

Binary Search



$$\text{arr} = [2, 4, 9, 10, 12, 14, 18, 19]$$

→ ascending

$$\text{arr2} = [19, 12, 6, 5, 3, 2, -8, -10]$$

→ descending order

max comparisons : $N \Rightarrow$ No. of elements

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

- ① find the middle element
- ② $\text{target} > \text{mid} \Rightarrow$ search in the right
else search in left
- ③ if middle element == target element // ans

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

Indices: 0 1 2 3 4 5 6 7 8 9

The element 11 at index 4 is circled in red and blue.

target = 36

Sensor 12

Σ $\begin{matrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 2, & 4, & 6, & 9, & 11, & 12, & 14, & 20, & 36, & 48 \end{matrix}$ ϵ

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

Σ $\begin{matrix} 12, 14 \end{matrix}$ $\begin{matrix} m \\ 20, 36, 48 \end{matrix}$ ϵ

Σ $\begin{matrix} 12, 14 \end{matrix}$ ϵ

Ans = 5

if $s > e$: element not found.



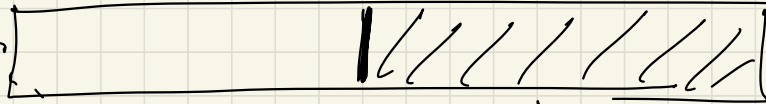
Best Case

$O(1)$

Why Binary Search?

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

0



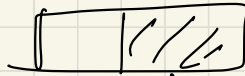
$$N = \frac{N}{2^0}$$

1

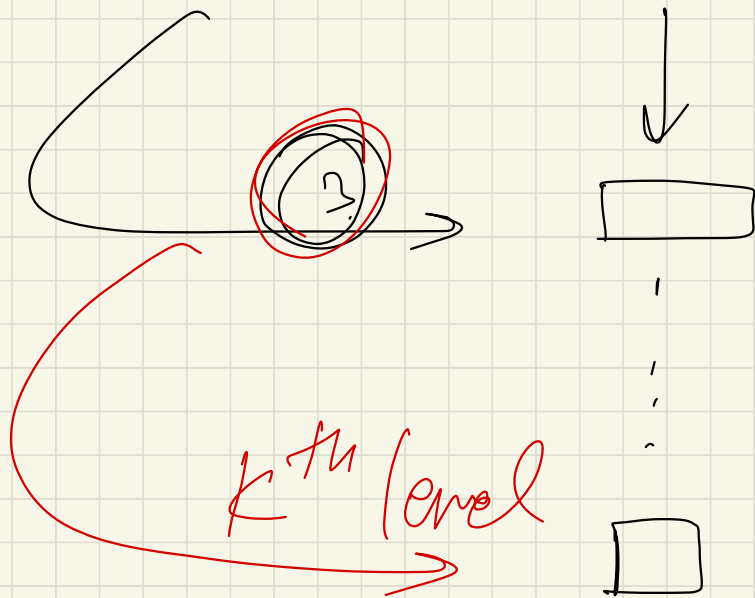


$$\frac{N}{2} = \frac{N}{2^1}$$

2



$$\frac{\frac{N}{2}}{2} = \frac{N}{4} = \frac{N}{2^2}$$



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

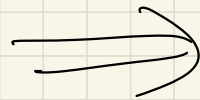
$$1 = \frac{N}{2^k}$$

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

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

$$\log N = k \log 2$$

$$k = \frac{\log N}{\log 2}$$



$$k = \log_2 N$$

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

Search in a 1, 000, 000

Linear

1 mill

Binary search

20 comparisons

$O(\log N)$

// better way to find mid

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

This may exceed
the int range

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

$$\begin{aligned} & s + \frac{e - s}{2} \\ & \cancel{s + e} \cancel{- s} \\ & = \frac{s + e}{2} \end{aligned}$$

#dsa with Kunal

@commclassroom

@Kunalstwt

Order agnostic Binary Search

arr = [90, 75, 18, 12, 6, 4, 3, 17]

Indices: 0 1 2 3 4 5 6 7

Annotations: Red '2' above 90, red '3' above 12, blue 'm' above 12, red '4' above 6, red '5' above 4, red '6' above 3, red '7' above 17. A blue circle is drawn around the element 6 at index 4.

target = 75

target > middle \Rightarrow left

$$e = m - 1$$

target < middle \Rightarrow right

$$s = m + 1$$

arr = [3, 2, 3, 3, 3, 3, 3, 14, 20, 33]

Handwritten annotations:

- A red circle around the first element '3'.
- A red arrow pointing from the circled '3' to the second element '2'.
- A red arrow pointing from the '2' to the word 'Prob!' written above the array.
- A red 's' written above the last element '33'.

if $s > j \Rightarrow$ increasing

else \Rightarrow decreasing

Searching in Matrices

	0	1	2
0	18	9	12
1	36	-4	91
2	44	33	16

target = 91

Ans

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

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

if arr[r][c] == target;

< ans

return -1;

Ans = [1, 2]

$N \times N =$

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

Don't worry

$O(N \times M)$

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

lib

target

rows

0	10	20	30	40
1	11	25	35	45
2	28	29	37	49
3	33	34	38	50

cols

target = 37

Case 1: if element == target
 \Rightarrow ans found

Case 2: If element < target
 $\Rightarrow r++$

Case 3: If element > target
 $\Rightarrow c--$

$n + m = 2n$
 $O(N)$
 $O(1)$

Q: Search in a sorted matrix:

target = 2

start 0
middle 1
rfind 3

mid

0	1	2	3
1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

* 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

$$\frac{2S + (e-1)}{2}$$

2) If element $<$ target
// ignore above row

target = 3

In the end 2 rows are remaining:

$m-1$

5	1	2	3	4
5	6	7	8	

1) Check whether the mid col you are at contains the ans
i.e. [2, 6]

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

2) Consider the four parts

