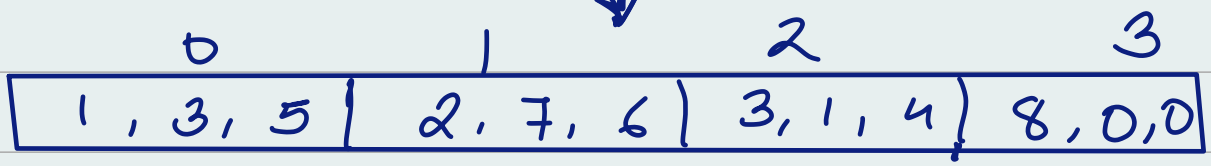
② Compute ans for every block

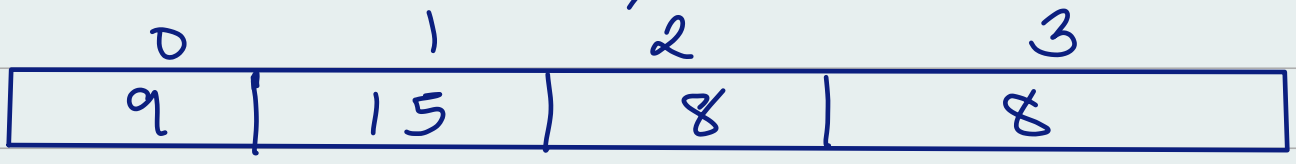
③ Given a question or query for left to Right, combine the blocks that contain this



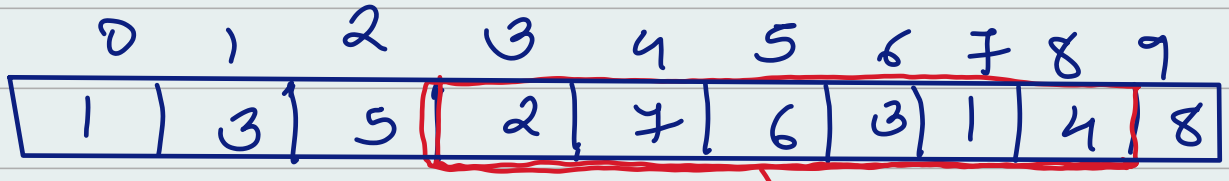
$\sqrt{N} = \sqrt{9} = 3$







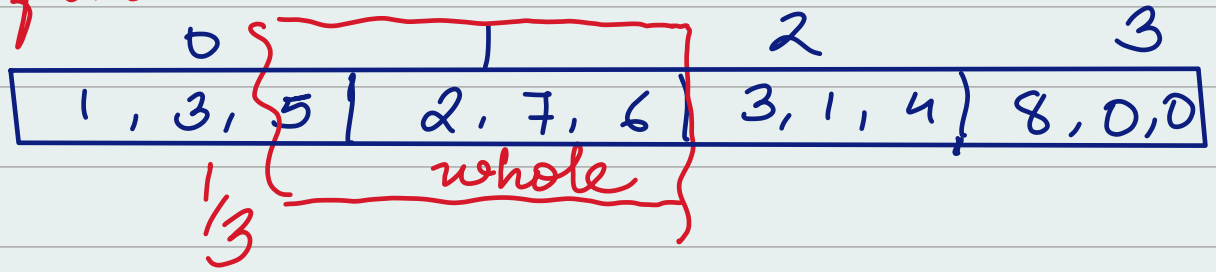
(Preprocessing)



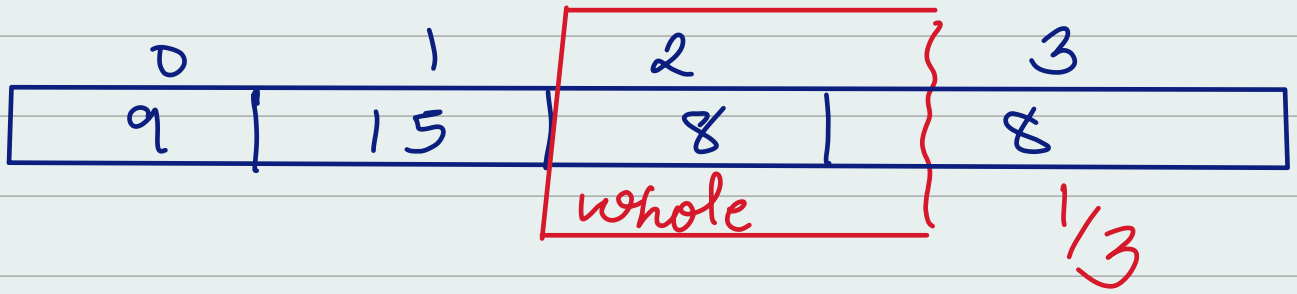
i-case

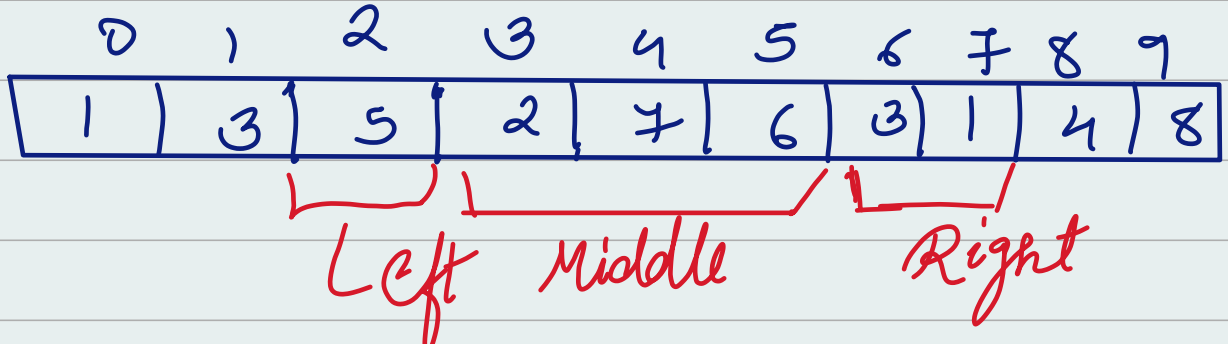
i) First case: taking whole blocks.
like indices from 3 till 8

ii) Second Case: taking 1/3 from first block and the whole from other block
from index 2 to 5



iii) Third case: Similarly taking last block as 1/3 and the other block as whole From index 6 till 9.



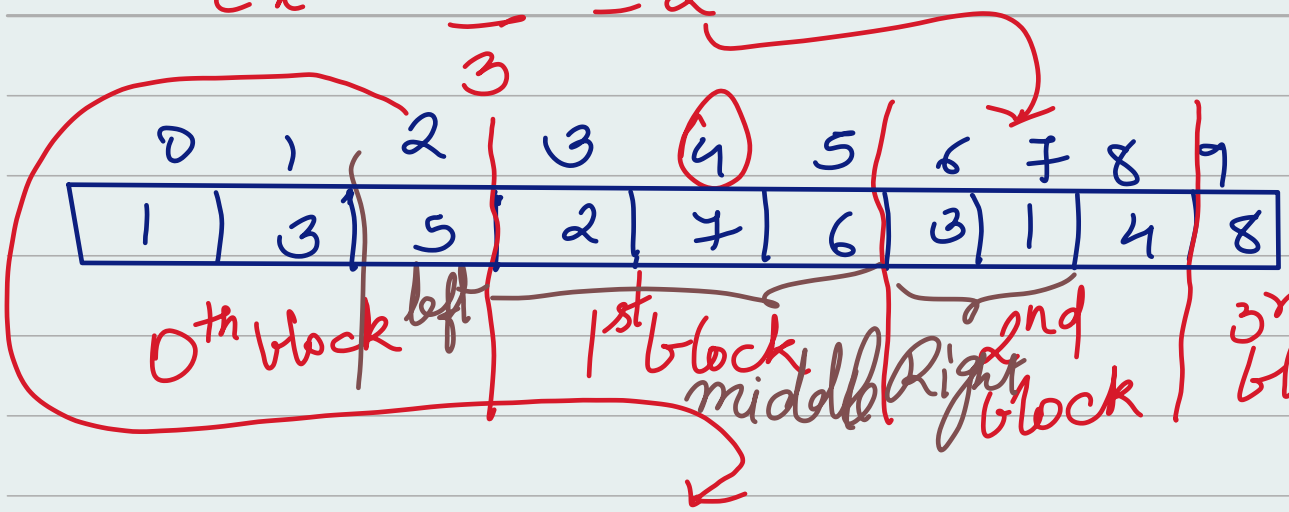


$(2, 7) = (2,2) + (3, 5) + (6, 7)$

Here $\boxed{q} = \frac{i}{\sqrt{N}} = \frac{4 \rightarrow \text{index}}{3} = 1$

which block does index no 7 lies in? Ans block 2.

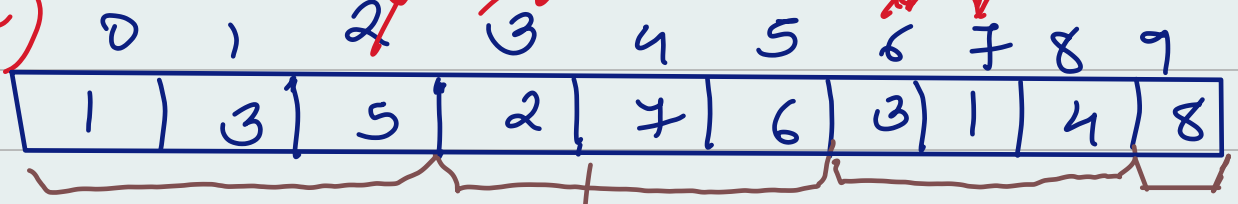
Ex: $\frac{7}{3} = 2$



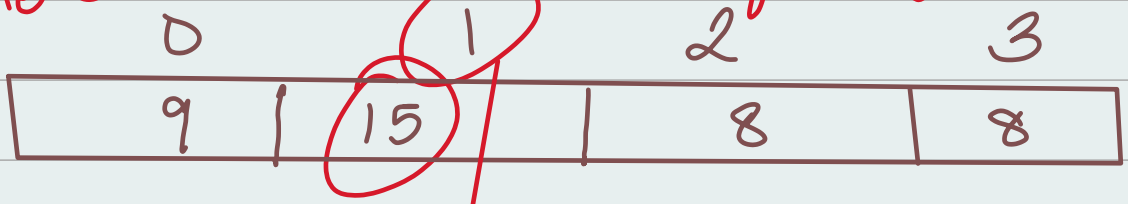
$i = \frac{2}{3} = 0$

(Jump left sqrt times)

(2,7)



Here chunks are size of ($\sqrt{N}$)



l	r	Sum
2	7	5
3		5+15=20
6		20+3=23
7		23+1=24

left i:

$l \% \sqrt{N} = 2 \% 3 = 2$

$l++$

$l \% \sqrt{N} = 3 \% 3 = 0$

left become 0 come out of the loop left part is done, now for middle part.

while (middle ≥ 1 ✓

if $l + \sqrt{N} \leq r$

if (3) + (3) ≤ 7 (table)

Sum = Sum + block $\left(\frac{l}{\sqrt{N}}\right)$.

= Sum + $\left(\frac{3}{3}\right)$

= Sum + index(1)

= 5 + 15 = 20

Here we don't have to calculate the sum of each & every block In the above array it is already preprocessed calculated. Take the whole block sum as it is

Now jump the left index by sqrt times

Now, right part:

while (l ≤ r)

(6 ≤ 7 → 3
7 ≤ 7 → 1) 4+ → 4

Now left = Right

break.

Done

Suppose we want to update the index number 4 value 7 to 8

arr [4] = 8 update

$i = l/\sqrt{N}$

$i = 4/3 = 1$

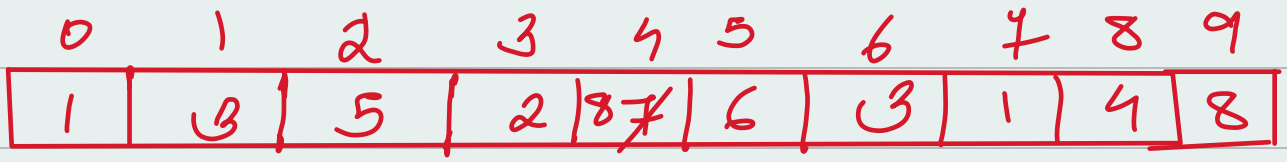
block[1]

then

block [1] = 15 - 7 + 8

= 15 + (8 - 7)

block [i] + (value - arr [i])



updated.

Code :-

```
public class Main {

    public static void main(String[] args) {
        int[] arr = {1, 3, 5, 2, 7, 6, 3, 1, 4, 8};
```

```
int n = arr.length;
// build a block array.
int sqrt = (int) Math.sqrt(n);
int block_id = -1;
int blocks = new int[sqrt+1];
for(int i=0; i<n; i++){
    // new block is starting
    if(i % sqrt == 0){
        block_id++;
    }
    blocks[block_id] = blocks
        [block_id] + arr[i];
}
System.out.println(query(blocks,
        arr, 2, 7, 3));
}
```

```
private static int query(int[] blocks,
    int[] arr, int l, int r, int sqrt){
    int sum = 0;
    // left part

    while (l % sqrt != 0 && l < r &&
                        l != 0){

        sum = sum + arr[l];
        l++;

    }
    // middle part

    while (l + sqrt <= r){
        sum = sum + blocks[l/sqrt];
        l += sqrt;
    }

    // right part

    while (l <= r){
        sum = sum + arr[l];
        l++;
    }
```

```
        return sum;
    }

    public void update(int[] blocks,
        int[] arr, int i, int val, int sqrt){

        int block_id = i/sqrt;

        blocks[block_id] = block
                [block_id] +(val- arr[i]));

    }
}
```