Binary Search

arr = [2, 4, 9, 10, 12, 14, 18, 19]

→ ascending

arr2 = [19, 12, 6, 5, 3, 2, −8, −16]

→ descending order

max comparisons : N ⇒ No. of elements

① find the middle element

② target > mid ⟹ search in the right
else search in left

③ if middle element == target element //ans

$\frac{0+9}{2} = 4$

s m e

0	1	2	3	4	5	6	7	8	9
2	4	6	9	11	12	14	20	36	48

arr = [2, 4, 6, 9, 11, 12, 14, 20, 36, 48]

target = 36

Search (12)

s				m					e
0	1	2	3	4	5	6	7	8	9
2,	4,	6,	9,	11,	12,	14,	20,	36,	48

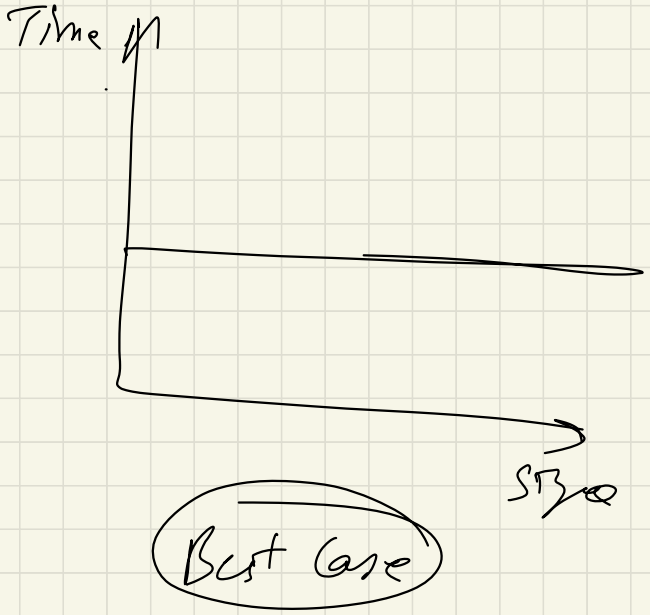
$\frac{5+9}{2} = 7$

s		m		e
5	6	7	8	9
12,	14,	20,	36,	48

s	e
5	6
12,	14

Ans = 5

if s > e : element not found.

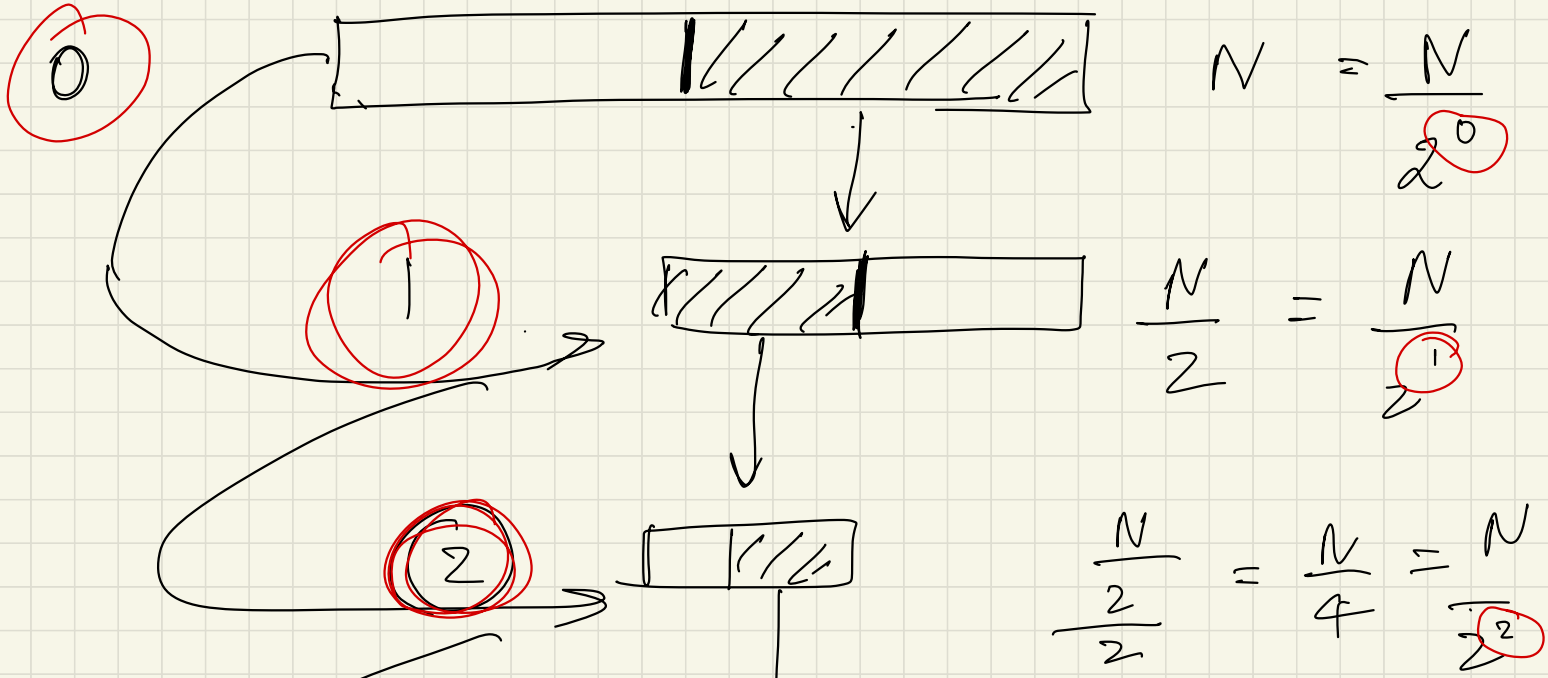


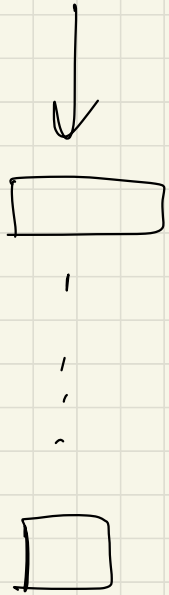
Best Case

O(1)

Why Binary Search?

Q: Find the max number of such comparisons in worst case.





$$\frac{N}{8} = \frac{N}{2^3}$$

k^{th} level

$$\frac{N}{2^k}$$

$$\frac{N}{2^K} = 1 \Rightarrow N = 2^K$$

$$\log(N) = \log(2^K)$$

$$\log N = K \log 2$$

$$K = \frac{\log N}{\log 2} \Rightarrow \boxed{K = \log_2 N}$$

Total comparisons in the worst case = $\log N$

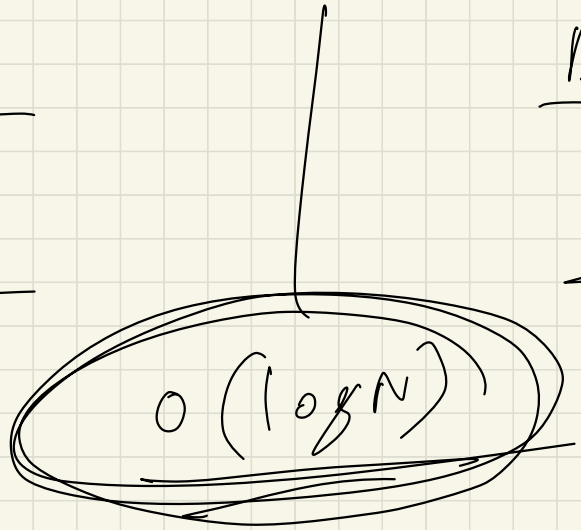
Search in a 1,000,000

Linear

1 mill

Binary search

20 comparisons



// better way to find mid

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

This may exceed the int range

$$m = s + \frac{(e-s)}{2}$$

$$s + \frac{e-s}{2}$$

$$\frac{2s + e - s}{2}$$

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

#dsa with Kunal

@ commclassroom

@ Kunalstut

Order agnostic Binary Search

arr = [90, 75, 18, 12, 6, 4, 3, 1]

s, m, e

target = 75

target > middle ⇒ left

e = m - 1

target < middle ⇒ right

s = m + 1

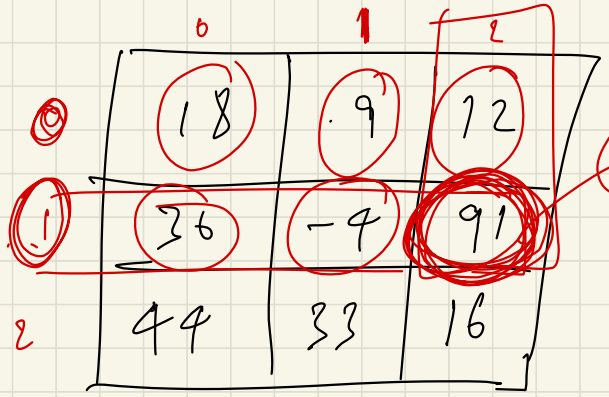
arr = [3, 2, 3, 5, 8, 9, 12, 14, 20, 33]

Prob 1

if s > f ⇒ increasing

else ⇒ decreasing

Searching in Matrices



target = 91

ans

for r=0; r<n; r++;

for c=0; c<n; c++;

if arr[r][c] == target

ans = [1, 2]

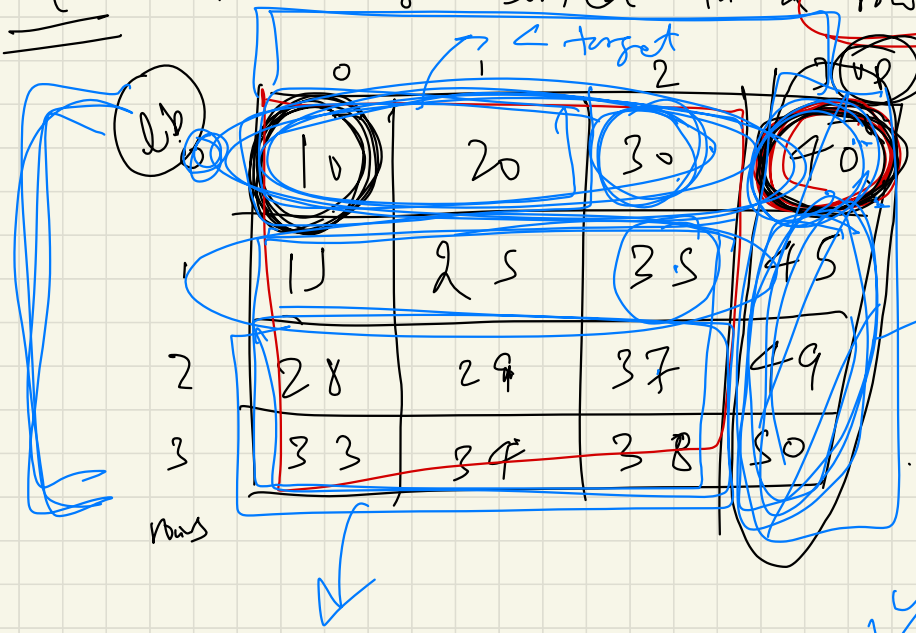
N * N = N^2 = $O(N^2)$

O (N * N)

Don't worry

return -1

Q: Matrix is sorted in a row wise & col wise manner.



target = 37

Case 1: if element == target
⇒ ans found

Case 2: If element < target
⇒ r++

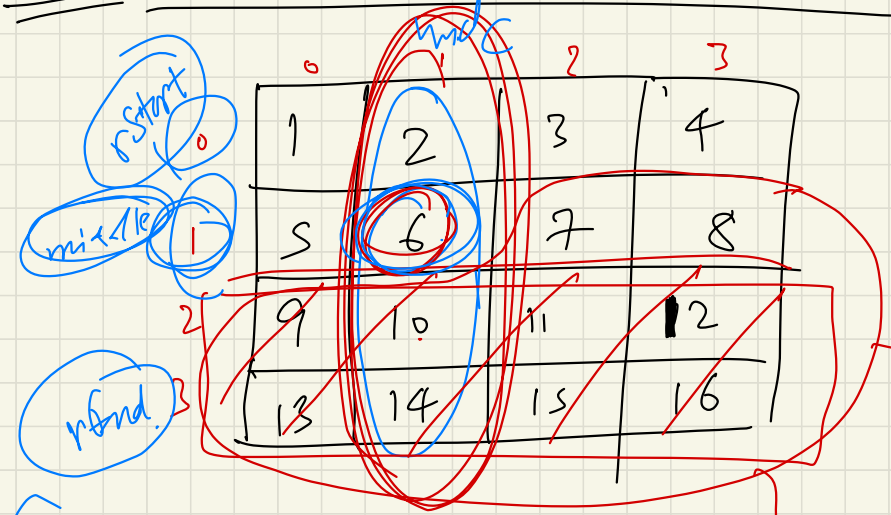
Case 3: If element > target
⇒ c--

n + n = 2n

O(N)

O(1)

Q: Search in a sorted matrix:



target = 2

* Take middle col & perform BS on it

--> mid = 6

6 > 2

--> 2 ≠ target

(1) If element == target
// ans

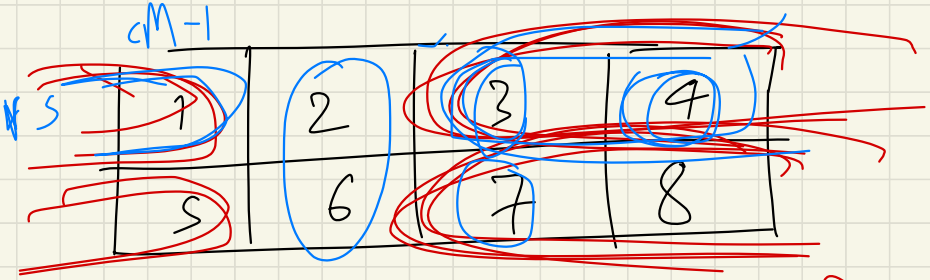
(2) If element > target
// Ignore rows after it

$\frac{rS + (cE - 1)}{2}$

(5) If element < target
// Ignore above rows

target = 3

In the end 2 rows are remaining:



$$O(\log(N) + \log(M))$$

(1) Check whether the middle column are at containing the ans i.e [2, 6]

(2) Consider the four parts