Searching Algo

1) Linear Search — O(n)

2) Binary Search — O(log n)

Interviews → Binary Search

↳ Lower Bound

↳ Square Root finding

Interview based Questions

Good Point

high = num/2 = 9/2 = 4

num = 9

high = num

low = 0, high = 9

mid = 4

0	1	2	3	4	5	6	7	8	9

Value > num — value = 4 * 4 = 16

```
if(value == num){
    return mid;
}
if (value < num){
```

result = mid; — Non perfect square

left = mid + 1; — Right

}

if (value > num) {

high = mid - 1; — Left

}

num = 9

low = 0 high = 9/2 = 4

0 1 2 3 4

mid = 2 3

value = 2 * 2 = 4

3 * 3 = 9

if (value == num) {

return mid;

}

if (value < num) {

result = mid

left = mid + 1;

}

Reason

Square root of any number n cannot be more than n/2.

Binary Search

↳ Coding Part

0	1	2	3	4	5
2	4	8	12	20	27

target = 2

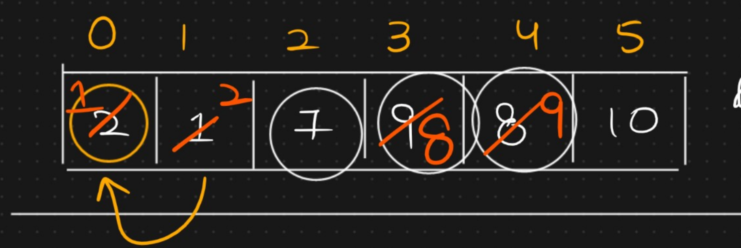
result = 0

priyabhatia@ineuron.ai

DSA —— LinkedIn

Sorting

- Bubble
- Selection
- Insertion



← sorted array

AND operator

i = 1 2 4 5 3 5

J = i = 1 0 J = 2 4

10 < 9

while(J>0 && arr(J) < arr(J-1)){

T 1 < 2 T

swap (arr(J), arr(J-1))

J--

Number system & Bit manipulation

Intution

Left shift —— *2

Right shift —— /2

2D Array

↳ Prefix Sum Approach

Query = 1

$r_1 = 2, c_1 = 2$

$r_2 = 3, c_2 = 3$

Brute force Approach

Optimized approach

	0	1	2	3
0	1	1	1	1
1	1	1	1	1
2	1	1	1	1
3	1	1	1	1

q* O(m * n)

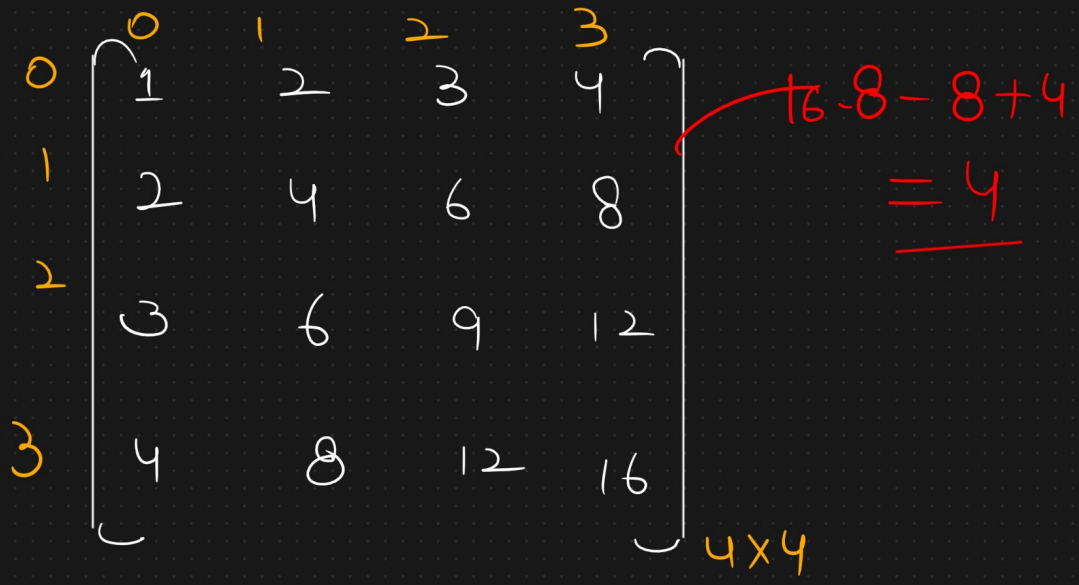
columns

rows

4

Row-Wise Prefix Sum

1	2	3	4
1	2	3	4
1	2	3	4
1	2	3	4



$r_1 = 0, \; c_1 = 0$

$r_2 = 1, \; c_2 = 1$