

## Search in a Sorted 2D matrix

mat

	0	1	2	3
0	00	01	02	03
1	10	11	12	13
2	20	21	22	23

  

5	7	11	14
25	28	30	39
45	52	68	74

$m = 3$  (row)  
 $n = 4$  (col)

### Brute Force :-

Search ~~at~~ in entire matrix and see if we find target element

$$T = O(mn), \quad S = O(1)$$

### Better Approach :-

We will traverse through the last column and see if last element of each row is greater or equal to target element.

↳ The row which satisfies the above condition, on that row we will apply binary-search to find target.

$$S_0, T = O(m) + O(\log(n)), \quad S = O(1)$$

### Best Approach :-

We will flatten the 2D matrix (In our mind)

flat

5	7	11	14	25	28	30	39	45	52	68	74
0	1	2	3	4	5	6	7	8	9	10	11

index

$(m \times n - 1)$

For any index,  $\text{flat}[\text{index}] = \text{mat}[\text{index}/n][\text{index}\%n]$

$$\text{So, row} = \text{index}/n$$

$$\text{col} = \text{index}\%n$$

$$(\text{index}/n, \text{index}\%n)$$

Then, we will apply binary search on flat array which will take  $O(\log(mn))$  time.

$$T = O(\log(mn))$$

$$S = O(1)$$